Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 8, No. 1, 2020, pp. 155-172
DOI:10.22034/cmde.2019.9464

Meshless local radial point interpolation (MLRPI) to two dimensional wave equation with Neumann's boundary conditions

Elyas Shivanian*
Department of Applied Mathematics,
Imam Khomeini International University,
Qazvin, 34149-16818, Iran.
E-mail: shivanian@sci.ikiu.ac.ir

Mostafa Hosseini
Department of Applied Mathematics,
Imam Khomeini International University,
Qazvin, 34149-16818, Iran.
E-mail: smh.6747@yahoo.com

Asghar Rahimi
Department of Applied Mathematics,
Imam Khomeini International University,
Qazvin, 34149-16818, Iran.
E-mail: a.rahimi.mt@gmail.com

Abstract In this article, the meshless local radial point interpolation (MLRPI) methods are applied to simulate two-dimensional wave equation subject to given appropriate initial and Neumann's boundary conditions. In MLRPI method, all integrations are carried out locally over small quadrature domains of regular shapes such as square or circle. The radial point interpolation method is proposed to construct shape functions for MLRPI. A weak formulation with a Heaviside step function transforms the set of governing equations into local integral equations on local sub domains where Neumann's boundary condition is imposed naturally. A two-step time discretization method with the help of Crank-Nicolson technique is employed to approximate the time derivatives. Convergence studies in the numerical example show that MLRPI method possesses excellent rates of convergence.

Keywords. Meshless local radial point interpolation (MLRPI), Local weak formulation, Radial basis function, 2-D wave equation, Neumanns boundary conditions, Finite difference.

2010 Mathematics Subject Classification. 65M99.

1. Introduction

The wave equations display vibrations of structures such as buildings, beams and machines, and more are the basis for fundamental equations of atomic physics. The wave equation usually depicts water waves, the vibrations of a string, the propagation of sound waves, and the transmission of electric signals in a cable, etc. The typical model of wave equation is an initial boundary value problem being valid in a bounded

Received: 06 May 2018 ; Accepted: 23 September 2018.

* Corresponding author.

domain or an initial value problem being valid in an unbounded domain. It is worth to mention that two initial conditions should be prescribed, namely the initial displacement $u\left(\mathbf{x},0\right)=\varphi\left(\mathbf{x}\right)$ and the initial velocity $u_t\left(\mathbf{x},0\right)=\psi(\mathbf{x})$ which describes the initial displacement and the initial velocity at the starting time, respectively. Unlike the heat equation, the wave equation includes the term u_{tt} which represents the vertical acceleration at point $\mathbf{x}$. The wave equation plays an important role in various physical problems and it is required in diverse areas of science and engineering [11, 15, 18, 27, 28, 29, 31, 32].

Consider the following two-space dimensional hyperbolic partial differential equation

$$\frac{\partial^2 u}{\partial t^2}+\alpha\frac{\partial u}{\partial t}+\beta u=\Delta u+f\left(\mathbf{x}\right),\quad \mathbf{x}\in\mathit{\Omega}=\left[0,1\right]^2, \tag{1.1}$$

subject to the initial conditions

$$u\left(\mathbf{x},0\right)=\varphi\left(\mathbf{x}\right),\quad \frac{\partial u}{\partial t}\left(\mathbf{x},0\right)=\psi\left(\mathbf{x}\right), \tag{1.2}$$

and the Neumann's boundary condition

$$\frac{\partial u}{\partial n}\left(\mathbf{x},t\right)=0,\qquad \mathbf{x}\in\partial\mathit{\Omega}. \tag{1.3}$$

The main shortcoming of mesh-based methods such as the finite element method (FEM), the finite volume method (FVM) and the boundary element method (BEM) are that these numerical methods rely on meshes or elements. In order to overcome the mentioned difficulties some techniques so-called meshless methods have been proposed recently [22]. Overcoming mesh based methods difficulties is an advantages of meshless methods which encouraged many researchers to work on different areas of applied mathematics based on these methods like integral equations[1, 2, 3] as good as differential equations. There are three types of meshless methods: Meshless methods based on weak forms such as the element free Galerkin (EFG) method [5, 6], meshless methods based on collocation techniques (strong forms) such as the meshless collocation method based on radial basis functions (RBFs) [14, 19] and meshless methods based on the combination of weak forms and collocation technique.

In the literature, several meshless weak form methods have been reported such as: diffuse element method (DEM) [26], smooth particle hydrodynamic (SPH) [7, 9], the reproducing kernel particle method (RKPM)[23], boundary node method (BNM) [25], partition of unity finite element method (PUFEM) [24], finite sphere method (FSM) [12], boundary point interpolation method (BPIM) [17] and boundary radial point interpolation method (BRPIM) [10, 16]. G.R. Liu applied the concept of MLPG and developed meshless local radial point interpolation (MLRPI) method [13, 20, 21, 30].

It is notable using radial basis functions have been shown to be most useful. Especially thin-plate splines, because of their ability to approximate locally, could help to improve the accuracy of solving hyperbolic equations [8]. Some basic formulations and properties of TPSs and the error estimate for them on the Sobolev spaces have been investigated in [4]. In this paper, MLRPI method is applied to the problem (1.1)-(1.3). We give an example to show that MLRPI method possesses excellent rates of convergence. For numerical results the famous MATLAB package R2013b is used.

2. Point interpolation using radial basis functions

Using polynomials as basis functions in the interpolation is one of the earliest interpolation schemes. Consider a continuous function $u(\mathbf{x})$ defined in a domain Ω, which is represented by a set of field nodes. The $u(\mathbf{x})$ at a point of interest $\mathbf{x}$ is approximated in the form:

$$u\left(\mathbf{x}\right)=\sum_{i=1}^{m}p_i(\mathbf{x})a_i=\left\{p_1\left(\mathbf{x}\right)p_2\left(\mathbf{x}\right)\ldots p_m\left(\mathbf{x}\right)\right\}\left\{\begin{matrix}a_1\\a_2\\\vdots\\a_m\end{matrix}\right\}=\mathbf{P}^{\mathrm{T}}\mathbf{a,}\tag{2.1}$$

where $p_i(\mathbf{x})$ is the given monomial in the polynomial basis function in the space coordinate $\mathbf{x}^{\mathrm{T}}=[x,y]$, m is the number of monomials, and a_i is the coefficient for $p_i(\mathbf{x})$ which is yet to be determined. The $p_i(\mathbf{x})$ in Eq. (2.1) is built using Pascal's triangle and a complete basis is usually preferred. For the two-dimensional (2-D) space, the linear basis functions are given by

$$\mathbf{P}^{T}\left(\mathbf{x}\right)=\left\{1,x,y\right\},\qquad m=3,\tag{2.2}$$

the quadratic basis functions are

$$\mathbf{P}^{T}\left(\mathbf{x}\right)=\left\{1,x,y,x^2,y^2,xy\right\},\qquad m=6,\tag{2.3}$$

and the cubic basis functions are presented as

$$\mathbf{P}^{T}\left(\mathbf{x}\right)=\left\{1,x,y,x^2,y^2,xy,x^3,y^3,x^2y,xy^2\right\},\qquad m=10.\tag{2.4}$$

