Journal of Mathematical Modeling
Vol. 9, No. 2, 2021, pp. 185–199

JMM

Solution of Kawahara equation using a predictor-corrector and RBF-QR method

Zahra Dehghan, Jalil Rashidinia*

Department of Mathematics, Central Tehran Branch, Islamic Azad University, Tehran, Iran

Email(s): kntu.dehghan@gmail.com, rashidinia@iust.ac.ir

Abstract. Two different methods based on radial basis functions (RBFs) for one-dimensional Kawahara equation are presented. In the first one, we use MQ-RBF with predictor-corrector scheme. Then the statistical tool LOOCV is implemented for selecting good value of shape parameter. In the second one a different scheme is constructed for time and then the RBF-QR method is implemented. In the both of two approaches, the Not-a-Knot method is used to improve the accuracy at the boundaries. The purpose of this paper is to devot suitable strategies to obtain more accurate and efficient solutions specially for arising fifth order time-dependent nonlinear equations comparing with the results from the relevant papers.

Keywords: Kawahara equation, multiquadric Radial basis functions, θ-weighted scheme, RBF-QR, LOOCV strategy.
AMS Subject Classification 2010: 65M99.

1 Introduction

Nonlinear partial differential equations play major role in various field of physics, applied mathematics, fluid mechanics, plasma physics, optical fibers, chemical physics, petroleum reservoir and geochemistry [2]. So finding an efficient method for solving these nonlinear equations with good accuracy would be important for researchers. A suitable numerical method for solving partial differential equations (PDEs) should be easy to implement, high-order accurate, flexible with respect to geometry and be efficient in computations. The methods that are usually imposed, fulfill one or two at those characteristics, but not all of them [21].

A new approach for solving PDEs is radial basis functions (RBFs). A radial basis function depends only on the distance to a center point x_j^c, $(j = 1, 2, \ldots, N)$ and is defined as $\psi(\|x - x_j^c\|)$. An important property of a RBF method is that it doesn't require a mesh grid. Also that RBFs may have a shape parameter ε which play critical role in accurate solution.

*Corresponding author.
Received: 26 July 2020 / Revised: 7 September 2020 / Accepted: 9 September 2020
DOI: 10.22124/jmm.2020.17221.1497

For the first time RBFs were used by Kansa in 1990 for solving PDEs [15,16]. In last decade the development of the RBFs as truly meshless for approximating the solution of PDEs has drawn the attention of many researchers in engineering and science [6,25,29,31].

In this work, we use two different methods based on RBFs for the numerical solution of Kawahara equation which was studied by Kawahara in 1972 in context of shallow water waves [17]. The Kawahara equation also occurs in the theory of magneto-acoustics waves in plasma, shallow water waves which surface tension and capillary gravity water waves [3,30]. In recent decades, some works have been doing in order to find the numerical solution and approximate the analytical solution of this equation such as: Adomian decomposition method [18], tanh-function method [33], sine-cosine method [34], homotopy perturbation method and variational iteration method [14], Crank-Nicolson differential quadrature method [19], homotopy analysis method [20], reduced differential transform method [1], Predictor-Corrector method [7], Dual-Petrov Galekin method [32] and RBF collocation method [13,23].

In this paper we consider the numerical solution of Kawahara equation

$$u_t + uu_x + u_{3x} - u_{5x} = 0, \qquad x \in \Omega = [a,b],\ t \geq 0. \tag{1}$$

The initial and boundary conditions for implementing numerical solution of equation (1) are as follows:

$$\begin{cases} u(x,0) = u_0(x),\ x \in \Omega = [a,b], \\ u(a,t) = g_1(t),\ u(b,t) = g_2(t), \\ u_x(a,t) = g_3(t),\ u_x(b,t) = g_4(t),\ u_{xx}(a,t) = g_5(t). \end{cases} \tag{2}$$

The local and global existence and uniqueness of solution of equation (1) with (2) are proved in [8,26].

The outline of this paper is as follows. In Section 2 we begin with reviewing the radial basis functions to approximate the solution of the equation. Section 3 is devoted to show the implementation of the numerical methods for equation (1) with (2). Also an efficient strategy for selecting good value for shape parameter are mentioned in this section. The numerical illustrations including comparison between available exact solution and the numerical results are given in Section 4. Some concluding remarks are given at the end.

2 Multiquadric radial basis function

Considering a spatial domain Ω and time interval $[0,T]$, we want to obtain a scalar function $u : \Omega \times [0,T] \to \mathbb{R}$, which satisfying the equation (1) with (2). The RBF approximation of a function $u(x,t)$ for Eq. (1) is given as:

$$u(x,t) = \sum_{j=1}^{N} \lambda_j(t)\psi(r_j) + \varphi(x),\ x \in \Omega, \tag{3}$$

where $r_j = \|x - x_j^c\|$ is the Euclidian norm. Also $\lambda_j(t)$, $(j = 1,2,\ldots,N)$ are coefficients to be determined. We can write the Eq. (3) without additional polynomial $\varphi(x)$. We have this condition when the radial basis function $\psi(x)$ is unconditionally positive definite to force the

interpolation matrix be solvable. Furthermore, $\varphi(x)$ is always required if $\psi(x)$ is defined as conditionally positive definite. we have implemented the multiquadric radial basis function (MQ):

$$\psi(r,\varepsilon) = \sqrt{(1+(\varepsilon r)^2)}. \tag{4}$$

MQ-RBFs are especially attractive for researchers because they have exponential convergence [22]. Due to these reasons, MQ-RBF will be used as radial basis function in Scheme 3.1 in section 3.

