Numerical Solutions of the General Rosenau—RLW Equation Using Meshless Kernel Based Method of Lines

Numerical Solutions of the General Rosenau–RLW Equation Using Meshless Kernel Based Method of Lines

Murat Arı, Yılmaz Dereli
Anadolu University, Mathematics Department, 26470
Eskişehir, Turkey
E-mail: murat_ari@anadolu.edu.tr, ydereli@anadolu.edu.tr

Abstract. In this paper, the General Rosenau–RLW equation is solved by using meshless kernel based method of lines. To test the accuracy of the method some numerical experiments are presented. Obtained numerical results are compared with some earlier works. It is seen that the method is very efficient and reliable due to obtained numerical results are very satisfactory.

1. Introduction

The General Rosenau RLW equation has following form

$$u_t - u_{xxt} + u_{xxxxt} + u_x + (u^p)_x = 0 \tag{1}$$

where $p \geq 2$ is an integer.

If $p = 2$, then Eq. (1) is called the usual Rosenau–RLW equation, [1, 2]. Moreover, if $p = 3$, then Eq. (1) is called the modified Rosenau–RLW equation and if $p \geq 4$ then it is called as the General Rosenau–RLW equation.

It is known that the solitary wave solution os Eq.(1) is

$$\begin{aligned} u(x,t) = e^{ln\{(p+3)(3p+1)(p+1)/[2(p^2+3)(p^2+4p+7)]\}/(p+1)} \\ \times \operatorname{sech}^{4/(p+1)}[\frac{(p-1)}{\sqrt{4p^2+8p+20}}(x-ct)] \end{aligned} \tag{2}$$

where $p \geq 2$ is an integer and

$$c = (p^4 + 4p^3 + 14p^2 + 20p + 25)/(p^4 + 4p^3 + 10p^2 + 12p + 21).$$

In the literature there a lot of studies about the Rosenau RLW equation see also references [3] - [10]. In this study as different from other publications evaluated numerical solutions of the General Rosenau–RLW equation by using the meshless kernel based method of lines (MKBMOL) will be presented. In this paper, we consider the following initial–boundary value problem of the general Rosenau–RLW equation with an initial condition:

$$u(x,0) = u_0(x) = 0, x_l \leq x \leq x_r \tag{3}$$

and boundary conditions

$$u(x_l, t) = u(x_r, t) = 0, \tag{4}$$

The initial–boundary value problem possesses the following conservative properties:

$$Q(t) = \int_{x_l}^{x_r} u(x,t)dx = \int_{x_l}^{x_r} u_0(x,0)dx = Q(0),$$

and

$$E(t) = \|u\|_{L_2}^2 + \|u_x\|_{L_2}^2 + \|u_{xx}\|_{L_2}^2 = E(0).$$

When $-x_l \gg 0$ and $x_r \gg 0$, the initial boundary value problem (1)-(3) is consistent, so the boundary condition (3) is reasonable.

2. Application of the MKBMOL

In this section, the MKBMOL will be used to obtain the numerical solution of the Eq. (1) with initial and boundary conditions (3) and (4). In the method, time discretization isn't necessary and radial basis functions are used as kernel function. The method of lines approximates the solution u by a linear combination [11]

$$u(x,t) = \sum_{j=1}^{N} \alpha_j(t)\upsilon_j(x), \tag{5}$$

where $\alpha_j(t)$ is unknown and $\upsilon_j(x)$ is obtained by kernel functions such as Multiquadric [12], Gaussian or Wendland's [13] functions.
Multiquadric and Gaussian radial basis functions are defined as follows:

Multiquadric(MQ) $\quad \phi(r) = \sqrt{(\varepsilon r)^2 + 1}$

Gaussian(GA) $\quad \phi(r) = \exp(-r^2/\varepsilon^2)$

where $r = |x - x_j|$ is the Euclidean distance between collocation points x and x_j. The general form of compactly supported radial basis functions are defined as follows:

$$\phi_{l,k}(r) = (1-r)_+^n p(r) \tag{6}$$

where p is a prescribed polynomial for $k \geq 1$ with following conditions:

$$(1-r)_+^n = \begin{cases} (1-r)^n, & \text{if } 0 \leq r < 1 \\ 0, & \text{if } r \geq 1 \end{cases} \tag{7}$$