In order to determine the coefficients a_i, a support domain is formed for the point of interest at $\mathbf{x}$, with a total of n field nodes included in the support domain. Note that in the conventional PIM, the number of nodes in the local support domain always equals the number of basis functions of m, i.e., $n=m$. The coefficients a_i in Eq. (2.1) can then be determined by enforcing $u\left(\mathbf{x}\right)$ in Eq. (2.1) to pass through the nodal values at these n nodes. This yields n equations with each for one node, i.e.,

$$\left\{\begin{array}{c}u_1=\sum_{i=1}^{m}a_ip_i\left(\mathrm{x}_1\right)=a_1+a_2x_1+a_3y_1+\cdots+a_mp_m\left(\mathrm{x}_1\right),\\[2ex]u_2=\sum_{i=1}^{m}a_ip_i\left(\mathrm{x}_2\right)=a_1+a_2x_2+a_3y_2+\cdots+a_mp_m\left(\mathrm{x}_2\right),\\\vdots\\u_n=\sum_{i=1}^{m}a_ip_i\left(\mathrm{x}_n\right)=a_1+a_2x_n+a_3y_n+\cdots+a_mp_m\left(\mathrm{x}_n\right),\end{array}\right.\tag{2.5}$$

which can be written in the following matrix form:

$$\mathbf{U}_{\mathrm{s}}=\mathbf{P}_{\mathrm{m}}\mathbf{a},\tag{2.6}$$

where

$$\mathbf{U}_{\mathrm{s}}=\left\{u_1\quad u_2\quad u_3\ldots u_n\right\}^{T}\tag{2.7}$$

is the vector of function values, and

$$\mathbf{a} = \{a_1 \quad a_2 \quad a_3 \ \ \ldots \ \ \ a_n\}^T \tag{2.8}$$

is the vector of unknown coefficients, and

$$\mathbf{P}_{\mathrm{m}}= \begin{bmatrix} 1 & x_1 y_1 & \ldots & p_m\left(\mathbf{x}_1\right) \\ 1 & x_2 y_2 & \ldots & p_m\left(\mathbf{x}_2\right) \\ 1 & x_3 y_3 & \ldots & p_m\left(\mathbf{x}_3\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n y_n & \ldots & p_m\left(\mathbf{x}_n\right) \end{bmatrix} \tag{2.9}$$

is the so-called moment matrix. Because of $n = m$ in PIM, P_{m} is hence a square matrix with the dimension of $(n \times n \ or \ m \times m)$. By solving Eq. (2.6) for **a**, we obtain

$$\mathbf{a}{=}\mathbf{P}_{\mathrm{m}}^{-1}\mathbf{U}_{\boldsymbol{s}}. \tag{2.10}$$

In obtaining the foregoing equations, we have assumed that $\mathbf{P}_m^{-1}$ exists. It is noted that coefficients **a** are constant even if the point of interest at **x** changes, as long as the same set of n nodes are used in the interpolation, because P_m is a matrix of constants for this given set of nodes. Substituting Eq. (2.10) back into Eq. (2.1) and considering $n = m$ yields

$$u\left(\mathbf{x}\right) = \mathbf{P}^T\left(\mathbf{x}\right)\mathbf{P}_m^{-1}\mathbf{U}_s = \sum_{i=1}^{n}\phi_i u_i = \mathbf{\Phi}^{\mathbf{T}}\left(\mathbf{x}\right)\mathbf{U}_{\mathrm{s}}, \tag{2.11}$$

where $\mathbf{\Phi}^{\mathbf{T}}\left(\mathbf{x}\right)$ is a vector of shape functions defined by

$$\mathbf{\Phi}^{\mathbf{T}}\left(\mathbf{x}\right){=}\mathbf{P}^T\left(\mathrm{x}\right)\mathbf{P}_m^{-1} = \{\phi_1\left(\mathbf{x}\right) \ \ \phi_2\left(\mathbf{x}\right)\ldots \ \ \ \phi_n\left(\mathbf{x}\right)\}. \tag{2.12}$$

Derivatives of the shape functions can be easily obtained because the PIM shape function is of polynomial form. The $\boldsymbol{l}$th derivatives of PIM shape functions can be written as

$$\mathbf{\Phi}^{\boldsymbol{l}}\left(\mathbf{x}\right) = \left\{\begin{array}{c} \phi_1^{\boldsymbol{l}}(\mathbf{x}) \\ \phi_2^{\boldsymbol{l}}(\mathbf{x}) \\ \vdots \\ \phi_n^{\boldsymbol{l}}(\mathbf{x}) \end{array}\right\} = \frac{\boldsymbol{\partial}^{\boldsymbol{l}}\mathbf{P}^{\mathbf{T}}(\mathbf{x})}{\boldsymbol{\partial}\mathbf{x}^{\boldsymbol{l}}}\mathbf{P}_m^{-1}. \tag{2.13}$$

Note that our discussion is based on the assumption that $\mathbf{P}_m^{-1}$ exists. This condition cannot always be satisfied depending on the locations of the nodes in the support domain and the terms of monomials used in the basis. If an inappropriate polynomial basis is chosen for a given set of nodes, it may yield in a badly conditioned or even singular moment matrix. In order to avoid the singularity of the moment matrix, several strategies have been proposed, such as the randomly moving node method and the method of transformation of local coordinate system. However, these methods cannot completely overcome the singularity problem. The matrix triangularization algorithm (MTA) is another technique to overcome the singularity. Kansa has also solved this kind of singularity problem using radial basis functions. The Kansa model is a global collocation method that uses all the grids in the problem domain, which leads to a fully populated system matrix. Since the RBFs are used, the moment matrix is not singular in general. A more stable symmetric formulation has also been proposed by Z. Wu. Also Liu et al, to overcome this deficiency used RBFs as the augmented terms in the point interpolation method. In the following section the conventional radial point interpolation method (RPIM) is described in details.

In order to avoid the singularity problem in the polynomial point interpolation (PIM), the radial basis function (RBF) is used to develop the radial point interpolation method (RPIM) shape functions for meshless weak form methods. The RPIM interpolation augmented with polynomials can be written as:

$$u\left(\mathbf{x}\right)=\sum_{i=1}^{n}R_i\left(\mathbf{x}\right)a_i+\sum_{j=1}^{m}p_j\left(\mathbf{x}\right)b_j=\mathbf{R}^T\left(\mathbf{x}\right)\mathbf{a}+\mathbf{P}^T\left(\mathbf{x}\right)\mathbf{b}, \tag{2.14}$$

where $R_i\left(\mathbf{x}\right)$ is a radial basis function (RBF), n is the number of RBFs, $p_j\left(\mathrm{x}\right)$ is monomial in the space coordinate $\mathbf{x}^T=[x,y]$, and m is the number of polynomial basis functions. When $m=0$, only RBFs are used. Otherwise, the RBF is augmented with m polynomial basis functions. Coefficients a_i and b_j are unknown which should be determined. There are a number of types of RBFs, and the characteristics of RBFs have been widely investigated. In the current work, we have chosen the thin plate spline (TPS) as radial basis functions in Eq. (2.14). This RBF is defined as follows:

$$R\left(\mathbf{x}\right)=r^{2m}\ln\left(r\right)\ , \qquad m=1,2,3,\ldots \tag{2.15}$$

