Journal of Computational and Applied Mathematics 236 (2012) 2367–2377

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

The numerical solution of the non-linear integro-differential equations based on the meshless method

Mehdi Dehghan *, Rezvan Salehi

Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave., 15914, Tehran, Iran

ARTICLE INFO

Article history:
Received 19 June 2011
Received in revised form 9 August 2011

MSC:
65R20
45D05
45G10
47G20

Keywords:
Moving least square method
Fredholm integro-differential equation
Volterra integro-differential equation
Error analysis

ABSTRACT

This article investigates the numerical solution of the nonlinear integro-differential equations. The numerical scheme developed in the current paper is based on the moving least square method. The moving least square methodology is an effective technique for the approximation of an unknown function by using a set of disordered data. It consists of a local weighted least square fitting, valid on a small neighborhood of a point and only based on the information provided by its n closet points. Hence the method is a meshless method and does not need any background mesh or cell structures. The error analysis of the proposed method is provided. The validity and efficiency of the new method are demonstrated through several tests.

1. Introduction

Mathematical modeling of real problems usually results in functional equations, e.g. partial differential equations, integral equations, integro-differential equations, stochastic differential equations, delay differential equations, partial integro-differential equations, differential algebraic equations and others. Many mathematical formulations of physical phenomena contain integro-differential equations. These equations arise in fluid dynamics, biological models, chemical kinetic, ecology, control theory of financial mathematics, aerospace systems, industrial mathematics etc. Especially, one always can describe a model which possesses hereditary properties by integro-differential equations in practice. Integro-differential equations are usually difficult to solve analytically so it is required to obtain an efficient approximate solution. Recently, much interest of scientists and engineers have been paid on nontraditional methods for nonlinear problems. Nowadays the Chebyshev and Taylor polynomial approximation methods are used and the numerical solutions are obtained in [1–3]. The Wavelet-Galerkin method is applied to solve the second kind integral equation [4]. In [5,6], the variational iteration method (VIM) is considered to solve integral and integro-differential equations. In addition, iterative and non-iterative methods for the solution of nonlinear Volterra integro-differential equations are presented and their local convergence is proved. The iterative methods provide a sequence solution and make use of fixed-point theory whereas the non-iterative ones result in series solutions and also make use of the fixed-point principles [7]. A one-step algorithm for the construction of approximation solution of a Volterra integro-differential equation has been presented in [8]. The authors of [9] have used the sine–cosine wavelets on interval [0, 1] to solve the integro-differential equations.

Moreover, the interested reader is referred to [10,11] for numerical or semi-numerical approaches in the solution of integral equations, respectively. In recent years the meshless methods have gained more attention not only by

* Corresponding author. Fax: +98 21 66497930.
E-mail addresses: mdehghan@aut.ac.ir, mdehghan.aut@gmail.com, mdehghan_aut@yahoo.com (M. Dehghan), rsalehi@aut.ac.ir (R. Salehi).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.11.022

mathematicians but also in the engineering community. Moving least-square methods are the basis of numerous mesh-free approximation methods for the solution of partial differential equations that have recently been suggested by practitioners as an alternative to the traditional finite element method. During the past decade the moving least square (MLS) method proposed in [12] has become a very popular approximation scheme. We can find a wide range of applications in the framework of function approximation and surface construction [13]. Recently, this method has been employed in response surface models in order to accelerate optimization procedures [14], and stochastic analyses [15]. Further important applications of meshless moving least square method are the diffuse element method (DEM) proposed in [16], the well-known element-free Galerkin (EFG) method introduced in [17] and the meshless local Petrov Galerkin method developed in [18]. We refer the interested reader to [19–24] for applications of meshless methods to partial differential equations.

In this article, we employ the meshless moving least square method to solve the nonlinear integro-differential equations. The moving least square methodology is an effective approach for the approximation of an unknown function by using a set of disordered data. It consists of a local weighted least square fitting, valid on a small neighborhood of a point and only based on the information provided by its n closet points. The method can easily be implemented and the technique is flexible for most classes of integral equations, because of easy adaptation of the nodal density. Also we refer the interested reader to [25].

The rest of this paper is organized as follows: the outline of the MLS method is discussed in Section 2. In Section 3, the proposed method is employed on nonlinear Fredholm integro-differential equations. An error analysis of the method is demonstrated in Section 4. In Section 5, the method is implemented for Volterra integro-differential equations. Several test problems are solved and the results are shown in Section 6. Section 7 completes this paper with a brief conclusion.

2. The moving least square approximation

The moving least square (MLS) approximation may be considered as a member of the class of meshless schemes that have the properties of local interpolation, high accuracy approximation and can easily be extended to n-dimensional problems. The idea behind MLS consists of a better control of the shape function smoothness and continuity as opposed to the finite element method (FEM). This is obtained through the use of the weight functions, that allows control of the locality and the continuity of the approximation. The MLS method was started with Shepard's method [26] and was extended by McLain [27,28], Franke and Nielson [29] and Lancaster and Salkauskas [12]. Now we follow the well known works on MLS to give an outline of this method.

Consider a sub-domain $\Omega_{\mathbf{x}}$, the neighborhood of a point $\mathbf{x}$ and the domain of definition of the MLS approximation for the trial function at $\mathbf{x}$, which is located in the problem domain Ω. To approximate the unknown function, u in $\Omega_{\mathbf{x}}$, over a number of randomly located nodes $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, the MLS approximant $u^h(\mathbf{x})$ of u, $\forall \mathbf{x} \in \Omega_x$, can be defined as

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{2.1}$$

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})]$ is a complete monomial basis of $\mathcal{P}_q$ of order m, and $\mathbf{a}(\mathbf{x})$ is a vector containing coefficients $\mathbf{a}_j(\mathbf{x})$, $j = 1, 2, \ldots, m$, which are functions of the space coordinate $\mathbf{x}$. For a 1D example, the linear basis is $\mathbf{p}^T(x) = \begin{bmatrix} 1 & x \end{bmatrix}$, and the quadratic basis is $\mathbf{p}^T(x) = \begin{bmatrix} 1 & x & x^2 \end{bmatrix}$.