3 The presented methods

3.1 Scheme 1: The MQ-RBFs based on a predictor-corrector scheme

Firstly, let us discretize the time derivative in equation (1) by using a θ-weighted scheme for temporal approximation to obtain:

$$u^{n+1} - u^n = \Delta t \cdot \theta(u^n_{5x} - u^n_{3x} - (uu_x)^n) + \Delta t \cdot (1-\theta)(u^{n+1}_{5x} - u^{n+1}_{3x} - (uu_x)^{n+1}),\ n \geq 0, \tag{5}$$

where $u^n = u(x,t_n)$, $t_n = n\Delta t$, Δt is the length of the time step, $u^n_x = \dfrac{d}{dx}u(x,t_n)$ etc. If $\theta = \dfrac{1}{2}$ then the method coincides with the unconditionally stable Crank-Nicolson formula. Suppose that there are $N-2$ interpolation points in the domain Ω, so we can approximate $u^n(x)$ by:

$$u^n(x) = u(x,t_n) = \sum_{j=1}^{N-2} \lambda^n_j \psi(r_j) + \varphi(x). \tag{6}$$

All the $N-2$ unknown coefficients $\lambda_j(t)$ $(j = 1,2,\ldots,N-2)$ are imposed to the first term of (6), so we will have

$$u^n(x) = \sum_{j=1}^{N-2} \lambda^n_j \psi(r_j) + \lambda^n_{N-1}(x) + \lambda^n_N. \tag{7}$$

To determine the unknown coefficients λ_1, λ_2, $\ldots$, λ_{N-1}, λ_N, we can impose the collocation method by using (7) at every data point x_i, $i = 1,2,\ldots,N-2$. Thus we obtain:

$$u^n(x_i) = \sum_{j=1}^{N-2} \lambda^n_j \psi(r_{ij}) + \lambda^n_{N-1}x_i + \lambda^n_N, \qquad i = 1,2,\ldots,N-2. \tag{8}$$

where $r_{ij} = \sqrt{(x_i - x^c_j)^2}$. According to relation (8) the additional condition can be written as

$$\sum_{j=1}^{N-2} \lambda^n_j = \sum_{j=1}^{N-2} \lambda^n_j x_j = 0. \tag{9}$$

In order to write Eq. (8) with (9) in terms of matrix form we have:

$$u^n = A\lambda^n, \tag{10}$$

where $u^n = \begin{bmatrix} u_1^n & u_2^n & \dots & u_{N-2}^n & 0 & 0 \end{bmatrix}^T$, $\lambda^n = \begin{bmatrix} \lambda_1^n & \lambda_2^n & \dots & \lambda_{N-1}^n & \lambda_N^n \end{bmatrix}$ are $N \times 1$ vectors $A = \begin{bmatrix} a_{ij} \end{bmatrix}$ is the $N \times N$ matrix of the form

$$A = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{(N-2)} & x_1 & 1 \\ \psi_{21} & \psi_{22} & \cdots & \psi_{(N-2)} & x_2 & 1 \\ \vdots & \vdots & & \vdots & \vdots & \\ \psi_{(N-2)1} & \psi_{(N-2)2} & \cdots & \psi_{(N-2)(N-2)} & x_{N-2} & 1 \\ x_1 & x_2 & \cdots & x_{N-2} & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 \end{bmatrix}. \tag{11}$$

Suppose that in our calculations we have $d < N-2$ internal points, $b = N-2-d$ boundary points and two additional conditions (9). Thus we can split the matrix A into: $A = A_d + A_e + A_b$ for simplicity of computations. So applying (5) together with (10) we obtain:

$$B\lambda^{n+1} = C\lambda^n - \Delta t \cdot \theta[u_d(u_d)_x)^n] - \Delta t \cdot (1-\theta)[u_d(u_d)_x)^{n+1}] + G^{n+1}, \tag{12}$$

where

$$\begin{aligned} &A_d = [a_{ij}, \quad \text{for } 1 \le i \le d, \ 1 \le j \le N, \quad \text{and} \quad 0, \quad \text{elsewhere}], \\ &A_b = [a_{ij}, \quad \text{for } d < i \le N-2, \ 1 \le j \le N, \quad \text{and} \quad 0, \quad \text{elsewhere}], \\ &A_e = [a_{ij}, \quad \text{for } N-1 \le i \le N, \ 1 \le j \le N, \quad \text{and} \quad 0, \quad \text{elsewhere}], \\ &B = A_d + \Delta t(1-\theta)[(A_d)_{3x} - (A_d)_{5x}] + A_b + A_e, \\ &C = A_d - \Delta t\theta[(A_d)_{3x} - (A_d)_{5x}], \\ &G^{n+1} = \begin{bmatrix} 0 & \dots & 0 & g_{d+1}^{n+1} & \dots & g_{N-2}^{n+1} & 0 & 0 \end{bmatrix}^T, \end{aligned}$$

and $u_d^n = A_d\lambda^n$. For defining the matrices B and C, we need to compute matrices A_d, $(A_d)_{3x}$, $(A_d)_{5x}$ [5,6] defined as

$$(A_d)_\alpha(x_i) = \frac{\partial^\alpha}{\partial x^\alpha} A_d(x_i), \quad \alpha = 1,3,5, \quad i = 1,2,\dots,N.$$