Since $R(\mathbf{x})$ in Eq. (2.15) belongs to C^{2m-1}(The set of $2m-1$ times continuously differentiable funtions), so higher-order thin plate splines must be used for higher-order partial differential operators. For the second-order partial differential equation (1.1), $m=2$ is used for thin plate splines (i.e. second-order thin plate splines). In radial basis function $R_i\left(\mathbf{x}\right)$, the variable is only the distance between the point of interest $\mathbf{x}$ and a node at, $\mathbf{x}_i=(x_i,y_i)$ i.e. $r=\sqrt{\left(x-x_i\right)^2+\left(y-y_i\right)^2}$. In order to determine a_i and b_j in Eq. (2.14), same as PIM, a support domain is formed for the point of interest at $\mathbf{x}$, and n field nodes are included in the support domain. Coefficients a_i and b_j in Eq. (2.14) can be determined by enforcing Eq. (2.14) to be satisfied at these n nodes surrounding the point of interest $\mathbf{x}$. This leads to the system of n linear equations, one for each node. The matrix form of these equations can be expressed as:

$$\mathbf{U}_{\mathrm{s}}=\mathbf{R}_n\mathbf{a}+\mathbf{P}_m\mathbf{b}, \tag{2.16}$$

where the vector of function values $\mathbf{U}_{\mathrm{s}}$ is

$$\mathbf{U}_{\mathrm{s}}=\{u_1\ \ u_2\ \ u_3\ldots u_n\}^T, \tag{2.17}$$

the RBFs moment matrix is

$$\mathbf{R}_n=\begin{bmatrix} R_1(r_1) & R_2(r_1) & \ldots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \ldots & R_n(r_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(r_n) & R_2(r_n) & \ldots & R_n(r_n) \end{bmatrix}_{n\times n}, \tag{2.18}$$

and the polynomial moment matrix is

$$\mathbf{P}_m = \begin{bmatrix} 1 & x_1 y_1 \ldots & p_m\left(\mathbf{x}_1\right) \\ 1 & x_2 y_2 \ldots & p_m\left(\mathbf{x}_2\right) \\ 1 & x_3 y_3 \ldots & p_m\left(\mathbf{x}_3\right) \\ \vdots & \vdots & \ddots \quad \vdots \\ 1 & x_n y_n \ldots & p_m\left(\mathbf{x}_n\right) \end{bmatrix}_{n\times m} . \tag{2.19}$$

Also, the vector of unknown coefficients for RBFs is

$$\mathbf{a}^T = \{a_1 \quad a_2 \quad a_3 \ldots a_n\} \tag{2.20}$$

and the vector of unknown coefficients for polynomial is

$$\mathbf{b}^T = \{b_1 \quad b_2 \quad b_3 \ldots b_m\}. \tag{2.21}$$

We notify that in Eq. (2.18), r_k in $R_i(r_k)$ is defined as

$$r_k = \sqrt{\left(x_k - x_i\right)^2 + \left(y_k - y_i\right)^2}. \tag{2.22}$$

We mention that there are $(m + n)$ variables in Eq. (2.16). The additional m equations can be added using the following m constraint conditions:

$$\sum_{i=1}^{n} p_j\left(\mathbf{x}_\mathrm{i}\right) a_i = \mathbf{P}_m^T \mathbf{a} = 0, \qquad j = 1, 2, \ldots, m. \tag{2.23}$$

Combining Eqs. (2.16) and (2.23) yields the following system of equations in the matrix form

$$\widetilde{\mathbf{U}}_\mathrm{s} = \begin{bmatrix} \mathbf{U}_s \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_n & \mathbf{P}_m \\ \mathbf{P}_m^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{G}\widetilde{\mathbf{a}}, \tag{2.24}$$

where

$$\widetilde{\mathbf{U}}_\mathrm{s} = \{u_1 \quad u_2 \quad u_3 \ldots u_n \quad 0 \quad 0 \quad \ldots \quad 0\}^T, \tag{2.25}$$

and

$$\widetilde{\mathbf{a}}^T = \{a_1 \quad a_2 \quad a_3 \ldots a_n \quad b_1 \quad b_2 \quad b_3 \ldots b_m\}. \tag{2.26}$$

Because the matrix $\mathbf{R}_n$ is symmetric, the matrix $\mathbf{G}$ will also be symmetric. Solving Eq. (2.24), we obtain

$$\widetilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{G}^{-1}\widetilde{\mathbf{U}}_s. \tag{2.27}$$

Eq. (2.14) can be re-written as

$$u\left(\mathrm{x}\right) = \mathbf{R}^T\left(\mathbf{x}\right)\mathbf{a} + \mathbf{P}^T\left(\mathbf{x}\right)\mathbf{b} = \left\{\mathbf{R}^T\left(\mathbf{x}\right)\mathbf{P}^T\left(\mathbf{x}\right)\right\} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}. \tag{2.28}$$

Now using Eq. (2.27) we obtain:

$$u\left(\mathrm{x}\right) = \left\{\mathbf{R}^T\left(\mathbf{x}\right)\mathbf{P}^T\left(\mathbf{x}\right)\right\}\mathbf{G}^{-1}\widetilde{\mathbf{U}}_s = \widetilde{\mathbf{\Phi}}^{\mathbf{T}}\left(\mathbf{x}\right)\widetilde{\mathbf{U}}_\mathrm{s}, \tag{2.29}$$

where $\widetilde{\mathbf{\Phi}}^{\mathbf{T}}(\mathbf{x})$ can be re-written as

$$\begin{aligned}\widetilde{\mathbf{\Phi}}^{\mathbf{T}}(\mathbf{x}) &= \left\{\mathbf{R}^{T}(\mathbf{x})\,\mathbf{P}^{T}(\mathbf{x})\right\}\mathbf{G}^{-1}\\ &= \{\phi_1(\mathbf{x})\,\phi_2(\mathbf{x})\ldots\quad \phi_n(\mathbf{x})\,\phi_{n+1}(\mathbf{x})\quad\ldots\quad \phi_{n+m}(\mathbf{x})\}\,.\end{aligned} \tag{2.30}$$

The first n functions of the above vector function are called the RPIM shape functions corresponding to the nodal displacements and we show by the vector $\mathbf{\Phi}^{\mathbf{T}}(\mathbf{x})$ so that it is

$$\mathbf{\Phi}^{\mathbf{T}}(\mathbf{x}) = \{\phi_1(\mathbf{x})\,\phi_2(\mathbf{x})\ldots\quad \phi_n(\mathbf{x})\}\,, \tag{2.31}$$

then Eq. (2.29) is converted to the following one

$$u(\mathbf{x}) = \mathbf{\Phi}^{\mathbf{T}}(\mathbf{x})\,\mathbf{U}_{\mathrm{s}} = \sum_{i=1}^{n}\phi_i(\mathbf{x})u_i. \tag{2.32}$$

The derivatives of $u(\mathbf{x})$ are easily obtained as

$$\frac{\partial u(\mathbf{x})}{\partial x} = \sum_{i=1}^{n}\frac{\partial\phi_i(\mathbf{x})}{\partial x}u_i, \qquad \frac{\partial u(\mathbf{x})}{\partial y} = \sum_{i=1}^{n}\frac{\partial\phi_i(\mathbf{x})}{\partial y}u_i. \tag{2.33}$$

Note that $\mathbf{R}_n^{-1}$ usually exists for arbitrary scattered nodes. In addition, the order of polynomial used in Eq. (2.14) is relatively low. It is remarkable that the RPIM shape functions have the Kronecker delta function property, that is

$$\phi_i(\mathbf{x}_j) = \begin{cases} 1, \quad i = j & \qquad j = 1, 2, \ldots, n,\\ 0, \quad i \neq j & \qquad i, j = 1, 2, \ldots, n.\end{cases} \tag{2.34}$$

This is because the RPIM shape functions are created to pass thorough nodal values.

