Available online at http://ijim.srbiau.ac.ir

Int. J. Industrial Mathematics Vol. 3, No. 2 (2011) 111-123

Using Multiquadric Quasi-Interpolation for Solving Kawahara Equation

R. Ezzati *, K. Shakibi, M. Ghasemimanesh

Department of Mathematics, Karaj Branch, Islamic Azad University, Karaj, Iran.

Received 12 June 2010; revised 20 December 2010; accepted 22 December 2011.

Abstract

Multiquadric quasi-interpolation is a useful instrument in approximation theory and its applications. In this paper, a numerical approach for solving Kawahara equation (KE) is developed by using multiquadric quasi-interpolation method. Obtaining numerical solution of KE by multiquadric quasi-interpolation is done by a recurrence relation. In this recurrence relation, the approximation of derivative is evaluated directly without the need to solve any linear system of equation. Also, by combining Hermite interpolation and quasi-interpolation L_D, another way to solve KE is obtained. The KE occurs in the theory of magneto-acoustic waves in a plasma and in the theory of shallow water waves with surface tension. We test the method in two examples and compare the numerical and exact results.

Keywords : Radial basis function; Quasi-interpolation; Preserving monotonicity; Linear reproducing, The Kawahara equation; Hermite interpolating polynomial.

---

1 Introduction

Nonlinear equations play an important role in various fileds of sciences. The actual world is nonlinear, so these equations are a model to describe the physical phenomena. Unfortunately, solving the nonlinear equations is harder than the linear ones, so we have always been looking for ways to solve them more easily. The KE is a nonlinear partial differential equation. It was first proposed by Kawahara in 1972 as a model equation describing solitary-wave propagation in media [1]. This equation occurs in the theory of magneto-acoustic waves in a plasma and in the theory of shallow water waves with surface tension [5]. The KE has been the subject of wide research work [1, 10, 11, 12]. Abbasbandy [1] solved the KE with homotopy analysis method (HAM) and proved that

*Corresponding author. Email address: ezati@kiau.ac.ir.

obtained solution of this equation by using (HAM) has a reasonable residual error. This method is a powerful analytical tool for nonlinear problems. The approximate solution of KE with this method is obtained as a series of exponentials. Moreover we can find the exact and numerical solution of KE by the Variational Iteration Method (VIM) [5]. The VIM is based on Lagrange multipliers. Using this method creates a sequence which tends to the exact solution of the problem. Existence and uniqueness of solution of the KE is considered in [9]. Recently, many authors have applied the direct algebraic method to find the exact solution of nonlinear PDE such as KE [13, 14, 15]. The KE is as follows:

$$u_t + \alpha u u_x + \beta u_{3x} + \gamma u_{5x} = 0 \tag{1.1}$$

where α, β, γ are arbitrary constants. The numerical solution of Eq. (1) is obtained subject to the initial condition:

$$u(x, 0) = f(x), \quad x \in R.$$

The rest of this paper is as follows: In Section 2, multiquadric quasi interpolation is introduced. In Section 3, mathematical formulation of our method is explained. In Section 4, Hermite quasi-interpolation is mentioned and mathematical formulation for Hermite quasi-interpolation is expressed. In Section 5, two examples for testing our methods are shown and in the last section the conclusion is derived.

2 Multiquadric quasi-interpolation method

Hardy [6] proposed multiquadric (MQ) in 1968 as a kind of radial basis function (RBF). For the first time, Kansa [8] successfully used modified MQ for solving partial differential equation (PDE). In 1992, Betson and Powell [2] proposed three univariate multiquadric quasi-interpolations. They named them L_A, L_B, L_C to approximate a function$\{f(x)|x_0 \leq x \leq x_n\}$. Afterwards, Schaback and Wu [3] proposed a multiquadric quasi-interpolation L_D to improve L_A, L_B, L_C. Multiquadric quasi-interpolation L_D possesses preserving monotonicity, convexity preserving and linear reproducing on $[x_0, x_n]$, but for example L_A and L_B cannot preserve both linearity and convexity. Quasi-interpolation is an appropriate instrument in approximation theory and its applications. Multiquadric is used in geodesy, geophysics, photogrammetry, hydrology and mining and so on [7] but the most important advantage of quasi-interpolation is that one can evaluate the approximation directly without the need to solve any linear system of equations. In this section we introduce the multi-quadric quasi-interpolation L_D to approximate a function $\{f(x)|x_0 \leq x \leq x_n\}$. Given the points $\{(x_j, f_j)\}_{j=0}^{n}$ where $x_0 \leq x_1 \leq ... \leq x_n$, the form of univariate quasi-interpolation is as follows:

$$f^*(x) = \sum_{j=0}^{n} f_j \psi_j(x) \tag{2.2}$$

where, $\psi_j(x)$ $(j = 0, 1, ...n)$ is a linear combination of radial basis functions. Buhmann [4] considered $\psi(x)$ as a second divided difference of ϕ as follows:

$$\psi_j(x) = \frac{\phi_{j+1}(x) - \phi_j(x)}{2(x_{j+1} - x_j)} - \frac{\phi_j(x) - \phi_{j-1}(x))}{2(x_j - x_{j-1})}. \tag{2.3}$$