For initial value problems the predictor-corrector methods are the application of multistep methods. For solving the nonlinear system (12), we use the following predictor-corrector scheme. As we cannot find $(uu_x)^{n+1}$ and then λ^{n+1} explicitly, we use an iterative procedure for finding λ^{n+1} in the system of equations (12): P: Predict some value $\lambda_{(0)}^{n+1}$ for λ^{n+1}, E: Evaluate the system of equations for $\lambda_{(0)}^{n+1}$, C: Correct $\lambda_{(0)}^{n+1}$ to obtain a new $\lambda_{(1)}^{n+1}$ for λ^{n+1}. The sequence of operations could be written in the form PECECE.... When $n = 0$ at the first time level t_0, we can determine vector u^0 by using the initial condition and then vector λ^0 can be determined by the system (10).

For dealing with the nonlinearity at the other time steps, (for example, in time level $n = 1,2,3,\dots$), we propose the following iterative algorithm:
Step 1: In the nonlinear term we set $u_d^{n+1} = u_d^n$.
Step 2 : Calculate an "intermediate" value $\lambda_{(0)}^{n+1}$ which is called *predictor* from system of equations (12). So we obtain the following system of equations:

$$B\lambda_{(0)}^{n+1} = C\lambda^n - \Delta t \cdot \theta[u_d(u_d)_x)^n] - \Delta t(1-\theta)[u_d(u_d)_x)^n] + G^{n+1}. \tag{13}$$

Notice that the value of λ^n and $[u_d(u_d)_x)^n]$ are known from the previous time level n. So by using the predicted value $\lambda_{(0)}^{n+1}$ we can approximate $\lambda_{(1)}^{n+1}$.
Step 3: we observe that the general iteration of predictor-corrector scheme for $\mu = 0, 1, 2, 3, \ldots$ and $n = 1, 2, 3, \ldots$ is as follows:

$$B\lambda_{(\mu+1)}^{n+1} = C\lambda^n - [\Delta t\theta[u_d(u_d)_x)^n] - \Delta t(1-\theta)[u_d(u_d)_x)_{(\mu)}^{n+1}] + G^{n+1}. \tag{14}$$

Step 4: calculate the system of linear equations (14) until getting:

$$\frac{\|u_{(\mu+1)}^{n+1} - u_{(\mu)}^n\|_\infty}{\|u_{(\mu)}^{n+1}\|_\infty} \leq e, \tag{15}$$

for prescribed tolerance e or until the maximum number of iterations ≤ 65. When the convergence condition is achieved, we can move to the next time level. This procedure is continued until reaching to the requested time T.

3.2 Choosing the shape parameter

The accuracy of the RBFs approximation depends heavily on the choice of a shape parameter ε. Theoretically, for a fixed number of centers N, smaller shape parameters produce the more accurate approximation, but is caused a poorly or ill-conditioned interpolation matrix. The optimal choice of the shape parameter is an open problem yet. A number of researchers have been suggested optimal forms of choosing shape parameters by different formulas [9]. In the scheme 3.1 ($MQ - RBF$ based a predictor-corrector) we use statistical tool, leave one out cross validation (LOOCV) strategy for selecting good values of the shape parameter. In this algorithm, an optimal value of ε is selected by minimizing a cost function $E = [E_1, E_2, \ldots, E_n]^T$ where $E_k = |u_k - s^k(x_k)|$ and $s^k(x)$ is the approximate function to a reduced data obtained by removing the point x_k from the original data set. Rippa [24] proved that the kth element of the error vector E_k can be computed by the formula $E_k = \lambda_k / A_{kk}^{-1}$ where A_{kk}^{-1} and λ_k are kth diagonal element of A^{-1} and kth element of λ respectively. According to Rippa's strategy, a good value for shape parameter ε is obtained by minimizing $\|E\|$ [28].

3.3 Scheme 2: The RBF-QR method

The best accuracy of using RBFs is achieved when the shape parameter ε is small. In recent years, two numerical methods have been implemented that are able to compute stably when $\varepsilon \to 0$: the Contour-Pade algorithm and the RBF-QR method [11,12]. The Gaussian radial basis function leads to the best accuracy when the shape parameter is small. But the obstacle is that the coefficient matrix will be ill-conditioned as the small shape parameter ε is used. The RBF-QR method is presented here which are able to present stable computations. To implement the RBF-QR method for solving equation (1) with (2), let us first discretize the Kawahara equation (1) by a second-order compact finite difference scheme for temporal approximation. So, we have:

$$\frac{u^{n+1} - u^n}{\Delta t} - e^n + u^n u_x^n + u_{3x}^n - u_{5x}^n = 0, \qquad n \geq 0, \tag{16}$$

where e^n is the truncation error given by:

$$e_n = \frac{\Delta t}{2} u_{tt}^n + O(\Delta t^2). \tag{17}$$

To obtain a higher order compact scheme, we can differentiate (16) to construct u_{tt}^n:

$$u_{tt}^n = -u_t^n u_x^n - u_{xt}^n u^n - u_{3xt}^n + u_{5xt}^n. \tag{18}$$

By discretizing the time derivatives in the right hand side of (18) with a forward difference scheme we get:

$$u_{tt}^n = \left(\frac{u_{5x}^{n+1} - u_{5x}^n}{\Delta t}\right) - \left(\frac{u_{3x}^{n+1} - u_{3x}^n}{\Delta t}\right) - u_x^n\left(\frac{u^{n+1} - u^n}{\Delta t}\right) - u^n\left(\frac{u_x^{n+1} - u_x^n}{\Delta t}\right) + O(\Delta t). \tag{19}$$