3. The time discretisation approximation

In the current work, we employ a time-stepping scheme to overcome the time derivatives. For this purpose, the following finite difference approximations for the time derivative operators are used:

$$\frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} \cong \frac{1}{(\Delta t)^2}\left(u^{k+1}(\mathbf{x}) - 2u^{k}(\mathbf{x}) + u^{k-1}(\mathbf{x})\right), \tag{3.1}$$

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} \cong \frac{1}{2\Delta t}\left(u^{k+1}(\mathbf{x}) - u^{k-1}(\mathbf{x})\right). \tag{3.2}$$

Also we employ the following approximations by using the Crank-Nicolson technique:

$$u(\mathbf{x}, t) \cong \frac{1}{3}\left(u^{k+1}(\mathbf{x}) + u^{k}(\mathbf{x}) + u^{k-1}(\mathbf{x})\right), \tag{3.3}$$

$$\Delta u(\mathbf{x}, t) \cong \frac{1}{3}\left(\Delta u^{k+1}(\mathbf{x}) + \Delta u^{k}(\mathbf{x}) + \Delta u^{k-1}(\mathbf{x})\right), \tag{3.4}$$

where $u^k(\mathbf{x}) = u(\mathbf{x}, k\Delta t)$ and $\mathbf{x} = (x, y)$. Using the above discussion, Eq. (1.1) can be written as:

$$\begin{aligned}
&\frac{1}{(\Delta t)^2}\left(u^{k+1}(\mathbf{x}) - 2u^k(\mathbf{x}) + u^{k-1}(\mathbf{x})\right) + \frac{\alpha}{2\Delta t}\left(u^{k+1}(\mathbf{x}) - u^{k-1}(\mathbf{x})\right)\\
&+\frac{\beta}{3}\left(u^{k+1}(\mathbf{x}) + u^k(\mathbf{x}) + u^{k-1}(\mathbf{x})\right)\\
&=\frac{1}{3}\left(\Delta u^{k+1}(\mathbf{x}) + \Delta u^k(\mathbf{x}) + \Delta u^{k-1}(\mathbf{x})\right)\\
&+\frac{1}{3}\left(f(\mathbf{x}, (k+1)\Delta t) + f(\mathbf{x}, k\Delta t) + f(\mathbf{x}, (k-1)\Delta t)\right).
\end{aligned} \tag{3.5}$$

Suppose that $\lambda{=}\frac{3}{(\Delta t)^2}$, $\eta = \frac{3\alpha}{2\Delta t}$and $F(\mathbf{x}; k) = f(\mathbf{x}, (k+1)\Delta t) + f(\mathbf{x}, k\Delta t) + f(\mathbf{x}, (k-1)\Delta t)$, then we obtain

$$\begin{aligned}
&(\lambda + \eta + \beta)\, u^{k+1} - \Delta u^{k+1} = (2\lambda - \beta)\, u^k + \Delta u^k\\
&+ (\eta - \lambda - \beta)\, u^{k-1} + \Delta u^{k-1} + F(\mathbf{x}; k).
\end{aligned} \tag{3.6}$$

4. The meshless local weak form formulation

Let us now before continuing recall a certain Green's formula in 2-dimension which will be of fundamental importance in what follows. Let us start from the divergence theorem (in two dimensions):

$$\int_\Omega \mathrm{div}\mathbf{A}\mathrm{d}\mathbf{x}{=}\int_\Gamma \mathbf{A}.\mathbf{n}\mathrm{d}\mathbf{s}, \tag{4.1}$$

in which $\mathbf{A} = (A_1, A_2)$ is a vector-valued function defined on Ω, $\mathbf{x} = (x, y)$, $div\mathbf{A} = \frac{\partial A_1}{\partial x} + \frac{\partial A_2}{\partial y}$ and $\mathrm{n} = (n_1, n_2)$ is the outward unit normal to Γ. Here $d\mathbf{x}$ denotes the element of area in $\mathbb{R}^2$ and $\mathrm{d}\mathbf{s}$ the element of length on Γ. If we apply the divergence theorem to $\mathbf{A}{=}(vw,\ 0)$ and $\mathbf{A}{=}(0,\ vw)$ we find that

$$\int_\Omega \frac{\partial v}{\partial x} w\mathrm{d}\mathbf{x}{+}\int_\Omega v\frac{\partial w}{\partial x}\ \mathrm{d}\mathbf{x}{=}\int_\Gamma vwn_1\mathrm{d}\mathbf{s}, \tag{4.2}$$

$$\int_\Omega \frac{\partial v}{\partial y} w\mathrm{d}\mathbf{x}{+}\int_\Omega v\frac{\partial w}{\partial y}\ \mathrm{d}\mathbf{x}{=}\int_\Gamma vwn_2\mathrm{d}\mathbf{s}, \tag{4.3}$$

Denoting by ∇v the gradient of v, i.e. $\nabla v{=}\left(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}\right)$, we get from (4.2) and (4.3) the following 2-dimensional Green's formula:

$$\begin{aligned}
&\int_\Omega \nabla v.\ \nabla w\mathrm{d}\mathbf{x}{=}\int_\Omega \left(\frac{\partial v}{\partial x}\frac{\partial w}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial w}{\partial y}\right)\mathrm{d}\mathbf{x}\\
&= \int_\Gamma \left(v\frac{\partial w}{\partial x}n_1 + v\frac{\partial w}{\partial y}n_2\right)\mathrm{d}\mathbf{s} - \int_\Omega v\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right)\mathrm{d}\mathbf{x}\\
&= \int_\Gamma v\frac{\partial w}{\partial \mathbf{n}}\mathrm{d}\mathbf{s}{-}\int_\Omega v\Delta w\mathrm{d}\mathbf{x},
\end{aligned} \tag{4.4}$$

where $\frac{\partial w}{\partial \mathrm{n}} = \frac{\partial w}{\partial x} n_1 + \frac{\partial w}{\partial y} n_2$ is the normal derivative, i.e. the derivative in the outward normal direction to the boundary Γ.

Instead of giving the global weak form, the MLRPI method constructs the weak form over local quadrature cell such as Ω_q, which is a small region taken for each node in the global domain Ω. The local quadrature cells overlap each other and cover the whole global domain Ω. The local quadrature cells could be of any geometric shape and size. In this paper they are taken to be of circular shape (see Figure 1). The local weak form of Eq. (3.6) for $\mathbf{x}_i = (x_i,\ y_i) \in \Omega_q^i$ can be written as

$$\begin{aligned}
&\int_{\Omega_q^i} \left((\lambda+\eta+\beta)\, u^{k+1} - \Delta u^{k+1} \right) v(\mathbf{x}) \mathrm{d}\Omega = \int_{\Omega_q^i} \left((2\lambda-\beta)\, u^{k} + \Delta u^{k} \right) v(\mathbf{x}) \mathrm{d}\Omega \\
&+ \int_{\Omega_q^i} \left((\eta-\lambda-\beta)\, u^{k-1} + \Delta u^{k-1} \right) v(\mathbf{x}) \mathrm{d}\Omega + \int_{\Omega_q^i} F\left(\mathbf{x}; k\right) v\left(\mathbf{x}\right) \mathrm{d}\Omega,
\end{aligned} \tag{4.5}$$

where Ω_q^i is the local quadrature domain associated with the point i, i.e., it is a circle centered at $\mathbf{x}_{\boldsymbol{i}}$ of radius r_q and, $v(\mathbf{x})$ is the Heaviside step function,