The operator L_D is introduced as follows [3, 4] :

$$(L_D f)(x) = f_0\alpha_0(x) + f_1\alpha_1(x) + \sum_{j=2}^{n-2} f_j\psi_j(x) + f_{n-1}\alpha_{n-1}(x) + f_n\alpha_n(x), \tag{2.4}$$

where

$$\alpha_0(x) = \frac{1}{2} + \frac{\phi_1(x) - (x - x_0)}{2(x_1 - x_0)},$$

$$\alpha_1(x) = \frac{\phi_2(x) - \phi_1(x)}{2(x_2 - x_1)} - \frac{\phi_1(x) - (x - x_0)}{2(x_1 - x_0)},$$

$$\alpha_{n-1}(x) = \frac{(x_n - x) - \phi_{n-1}(x)}{2(x_n - x_{n-1})} - \frac{\phi_{n-1}(x) - \phi_{n-2}(x)}{2(x_{n-1} - x_{n-2})},$$

$$\alpha_n(x) = \frac{1}{2} + \frac{\phi_{n-1}(x) - (x_n - x)}{2(x_n - x_{n-1})},$$

$$\phi_j(x) = \sqrt{(x - x_j)^2 + c^2}, \quad j = 1, ..., n-1, \quad c \in R,$$

$$\psi_j(x) = \frac{\phi_{j+1}(x) - \phi_j(x)}{2(x_{j+1} - x_j)} - \frac{\phi_j(x) - \phi_{j-1}(x)}{2(x_j - x_{j-1})}, \quad j = 2, ..., n-2.$$

Let

$$\phi_{-1}(x) = |x - x_{-1}|, \quad \phi_0(x) = |x - x_0|, \quad \phi_n(x) = |x - x_n|, \quad \phi_{n+1}(x) = |x - x_{n+1}|$$

then $(L_D f)(x)$ can be written as follows:

$$(L_D f)(x) = \sum_{j=0}^{n} f_j\psi_j(x).$$

Quasi-interpolation $f^*(x)$ has the following properties.

Theorem 2.1. *[4], If*

$$\begin{aligned} \phi_{-1}(x) &= \phi_0(x) + x_0 - x_{-1}, \\ \phi_n(x) &= \phi_0(x) - 2x + x_0 + x_n, \\ \phi_{n+1}(x) &= \phi_n(x) + x_{n+1} - x_n \end{aligned} \tag{2.5}$$

then the multiquadric quasi-interpolation $f^*(x)$ can be written as three equivalent forms as follows:

$$f^*(x) = \frac{1}{2}\sum_{j=1}^{n-1}\left(\frac{\phi_{j+1}(x) - \phi_j(x)}{x_{j+1} - x_j} - \frac{\phi_j(x) - \phi_{j-1}(x)}{x_j - x_{j-1}}\right)f_j$$

$$+\frac{1}{2}\left(1 + \frac{\phi_1(x) - \phi_0(x)}{x_1 - x_0}\right)f_0 + \frac{1}{2}\left(1 - \frac{\phi_n(x) - \phi_{n-1}(x)}{x_n - x_{n-1}}\right)f_n,$$

$$f^*(x) = \frac{f_0 + f_n}{2} + \frac{1}{2}\sum_{j=0}^{n-1}\frac{\phi_j(x) - \phi_{j+1}(x)}{x_{j+1} - x_j}(f_{j+1} - f_j),$$

$$f^*(x) = \frac{1}{2}\sum_{j=1}^{n-1}(\frac{f_{j+1}-f_j}{x_{j+1}-x_j} - \frac{f_j - f_{j-1}}{x_j - x_{j-1}})\phi_j(x)$$

$$+\frac{f_0+f_n}{2} + \frac{f_1 - f_0}{2(x_1 - x_0)}\phi_0(x) - \frac{f_n - f_{n-1}}{2(x_n - x_{n-1})}\phi_n(x).$$

In addition on $[x_0, x_n]$, we have

$$(f^*(x))^{(k)} = \frac{1}{2}\sum_{j=0}^{n-1}\frac{\phi_j^{(k)} - \phi_{j+1}^{(k)}}{x_{j+1}-x_j}(f_{j+1}-f_j)$$

where, $x_{-1} < x_0$ and $x_n < x_{n+1}$. With the Eqs. (2.5) and the definition of $\phi_j(x)$, $\psi_j(x)$, we can say that the quasi-interpolation $f^*(x)$ defined by Eq. (2.2) is just the multiquadric quasi-interpolatin which we use in this paper.