If we substitute (19) into (16) and put the result of e_n in (17), after neglecting $O(\Delta t^2)$, we can get the new numerical scheme as follows:

$$u^{n+1}[1 + \frac{\Delta t}{2} u_x^n] - \frac{\Delta t}{2} u_{5x}^{n+1} + \frac{\Delta t}{2} u_{3x}^{n+1} + \frac{\Delta t}{2} u^n u_x^{n+1} = u^n + \frac{\Delta t}{2} u_{5x}^n - \frac{\Delta t}{2} u_{3x}^n. \tag{20}$$

Assuming that there are N interpolation points in $\Omega = [a, b]$, we can approximate $u(x, n)$ at every point x_i by

$$u(x_i, n) = \sum_{j=1}^{N} a_{ij}\psi(r_{ij}), \quad i = 1, 2, \ldots, N,$$

where in matrix form we have

$$u^n = A\lambda^n, \; A = \begin{bmatrix} \psi(r_{11}) & \psi(r_{12}) & \ldots & \psi(r_{1N}) \\ \psi(r_{21}) & \psi(r_{22}) & \ldots & \psi(r_{2N}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(r_{N1}) & \psi(r_{N2}) & \ldots & \psi(r_{NN}) \end{bmatrix} \tag{21}$$

Then equation (20) together with (2), can be written in the matrix form as

$$B_1\lambda^{n+1} = C_1\lambda^n + G^{n+1}, \tag{22}$$

$$\begin{cases} B_1 = A + \dfrac{\Delta t}{2}(A^* u_x^n - A_{5x} + A_{3x} + u_n * A_x), \\ C_1 = A + \dfrac{\Delta t}{2}(A_{5x} - A_{3x}), \end{cases} \tag{23}$$

where $G^{n+1} = \begin{bmatrix} 0 & \ldots & 0 & g_{d+1}^{n+1} & \ldots & g_{N-2}^{n+1} & 0 & 0 \end{bmatrix}^T$ is the vector of boundary conditions (2), symbol $(*)$ denotes component by component multiplication and

$$A_{jx} = [\frac{\partial^j}{\partial x^j} A(x_j)]_{N\times N}, \; i = 1, 2, \ldots, N, \; j = 3, 5.$$

In one-dimensional case, the Gaussian radial basis function $\psi(r,\varepsilon) = e^{-\varepsilon^2 r^2}$ is written as the expansion in terms of the Chebyshev polynomials $T_j(x)$. This leads to

$$\psi_k(x) = \psi(\|x - x_k\|) = \sum_{j=0}^{\infty} d_j c_j(x_k) e^{-\varepsilon^2 x^2} T_j(x), \tag{24}$$

where the coefficients are

$$d_j = \frac{2c^{2j}}{j!}, \quad c_j(x_k) = t_j e^{-\varepsilon^2 x_k^2} x_{k0}^j F_1([], j+1, \varepsilon^4 x_k^2),$$

with $t_j = 1$ for $j > 0$ and $t_0 = \frac{1}{2}$. ${}_mF_n(z)$ is the hypergeometric function. The function of the form ${}_0F_1(;a;z)$ are called confluent hypergeometric limit functions and are closely related to Bessel functions [4]. After truncating the series for p terms, we can write the coefficient matrix $\psi(\|x - x_j^c\|)$ as follows:

$$\underbrace{\begin{bmatrix} \psi(\|x-x_1^c\|) \\ \psi(\|x-x_2^c\|) \\ \vdots \\ \psi(\|x-x_N^c\|) \end{bmatrix}}_{\overline{\psi(x)}}_{N\times N} = \underbrace{\begin{bmatrix} c_0(x_1) & c_1(x_1) & \dots & c_p(x_1) \\ c_0(x_2) & c_1(x_2) & \dots & c_p(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ c_0(x_N) & c_1(x_N) & \dots & c_p(x_N) \end{bmatrix}}_{C} \cdot \underbrace{\begin{bmatrix} \varepsilon^0 & & & \\ & \varepsilon^2 & & \\ & & & \varepsilon^{2p} \end{bmatrix}}_{E} \cdot \underbrace{\begin{bmatrix} e^{-\varepsilon^2 x^2} T_0(x) \\ e^{-\varepsilon^2 x^2} T_1(x) \\ \vdots \\ e^{-\varepsilon^2 x^2} T_p(x) \end{bmatrix}}_{\overline{F(x)}} \tag{25}$$