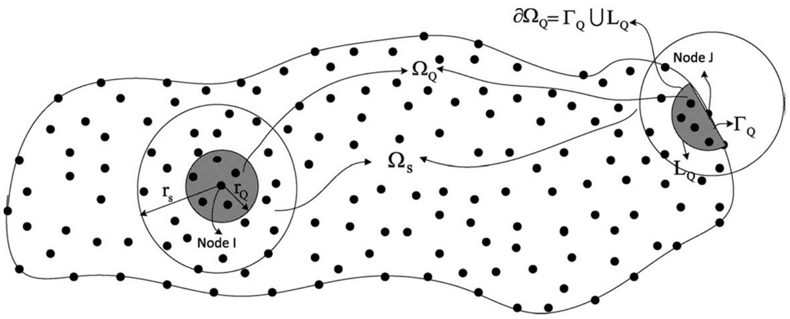
$$v\left(\mathbf{x}\right) = \begin{cases} 1, & \mathbf{x} \in \Omega_q, \\ 0, & \mathbf{x} \notin \Omega_q, \end{cases} \tag{4.6}$$

as the test function in each local quadrature domain. Using the 2-dimensional Green's formula (4.4) and Eq. (4.5) yields the following expression:

$$\begin{aligned}
&(\lambda+\eta+\beta) \int_{\Omega_q^i} u^{(k+1)} v(\mathbf{x}) \mathrm{d}\Omega + \int_{\Omega_q^i} \nabla u^{(k+1)} \nabla v \ \mathrm{d}\Omega - \int_{\partial\Omega_q^i} v \frac{\partial u^{(k+1)}}{\partial \mathbf{n}} \mathrm{d}\Gamma \\
&= (2\lambda-\beta) \int_{\Omega_q^i} u^{k} v\left(\mathbf{x}\right) \mathrm{d}\Omega - \int_{\Omega_q^i} \nabla u^{k} \nabla v \ \mathrm{d}\Omega + \int_{\partial\Omega_q^i} v \frac{\partial u^{k}}{\partial \mathbf{n}} \mathrm{d}\Gamma \\
&+ (\eta - \lambda - \beta) \int_{\Omega_q^i} u^{(k-1)} v\left(\mathbf{x}\right) \mathrm{d}\Omega - \int_{\Omega_q^i} \nabla u^{(k-1)} \nabla v \mathrm{d}\Omega \\
&+ \int_{\partial\Omega_q^i} v \frac{\partial u^{(k-1)}}{\partial \mathbf{n}} \mathrm{d}\Gamma + \int_{\Omega_q^i} F\left(\mathbf{x}; k\right) v\left(\mathbf{x}\right) \mathrm{d}\Omega,
\end{aligned} \tag{4.7}$$

where $\partial\Omega_q^i$ is the boundary of Ω_q^i, $\mathrm{n} = (n_1,\ n_2)$ is the outward unit normal to the boundary $\partial\Omega_q^i$, and $\frac{\partial u}{\partial \mathrm{n}} = \frac{\partial u}{\partial x} n_1 + \frac{\partial u}{\partial y} n_2$ is the normal derivative, i.e., the derivative in the outward normal direction to the boundary $\partial\Omega_q^i$. Because the derivative of the Heaviside step function $v(\mathbf{x})$ is equal to zero, then the local weak form equation (4.7)

FIGURE 1. Node I is an interior node and J is a node on the Neumann boundary of the problem domain. $\varOmega_s$ and $\varOmega_q$ are local support and local quadrature domains, respectively.



is transformed into the following simple integral equation:

$$\begin{aligned}&(\lambda{+}\eta{+}\beta)\int_{\varOmega_q^i} u^{(k+1)}\mathrm{d}\Omega-\int_{\partial\varOmega_q^i}\frac{\partial u^{(k+1)}}{\partial\mathbf{n}}\mathrm{d}\Gamma\\&=(2\lambda{-}\beta)\int_{\varOmega_q^i} u^{k}\mathrm{d}\Omega+\int_{\partial\varOmega_q^i}\frac{\partial u^{k}}{\partial\mathbf{n}}\mathrm{d}\Gamma\\&+(\eta-\lambda-\beta)\int_{\varOmega_q^i} u^{(k-1)}\mathrm{d}\Omega+\int_{\partial\varOmega_q^i}\frac{\partial u^{(k-1)}}{\partial\mathbf{n}}\mathrm{d}\Gamma+\int_{\varOmega_q^i} F\left(\mathbf{x};k\right)\mathrm{d}\Omega,\end{aligned}\tag{4.8}$$

As it is obvious from Figure 1, for node $\mathbf{x}_{\mathrm{i}}$ on the natural boundary, $\partial\varOmega_q^i$ and $\varOmega_q^i$ are replaced by $L_q^i\cup\varGamma_q^i$and $\acute{\varOmega}_q^i$, respectively. Therefore, the local weak form equation for this node is:

$$\begin{aligned}&(\lambda{+}\eta{+}\beta)\int_{\acute{\varOmega}_q^i} u^{(k+1)}\mathrm{d}\Omega-\int_{L_q^i}\frac{\partial u^{(k+1)}}{\partial\mathbf{n}}\mathrm{d}\Gamma-\int_{\varGamma_q^i}\frac{\partial u^{(k+1)}}{\partial\mathbf{n}}\mathrm{d}\Gamma\\&=(2\lambda{-}\beta)\int_{\acute{\varOmega}_q^i} u^{k}\mathrm{d}\Omega+\int_{L_q^i}\frac{\partial u^{k}}{\partial\mathbf{n}}\mathrm{d}\Gamma+\int_{\varGamma_q^i}\frac{\partial u^{k}}{\partial\mathbf{n}}\mathrm{d}\Gamma\\&+(\eta-\lambda-\beta)\int_{\acute{\varOmega}_q^i} u^{(k-1)}\mathrm{d}\Omega\\&+\int_{L_q^i}\frac{\partial u^{(k-1)}}{\partial\mathbf{n}}\mathrm{d}\Gamma+\int_{\varGamma_q^i}\frac{\partial u^{(k-1)}}{\partial\mathbf{n}}\mathrm{d}\Gamma+\int_{\acute{\varOmega}_q^i} F\left(\mathbf{x};k\right)\mathrm{d}\Omega,\end{aligned}\tag{4.9}$$

Applying the radial point interpolation (RPI) for the unknown functions, the local integral equation (4.8) and (4.9) is transformed into a system of algebraic equations with used unknown quantities, as described in the next section.

5. Discretization and numerical implementation for MLRPI method

In this section, we consider Eq. (4.8) to see how to obtain discrete equations for the interior points. Consider N regularly located points in the interior of domain of the problem so that the distance between two consecutive nodes in each direction is constant and equal to h (see Figure1). Assuming that $u(\,\mathbf{x_i},(k-1)\Delta t)$ and $u(\,\mathbf{x_i},k\Delta t)$, $i=1,\ 2,\ ...,\ N$ are known, our aim is to compute $u\left(\,\mathbf{x_i},(k+1)\,\Delta t\right),\ \ i=1,\ 2,\ ...,\ N$. So, we have N unknowns and to compute these unknowns we need N equations. As it will be described, corresponding to each node we obtain one equation. For nodes which are located in the interior of the domain, i.e., for $\mathbf{x}_i\in$ interior Ω, to obtain the discrete equations from the locally weak forms (4.8), substituting approximation formulas (2.32) and (2.33) into local integral equations (4.8) yields:

$$\begin{aligned}
&(\lambda{+}\eta{+}\beta)\sum_{j=1}^{N}\left(\int_{\Omega_q^i}\phi_j\mathrm{d}\Omega\right)u_j^{(k+1)}-\sum_{j=1}^{N}\left(\int_{\partial\Omega_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma\right)u_j^{(k+1)}\\
&=(2\lambda{-}\beta)\sum_{j=1}^{N}\left(\int_{\Omega_q^i}\phi_j\mathrm{d}\Omega\right)u_j^{k}+\sum_{j=1}^{N}\left(\int_{\partial\Omega_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma\right)u_j^{k}\\
&+(\eta-\lambda-\beta)\sum_{j=1}^{N}\left(\int_{\Omega_q^i}\phi_j\mathrm{d}\Omega\right)u_j^{(k-1)}+\sum_{j=1}^{N}\left(\int_{\partial\Omega_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma\right)u_j^{(k-1)}\\
&+\int_{\Omega_q^i}F\left(\mathbf{x};k\right)\mathrm{d}\Omega,
\end{aligned}\tag{5.1}$$

or equivalently

$$\begin{aligned}
&\left[(\lambda{+}\eta{+}\beta)\sum_{j=1}^{N}\left(\int_{\Omega_q^i}\phi_j\mathrm{d}\Omega\right)-\sum_{j=1}^{N}\left(\int_{\partial\Omega_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma\right)\right]u_j^{(k+1)}\\
&=\left[(2\lambda{-}\beta)\sum_{j=1}^{N}\left(\int_{\Omega_q^i}\phi_j\mathrm{d}\Omega\right)+\sum_{j=1}^{N}\left(\int_{\partial\Omega_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma\right)\right]u_j^{k}\\
&+\left[(\eta-\lambda-\beta)\sum_{j=1}^{N}\left(\int_{\Omega_q^i}\phi_j\mathrm{d}\Omega\right)+\sum_{j=1}^{N}\left(\int_{\partial\Omega_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma\right)\right]u_j^{(k-1)}\\
&+\int_{\Omega_q^i}F\left(\mathbf{x};k\right)\mathrm{d}\Omega,
\end{aligned}\tag{5.2}$$

Now for nodes which are located on the boundary, in this step we write the discretized form of Eq. (4.9). But before this, the natural boundary condition can be easily imposed as follows. From Eq. (1.3) we understand:

$$\int_{\mathit{\Gamma}_q^{??}} \frac{\partial u^{(k-1)}}{\partial \mathbf{n}} \mathrm{d}\Gamma = \int_{\mathit{\Gamma}_q^i} \frac{\partial u^{k}}{\partial \mathbf{n}} \mathrm{d}\Gamma = \int_{\mathit{\Gamma}_q^i} \frac{\partial u^{(k+1)}}{\partial \mathbf{n}} \mathrm{d}\Gamma = 0. \tag{5.3}$$

Substituting the relation (5.3) and also the approximation formulas (2.32) and (2.33) into local integral equation (4.9) yields:

$$\begin{aligned}
&\left[(\lambda+\eta+\beta)\sum_{j=1}^{N}\left(\int_{\acute{\mathit{\Omega}}_q^i}\phi_j \mathrm{d}\Omega\right)-\sum_{j=1}^{N}\left(\int_{L_q^i}\frac{\partial \phi_j}{\partial \mathbf{n}}\mathrm{d}\Gamma\right)\right]u_j^{(k+1)}\\
&=\left[(2\lambda-\beta)\sum_{j=1}^{N}\left(\int_{\acute{\mathit{\Omega}}_q^i}\phi_j \mathrm{d}\Omega\right)+\sum_{j=1}^{N}\left(\int_{L_q^i}\frac{\partial \phi_j}{\partial \mathbf{n}}\mathrm{d}\Gamma\right)\right]u_j^{k}\\
&+\left[(\eta-\lambda-\beta)\sum_{j=1}^{N}\left(\int_{\acute{\mathit{\Omega}}_q^i}\phi_j \mathrm{d}\Omega\right)+\sum_{j=1}^{N}\left(\int_{L_q^i}\frac{\partial \phi_j}{\partial \mathbf{n}}\mathrm{d}\Gamma\right)\right]u_j^{(k-1)}\\
&+\int_{\acute{\mathit{\Omega}}_q^i} F\left(\mathbf{x};k\right)\mathrm{d}\Omega,
\end{aligned} \tag{5.4}$$

The matrix forms of Eqs. (5.2) and (5.4) for all N nodal points in domain and on boundary of the problem are given in below in Eqs. (5.5) and (5.7) respectively:

$$\begin{aligned}
&\left[(\lambda+\eta+\beta)\sum_{j=1}^{N}A_{ij}-\sum_{j=1}^{N}B_{ij}\right]u_j^{(k+1)}\\
&=\left[(2\lambda-\beta)\sum_{j=1}^{N}A_{ij}+\sum_{j=1}^{N}B_{ij}\right]u_j^{k}\\
&+\left[(\eta-\lambda-\beta)\sum_{j=1}^{N}A_{ij}+\sum_{j=1}^{N}B_{ij}\right]u_j^{(k-1)}+F_i^k,
\end{aligned} \tag{5.5}$$

where

$$A_{ij}=\int_{\mathit{\Omega}_q^i}\phi_j \mathrm{d}\Omega,\;\; B_{ij}=\int_{\partial\mathit{\Omega}_q^i}\frac{\partial \phi_j}{\partial \mathbf{n}}\mathrm{d}\Gamma,\;\; F_i^k=\int_{\mathit{\Omega}_q^i}F\left(\mathbf{x};k\right)\mathrm{d}\Omega, \tag{5.6}$$

and

$$\begin{aligned}&\left[(\lambda{+}\eta{+}\beta)\sum_{j=1}^{N}\acute{A}_{ij}-\sum_{j=1}^{N}\acute{B}_{ij}\right]u_j^{(k+1)}\\ &=\left[(2\lambda{-}\beta)\sum_{j=1}^{N}\acute{A}_{ij}+\sum_{j=1}^{N}\acute{B}_{ij}\right]u_j^{k}\\ &+\left[(\eta-\lambda-\beta)\sum_{j=1}^{N}\acute{A}_{ij}+\sum_{j=1}^{N}\acute{B}_{ij}\right]u_j^{(k-1)}+\acute{F}_i^{k},\end{aligned} \tag{5.7}$$

where

$$\acute{A}_{ij}=\int_{\acute{\Omega}_q^i}\phi_j\mathrm{d}\Omega,\ \ \acute{B}_{ij}=\int_{L_q^i}\frac{\partial\phi_j}{\partial\mathbf{n}}\mathrm{d}\Gamma,\ \ \acute{F}_i^{k}=\int_{\acute{\Omega}_q^i}F\left(\mathbf{x};k\right)\mathrm{d}\Omega. \tag{5.8}$$

Now, assuming

$$\forall j:\quad \mathbf{A}_{ij}=\begin{cases}\ \ (\lambda{+}\eta{+}\beta)\,A_{ij}-B_{ij}, & i\in \textit{interior}\ \Omega\\ \\ (\lambda{+}\eta{+}\beta)\,\acute{A}_{ij}-\acute{B}_{ij} & \qquad\qquad i\in\partial\Omega\end{cases}$$

$$\forall j:\quad \mathbf{B}_{ij}=\begin{cases}\ \ (2\lambda{-}\beta)\,A_{ij}+B_{ij}, & i\in \textit{interior}\ \Omega\\ \\ (2\lambda{-}\beta)\,\acute{A}_{ij}+\acute{B}_{ij} & \qquad\qquad i\in\partial\Omega\end{cases}$$

$$\forall j:\quad \mathbf{C}_{ij}=\begin{cases}\ \ (\eta-\lambda-\beta)\,A_{ij}-B_{ij}, & i\in \textit{interior}\ \Omega\\ \\ (\eta-\lambda-\beta)\,\acute{A}_{ij}-\acute{B}_{ij} & \qquad\qquad i\in\partial\Omega\end{cases}$$

$$\mathcal{F}_i^k=\begin{cases}\ \ F_i^k, & i\in \textit{interior}\ \Omega\\ \\ \acute{F}_i^k & \qquad\qquad i\in\partial\Omega\end{cases}$$

with $\mathcal{F}^k=\left[\mathcal{F}_1^k,\mathcal{F}_2^k\ ,\ \ldots,\ \mathcal{F}_N^k\right]^T$, $U=(u_i)_{N\times 1}$, and combining Eqs. (5.5) and (5.7) yields:

$$\mathbf{A}U^{(k+1)}=\mathbf{B}U^{k}+\mathbf{C}U^{(k-1)}+\mathcal{F}^{k}. \tag{5.9}$$