The coefficient vector $\mathbf{a}(\mathbf{x})$ is determined by minimizing a weighted discrete L_2 norm, defined as

$$\mathcal{J}(\mathbf{x}) = \sum_{t=1}^{n} w_i(\mathbf{x})(\mathbf{p}^T(\mathbf{x}_j)\mathbf{a}(\mathbf{x}) - \hat{u}_i)^2 = [\mathbf{P}\cdot\mathbf{a} - \hat{\mathbf{u}}]^T \cdot \mathbf{W} \cdot [\mathbf{P}\cdot\mathbf{a} - \hat{\mathbf{u}}], \tag{2.2}$$

where $w_i(\mathbf{x})$ is the weight function associated with the node i, $\mathbf{x}_i$ denotes the value of $\mathbf{x}$ at node i, n is the number of nodes in $\Omega_{\mathbf{x}}$ with $w_i(\mathbf{x}) > 0$, the matrices $\mathbf{P}$ and $\mathbf{W}$ are defined as

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_n) \end{bmatrix}_{n\times(m+1)}, \qquad \mathbf{W} = \begin{bmatrix} w_1(\mathbf{x}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n(\mathbf{x}) \end{bmatrix}_{n\times n}, \tag{2.3}$$

and

$$\hat{\mathbf{u}} = [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n]. \tag{2.4}$$

Here it should be noted that $\hat{u}_i$, $i = 1, 2, \ldots, n$, in Eqs. (2.2) and (2.4) are the fictitious nodal values, and not the nodal values of the unknown trial function $u^h(\mathbf{x})$ in general.

The stationarity of $\mathcal{J}$ in Eq. (2.2) with respect to $\mathbf{a}(\mathbf{x})$ leads to the following linear relation between $\mathbf{a}(\mathbf{x})$ and $\hat{\mathbf{u}}$,

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\hat{\mathbf{u}}, \tag{2.5}$$

where the matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are defined by

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T\mathbf{W}\mathbf{P} = \mathbf{B}(\mathbf{x})\mathbf{P} = \sum_{i=1}^{n} w_i(\mathbf{x})\mathbf{p}^T(\mathbf{x}_i)\mathbf{p}(\mathbf{x}_i), \tag{2.6}$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^T\mathbf{W} = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \ldots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)]. \tag{2.7}$$

The MLS approximation is well defined only when the matrix $\mathbf{A}$ in Eq. (2.5) is non-singular. It can be seen that this is the case if and only if the rank of $\mathbf{P}$ equals m. Computing $\mathbf{a}(\mathbf{x})$ from Eq. (2.5) and substituting it into Eq. (2.1) give

$$u^h(\mathbf{x}) = \mathbf{\Phi}^T(\mathbf{x}) \cdot \hat{\mathbf{u}} = \sum_{j=1}^{n} \phi_j(\mathbf{x})\hat{u}_j, \quad \mathbf{x} \in \Omega_{\mathbf{x}}, \tag{2.8}$$

where

$$\mathbf{\Phi}^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}), \tag{2.9}$$

or

$$\phi_j(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{p}_i(\mathbf{x})[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})]_{ij}. \tag{2.10}$$

$\phi_i(\mathbf{x})$ is usually called the shape function of the MLS approximation corresponding to the nodal point $\mathbf{x}_i$. The smoothness of the shape functions $\phi_i(\mathbf{x})$ is determined by that of the basis functions and of the weight functions. Let $\mathcal{C}^k(\Omega)$ be the space of k-th continuously differentiable functions. If $w_i(\mathbf{x}) \in \mathcal{C}^k(\Omega)$ and $p_j(\mathbf{x}) \in \mathcal{C}^l(\Omega)$ $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$ then $\phi_{\mathbf{x}} \in \mathcal{C}^{\min(k,l)}(\Omega)$. The partial derivatives of $\phi_i(\mathbf{x})$ are obtained as [17]

$$\phi_{i,k} = \sum_{j=1}^{m}[p_{j,k}(\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}))_{ji} + p_j(\mathbf{A}^{-1}_{,k}(\mathbf{x})\mathbf{B}(\mathbf{x}) + \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}_{,k}(\mathbf{x}))_{ji}], \tag{2.11}$$

where $\mathbf{A}^{-1}_{,k} = (\mathbf{A}^{-1})_{,k}$ represents the derivative of the inverse of $\mathbf{A}$ with respect to x_k, which is given by

$$\mathbf{A}^{-1}_{,k} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}, \tag{2.12}$$

where $()_{,i}$ denotes $\partial()/\partial x_i$.

The Gaussian weight function corresponding to the node i may be written as

$$w_i(\mathbf{x}) = \begin{cases} \dfrac{\exp\left[-\left(\frac{d_i}{\alpha}\right)^2\right] - \exp\left[-\left(\frac{h_i}{\alpha}\right)^2\right]}{1 - \exp\left[-\left(\frac{h_i}{\alpha}\right)^2\right]}, & 0 \le d_i \le h_i, \\ 0, & d_i \ge h_i, \end{cases} \tag{2.13}$$

where $d_i = |\mathbf{x} - \mathbf{x}_i|$ is the distance from node $\mathbf{x}_i$ to point $\mathbf{x}$, α is a constant controlling the shape of the weight function w_i and therefore the relative weights, and h_i is the size of the support for the weight function w_i and determines the support of node $\mathbf{x}_i$.

3. The Fredholm integro-differential equation

In this section, we employ the MLS method for solving the 1D Fredholm integro-differential equation. The Fredholm integro-differential equation can be considered as follows:

$$\begin{cases} \acute{u}(x) = \mathcal{F}\left(x, u(x), \displaystyle\int_a^b \mathcal{K}(x, t, u(t))\mathrm{d}t\right), \\ u(a) = u_0, \end{cases} \tag{3.1}$$

where u is the unknown function, a and b are real finite numbers, $\mathcal{K}$ is the so-called kernel function, $\mathcal{F} : \mathbb{R} \times \mathcal{C}^p \times \mathcal{C}^p \longrightarrow \mathcal{C}^p$ and kernel function $\mathcal{K} : \mathbb{R} \times \mathbb{R} \times \mathcal{C}^p \longrightarrow \mathcal{C}^p$ are given continuous mappings and satisfying in the following conditions:

$$\|\mathcal{F}(x, u_1, u_2) - \mathcal{F}(x, \nu_1, \nu_2)\| \le \beta_1\|u_1 - \nu_1\| + \beta_2\|u_2 - \nu_2\|, \quad \forall u_1, u_2, \nu_1, \nu_2 \in \mathcal{C}^p, \tag{3.2}$$

$$\|\mathcal{K}(x, u(x), \theta_1) - \mathcal{K}(x, u(x), \theta_1)\| \le c\|\theta_1 - \theta_2\|, \quad \forall \theta_1, \theta_2 \in \mathcal{C}^p, \tag{3.3}$$

$$\|\mathcal{K}(x, u(x), \theta)\| \le c_1\|\theta\|, \quad \forall \theta \in \mathcal{C}^p, \tag{3.4}$$

where β_1, β_2 and c are real constant values.