At this stage based on the QR-factorization, we are going to find new basis functions that can overcome the ill-conditioning of the matrix $\psi(x)$. To achieve this, we split $C = QR$, where Q is unitary and R is an upper triangular matrix. So we have

$$\psi(x) = C.E.F(x) = Q.R.E.F(x). \tag{26}$$

After multiplying the relation (26) by $E^{-1}Q^T$ from the left, where E denotes the diagonal matrix with the increasing powers of ε, we create a well-conditioned basis in exactly the same space. It results in the new set of basis function

$$\underbrace{E^{-1}.Q^T.\psi(x)}_{\phi(x)} = E^{-1}.\underbrace{Q^T.Q}_{I}.R.E.F(x) \Rightarrow \phi(x) = E^{-1}.R.E.F(x). \tag{27}$$

The matrix $E^{-1}.R.E$ provides the well-conditioned, upper triangular matrix that we will use numerically. It should be noted that we obtain new basis functions but dont change the space that is spanned by them. For applying the RBF-QR for solving Kawahara equation (1), only the collocation matrix A in (22) is replaced by the new basis function in (27). Then the obtained linear system of equations can be solved by LU the factorization to find u^{n+1}.

3.4 The Not-a-Knot scheme

A usual feature in all RBFs approximation is how they are inaccurate at boundaries. When we use RBFs for equation (1) if centers cover a domain in a uniform manner, the largest errors will appear near the boundaries. Also edge errors start to dominate over interior errors. Thus it is necessary to know how RBFs behave near boundaries and whether there is a way to improve accuracy there. One of the strategies to reduce errors in the boundary region is to separate the location of the centers and the data points. It means that the centers which are nearest to the boundary are moved outside the domain. This strategy is referred to as a Not-a-Knot (NaK) scheme due to its connection with polynomial splines. The Not-a-Knot method is used to improve the accuracy at the boundaries especially in nonlinear equation with large N. If the two extra conditions are chosen so that there is no jump in the third derivative at the first and last interior data point, tt means that moving the center points at those points outside the interval. For this reason, in the Not-a-knot method, the set of points at which PDE conditions (2) are imposed is chosen a little different from the set of RBF centers. The approximation in the interior is independent of how far the centers are moved outside of the interval [10]. In our proposed method two nearest centers to the boundary are moved at spatial domain $\Omega = [a, b]$ outside of interval as measure as distance between data points or as much as very small amount. We should consider the condition that the number of data points and centers remain unchanged. Table 1 illustrates this fact, with the new centers in the domain.

Table 1: illustrating Not-a-Knot scheme with two new centers in domain $[a, b]$.

New center	First of interval	Moved center	Interior center	Interior center	-	-	-	-	Interior center	Interior center	Moved center	End of interval	New center

4 Numerical illustrations

Problem 1. Consider the Kawahara equation

$$\begin{cases} u_t + uu_x + u_{3x} - u_{5x} = 0, & x \in [-20, 30],\ t \in [0, T], \\ u(x, 0) = \frac{105}{169} \sec h^4[k'(x - x_0)], \end{cases} \tag{28}$$

with the boundary conditions [27]:

$$\begin{cases} u(-20, t) = 0,\ u(30, t) = 0, \\ u_x(-20, t) = 0,\ u_x(30, t) = 0, u_{xx}(-20, t) = 0. \end{cases} \tag{29}$$

The exact solution given in [19] is:

$$u(x, y) = \frac{105}{169}\text{sech}^4[k'(x - \frac{36}{169}t - x_0].$$

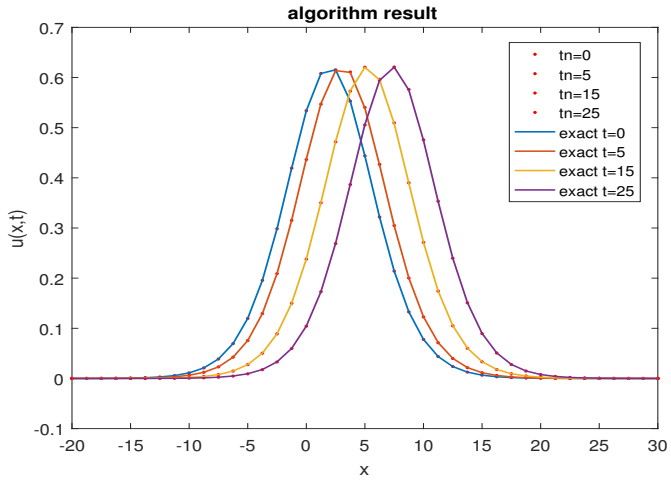


Figure 1: Traveling wave solution of Kawahara equation (28), numerical solution (full line) and by Scheme 3.1.

Accuracy of the estimated solutions can be calculated by L_2 and L_∞ error norms which are:

$$L_2 = \|u_{ex} - u_{num}\|_2 = \left(\sum_{i=1}^{N} |u_{ex}(i) - u_{num}(i)| \right)^{\frac{1}{2}},$$
$$L_\infty = \|u_{ex} - u_{num}\|_\infty = \max_i |u_{ex}(i) - u_{num}(i)|.$$

We have three invariants I_1, I_2 and I_3 defined as lowest conserved quantities which are explained in [19]