Theorem 2.2. *[4]. Let*

$$h = \max\{x_j - x_{j-1}\}, \quad 1 \le j \le n.$$

For any real number $c > 0$, $x \in [x_0, x_n]$ and function $f(x) \in C^2(x_0, x_n)$, the multiquadric quasi-interpolation L_D satisfies:

$$\|(L_D)(x) - f(x)\|_\infty \le k_1 h^2 + k_2 ch + k_3 c^2 \log h,$$

where k_1, k_2, k_3 are constants independent of h and c.

3 Mathematical formulation for MQ quasi-interpolation

As [16], we can present the numerical method for solving the KE . The KE is as follows:

$$u_t + \alpha u u_x + \beta u_{3x} + \gamma u_{5x} = 0,$$

where α, β, γ are arbitrary constants. u_j^n is the approximation of the value of $u(x,t)$ at point (x_j, t_n), $t_n = n\tau$, τ is time step. We approximate u_t with $\frac{u_j^{n+1}-u_j^n}{\tau}$ so we get:

$$\frac{u_j^{n+1} - u_j^n}{\tau} \simeq -\alpha u_j^n (u_x)_j^n - \beta (u_{3x})_j^n - \gamma (u_{5x})_j^n.$$

Clearly we have:

$$u_j^{n+1} = u_j^n - \alpha\tau(u_j^n)(u_x)_j^n - \beta\tau(u_{3x})_j^n - \gamma\tau(u_{5x})_j^n.$$

The above equation means that the value of u can be obtained in time step $(n+1)$ according to time step (n). So, it dose not need to solve a system of equations. According to Theorem (2.1), the approximate values of u_x, u_{3x} and u_{5x} can be obtained. The derivative of the multiquadric quasi-interpolation to approximate u_x, u_{3x} and u_{5x} is used. And also we have:

$$\begin{aligned}
(u_x)_j^n &= \tfrac{1}{2}\textstyle\sum_{m=0}^{n-1} \frac{\phi_m^{(1)}(x_j)-\phi_{m+1}^{(1)}(x_j)}{x_{m+1}-x_m}(u(x_{m+1},t_n) - u(x_m,t_n)),\\
(u_{3x})_j^n &= \tfrac{1}{2}\textstyle\sum_{m=0}^{n-1} \frac{\phi_m^{(3)}(x_j)-\phi_{m+1}^{(3)}(x_j)}{x_{m+1}-x_m}(u(x_{m+1},t_n) - u(x_m,t_n)),\\
(u_{5x})_j^n &= \tfrac{1}{2}\textstyle\sum_{m=0}^{n-1} \frac{\phi_m^{(5)}(x_j)-\phi_{m+1}^{(5)}(x_j)}{x_{m+1}-x_m}(u(x_{m+1},t_n) - u(x_m,t_n)).
\end{aligned} \tag{3.6}$$

4 MQ interpolating operator using Hermite interpolating polynomial

In this section, first, the quasi-interpolation $L_{H_{2m-1}}$ is recalled [17] and the problem operators $L_{H_{2m-1}}$ for solving KE is expressed. This defect is removed by combining operators L_D and $L_{H_{2m-1}}$. The quasi-interpolation operator L_B is defined as follows [2]:

$$(L_B f)(x) = f(x_0)\psi_0(x) + \sum_{i=1}^{n-1} f(x_i)\psi_i(x) + f(x_n)\psi_n(x), \quad x \in [a,b], \tag{4.7}$$

where

$$\begin{aligned} \psi_0(x) &= \tfrac{1}{2} + \tfrac{\phi_1(x)-\phi_0(x)}{2(x_1-x_0)}, \\ \psi_n(x) &= \tfrac{1}{2} - \tfrac{\phi_n(x)-\phi_{n-1}(x)}{2(x_n-x_{n-1})}, \\ \psi_i(x) &= \tfrac{\phi_{i+1}(x)-\phi_i(x)}{2(x_{i+1}-x_i)} - \tfrac{\phi_i(x)-\phi_{i-1}(x)}{2(x_i-x_{i-1})}, \end{aligned} \tag{4.8}$$