Now, to employ the MLS method let us consider n nodal points in the interval $[a, b]$ as $a = x_1 \leq x_2 \leq \cdots \leq x_n = b$. Considering Eqs. (2.8) and (2.11), we replace u_h and u_h' with u and $\acute{u}$ in Eq. (3.1), respectively. Therefore Eq. (3.1) becomes

$$u_h{}'(x) = \mathcal{F}\left(x, u_h(x), \int_a^b \mathcal{K}(x, t, u_h(t))\mathrm{d}t\right), \tag{3.5}$$

or correspondingly

$$\sum_{i=1}^{n} \phi_{i,x}(x_j)\hat{u}_i = \mathcal{F}\left(x_j, \sum_{i=1}^{n} \phi_i(x_j)\hat{u}_i, \int_a^b \mathcal{K}\left(x_j, t, \sum_{i=1}^{n} \phi_i(t)\hat{u}_i\right)\mathrm{d}t\right), \quad j = 1, 2, \ldots, n. \tag{3.6}$$

Choosing an m-point quadrature formula with the coefficients $\{\xi_j\}$ and weights $\{\omega_j\}$ in the interval $[a, b]$ for solving integral in (3.5) yield:

$$\sum_{i=1}^{n} \phi_{i,x}(x_j)\hat{u}_i = \mathcal{F}\left(x_j, \sum_{i=1}^{n} \phi_i(x_j)\hat{u}_i, \sum_{k=1}^{m} \omega_k \mathcal{K}\left(x_j, \xi_k, \sum_{i=1}^{n} \phi_i(\xi_k)\hat{u}_i\right)\right), \quad j = 1, 2, \ldots, n. \tag{3.7}$$

Finding the values of $\hat{u}_j$ by solving Eq. (3.7) with an appropriate procedure for investigating the solution of the non-linear systems yields the following approximate solution:

$$u(x) \cong u_h(x) = \sum_{j=1}^{n} \phi_j(x)\hat{u}_j, \quad \forall x \in [a, b]. \tag{3.8}$$

4. The error analysis

In this section, the error estimate for the proposed method is obtained. Since the error estimate of the method is strictly connected to the error estimate of the MLS method, first we introduce the error estimate of the MLS method. Levin [13] studied the MLS method for a special weight function and gained the error estimate in the uniform norm for the approximation of a regular function in N dimension but, author of [13] did not investigate the error estimate for the derivatives. Authors of [30], proved the uniform norm error estimate for the approximation of the first and second derivatives of a function with the MLS method in the one-dimensional case. Armentano [31] obtained the error estimates in the L^2 and L^∞ norms for the first and second derivatives of a function approximated by the MLS method in the N-dimensional under the optimal regularity assumptions. In [32] Zuppa proved the error estimates for approximation of the function and the first and second order derivatives in L_∞ norm. In [33] the authors report the error estimates for the MLS approximation in the H^k norm in 1D when nodes and weight functions satisfy certain conditions, and then derive the convergence rate of a MLS-based meshless Galerkin method, the Galerkin boundary node method, for general integral equations. In the current work, we employ the results for error estimates of the MLS method in [32]. Furthermore the interested readers are referred to the interesting papers [30–32,34].

4.1. Preliminaries

In the n dimensional space $\mathbb{R}^n$, given any multi-index $\beta = (\beta_1, \beta_2, \ldots, \beta_n) \in \mathbb{N}^n$, $|\beta|$ denotes the sum $\beta_1 + \beta_2 + \cdots + \beta_n$, and, if u is a sufficiently smooth function, $\mathcal{D}^\beta u$ denotes the partial derivative

$$\frac{\partial^{|\beta|}}{\partial\beta_1\partial\beta_2\cdots\partial\beta_n}u.$$

Let Ω be an open bounded domain in $\mathbb{R}^n$ and $\mathcal{Q}_N$ denotes an arbitrarily chosen set of N points of $\mathbf{x}_\alpha \in \overline{\Omega}$ referred to as *nodes*:

$$\mathcal{Q}_N = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad x_i \in \overline{\Omega}.$$

Let $\mathcal{U}_N := \{\Lambda_j\}_{j=1}^N$ denotes a finite open covering of $\overline{\Omega}$ consisting of N clouds Λ_j such that $\mathbf{x}_j \in \Lambda_j$ and Λ_j is centered around $\mathbf{x}_j$ in some way and

$$\bar{\Omega} \subset \bigcup_{j=1}^{N} \Lambda_j.$$

The radius d_j of Λ_j is defined as $\max_{\mathbf{x}\in\partial\Lambda_j} \|\mathbf{x} - \mathbf{x}_j\|$.

A class of functions $\mathcal{S}_N := \{\omega_i\}_{i=1}^N$ is called a partition of unity subordinate to the open covering $\mathcal{U}_N$ if it possesses the following properties:

- $\omega_i \in \mathcal{C}_0^s(\mathbb{R}^n)$, $s \geq 0$, or $s = \infty$.
- $\mathrm{supp}(\omega_i) \subseteq \overline{\Lambda_i}$.
- $\omega_i(\mathbf{x}) > 0, \mathbf{x} \in \Lambda_i$.
- $\sum_{i=1}^N \omega_i(\mathbf{x}) = 1, \forall \mathbf{x} \in \overline{\Omega}$.

There is no unique way to build a partition of unity as defined above.

A function u is said to be of class $\mathcal{C}^{p,1}$ in $\overline{\Omega}$ if and only if u is of class $\mathcal{C}^p$ in $\overline{\Omega}$ and the partial derivatives $\mathcal{D}^\mu u$ of u of order p ($|\mu| = p$) are Lipschitz continuous in $\overline{\Omega}$. The semi-norm $|.|_{p,1}$ is defined as

$$|u|_{p,1} = \sup\left\{\frac{\|\mathcal{D}^\mu u(\mathbf{x}) - \mathcal{D}^\mu u(\mathbf{y})\|}{\|\mathbf{x}-\mathbf{y}\|} : \mathbf{x}, \mathbf{y} \in \overline{\Omega}, \mathbf{x} \neq \mathbf{y}, |\mu| = p\right\}.$$

In order to have the moving least square approximation well defined we need that the minimization problem has a unique solution at $\mathbf{y} \in \overline{\Omega}$ and this is equivalent to the non-singularity of matrix $\mathbf{A}(\mathbf{y})$. The error estimates are obtained [34] with the following assumptions about the system of nodes and the weight function $\{\mathcal{Q}_N, \mathcal{W}_N\}$.