In our algorithms used following Wendland's functions are as follows:

$$\begin{aligned} \phi_{6,4}(r) &= (1-r)_+^{10}(5 + 50r + 210r^2 + 450r^3 + 429r^4), \\ \phi_{7,5}(r) &= (1-r)_+^{12}(9 + 108r + 566r^2 + 1644r^3 + 2697r^4 + 2048r^5). \end{aligned} \tag{8}$$

Derivatives in (1) with respect to time and space variables can be expressed as follows

$$u_t(x,t) = \sum_{j=1}^{N} \alpha_j'(t)\upsilon_j(x), \tag{9}$$

$$u_x(x,t) = \sum_{j=1}^{N} \alpha_j(t)\upsilon_j'(x), \tag{10}$$

$$u_{xxt}(x,t) = \sum_{j=1}^{N} \alpha_j'(t)\upsilon_j''(x), \tag{11}$$

$$u_{xxxxt}(x,t) = \sum_{j=1}^{N} \alpha_j'(t)\upsilon_j^{iv}(x), \tag{12}$$

Substituting the Eqs.(9-12) in to (1) we get

$$\begin{aligned}\sum_{j=1}^{N} \alpha_j'(t)\upsilon_j(x) + \sum_{j=1}^{N} \alpha_j'(t)\upsilon_j^{iv}(x) - \sum_{j=1}^{N} \alpha_j'(t)\upsilon_j''(x) + \sum_{j=1}^{N} \alpha_j(t)\upsilon_j'(x) \\ + [\sum_{j=1}^{N} \alpha_j(t)\upsilon_j'(x)]^p = 0\end{aligned} \tag{13}$$

The Eq.(13) can be written as follows

$$\sum_{j=1}^{N}(\upsilon_j(x) + \upsilon_j^{iv}(x) - \upsilon_j''(x))\alpha_j'(t) = -\sum_{j=1}^{N} \alpha_j(t)\upsilon_j(x) - [\sum_{j=1}^{N} \alpha_j(t)\upsilon_j'(x)]^p \tag{14}$$

The Eq.(13) is written as follows by using MATLAB notations

$$(V + V^{iv} - V'') * \alpha'(t) = -(V * \alpha(t)) - (V' * \alpha(t)). \wedge p \tag{15}$$

Also the Eq. (15) is rewritten as follows:

$$\alpha'(t) = -(V + V^{iv} - V'')^{-1}[(V * \alpha(t)) + (V' * \alpha(t)). \wedge p] \tag{16}$$

since matrices V and others are invertible. $V + V^{iv} - V''$ and vectors $\alpha(t)$ and $\alpha'(t)$ are defined as follows

$$\begin{aligned} &V := \upsilon_j(x_k) \\ &V' := \upsilon_j'(x_k) \\ &V'' := \upsilon_j''(x_k) \\ &V^{iv} = \upsilon_j^{iv}(x_k) \\ &\alpha(t) = [\alpha_1(t), \alpha_2(t), ...\alpha_n(t)]^T \\ &\alpha'(t) = [\alpha_1'(t), \alpha_2'(t), ...\alpha_n'(t)]^T \end{aligned} \tag{17}$$

for $1 \leq k \leq N$ and $1 \leq j \leq N$. Where the symbol "$*$" defines the pointwise product, the symbol "$. \wedge p$" defines the component-by-component multiplication of vectors. Eq.(16) is an ordinary differential equation with respect to $\alpha(t)$ and it is solved by using any ordinary differential equation solver in MATLAB.

3. Numerical examples

In this section, numerical experiments on a test problem to confirm and illustrate the accuracy of our proposed method will be presented. The accuracy of the method is measured by the root mean square error L_2 and maximum error L_∞. In the tables comparisons with numerical solutions references [9]-[10] are presented. Where L_2 and L_∞ are calculated as follows:

$$L_2 = \sqrt{h \sum_{j=1}^{N} \left| u_j^{exact} - u_j^{num.} \right|^2}, \quad L_\infty = \max_{1 \le j \le N} \left| u_j^{exact} - u_j^{num.} \right|.$$

The results in term of errors for different values of p at time $T = 40$ for $\Delta t = 0.1$ at the interval $(x_l, x_r) = (-60, 120)$ with $h = 0.5$ are reported in Tables 1-3. Obtained results with the MQ, G and Wendland's functions are compared with some previous results. The method by using MQ, G and Wendland's functions provides higher accuracy then compared with other numerical results in [9] and [10]. Especially method with the W(7,5) function provided least error and use of the MQ provided the highest errors between the used kernel functions interms of the error norms.