and $i = 1, 2, \ldots, n-1$.
By combining L_B and Hermite interpolating polynomials [17] the improved quasi-interpolation operator is defined as follows:

$$(L_{H_{2m-1}} f)(x) = \sum_{i=0}^{n} \psi_i(x) H_{2m-1}[f; x_i, x_{i+1}](x), \tag{4.9}$$

where $H_{2m-1}[f; x_i, x_{i+1}](x)$ is Hermite interpolating polynomial of degree $2m-1$ which agrees with the function f at the points

$$\underbrace{x_i, x_i, \ldots, x_i}_{m}, \underbrace{x_{i+1}, x_{i+1}, \ldots, x_{i+1}}_{m}.$$

Also Hermite interpolating polynomial is defined as follows [18]:

$$H_{2m-1}(f(x)) = \sum_{i=0}^{m-1} r_i(x) f(x_i) + \sum_{i=0}^{m-1} s_i(x) f'(x_i), \tag{4.10}$$

where

$$\begin{aligned} r_i(x) &= (1 - 2(x - x_i)L_i'(x_i))(L_i(x))^2, \\ s_i(x) &= (x - x_i)(L_i(x))^2, \\ L_i(x) &= \tfrac{(x-x_0)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_{m-1})}{(x_i-x_0)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_{m-1})}. \end{aligned}$$

If we want to use Eq. (4.9) for solving (KE), $u_x(x_i, t)$, $i = 0, 1, \ldots n$, should be calculated in Eq. (4.10). But, as it is seen the values of $u_x(x, t)$ at $x = x_i$ are not available. To remove this defect, the bellow steps are done.
The (KE) is as follows:

$$u_t + \alpha u u_x + \beta u_{3x} + \gamma u_{5x} = 0,$$

where α, β, γ are arbitrary constants. u_j^n is the approximation of the value of $u(x,t)$ at point (x_j,t_n), $t_n = n\tau$, τ is time step. We approximate u_t with $\frac{u_j^{n+1}-u_j^n}{\tau}$ so we get:

$$\frac{u_j^{n+1}-u_j^n}{\tau} \simeq -\alpha u_j^n (u_x)_j^n - \beta(u_{3x})_j^n - \gamma(u_{5x})_j^n.$$

Clearly we have:

$$u_j^{n+1} = u_j^n - \alpha\tau(u_j^n)(u_x)_j^n - \beta\tau(u_{3x})_j^n - \gamma\tau(u_{5x})_j^n.$$

The above equation means that the value of u can be obtained in time step $(n+1)$ according to time step (n). This time the values of u_x, u_{3x} and u_{5x} are approximated by Hermite quasi-interpolation as bellow:

$$Lu(x,t) = \sum_{i=0}^{n} \psi_i(x) H_{2m-1}(u(x_i,t)),$$

$$H_{2m-1}(u(x,t)) = \sum_{i=0}^{m-1} r_i(x)u(x_i,t) + \sum_{i=0}^{m-1} s_i(x)(u_x(x,t))_{x=x_i},$$

where $u_x(x,t)$ is the derivative of $L_D(u(x,t))$ as follows:

$$(u_x(x,t))_{x=x_i} = \frac{\partial}{\partial x}[\frac{1}{2}\sum_{j=1}^{n-1}(\frac{u(x_{j+1},t)-u(x_j,t)}{x_{j+1}-x_j} - \frac{u(x_j,t)-u(x_{j-1},t)}{x_j-x_{j-1}})\phi_j(x) -$$

$$\frac{u(x_0,t)+u(x_n,t)}{2} + \frac{u(x_1,t)-u(x_0,t)}{2(x_1-x_0)}(x-x_0) -$$

$$\frac{u(x_n,t)-u(x_{n-1},t)}{2(x_n-x_{n-1})}(x_n-x)]_{x=x_i}.$$

So

$$u(x_j,t_{n+1}) = u(x_j,t_n) - \alpha\tau u(x_j,t_n)(Lu(x,t_n))^{(1)}_{x=x_j} - \beta\tau(Lu(x,t_n))^{(3)}_{x=x_j} - \gamma\tau(Lu(x,t_n))^{(5)}_{x=x_j}.$$

Remark 4.1. *[4], The formula of $(L_D f)(x)$ can be rewritten as:*

$$\begin{aligned}(L_D f)(x) &= \tfrac{1}{2}\textstyle\sum_{j=1}^{n-1}(\frac{f_{j+1}-f_j}{x_{j+1}-x_j} - \frac{f_j-f_{j-1}}{x_j-x_{j-1}})\phi_j(x) - \frac{f_0+f_n}{2} \\ &+ \tfrac{f_1-f_0}{2(x_1-x_0)}(x-x_0) - \tfrac{f_n-f_{n-1}}{2(x_n-x_{n-1})}(x_n-x).\end{aligned}$$

5 Numerical examples

In this section we give two examples to test the methods. All through this section we suppose $\alpha = \beta = 1$, $\gamma = -1$.