Property $\mathcal{R}_q$. *For any* $\mathbf{x} \in \overline{\Omega}$, *the normal matrix* $\mathbf{A}(\mathbf{x})$ *is non-singular.*

Definition. Given $\mathbf{x} \in \overline{\Omega}$, the set $\mathcal{ST}(\mathbf{x}) := \{i|\omega_i(\mathbf{x}) \neq 0\}$ will be called the star at $\mathbf{x}$ [34].

Theorem 4.1. *A necessary condition for Property* $\mathcal{R}_q$ *is that for any* $\mathbf{x} \in \overline{\Omega}$

$$n = card(\mathcal{ST}(\mathbf{x})) \geq card(\mathcal{P}_q) = m + 1.$$

For a point $\mathbf{c} \in \overline{\Omega}$ *if* $\mathcal{ST}(\mathbf{c}) = \{i_1, i_2, \ldots, i_k\}$, *then the size of star* $\mathcal{ST}(\mathbf{c})$ *is defined by the number*

$$h(\mathcal{ST}(\mathbf{c})) = \max\{d_{i_1}, d_{i_2}, \ldots, d_{i_k}\}.$$

Assumptions. The following assumptions are made about the parameters employed in the current work [34,35]. There exist

(A1): An upper bound of the overlap of clouds:

$$\mathcal{M} = \sup_{\mathbf{c}\in\overline{\Omega}}\{card(\mathcal{ST}(\mathbf{c}))\}.$$

(A2): Upper bound for the condition number:

$$\mathcal{CB}_q = \sup_{\mathbf{c}\in\overline{\Omega}}\{\mathcal{CN}_q(\mathcal{ST}(\mathbf{c}))\}, \quad q = 1, 2,$$

where the numbers $\mathcal{CN}_q(\mathcal{ST}(\mathbf{c}))$ are computable measures of the quality of the star $(\mathcal{ST}(\mathbf{c}))$ were defined in Theorem 7 of [32].

(A3): An upper bound of the mesh-size of stars:

$$d = \sup_{\mathbf{c}\in\overline{\Omega}}\{h(\mathcal{ST}(\mathbf{c}))\}.$$

(A4): An uniform bound of the derivatives of $\{\omega_j\}$. That is, a constant $\mathcal{G}_q \geq 0$, $q = 1, 2$, such that

$$\|\mathcal{D}^\mu \mu_j\| \leq \frac{\mathcal{G}_q}{h^{|\mu|}}, \quad 1 \leq |\mu| \leq q.$$

(A5): There exists number $\gamma \geq 0$ such that any two points $\mathbf{x}, \mathbf{y} \in \overline{\Omega}$ can be joined by a rectified curve Γ in $\overline{\Omega}$ with length $|\Gamma| \leq \gamma\|\mathbf{x}-\mathbf{y}\|$.

Considering the above mentioned conditions, Zuppa [32] proved the following theorem.

Theorem 4.2. *There exist constants* $\mathcal{C}_q$, $q = 1$ *or* 2

$$\mathcal{C}_1 = \mathcal{C}_1(\gamma, n, \mathcal{M}, \mathcal{G}_1, \mathcal{CB}_1),$$
$$\mathcal{C}_2 = \mathcal{C}_2(\gamma, n, \mathcal{M}, \mathcal{G}_1, \mathcal{CB}_1, \mathcal{CB}_2),$$

such that, for each $u \in \mathcal{C}^{q,1}(\overline{\Omega})$

$$\|\mathcal{D}^\mu u - \mathcal{D}^\mu u_h\|_{L^\infty(\Omega)} \leq \mathcal{C}_q d^{q+1-|\mu|}|u|_{q,1}, \quad 0 \leq |\mu| \leq q. \tag{4.1}$$

Also, we assume the numerical quadrature satisfies the condition described in the following [35].

QA: There exists positive number η, small enough and independent of i and mesh-size, such that

$$\left|\int_\omega \varrho\, dx - \int_\omega^* \varrho\, dx\right| \leq \eta|\omega|\|\varrho\|_{L_\infty(\omega)}, \tag{4.2}$$

where the notation $\int^*(,)$ denotes the integrals which are computed using a quadrature formula and the constant η is a bound for the error of the numerical quadrature formula [36].

4.2. The error estimate

Let us consider the non-linear Fredholm integro-differential equation (3.1) with conditions (3.2)–(3.4). By defining in the Banach space $\mathcal{C}^p$, the integral operator $\mathcal{T}(u(x))$ as

$$\mathcal{T}(u(x)) = u(a) + \int_a^x \mathcal{F}\left(t, u(t), \int_a^b \mathcal{K}(t, \xi, u(\xi))\mathrm{d}\xi\right)\mathrm{d}t, \quad [a, x] \subseteq [a, b], \tag{4.3}$$

then the Banach fixed point theorem guarantees that, under certain assumptions ((3.2)–(3.4)), $\mathcal{T}$ has a unique fixed point, that is, the Fredholm integro-differential equation has exactly one solution. Also, assume that $u(x)$ is the solution of Eq. (3.1), $u_h(x)$ is the solution of Eq. (3.5) and $u_h^*(x)$ is the solution of Eq. (3.7).

Theorem 4.3. *Let $u \in \mathcal{C}^{q,1}(\overline{\Omega})$ where Ω is a bounded set in $\mathbb{R}$. Assume that $\mathcal{F}$ and $\mathcal{K}$ satisfy the conditions (3.1)–(3.2) and $[a, x] \subseteq \Omega = [a, b]$. Also, assume that the numerical quadrature formula satisfies (4.2). Moreover, take a suitable approximation u_h of u. Then we have*

$$\|\mathcal{T}(u(x)) - \mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)} \leq \mathcal{M}_1|u|_{q,1} + \mathcal{M}_2\|u\|_{L^\infty(\Omega)}. \tag{4.4}$$

Consequently, we obtain

$$\|u - u_h^*\|_{L^\infty(\Omega)} \leq \mathcal{M}_1\,|u|_{q,1} + \mathcal{M}_2\,\|u\|_{L^\infty(\Omega)}, \tag{4.5}$$

where $\mathcal{M}_1 = ((b-a)\beta_1 + c(b-a)^2\beta_2 + \beta_2\, c_1\, \eta\, (b-a)^2)\mathcal{C}_q d^{q+1}$ and $\mathcal{M}_2 = \beta_2\, c_1\, \eta\, (b-a)^2$.