In general, we have the above equation for each time level and to obtain $U^{(k+1)}$, we need to solve the above system of linear equations but at first step, when $k=0$, according to the initial conditions that were introduced in Eqs. (1.2), we apply the following assumptions:

$$U^{(0)}=\varPhi=\left[\varphi\left(\mathbf{x_1}\right),\varphi\left(\mathbf{x_2}\right)\boldsymbol{,\ldots,\varphi(\mathbf{x_N})}\right]^T, \tag{5.10}$$

$$U^{(-1)}=U^{(1)}-2\Delta t\varPsi=U^{(1)}-2\Delta t[\psi\left(\mathbf{x_1}\right),\psi\left(\mathbf{x_2}\right)\boldsymbol{,\ldots,\psi(\mathbf{x_N})}]^T. \tag{5.11}$$

Therefore Eq. (5.9) is converted to the following

$$\mathbf{A}U^{(1)} = \mathbf{B}U^{(0)}+\mathbf{C}\left(U^{(1)} - 2\Delta t\,\Psi\right) + \mathcal{F}^{0}, \tag{5.12}$$

or equivalently

$$(\mathbf{A}-\mathbf{C})U^{(1)} = \mathbf{B}U^{(0)}+\mathbf{C}\left(-2\Delta t\,\Psi\right) + \mathcal{F}^{0}. \tag{5.13}$$

6. Numerical experiment

In this section, we show the result obtained for an example using the meshless method described above. In this example, the domain integrals and the boundary integrals are both evaluated with 6 points Gaussian quadrature rule. To show the behavior of the solution and the efficiency of the proposed method, the following root mean square (RMS) error is applied to make comparison:

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{N}\left(U_{exact}\left(\mathrm{x}_i\right) - U_{approx}\left(\mathrm{x}_i\right)\right)^2}{N}},$$

where $U_{exact}\left(\mathrm{x}_i\right)$ and $U_{approx}\left(\mathrm{x}_i\right)$ are achieved by exact and approximate solutions on points $\mathbf{x}_i$ and N is number of nodal points. Also in order to implement the meshless local weak form, the radius of the local quadrature domain $r_q = 0.7h$ is selected, where h is the distance between the nodes in $x \; or \; y$ direction. The size of r_q is such that the union of these sub-domains must cover the whole global domain. The radius of support domain to obtain shape functions in the frame of local radial point interpolation method is set as $r_s = 4r_q$. This size is significant enough to have sufficient number of nodes (N) and gives an appropriate approximation. Also, in Eq. (2.14), we set $m = 6$, i.e., the quadratic basis function (2.3) is used. We set $\alpha = 70$, $\beta = 50$ in Eq. (1.1) and, the exact solution of the problem (1.1)-(1.3) is taken as:

$$u\left(x, y, t\right) = \cos\left(\pi x\right)\,\cos\left(\pi y\right)\,exp\left(-t\right),$$

where $\varphi\left(x, y\right)$, $\psi\left(x, y\right)$ are defined accordingly and

$$f\left(x,\; y,\; t\right) = cos\left(\pi x\right)\,cos\left(\pi y\right)\,exp\left(-t\right)\left(1 - \alpha + \beta + 2\pi^2\right).$$

In Table 1, The RMS error of numerical solution for different time steps and different number of nodal points for t=1.5 have been reported. It is clear from this table, the RMS error decreases when nodal points increases and also when time steps decreases. In Table 2, the RMS error of numerical solution for different number of nodal points for some time levels has been given. The spatial convergence could be seen in this table. The descending behavior of the RMS error for long time by increasing time with $N = 1681$ and $\Delta t = 0.01$ is observed in Table 3. It can be seen that the RMS error goes to zero by passing time, in other words the MLRPI method is unconditionally stable for this example. In Fig.2, the RMS error has been plotted at different time levels with Δt=0.1, 0.05 and 0.001 and $N = 289$. As it is clear from this figure, the RMS error decreases when Δt decreases and also the RMS error decreases by passing time. In Fig.3, the RMS error has been plotted for increasing nodal points with time levels t=3.8, 4.8 and 5.8 and Δt=0.01. As it is clear from this figure, the RMS error decreases when nodal points increases and this refers to spatial convergence.

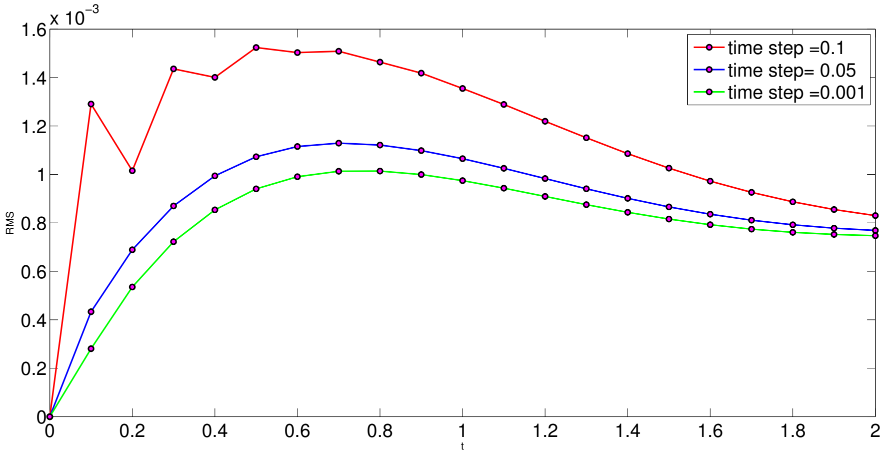
Table 1. The RMS error for different time steps and different nodal points for t =1.5.

Δt	N=36	N=121	N=441
0.1	1.048641e-02	2.027650e-03	1.069825e-03
0.05	9.575349e-03	1.877718e-03	8.566803e-04
0.001	9.216767e-03	1.813486e-03	7.918458e-04

Table 2. The RMS error for different time levels and different nodal points with Δt =0.01.

N	t=1.8	t=3.6	t=5.4
16	2.954136e-02	1.460926e-02	5.276435e-03
81	3.216066e-03	2.335722e-03	1.031264e-03
169	1.295624e-03	1.134789e-03	5.520654e-04
225	9.446468e-04	8.581564e-04	4.364497e-04
400	6.633804e-04	4.831297e-04	2.728478e-04

Figure 2. The behavior of RMS error via time with N=289.