Example 5.1. *Consider the KE:*

$$u_t + uu_x + u_{3x} - u_{5x} = 0,$$

with initial condition

$$u(x,0) = -\frac{72}{169} + \frac{105}{169} sech^4(\frac{1}{2\sqrt{13}}x).$$

The authors of [5] obtain the exact solution of this equation as follows:

$$u(x,t) = -\frac{72}{169} + \frac{105}{169} sech^4(\frac{1}{2\sqrt{13}}(x + \frac{36}{169}t)).$$

According to $u_t \simeq \frac{u_j^{n+1} - u_j^n}{\tau}$ *we get:*

$$\frac{u_j^{n+1} - u_j^n}{\tau} \simeq -uu_x - u_{3x} + u_{5x}.$$

So we can write:

$$u(x_j, t_{n+1}) = u(x_j, t_n) + \tau(-u(x_j, t_n)u_x(x_j, t_n) - u_{3x}(x_j, t_n) + u_{5x}(x_j, t_n)).$$

In the above relation u_x, u_{3x} *and* u_{5x} *can be obtained in each time step with Eqs. (3.6). For example if we put* $t = 0.1$*, the values of* $u(x_j, 0.1)$ *can be obtained as follows:*

$$u(x_j, 0.1) = u(x_j, 0) + 0.0001(-u(x_j, 0)u_x(x_j, 0) - u_{3x}(x_j, 0) + u_{5x}(x_j, 0)),$$

where

$$u_x(x_j, 0) = \frac{1}{2}\sum_{m=0}^{n-1} \frac{\phi^{(1)}(x_j, m) - \phi^{(1)}(x_j, m+1)}{x_{m+1} - x_m}(u(x_{m+1}, 0) - u(x_m, 0)),$$

$$u_{3x}(x_j, 0) = \frac{1}{2}\sum_{m=0}^{n-1} \frac{\phi^{(3)}(x_j, m) - \phi^{(3)}(x_j, m+1)}{x_{m+1} - x_m}(u(x_{m+1}, 0) - u(x_m, 0)),$$

$$u_{5x}(x_j, 0)) = \frac{1}{2}\sum_{m=0}^{n-1} \frac{\phi^{(5)}(x_j, m) - \phi^{(5)}(x_j, m+1)}{x_{m+1} - x_m}(u(x_{m+1}, 0) - u(x_m, 0)).$$

Similarly the values of $u(x_j, 0.2)$ *can be obtained according to the following:*

$$u(x_j, 0.2) = u(x_j, 0.1) + 0.0001(-u(x_j, 0.1)u_x(x_j, 0.1) - u_{3x}(x_j, 0.1) + u_{5x}(x_j, 0.1)),$$

and so on. The comparison of the exact and numerical solution is shown in Table 1. We show the results at t=0.1 in Fig. 1.

Table 1

Comparison of results at $t = 0.1$ for MQ.

x	*Exact solution*	*Numerical solution*	*Absolute error*
0.1	0.194915	0.195027	0.000112515
0.2	0.194097	0.194311	0.000224214
0.3	0.192805	0.19312	0.000315175
0.4	0.191042	0.191457	0.000415287
0.5	0.188812	0.189326	0.000514287
0.6	0.186121	0.186733	0.000611914
0.7	0.182977	0.183685	0.000707918
0.8	0.179389	0.180191	0.000802096
0.9	0.175364	0.176261	0.000897298
1	0.170914	0.171898	0.000983632

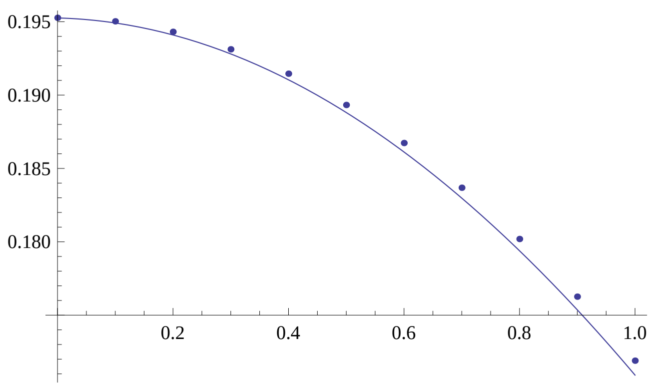

Fig. 1. Comparison of numerical solution $(\cdots)$, and exact solution $(---)$ of example (5.1) for t=0.1.