Proof. In the proof, we use Theorem 4.2 and Lipschitz condition for $\mathcal{F}$ and $\mathcal{K}$. From definition of $\mathcal{T}$, we have

$$\|\mathcal{T}(u(x)) - \mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)} \leq \|\mathcal{T}(u(x)) - \mathcal{T}(u_h(x))\|_{L^\infty(\Omega)} + \|\mathcal{T}(u_h(x)) - \mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)}.$$

First, we obtained

$$\begin{aligned}
\|\mathcal{T}(u(x)) - \mathcal{T}(u_h(x))\|_{L^\infty(\Omega)} &= \left\| \int_a^x \left(\mathcal{F}\left(t, u(t), \int_a^b \mathcal{K}(t, \xi, u(\xi))\mathrm{d}\xi\right)\right.\right. \\
&\quad \left.\left. - \mathcal{F}\left(t, u_h(t), \int_a^b \mathcal{K}(t, \xi, u_h(\xi))\mathrm{d}\xi\right)\right)\mathrm{d}t \right\|_{L^\infty(\Omega)} \\
&\leq (b-a)\left\| \mathcal{F}\left(t, u(t), \int_a^b \mathcal{K}(t, \xi, u(\xi))\mathrm{d}\xi\right)\right. \\
&\quad \left. - \mathcal{F}\left(t, u_h(t), \int_a^b \mathcal{K}(t, \xi, u_h(\xi))\,\mathrm{d}\xi\right)\right\|_{L^\infty(\Omega)} \\
&\leq (b-a)\left(\beta_1\|u(t) - u_h(t)\|_{L^\infty(\Omega)} + \beta_2\left\| \int_a^b \mathcal{K}(t, \xi, u(\xi))\mathrm{d}\xi\right.\right. \\
&\quad \left.\left. - \int_a^b \mathcal{K}(t, \xi, u_h(\xi))\mathrm{d}\xi\right\|_{L^\infty(\Omega)}\right) \\
&\leq (b-a)\left(\beta_1\|u(t) - u_h(t)\|_{L^\infty(\Omega)} + \beta_2\left\| \int_a^b (\mathcal{K}(t, \xi, u(\xi))\right.\right. \\
&\quad \left.\left. - \mathcal{K}(t, \xi, u_h(\xi)))\,\mathrm{d}\xi\|_{L^\infty(\Omega)}\right) \\
&\leq (b-a)(\beta_1\|u(t) - u_h(t)\|_{L^\infty(\Omega)} + c(b-a)\beta_2\|u(x) - u_h(x)\|_{L^\infty(\Omega)}) \\
&\leq \mathcal{C}_q(b-a)(\beta_1 + c(b-a)\beta_2)d^{q+1}|u|_{q,1},
\end{aligned}$$

on the other hand we can write

$$\begin{aligned}
\|\mathcal{T}(u_h(x)) - \mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)} &= \left\| \int_a^x \left(\mathcal{F}\left(t, u_h(t), \int_a^b \mathcal{K}(t, \xi, u_h(\xi))\mathrm{d}\xi\right)\right.\right. \\
&\quad \left.\left. - \mathcal{F}\left(t, u_h(t), \int_a^{b*} \mathcal{K}(t, \xi, u_h(\xi))\mathrm{d}\xi\right)\right)\mathrm{d}t \right\|_{L^\infty(\Omega)} \\
&\leq \beta_2(b-a)\left\| \int_a^b \mathcal{K}(t, \xi, u_h(\xi))\mathrm{d}\xi - \int_a^{b*} \mathcal{K}(t, \xi, u_h(\xi))\mathrm{d}\xi \right\|_{L^\infty(\Omega)}
\end{aligned}$$

$$\begin{aligned}
&\leq \beta_2\,\eta(b-a)^2\|\mathcal{K}(t,\xi,u_h(\xi))\|_{L^\infty(\Omega)}\\
&\leq \beta_2\,c_1\,\eta(b-a)^2\|u_h\|_{L^\infty(\Omega)}\\
&\leq \beta_2\,c_1\,\eta(b-a)^2(\|u\|_{L^\infty(\Omega)}+\|u-u_h\|_{L^\infty(\Omega)})\\
&\leq \beta_2\,c_1\,\eta(b-a)^2(\|u\|_{L^\infty(\Omega)}+\mathcal{C}_q\,d^{q+1}|u|_{q,1}).
\end{aligned}$$

Consequently, from the two last obtained inequalities, we can get the following inequality:

$$\begin{aligned}
\|\mathcal{T}(u(x))-\mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)} &\leq \|\mathcal{T}(u(x))-\mathcal{T}(u_h(x))\|_{L^\infty(\Omega)}+\|\mathcal{T}(u_h(x))-\mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)}\\
&\leq \mathcal{C}_q(b-a)(\beta_1+c(b-a)\beta_2)d^{q+1}|u|_{q,1}+\beta_2\,c_1\,\eta\,(b-a)^2(\|u\|_{L^\infty(\Omega)}+\mathcal{C}_q\,d^{q+1}|u|_{q,1})\\
&\leq ((b-a)\beta_1+c(b-a)^2\beta_2+\beta_2\,c_1\,\eta(b-a)^2)\mathcal{C}_q d^{q+1}|u|_{q,1}+\beta_2\,c_1\,\eta\,(b-a)^2\|u\|_{L^\infty(\Omega)}\\
&\leq \mathcal{M}_1|u|_{q,1}+\mathcal{M}_2\|u\|_{L^\infty(\Omega)}.
\end{aligned}$$