The invariants remain almost constant when compared with analytical values of invariants. Calculated numerical results are very satisfactorily. The motion of single solitary waves are plotted in Figures 1-3. The program is run up to time T=40 over the solution domain. Initially the peak of the solitary wave was positioned at $x = 0$ and at the end of running time location of peak position of the wave moves to the right-side with the preserved amplitude and shape while time increasing. In Figures 1-3, the motion of single solitary wave motion can be seen.

Table 1. Comparison of invariants and error norms with p=4 and h=0.5

Method	h	Δt	T	L_2	L_∞	Q	E
Analytical						6.265806	2.8676946
Wendland 7-5	0.5	0.1	40	0.00093510	0.00035069	6.266377	2.868226
Wendland 6-4	0.5	0.1	40	0.00002355	0.00001031	6.265844	2.867735
Gaussian	0.5	0.1	40	0.00066954	0.00029706	6.265806	2.867684
Multiquadric	0.5	0.1	40	0.00110452	0.00042250	6.265992	2.867617
[9]	0.5	0.1	40	0.00447881	0.00171122		
[10]	0.5	0.1	40	0.07451730	0.02787120		

Table 2. Comparison of invariants and error norms with p=8 and h=0.5

Method	h	Δt	T	L_2	L_∞	Q	E
Analytical						9.742086	4.735164
Wendland 7-5	0.5	0.1	40	0.00038078	0.00013784	9.742126	4.735346
Wendland 6-4	0.5	0.1	40	0.00007522	0.00002949	9.742181	4.735225
Gaussian	0.5	0.1	40	0.00170387	0.00062856	9.742146	4.735302
Multiquadric	0.5	0.1	40	0.00127623	0.00047892	9.742227	4.735082
[9]	0.5	0.1	40	0.00431841	0.00161891		
[10]	0.5	0.1	40	0.08037300	0.02953370		

Table 3. Comparison of invariants and error norms with p=16 and h=0.5

Method	h	Δt	T	L_2	L_∞	Q	E
Analytical						17.148841	8.375324
Wendland 7-5	0.5	0.1	40	0.00233340	0.00044109	17.168699	8.375376
Wendland 6-4	0.5	0.1	40	0.00231987	0.00044493	17.169258	8.375400
Gaussian	0.5	0.1	40	0.00302310	0.00053860	17.172776	8.375393
Multiquadric	0.5	0.1	40	0.00762184	0.00227095	17.116828	8.375272
[9]	0.5	0.1	40	0.00357250	0.00118750		
[10]	0.5	0.1	40	0.06130440	0.02254710		

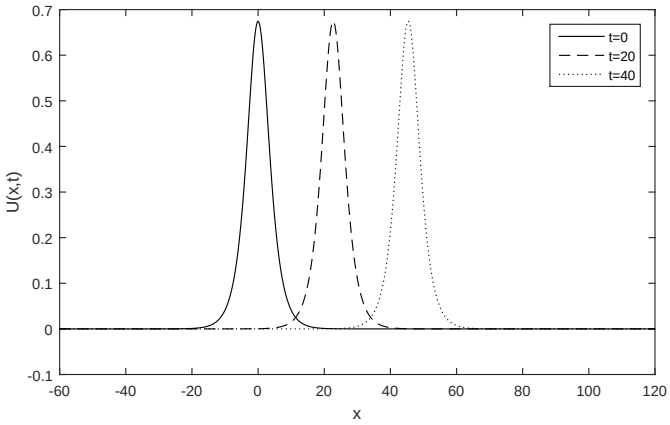

Figure 1. Motion of the single solitary wave for $p = 4$.