In Table 2, we compare the absolute error at different times for example (5.1).

Table 2
Absolute error of MQ at different times.

x	*t=0.1*	*t=0.2*	*t=0.3*	*t=0.4*
-25	1.11516×10^{-7}	2.24354×10^{-7}	3.38529×10^{-7}	4.54058×10^{-7}
30	6.912×10^{-9}	1.37429×10^{-8}	2.04936×10^{-8}	4.54058×10^{-7}
40	2.69876×10^{-11}	5.36582×10^{-11}	8.00156×10^{-11}	1.06063×10^{-10}

In Table 3, the values of $u(x,t)$ is calculated at $t = 0.1$ by Hermite quasi-interpolation as bellow:

$$u(x_j, 0.1) = u(x_j, 0) + 0.0001(-u(x_j, 0)u_x(x_j, 0) - u_{3x}(x_j, 0) + u_{5x}(x_j, 0)),$$

where

$$u_x(x_j, 0) = \frac{\partial}{\partial x}[\sum_{i=0}^{n} \psi_i(x) H_{2m-1}(u(x_i, 0))]_{x=x_j},$$

$$u_{3x}(x_j, 0) = \frac{\partial^3}{\partial x^3}[\sum_{i=0}^{n} \psi_i(x) H_{2m-1}(u(x_i, 0))]_{x=x_j},$$

$$u_{5x}(x_j,0)=\frac{\partial^5}{\partial x^5}[\sum_{i=0}^{n}\psi_i(x)H_{2m-1}(u(x_i,0))]_{x=x_j}.$$

And

$$H_{2m-1}(u(x,0))=\sum_{i=0}^{m-1}r_i(x)u(x_i,0)+\sum_{i=0}^{m-1}s_i(x)(u_x(x,0))_{x=x_i},$$

where

$$(u_x(x,0))_{x=x_i}=\frac{\partial}{\partial x}[\frac{1}{2}\sum_{j=1}^{n-1}(\frac{u(x_{j+1},0)-u(x_j,0)}{x_{j+1}-x_j}-\frac{u(x_j,0)-u(x_{j-1},0)}{x_j-x_{j-1}})\phi_j(x)-$$
$$\frac{u(x_0,0)+u(x_n,0)}{2}+\frac{u(x_1,0)-u(x_0,0)}{2(x_1-x_0)}(x-x_0)-$$
$$\frac{u(x_n,0)-u(x_{n-1},0)}{2(x_n-x_{n-1})}(x_n-x)].$$

Similarly

$$u(x_j,0.2)=u(x_j,0.1)+0.0001(-u(x_j,0.1)u_x(x_j,0.1)-u_{3x}(x_j,0.1)+u_{5x}(x_j,0.1)),$$

where

$$u_x(x_j,0.1)=\frac{\partial}{\partial x}[\sum_{i=0}^{n}\psi_i(x)H_{2m-1}(u(x_i,0.1))]_{x=x_j},$$
$$u_{3x}(x_j,0.1)=\frac{\partial^3}{\partial x^3}[\sum_{i=0}^{n}\psi_i(x)H_{2m-1}(u(x_i,0.1))]_{x=x_j},$$
$$u_{5x}(x_j,0.1)=\frac{\partial^5}{\partial x^5}[\sum_{i=0}^{n}\psi_i(x)H_{2m-1}(u(x_i,0.1))]_{x=x_j}.$$

And

$$H_{2m-1}(u(x,0.1))=\sum_{i=0}^{m-1}r_i(x)u(x_i,0.1)+\sum_{i=0}^{m-1}s_i(x)(u_x(x,0.1))_{x=x_i},$$

where

$$(u_x(x,0.1))_{x=x_i}=\frac{\partial}{\partial x}[\frac{1}{2}\sum_{j=1}^{n-1}(\frac{u(x_{j+1},0.1)-u(x_j,0.1)}{x_{j+1}-x_j}-\frac{u(x_j,0.1)-u(x_{j-1},0.1)}{x_j-x_{j-1}})\phi_j(x)-$$
$$\frac{u(x_0,0.1)+u(x_n,0.1)}{2}+\frac{u(x_1,0.1)-u(x_0,0.1)}{2(x_1-x_0)}(x-x_0)-$$
$$\frac{u(x_n,0.1)-u(x_{n-1},0.1)}{2(x_n-x_{n-1})}(x_n-x)],$$

and so on.