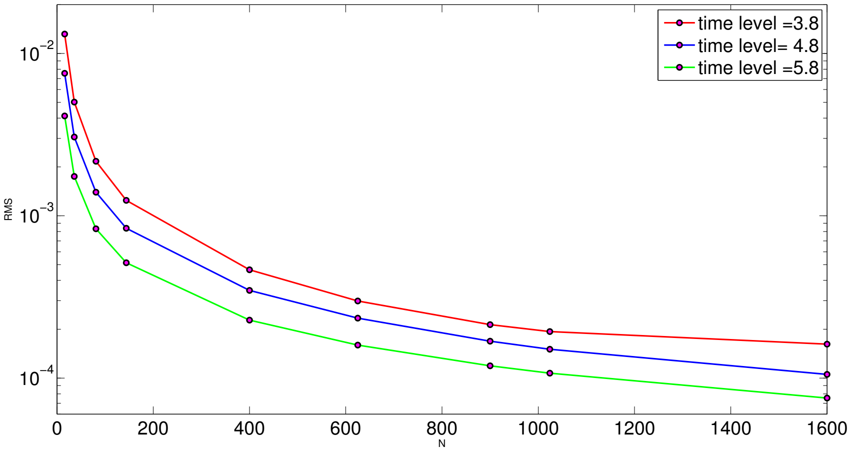
7. Conclusion

In this paper meshless local radial point interpolation (MLRPI) method has been applied to solve a class of two-space dimensional wave equations. The main challenge of current work is the improvement of meshless techniques, which relies on the robust and efficient implementation of effective integration rules, to impose Neumanns boundary conditions . In this method, the shape functions have been constructed by the radial point interpolation. Some time stepping schemes were employed to approximate the time derivatives.

TABLE 3. The RMS error for different time levels with Δt=0.01 and N=1681.

t	RMS error
0	0
0.5	1.065287e − 03
1	1.215302e − 03
1.5	1.008524e − 03
2	7.274216e − 04
2.5	4.838264e − 04
3	3.092164e − 04
3.5	2.010535e − 04
4	1.429591e − 04
4.5	1.135373e − 04
5	9.557862e − 05
10	8.125376e − 06
13	1.155090e − 06

FIGURE 3. The behavior of RMS error via increasing nodal point for some time levels.



Competing interests

The authors declare that they have no competing interests.

Authors contributions

All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

References

[1] P. Assari and M. Dehghan, *A meshless discrete collocation method for the numerical solution of singular logarithmic boundary integral equations utilizing radial basis functions*, Appl. Math. Comput., *315* (2017), 424–444.

[2] P. Assari and M. Dehghan, *Solving a class of nonlinear boundary integral equations based on the meshless local discrete Galerkin (MLDG) method*, Comput. Methods Differ. Appl. Numer. Math., *123*(2017), 137–158.

[3] P. Assari and M. Dehghan, *A meshless Galerkin scheme for the approximate solution of nonlinear logarithmic boundary integral equations utilizing radial basis functions*, J. Comput Appl. Math, *333* (2017), 362–381.

[4] P. Assari and M. Dehghan, *Application of thin plate splines for solving a class of boundary integral equations arisen from Laplace's equations with nonlinear boundary conditions*, Int. J. Comput. Appl. Math, *96*(1) (2019), 170–198.

[5] T. Belytschko, Y. Y. Lu, and L. Gu, *Element-free Galerkin methods*, Int. J. Numer. MethodsEng, *37*(2) (1994) 229-256.

[6] T. Belytschko, Y. Y. Lu, and L. Gu, *Element free Galerkin methods for static and dynamic fracture*, Int. J. Solids Struct., *32* (1995) 2547-2570.

[7] A. Bratsos, *An improved numerical scheme for the sine-Gordon equation in 2+1 dimensions*, Int. J. Numer. Meth. Eng., *75* (2008) 787-799.

[8] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge University Press, (2004).

[9] P. Clear, *Modeling conned multi-material heat and mass flows using SPH*, Appl. Math. Model., *22* (1998) 981-993.

[10] W. J. Costin and C. B. Allen, *Numerical study of radial basis function interpolation for data transfer across discontinuous mesh interfaces*, Int. J. Numer. Meth. Fluids, *72(10)*(2013) 1076-1095.

[11] W. Dai, H. Song,S. Su, and R. Nassar, *A stable finite difference scheme for solving a hyperbolic two-step model in a 3D micro sphere exposed to ultra short-pulsed lasers*, Int. J. Numer. Meth. H., *16* (2006) 693-717.

[12] S. De and K. Bathe, *The method of finite spheres*, Comput. Mech., *25* (2000) 329-345.

[13] M. Dehghan and A. Ghesmati, *Numerical simulation of two-dimensional sine-gordon solitons via a local weak meshless technique based on the radial point interpolation method (RPIM)*, Comput. Phys. Commun., *181* (2010) 772-786.

[14] L. M. J. S. Dinis, R. M. Natal Jorge, and J. Belinha, *Static and dynamic analysis of laminated plates based on an unconstrained third order theory and using a radial point interpolator meshless method*, Computers and Structures., *89* (2011) 1771-1784.

[15] S. T. Grilli, P. Guyenneb, and F. Diasc, *A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom*, Int. J. Numer. Meth. Fluids, *35* (2001) 829-867.

[16] Y. Gu and G. Liu, *A boundary radial point interpolation method (BRPIM) for 2-d structural analyses*, Struct. Eng. Mech, *15* (2003) 535-550.

[17] Y. Gu and G. Liu, *A boundary point interpolation method for stress analysis of solids*, Comput. Mech, *28* (2002) 47-54.

[18] A. Hirose, *Introduction to Wave Phenomena*, Wile, New York, (1985).

[19] E. Kansa, *Multiquadrics-a scattered data approximation scheme with applications to computational fluid-dynamics. I. surface approximations and partial derivative estimates*, Comput. Math. Appl, *19(8-9)* (1990) 127-145.

[20] G. Liu,L. Yan,J. Wang and Y. Gu, *Point interpolation method based on local residual formulation using radial basis functions*, Struct. Eng. Mech, *14* (2002) 713-732.

[21] G. Liu and Y. Gu, *A local radial point interpolation method (LR-PIM) for free vibration analyses of 2-D solids*, J. Sound Vib, *246(1)* (2001) 29-46.

[22] G. Liu and Y. Gu, *An Introduction to Meshfree Methods and Their Programing*, Springer, (2005).

[23] W. Liu,S. Jun and Y. Zhang, *Reproducing kernel particle methods*, Int. J. Numer. Meth. Eng, *20* (1995) 1081-1106.

[24] J. Melenk and I. Babuska, *The partition of unity finite element method: Basic theory and applications*, Comput. Meth. Appl. Mech. Eng, *139* (1996) 289-314.

[25] Y. Mukherjee and S. Mukherjee, *Boundary node method for potential problems*, Int. J. Numer.Meth. Eng, *40* (1997) 797-815.

[26] B. Nayroles, G. Touzot and P. Villon, *Generalizing the finite element method: diffuse approximation and diffuse elements*, Comput. Mech, *10* (1992) 307-318.

[27] S. Nettel,*Wave Physics: Oscillations-Solitons-Chaos*, Springer, Berlin, (2003).

[28] A. Selvadurai, *Partial Differential Equations in Mechanics*, Springer, Berlin, (2000).

[29] U. Senturk, *Modeling nonlinear waves in a numerical wave tank with localized meshless RBF method*, Comput. and Fluids, *44* (2011) 221-228.

[30] E. Shivanian, *Analysis of meshless local and spectral meshless radial point interpolation (MLRPI and SMRPI) on 3-D nonlinear wave equations*, Ocean Engineering, *89* (2014) 173-188.

[31] A. Wazwaz, *Partial differential equations: methods and applications*, Swets and Zeitlinger B.V., Lisse, Netherlands, (2002).

[32] Z. Yan and V. V. Konotop, *Exact solutions to three-dimensional generalized nonlinear schrdinger equations with varying potential and nonlinearities*, Phys. Rev, *(80)036607* E (2009)1-9.