In this problem we concern $k^{'} = 1/(2\sqrt{13})$, $x_0 = 2$. The computation is calculated up to time $T = 25$. L_2 and L_∞ error norms are obtained at desired time $t = 0, 5, 15, 25$ with time step $\Delta t = 0.001$ and $N = 51$. Numerical solution of (28) with (29) obtained by Scheme 3.1 are tabulated in Table 2, which shows that solving kawahara equation by Scheme 3.1 is considerable more accurate than [13]. Figure 1 shows the forward motion of the solitary wave which propagates from left to right with constant speed in different time level. In Figure 2, motion of solitary wave in 3-dimension is shown. Figure 3 shows error norms L_2 and L_∞ versus shape parameter. The good value $\varepsilon = 4.3$ for shape parameter has been selected by the LOOCV algorithm. We observe that in neighborhood of $\varepsilon = 4.3$ the error norms specially L_∞ behave more stably. The solitary wave motion of Kawahara equation (28) in large N with RBF-QR is shown in Figure 4. Using Not-a-Knot (NaK) scheme results stability at the boundaries specially in large number of data points N. The maximum error is not occurred at the boundaries because we use the Not-a-Knot method that improves the accuracy at the corners of the interval. It is clear from Table 3 and comparison with Table 2 that the results obtained by MQ-RBF (with small shape parameter) has good agreement with the exact solution and Scheme 3.1 and show better accuracy than [13].

Table 2: Comparison of result for Scheme 3.1 for problem (1) with [13].

Time	L_2	L_∞	I_1	I_2	I_3
Scheme 3.1(MQ-RBF based a predictor-corrector)					
0	0	0	5.97357	1.27250	-0.16458
5	5.856e (-7)	1.045e (-7)	5.97361	1.27250	-0.16458
15	4.168e (-7)	1.218e (-7)	5.97364	1.27250	-0.16459
25	5.418e (-7)	2.186e (-7)	5.97364	1.27250	-0.16459
Method in [13]					
0	0	0	5.97357	1.27250	-0.16459
5	6.278e (-5)	1.684e (-5)	5.97361	1.27250	-0.16459
15	4.466e (-5)	1.852e (-5)	5.97364	1.27250	-0.16459
25	9.370e (-5)	2.387e (-5)	5.97367	1.27250	-0.16459

Table 3: Comparison between the error norms L_2, L_∞ of Scheme 3.3 in time step $\Delta t = 0.001$,with $N = 51$, $t = 25$, $\varepsilon = 0.01$.

Time	L_2	L_∞
0	0	0
5	5.901e (-7)	3.344e (-7)
15	5.846e (-7)	3.363e (-7)
25	5.585e (-7)	3.157e (-7)

We observe from Table 4 that the condition number of matrix $B_1 \approx 1$ in Scheme 3.3 which shows the stability of our computations with small value of shape parameter. We calculate spatial rate of convergence [6] in Table 5 by keeping time step $\Delta t = 0.001$ fixed and varying the number of collocation points ($N = 20, 40, 80$). We can see that the order of convergence decreases with the smaller spatial step size.

Problem 2. Here we consider a Kawahara-type equation given as

$$\begin{cases} u_t + uu_x + u_{3x} + u_x - u_{5x} = 0, & x \in [-20, 50],\ t \in [0, T], \\ u(x, 0) = \frac{105}{169} \sec h^4[k'(x - x_0)]. \end{cases} \tag{30}$$

Other required data such as x_0, k' and Δt are the same as Problem (1). The error norms L_2, L_∞ and the conserved quantities I_i $(i = 1, 2)$ by Scheme 3.1 and Scheme 3.3, are given in Table 6. It is clear from Table 6 that all the conserved quantities are well preserved and the results of Table 6 shows that for Problem (30), Scheme 3.3 is more accurate and suitable than Scheme 3.1. Our schemes have acceptable accuracy. Also we can observe from Figure 5 that motion of solitary wave propagates from left to right with constant speed in spatial interval $[-20, 50]$.

Table 4: Condition number of the matrix B_1 of Scheme 3.3 in time step $\Delta t = 0.001$, $\varepsilon = 0.01$, with $N = 51$, $t = 25$.

Time	RBF-QR method
5	1.00009
10	1.00002
15	1.00013
25	1.00009

Table 5: Spatial rate of convergence for problem (1) with Scheme 3.1.

N	L_2	order	L_∞	order
Scheme 3.1				
20	0.0003048		0.0001693	
40	2.8553e(-4)	5.9004	2.1535e(-4)	5.9178
80	1.2107e(-7)	0.5361	1.1987e(-7)	0.0422

Table 6: Results for Problem (29) for $\Delta t = 0.001$, $T = 25$ by Scheme 3.1 and Scheme 3.3.

	Time	L_2	L_∞	I_1	I_2
	0	0	0	5.97361	1.27250
	10	0.666(-5)	0.416(-5)	5.97361	1.27250
S-3.1:	20	1.756(-5)	0.886(-5)	5.97362	1.27250
	30	4.784(-5)	1.726(-5)	5.97363	1.27250
	0	0	0	5.97371	1.27250
	10	$1.387e(-7)$	$2.934e(-7)$	5.97371	1.27250
S-3.3:	20	$2.920e(-7)$	$1.507e(-7)$	5.97380	1.27250
	30	$2.981e(-7)$	$1.298e(-7)$	5.97381	1.27250

Table 7: Results for problem (2) for $\Delta t = 0.001$, $\varepsilon = 0.001$, $T = 25$ by Scheme 3.1 and Gaussian-RBF.