Table 3
Comparison of results at $t=0.1$ for Hermite quasi-interpolation.

x	*Exact solution*	*Numerical solution*	*Absolute error*
0.1	*0.194915*	*0.1945*	*0.000414314*
0.2	*0.194097*	*0.194765*	*0.000667334*
0.3	*0.192805*	*0.193127*	*0.000321756*
0.4	*0.191042*	*0.191459*	*0.000417559*
0.5	*0.188812*	*0.189326*	*0.000514435*

In Table 4, the values of $u(x,t)$ are calculated at $t = 0.2$ by Hermite quasi-interpolation. In other time steps the same thing is done.

Table 4
Comparison of results at $t = 0.2$ for Hermite quasi-interpolation.

x	*Exact solution*	*Numerical solution*	*Absolute error*
0.1	*0.194781*	*0.193924*	*0.000856121*
0.2	*0.193862*	*0.195259*	*0.00139732*
0.3	*0.192469*	*0.193135*	*0.000666496*
0.4	*0.195606*	*0.191462*	*0.000856056*
0.5	*0.188277*	*0.189327*	*0.00104978*

Example 5.2. *Consider the KE:*

$$u_t + uu_x + u_{3x} - u_{5x} = 0,$$

with initial condition

$$u(x,0) = -\frac{72}{169} + \frac{420sech^2(\frac{1}{2\sqrt{13}}x)}{169(1 + sech^2(\frac{1}{2\sqrt{13}}x))}.$$

The authors of [5] obtained the exact solution of this equation as follows:

$$u(x,t) = -\frac{72}{169} + \frac{420sech^2(\frac{1}{2\sqrt{13}}(x + \frac{36}{169}))}{169(1 + sech^2(\frac{1}{2\sqrt{13}}(x + \frac{36}{169})))}.$$

The comparison of the exact and numerical solution is shown in Table 5. We show the results at t=0.1 in Fig. 2.

Table 5
Comparison of results at $t = 0.1$ of MQ.

x	*Exact solution*	*Numerical solution*	*Absolute error*
0.1	*0.816392*	*0.816449*	*0.000056432*
0.2	*0.815983*	*0.816091*	*0.000107586*
0.3	*0.815335*	*0.815494*	*0.000158618*
0.4	*0.814449*	*0.814658*	*0.000209586*
0.5	*0.813324*	*0.813584*	*0.000260473*
0.6	*0.811962*	*0.812273*	*0.000311259*
0.7	*0.810362*	*0.810724*	*0.000361924*
0.8	*0.808526*	*0.808939*	*0.000412469*
0.9	*0.806454*	*0.806919*	*0.000464483*
1	*0.804147*	*0.804659*	*0.000512132*

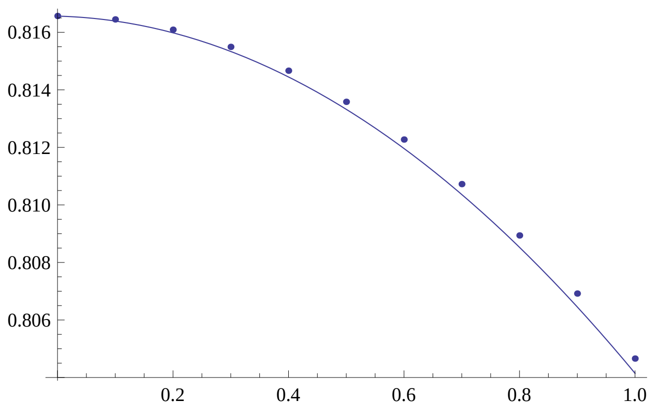

Fig. 2. Comparison of numerical solution $(\cdots)$*, and exact solution* $(---)$ *of example (5.2) for t=0.1.*

In Table 6 we compare the absolute error in different times for example (5.2).