On the other hand from the fixed point property of u and u_h^* we have

$$\begin{aligned}
\|u-u_h^*\|_{L^\infty(\Omega)} &= \|\mathcal{T}(u(x))-\mathcal{T}(u_h^*(x))\|_{L^\infty(\Omega)}\\
&\leq ((b-a)\beta_1+c(b-a)^2\beta_2+\beta_2\,c_1\,\eta(b-a)^2)\mathcal{C}_q d^{q+1}|u|_{q,1}+\beta_2\,c_1\,\eta\,(b-a)^2\|u\|_{L^\infty(\Omega)}\\
&\leq \mathcal{M}_1|u|_{q,1}+\mathcal{M}_2\|u\|_{L^\infty(\Omega)}.
\end{aligned}$$

This completes the proof. □

5. The Volterra integro-differential equation

The method can be easily extended for the Volterra integro-differential equations. For this purpose consider the following Volterra integro-differential equation

$$\begin{cases}\acute{u}(x)=\mathcal{F}\left(x,u(x),\displaystyle\int_a^x\mathcal{K}(x,t,u(t))\mathrm{d}t\right),\\ u(a)=u_0.\end{cases}\tag{5.1}$$

Let us consider n nodal points in the interval $[a,b]$ as $a=x_1\leq x_2\leq\cdots\leq x_n=b$. Noting to Eqs. (2.8) and (2.11), u_h and u_h' are replaced with u and $\acute{u}$ in Eq. (3.1), respectively. Therefore Eq. (5.1) becomes

$$u_h{}'(x)=\mathcal{F}\left(x,u_h(x),\int_a^x\mathcal{K}(x,t,u_h(t))\mathrm{d}t\right),\tag{5.2}$$

or equivalently

$$\sum_{i=1}^n\phi_{i,x}(x_j)\hat{u}_i=\mathcal{F}\left(x_j,\sum_{i=1}^n\phi_i(x_j)\hat{u}_i,\int_a^x\mathcal{K}\left(x_j,t,\sum_{i=1}^n\phi_i(t)\hat{u}_i\right)\mathrm{d}t\right),\quad j=1,2,\ldots,n.\tag{5.3}$$

The integral domain $[a,x]$ must be transferred to a fixed interval $[a,b]$. For this purpose, the following transformation has been considered

$$\rho(x,\theta)=\frac{x-a}{b-a}\theta+\frac{b-x}{b-a}a.\tag{5.4}$$

Employing this transformation, Eq. (5.3) becomes

$$\sum_{i=1}^n\phi_{i,x}(x_j)\hat{u}_i=\mathcal{F}\left(x_j,\sum_{i=1}^n\phi_i(x_j)\hat{u}_i,\int_a^b\mathcal{K}^\dagger\left(x_j,\rho(x_j,\theta),\sum_{i=1}^n\phi_i(t)\hat{u}_i\right)\mathrm{d}\theta\right),\quad j=1,2,\ldots,n,\tag{5.5}$$

where

$$\mathcal{K}^\dagger=\frac{x-a}{b-a}\mathcal{K}.$$

Applying an m-point quadrature formula with coefficients τ_k and the weights ω_k in the interval $[a,b]$ in Eq. (5.3) yields

$$\sum_{i=1}^n\phi_{i,x}(x_j)\hat{u}_i=\mathcal{F}\left(x_j,\sum_{i=1}^n\phi_i(x_j)\hat{u}_i,\sum_{k=1}^m\omega_k\,\mathcal{K}^\dagger\left(x_j,\rho(x_j,\tau_k),\sum_{i=1}^n\phi_i(\rho_k)\hat{u}_i\right)\right),\quad j=1,2,\ldots,n.\tag{5.6}$$

Solving Eq. (5.6) with an appropriate numerical solver for the nonlinear systems, we can obtain the values of $\hat{u}_j$. Then the value of $u(x)$ is approximated by

$$u(x)\cong u_h(x)=\sum_{j=1}^n\phi_j(x)\hat{u}_j,\quad\forall x\in[a,b].\tag{5.7}$$

Table 1
Maximum absolute errors, ratio of error and CPU times used for different values of N.

N	Linear ($q = 1$)			Quadratic ($q = 2$)		
	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time
5	8.12×10^{-4}	–	0.37	9.63×10^{-4}	–	0.42
9	2.11×10^{-4}	3.85	0.41	1.28×10^{-4}	7.54	0.58
17	5.79×10^{-5}	3.53	0.56	2.87×10^{-5}	4.45	0.73
33	1.51×10^{-5}	3.95	0.89	5.61×10^{-6}	5.12	1.27
65	3.60×10^{-6}	4.19	0.96	2.39×10^{-6}	2.35	1.54
129	9.05×10^{-7}	3.98	1.41	1.28×10^{-6}	1.86	2.15

6. Numerical results

In this section, the numerical results for the one-dimensional Fredholm and Volterra integro-differential equations have been depicted. To measure the accuracy of the method, the maximum error has been used with the following definition:

$$\text{Maximum error}: \ \|\mathbf{e}\|_\infty = \max_j |u(j) - u_{exact}(j)|. \tag{6.1}$$

To show the rate of convergence of the new method, the values of ratio with the following formula have been reported

$$\text{Ratio} = \frac{\|\mathbf{e}_\infty^{N-1}\|}{\|\mathbf{e}_\infty^{N}\|}.$$

For the tests we used the linear and the quadratic basis and Gaussian weight function. In the following computations, we take $d = h_N = \frac{1}{N-1}, \alpha = \frac{0.6}{N-1}$ where these values are obtained experimentally. In addition, the influence domain of the node x_i is the support of the weight function $\omega_i(x)$. Also, for the numerical quadrature rule we used the seven-point Gauss–Legendre quadrature formula. In the case of Volterra integro-differential equations, the approximate values and errors are tabulated for $x = 2$.

6.1. The Fredholm integro-differential equation

6.1.1. Example 1

Consider the integro-differential equation

$$\acute{u}(x) = -u(x) + \int_0^1 u(t)^2 \mathrm{d}t + \frac{1}{2}(e^{-2} - 1),$$
$$u(0) = 1.$$

The exact solution is

$$u(x) = e^{-x}.$$

The maximum error values and *CPU* times are given for different values of N in Table 1. Moreover, the rate of convergence is shown in Table 1 for both linear and quadratic cases. The ratio numbers show the quadratic convergence of the method in the linear case. But increasing the condition number $\mathcal{CN}_2$ causes increasing of the errors at the boundary which effects on the global errors in the quadratic case.