N	L_2	L_∞
Scheme 3.1		
51	1.1376e (+35)	2.9312e (+37)
71	4.0891e (+20)	3.9804e (+19)
91	2.1916e (+24)	6.9012e (+25)

As shown in Table 7, the RBF based a predictor-corrector scheme when using Gaussian RBF

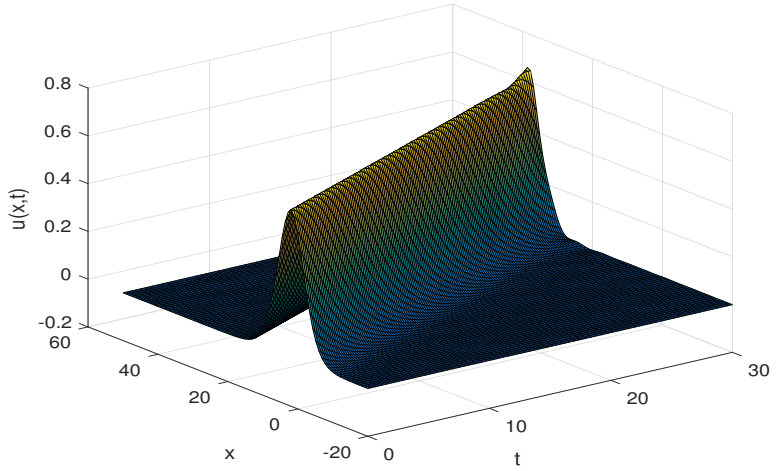
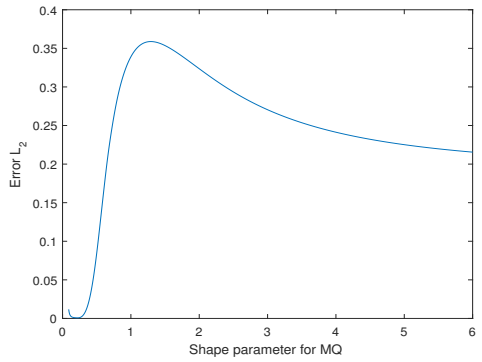
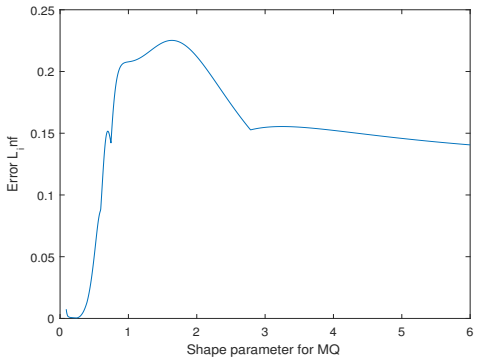


Figure 2: dimentional motion of solitary wave of Kawahara equation (28).



(a) Error L_2 verses shape parameter for Example 4.1



(b) Error L_∞ verses shape parameter

Figure 3: Error L_2, L_∞ versus shape parameter ε for problem (1) by Scheme 3.1.

isn't stable for $\varepsilon = 0.001$. So, for stability of computations for small value of shape parameter Scheme 3.3 is recomended.

5 Conclusion

Two numerical methods $MQ - RBF$ based on a predictor-corrector scheme and $RBF - QR$ method have been applied to find numerical solution of kawahara equation. In MQ-RBF based predictor-corrector we have used the LOOCV strategy for selecting shape parameter. In both methods we have used NaK scheme for increasing accuracy at the boundaries. The presented methods have provided better results and produce efficient and more accurate alternative for the solution of kawahara equation in comparison with work in [13].

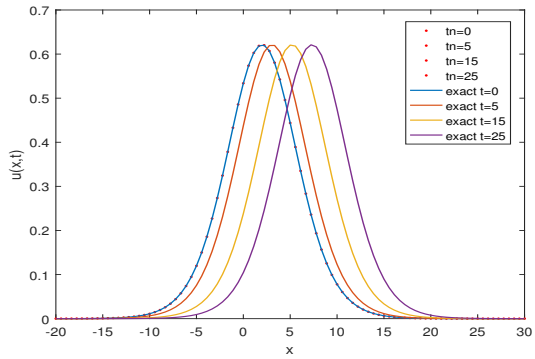


Figure 4: Travelling wave solution of Kawahara equation (28) for large $N = 210$, numerical solution (full line) and exact solution (dot line) by Scheme 3.3.

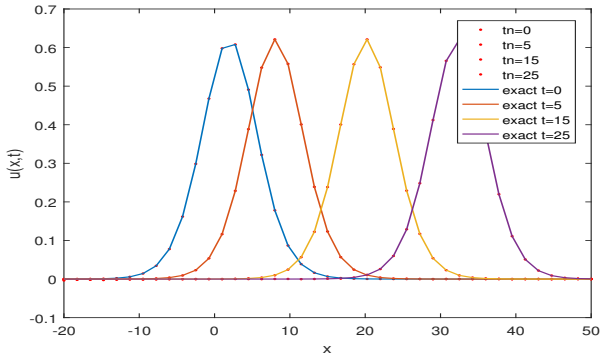


Figure 5: Travelling wave solution of Kawahara equation (29), numerical solution (full line) and exact solution (dot line) by Scheme 3.3 and $\Delta t = 0.001$, $N = 81$, $T = 25$.

References

[1] R. Abazari, B. Soltanalizadeh, *Reduced differential transform method and its application on Kawahara equation*, Thai J. Math. **11** (2013) 1199-1216.