Table 6
Absolute error of MQ in different times.

x	*t=0.1*	*t=0.2*	*t=0.3*	*t=0.4*
-25	*0.0000567258*	*0.000113784*	*0.000171176*	*0.000228904*
30	*0.0000142158*	*0.0000283481*	*0.0000423974*	*0.0000563642*
40	8.90104×10^{-7}	1.77497×10^{-6}	2.65462×10^{-6}	3.52909×10^{-6}

In Table 7. the values of $u(x,t)$ *is calculated at* $t = 0.1$ *and different* x *by Hermite quasi-interpolation.*

Table 7
Comparison of results at $t = 0.1$ *for Hermite quasi-interpolation.*

x	*Exact solution*	*Numerical solution*	*Absolute error*
0.1	*0.816392*	*0.816078*	*0.000314623*
0.2	*0.815983*	*0.816318*	*0.000335155*
0.3	*0.815335*	*0.815497*	*0.000161923*
0.4	*0.814449*	*0.814659*	*0.000210734*
0.5	*0.813324*	*0.813584*	*0.000260144*

In Table 8. the values of

$u(x,t)$ is calculated at $t = 0.2$ by Hermite quasi-interpolation. In other times steps the same thing is done.

Table 8
Comparison of results at $t = 0.2$ for Hermite quasi-interpolation.

x	*Exact solution*	*Numerical solution*	*Absolute error*
0.1	*0.815681*	*0.816325*	*0.000643864*
0.2	*0.816561*	*0.815865*	*0.000695903*
0.3	*0.815502*	*0.815166*	*0.000335528*
0.4	*0.814661*	*0.814229*	*0.000432152*
0.5	*0.813585*	*0.813054*	*0.000531075*

6 Conclusion

In this paper multiquadric quasi-interpolation was used to solve the KE. The numerical solutions are compared with exact solutions in two examples. The results show that the MQ quasi-interpolation is a reasonable method to solve the KE. In this method calculations is convenient because we don't have to solve a system of equations. Then the defect of Hermit quasi-interpolation for solving KE is removed by using L_D. The results showed that this methods is reasonable too. In our work we use the Mathematica software to our calculates.

References

[1] S. Abbasbandy, Homotopy analysis method for the Kawahara equation, Nonlinear analysis, Real world application 11 (2010) 307-312.

[2] R. K. Beatson, M. J. D. powell, Univariate MQ approximation quasi-interpolation to scattered data, constr. Approx. 8 (1992) 275–288.

[3] Z. M. Wu, R. Schaback, Shape preserving properties and convergence of univariate multiquadric quasi-interpolation. Acta Math. Appl.Sin. 10 (1994) 441-446.

[4] R. Chen, X. Han, Z. Wu, A multiquadric quasi-interpolation with linear reproducing and preserving monotonicity, Journal of computational and applied mathematics. 231 (2009) 517-525.

[5] M. Matinfar, H. Hosseinzadeh, M. Ghanbari, Exact and numerical solution of Kawahara equation by the variational iteration method. Applied mathematical sciences 2 (2008) 2119-2126.

[6] R. L. Hardy, Multiquadric equation of topography and other irregular surfaces, Geophys.Res. 76 (1971) 1905-1915.

[7] R. L. Hardy, Theory and application of the multiquadric-biharmonic method, 20 years of discovery 1968-1988, Math. Appl. 19 (1990) 163-208.

[8] E. J. Kansa, Multiquadrics-a scattered data approximation scheme with applications to computational fluid dynamics 1, Math. Appl. 19 (1990) 127-145.

[9] T. Shuangping, C. Shuangbin, Existence and uniqueness of Soulotion to nonlinear Kawahara equation, Chinese Annals of Mathematics, Series A. 23 (2002) 221-228.

[10] E. Fan, Extented tanh-function method and its applications to nonlinear equations, Physics Letters A. 277 (2002) 208-112.

[11] T. Kawahara, Oscillatory solitary waves in dispersive media, Journal of the Physical Society of Japan 33 (1972) 260-264.

[12] N. Berloff, L. Howard, Solitary and periodic soluyion of nonlinear nonintegrable equation, Studies in Applied Mathematics. 99 (1997) 1-24.

[13] A. Darwish, A. Ramady, Applications of algebraic method to exactly solve some nonlinear partial differential equation, SChaos, Solitons and Fractals 33 (2007) 1263-1274.

[14] H. Zhang, A direct algebraic method applied to obtain complex solution of some nonlinear partial differential equations, Chaos, Solitons and Fractals, 2007.

[15] H. Zhang, New exact travelling wave solutions to the complex coupled Kawahara equation and modified Kawahara equation, Communications in Nonlinear Science and numerical Simulation, 2007.

[16] R. Chen, Z. Wu, Applying multiquadric quasi-interpolation to solve Burger's equation, Applied Mathematics and Computation 172 (2006) 427-484.

[17] R. Wang, Q. Fan, M. Xu, A kind of improved univariate multiquadric quasi-interpolation operators, Computers and Mathematics with Application 59 (2010) 451-456

[18] J. Stoer, Introduction to Numerical Analysis, Springer-Verlage, 2002.