6.1.2. Example 2

As the second problem, consider the following equation

$$\acute{u}(x) = 1 - \frac{1}{3}x^3 + \int_0^1 x^3\, u(t)^2\, \mathrm{d}t,$$
$$u(0) = 0,$$

where the unknown function is

$$u(x) = x.$$

To measure the accuracy of the studied approach, the maximum errors, *CPU* times and the rate of convergence are presented in Table 2. Like the first example in this example, the obtained results show that the rate of convergence increases in the linear case, but because of the effect of the condition number $\mathcal{CN}_2$, the convergence rate of the quadratic case is not gradually increasing like the linear case.

Table 2
Maximum absolute errors, ratio of error and CPU times used for different values of N.

N	Linear ($q = 1$)			Quadratic ($q = 2$)		
	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time
5	3.26×10^{-3}	–	0.45	1.72×10^{-3}	–	0.35
9	8.44×10^{-4}	3.87	0.46	2.11×10^{-4}	8.12	0.39
17	2.16×10^{-4}	3.90	0.51	1.83×10^{-4}	1.15	0.42
33	5.32×10^{-5}	4.07	0.62	4.10×10^{-5}	4.47	0.51
65	1.27×10^{-5}	4.18	1.09	2.63×10^{-5}	1.56	0.68
129	3.19×10^{-6}	3.98	2.83	6.48×10^{-6}	4.06	1.42

Table 3
Maximum absolute errors, ratio of error and CPU times used for different values of N.

N	Linear ($q = 1$)			Quadratic ($q = 2$)		
	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time
5	2.02×10^{-4}	–	0.39	1.34×10^{-4}	–	0.38
9	5.07×10^{-5}	3.98	0.58	1.71×10^{-5}	7.86	0.55
17	1.26×10^{-5}	4.01	0.69	2.15×10^{-6}	7.94	0.55
33	2.97×10^{-6}	4.25	0.76	2.64×10^{-7}	8.12	0.79
65	6.88×10^{-7}	4.32	1.01	3.22×10^{-8}	8.20	1.21
129	1.51×10^{-7}	4.56	2.19	3.87×10^{-9}	8.33	3.31

Table 4
Maximum absolute errors, ratio of error and CPU time used for different values of N.

N	Linear ($q = 1$)			Quadratic ($q = 2$)		
	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time
5	2.83×10^{-3}	–	0.42	2.21×10^{-4}	–	0.58
9	7.22×10^{-4}	3.92	0.68	3.41×10^{-5}	6.49	0.79
17	2.25×10^{-4}	3.21	0.76	9.98×10^{-6}	3.41	1.16
33	6.93×10^{-5}	3.25	0.87	3.26×10^{-6}	3.06	1.85
65	2.19×10^{-5}	3.16	0.95	1.87×10^{-6}	1.75	2.15
129	6.58×10^{-6}	3.33	1.40	1.26×10^{-6}	1.48	2.91

6.1.3. *Example 3*

In this example, let us consider the following equation

$$\acute{u}(x) = u(x) + \frac{1}{x+1} - \frac{1}{2}x - \ln(x+1) + \frac{1}{(\ln(2))^2}\int_0^1 \frac{x}{t+1}u(t)\mathrm{d}t,$$
$$u(0) = 0,$$

with the exact solution

$$u(x) = \ln(x+1).$$

The maximum error, CPU times and the rate of convergence are shown in Table 3. As we expected from theory, the error increases with the order $\mathcal{O}(h^2)$ in the linear case approximately.

6.2. *Volterra integro-differential equation*

6.2.1. *Example 4*

In this example consider the following integro-differential equation

$$\acute{u}(x) = 1 - \frac{x}{2} + \frac{x\,e^{-x^2}}{2} + \int_0^x x\,t\,e^{-u^2(t)}\mathrm{d}t,$$
$$u(0) = 0.$$

The exact solution of this problem is

$$u(x) = x.$$

Table 4, shows the maximum errors, *CPU* times and the rate of convergence for different values of N.

Table 5
Maximum absolute errors, ratio of error and CPU times used for different values of N.

N	Linear ($q = 1$)			Quadratic ($q = 2$)		
	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time
5	4.25×10^{-3}	–	0.34	4.86×10^{-2}	–	0.45
9	1.34×10^{-3}	3.16	0.53	6.79×10^{-3}	7.15	0.58
17	3.58×10^{-4}	3.75	0.64	9.93×10^{-4}	6.84	0.77
33	8.28×10^{-5}	4.32	0.97	2.13×10^{-4}	4.06	1.27
65	2.35×10^{-5}	3.53	1.08	3.27×10^{-5}	6.50	1.79
129	6.04×10^{-6}	3.82	1.69	5.04×10^{-6}	6.53	2.34

Table 6
Maximum absolute errors, ratio of error and CPU times used for different values of N.

N	Linear ($q = 1$)			Quadratic ($q = 2$)		
	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time	$\|\mathbf{e}\|_{L_\infty}$	Ratio	Time
5	5.84×10^{-3}	–	0.45	1.96×10^{-4}	–	0.46
9	1.75×10^{-3}	3.33	0.57	2.42×10^{-5}	8.11	0.58
17	4.88×10^{-4}	3.59	0.65	3.48×10^{-6}	6.94	0.73
33	1.30×10^{-4}	3.74	0.72	6.55×10^{-7}	5.31	0.87
65	3.21×10^{-5}	4.05	1.68	5.75×10^{-7}	1.14	1.66
129	8.19×10^{-6}	3.93	4.83	6.05×10^{-7}	0.95	5.61

6.2.2. Example 5

Let us consider this integro-differential equation

$$\acute{u}(x) = -\frac{x^2}{3} + \frac{4}{3}\, e^{-u(x)} + \int_1^x \frac{1}{x}\, t\, e^{u(t)} \mathrm{d}t,$$
$$u(1) = 0,$$

where the unknown solution is

$$u(x) = \log(x).$$

The maximum errors, *CPU* times and rate of convergence are reported in Table 5. The ratio of errors increases approximately in the linear case and degrades in the quadratic case as it is expected.

6.2.3. Example 6

As the last example, we consider the following Volterra integro-differential equation

$$\acute{u}(x) = 2x - \frac{1}{2}\sin(x^4) + \int_0^x x^2\, t\, \cos(x^2\, u(t))\mathrm{d}t,$$
$$u(0) = 0,$$

where the unknown solution is

$$u(x) = x^2.$$

In Table 6, the maximum error and ratio estimate the accuracy of the method. The results show the efficiency of the method to approximate the nonlinear integro-differential equations.