[2] M.J. Ablowitz, P.A. Clarkson, *Nonlinear Evolution Equations and Inverse Scattering Form*, Cambridge University Press, England, 1991.

[3] M.J. Ablowitz, H. Segur, *Solitons and Inverse Scattering Transformation*, SIAM, Philadelphia, 1981.

[4] K. Aomoto, M. Kita, *Theory of Hypergeometric Functions*, Springer, Tokyo, 2011.

[5] N. Bib, S.I.A. Tirmizi, S. Haq, *Meshless method of lines for numerical solution of Kawahara type equations*, App. Math. **2** (2011) 608-618.

[6] 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.

[7] K. Djidjeli, W.G. Price, E.H. Twizel, Y. Wand, *Numerical methods for the solution of the third- and fifth-order dispersive KdV equations*, J. Sci. Comput. **58** (1995) 307-336.

[8] G. Doronin, N.A. Larkin, *Well and ill-posed problems for the KdV and Kawahara equations*, Bol. Soc. Parana. Mat. **26** (2008) 133-137.

[9] G.E. Fasshauer, J.G. Zhang, *On choosing optimal shape parameters for RBF approximation*, Numer. Algorithms **45** (2007) 345-368.

[10] B. Fornberg, T.A. Driscoll, G. Wright, R. Charles, *Observation on the behavior of radial basis functions near boundaries*, Comput. Math. Appl. **43** (2002) 473-490.

[11] B. Fornberg, C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comput. **30** (2007) 60-80.

[12] B. Fornberg, G. Wright, *Stable computations of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl. **48** (2004) 853-867.

[13] S. Haq, M. Uddin, *RBFs approximation method for Kawahara equation*, Eng. Anal. Bound. Elem. **35** (2011) 575-580.

[14] L. Jin, *Application of variational iteration method and homotopy perturbation method to the modified Kawahara Equation*, Math. Comput. Model. **34** (2009) 573-578.

[15] E.J. Kansa, *Multiquadrics a scattered data approximation scheme with applications to computational fluid dynamics I*, Comput. Math. Appl. **19** (1990) 127-145.

[16] E.J. Kansa, *Multiquadrics a scattered data approximation scheme with applications to computational fluid dynamics II*, Comput. Math. Appl. **19** (1990) 147-161.

[17] T. Kawahara, *Oscillatory solitary waves in dispersive media*, J. Phys. Soc. Japan. **33** (1972) 260-264.

[18] D. Kaya, *An explicit and numerical solutions of some fifth-order KdV equations by decomposition method*, Appl. Math. Comput. **144** (2003) 353-363.

[19] A. Korkmaz, I. Dag, *Crank Nicolson differential quadrature algorithms for the Kawahara equation*, Chaos Solitons Fractals. **42** (2009) 65-73.

[20] M. Kurulay, *Approximate analytic solutions of the modified Kawahara equation with homotopy analysis method*, Adv. Differ. Eq. **178** (2012).

[21] E. Larsson, B. Fornberg, *A numerical study of some radial basis function based solution methods for elliptic PDEs*, Comput. Math. Appl. **46** (2003) 891-902.

[22] W.R. Madych, S.A. Nelson, *Multivariate interpolation and conditionally positive definite functions II*, Math. Comput. **54** (1991) 211-230.

[23] J. Rashidinia, M. Rasoulizadeh, *Numerical methods based on radial basis function-generated finite difference (RBF-FD) for solution of GKdVB equation*, Wave Motion **99** (2019) 152-167.

[24] S. Rippa, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math. **11** (1999) 193-210.

[25] 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.

[26] T. Shuangping, C. Shuangbin, *Existence and uniqueness of solutions to nonlinear Kawahara equations*, Chinese Ann. Math. Ser. A. **23** (2002) 221-228.

[27] Sirendaoreji, *New exact traveling wave solutions for the Kawahara and modified Kawahara equations*, Chaos Solitons Fractals **19** (2004) 147–50.

[28] M. Uddin, *On the selection of a good value of shape parameter in solving time-dependent partial defferential equations using RBF approximation*, Appl. Math. Model. **38** (2014) 135-144.

[29] M. Uddin, S. Haq, *A mesh-free numerical method for solution of the family of Kuramoto-Sivashinsky equations*, Appl. Math. Comput. **212** (2009) 458-469.

[30] A.M. Wazwaz, *Compacton solutions of the Kawahara-type nonlinear dispersive equation*, Appl. Math. Comput. **145** (2003) 133-150.

[31] H. Xie, D. Li, *A meshless method for Burger's equation using MQ-RBF and high-order temporal approximation*, Appl. Math. Model. **37** (2013) 9215-9222.

[32] J.M. Yuan, J. Shen and W. Jiahong, *A dual petrov-galerkin method for the Kawahara-type equations*, J. Sci. Comput. **34** (2008) 48-63.

[33] E. Yusufoglu, A. Bekir, *Symbolic computation and new families of exact travelling solutions for the Kawahara and modified Kawahara equations*, Comput. Math. Appl. **55** (2008) 1113-1121.

[34] E. Yusufoglu, A. Bekir and M. Alp, *Periodic and solitary wave solutions of Kawahara and modified Kawahara equations by using sine-cosine method*, Chaos Solitons Fractals **37** (2008) 1193-1197.