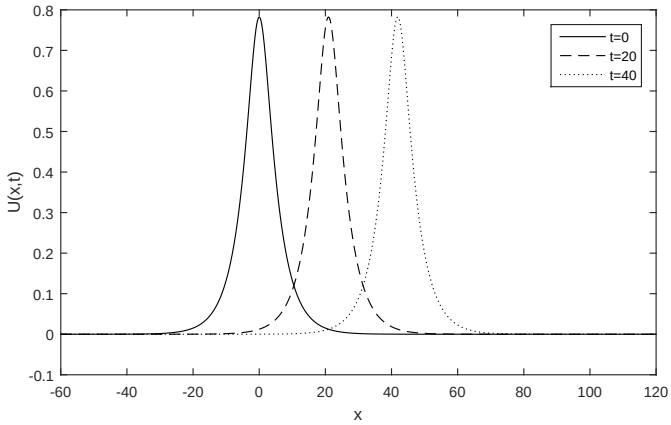

Figure 2. Motion of the single solitary wave for $p = 8$.

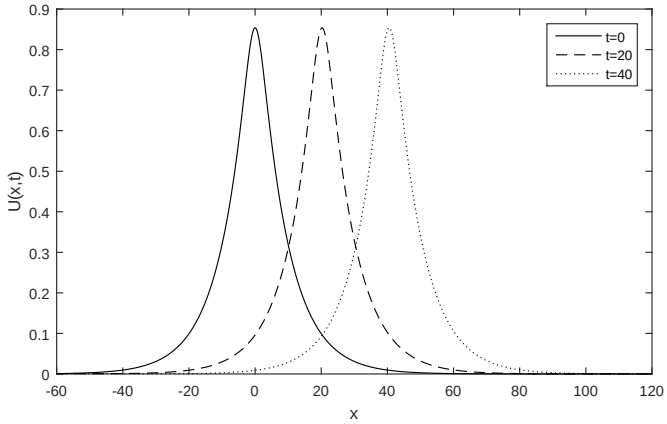

Figure 3. Motion of the single solitary wave for $p = 16$.

Conclusions

In this study the General Rosenau–RLW equation was solved by using a meshless technique which was MKBMOL. The main advantage of this technique there was no linearization to the nonlinear part of given equation. This equation was converted to a ordinary differential equation system and solved by using MATLAB codes. Here we present the numerical results of the Adams-Bashford multi-step method but any one can be present some numerical results by using other one step or multi step methods since MATLAB uses a ode solver package. From numerical results it is seen that the technique gives very reliable and sensitive result with all used kernel functions. Therefore it is said that this method is very reliable and applicable to this type equations. Also for different form of this type evaluation equations and systems this method can be applied at the future works.

References

[1] Rosenau P 1986 A quasi-continuous description of a nonlinear transmission line *Physica Scripta* **34** pp 827-29
[2] Rosenau P 1988 Dynamics of dense discrete systems *Progress of Theoretical Physics* **79** pp 1028-42
[3] Park M A 1990 On the Rosenau equation *Mathematica Aplicada e Computacional* **9** pp 145-52
[4] Park M A 1992 Pointwise decay estimate of solutions of the generalized Rosenau equation *Journal of the Korean Mathematical Society* **29** pp 26180
[5] Zuo J M Zhang Y M Zhang T D and Chang F 2010 A new conservative difference scheme for the general Rosenau-RLW equation *Boundary Value Problems* **2010** p 13
[6] Pan X Zhang L 2012 On the convergence of a conservative numerical scheme for the usual Rosenau-RLW equation *Applied Mathematical Modelling* **36** pp 3371-78
[7] Pan X Zheng K Zhang L 2013 Finite difference discretization of the Rosenau-RLW equation *Applicable Analysis* **92** pp 257889
[8] Atouani N Omrani K 2013 Galerkin finite element method for the Rosenau-RLW equation *Computers and Mathermatics with Applications* **66** pp 289-303
[9] Wongsaijai B Poochinapan K and Disyadej T 2014 A compact finite difference method for solving the general Rosenau-RLW equation *IAENG International Journal of Applied Mathematics* p 8
[10] X. Pan and L. Zhang 2012 Numerical simulation for general Rosenau-RLW equation: an average linearized conservative scheme *Mathematical Problems in Engineering* **2012** p 15
[11] Schaback, R. The Meshless Kernel-Based Method of Lines for Solving Nonlinear EvolutionEquations. Preprint, Gt-tingen, 2008.
[12] Rolland H L 1971 Multiquadric equations of topography and other irregular surfaces *J. Geophys. Res.* **176** pp 1905-15
[13] Wendland H 1995 Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree *Advances in Comput.Math.* **4** pp 389-96