7. Concluding remarks

In this paper the moving least square method with the point collocation approach presented for solving the nonlinear Fredholm and Volterra integro-differential equations. Error analysis was provided for sufficiently smooth kernel and source functions. The MLS method is a meshless based technique which does not need any domain discretization for approximation. The method can be easily implemented and its algorithm is simple and efficient to approximate the unknown function. The MLS method is a flexible approach to select the nodal density. The numerical results for different examples are reported which show the efficiency of the method for solving various types of nonlinear integro-differential equations.

Acknowledgments

The authors thank the anonymous reviewers for the constructive comments and suggestions.

References

[1] A. Akyüz, M. Sezer, A Taylor polynomial approach for solving high-order linear Fredholm integro-differential equations in the most general form, Int. J. Comput. Math. 84 (2007) 527–539.
[2] B. Bülbül, M. Sezer, Taylor polynomial solution of hyperbolic type partial differential equations with constant coefficients, Int. J. Comput. Math. 88 (2011) 533–544.
[3] I.P. Streltsov, Approximation of Chebyshev and Legendre polynomials on discrete point set to function interpolation and solving Fredholm integral equations, Comput. Phys. Comm. 126 (2000) 178–181.
[4] K. Maleknejad, M. Tavassoli Kajani, Solving second kind integral equation by Galerkin methods with hybrid Legendre and Block–Pulse functions, Appl. Math. Comput. 145 (2003) 623–629.
[5] N. Bildik, A. Konuralp, S. Yalçnbaş, Comparison of Legendre polynomial approximation and variational iteration method for the solutions of general linear Fredholm integro-differential equations, Comput. Math. Appl. 59 (2010) 1909–1917.
[6] S.H. Wang, J.H. He, Variational iteration method for solving integro-differential equations, Phys. Lett. A 367 (2007) 188–191.
[7] J.I. Ramos, Iterative and non-iterative methods for non-linear Volterra integro-differential equations, Appl. Math. Comput. 214 (2009) 287–296.
[8] A. Feldstein, J.R. Sopka, Numerical methods for nonlinear Volterra integro-differential equations, SIAM J. Numer. Anal. 11 (1974) 826–846.
[9] M. Tavassoli Kajani, M. Ghasemi, E. Babolian, Numerical solution of linear integro-differential equation by using sine-cosine wavelets, Appl. Math. Comput. 180 (2006) 569–574.
[10] K. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, 1997.
[11] A.M. Wazwaz, A First Course in Integral Equations, World Scientific Publishing, 1997.
[12] P. Lancaster, K. Salkauskas, Surface generated by moving least squares methods, Math. Comp. 37 (1981) 41–58.
[13] D. Levin, The approximation power of moving least-squares, Math. Comp. 67 (1998) 17–31.
[14] P. Breitkopf, H. Naceur, A. Rassineux, P. Villon, Moving least squares response surface approximation: formulation and metal forming applications, Comput. Struct. 83 (2005) 11–28.
[15] C. Bucher, M. Macke, T. Most, Approximate response functions in structural reliability analysis, in: P. Spanos, G. Deodatis (Eds.), Proceedings of the Fifth Computational Stochastic Mechanics Conference, Rhodos, Greece, June 21–23, 2006, Balkema, 2006.
[16] B. Nayroles, G. Touzot, P. Villon, Generalizing the FEM: diffuse approximations and diffuse elements, Comput. Mech. 10 (1992) 7–18.
[17] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, Internat. J. Numer. Methods Engrg. 37 (1994) 29–56.
[18] S.N. Atluri, T. Zhu, A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics, Comput. Mech. 22 (1998) 17–27.
[19] 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.
[20] M. Dehghan, D. Mirzaei, Meshless local Petrov–Galerkin (MLPG) method for the unsteady magnetohydrodynamic (MHD) flow through pipe with arbitrary wall conductivity, Appl. Numer. Math. 59 (2009) 1043–1058.
[21] M. Dehghan, A. Ghesmati, Combination of meshless local weak and strong (MLWS) forms to solve the two dimensional hyperbolic telegraph equation, Eng. Anal. Bound. Elem. 34 (2010) 324–336.
[22] M. Dehghan, R. Salehi, A boundary-only meshless method for numerical solution of the Eikonal equation, Comput. Mech. 47 (2011) 283–294.
[23] M. Dehghan, R. Salehi, The solitary wave solution of the two-dimensional regularized long-wave equation in fluids and plasmas, Comput. Phys. Comm. 182 (2011) 2540–2549.
[24] M. Tatari, M. Dehghan, On the solution of the non-local parabolic partial differential equations via radial basis functions, Appl. Math. Model. 33 (2009) 1729–1738.
[25] D. Mirzaei, M. Dehghan, A meshless based method for solution of integral equations, Appl. Numer. Math. 60 (2010) 245–262.
[26] D. Shepard, A two-dimensional interpolation function for irregularly spaced points, in: Proc. 23rd Nat. Conf. ACM, ACM Press, New York, 1968, pp. 517–524.
[27] D.H. McLain, Drawing contours from arbitary data points, Computing 17 (1974) 318–324.
[28] D.H. McLain, Two dimensional interpolation from random data, Computing 19 (1976) 178–181.
[29] R. Franke, G. Nielson, Smooth interpolation of large sets of scattered data, Internat. J. Numer. Methods Engrg. 15 (1980) 1691–1704.
[30] M.G. Armentano, R.G. Durán, Error estimates for moving least square approximations, Appl. Numer. Math. 37 (2001) 397–416.
[31] M.G. Armentano, Error estimates in Sobolev spaces for moving least square approximations, SIAM J. Numer. Anal. 39 (2001) 38–51.
[32] C. Zuppa, Error estimates for moving least square approximations, Bull. Braz. Math. Soc. (N.S.) 34 (2003) 231–249.
[33] X. Li, J. Zhu, A Galerkin boundary node method and its convergence analysis, J. Comput. Appl. Math. 230 (2009) 314–328.
[34] C. Zuppa, Good quality point sets and error estimates for moving least square approximations, Appl. Numer. Math. 47 (2003) 575–585.
[35] I. Babuška, U. Banerjee, J.E. Osborn, Q. Zhang, Effect of numerical integration on meshless methods, Comput. Methods Appl. Mech. Engrg. 198 (2009) 2886–2897.
[36] N.S. Kambo, Error of the Newton–Cotes and Gauss–Legendre quadrature formulas, Math. Comp. 24 (1970) 261–269.