

Article

# Meshless Local Petrov–Galerkin Formulation of Inverse Stefan Problem via Moving Least Squares Approximation

A. Karami, Saeid Abbasbandy \*  and E. Shivanian

Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin 34149-16818, Iran; karami\_960@yahoo.com (A.K.); e\_shivanian@yahoo.com (E.S.)

\* Correspondence: abbasbandy@ikiu.ac.ir or abbasbandy@yahoo.com; Tel.: +98-9121-305326

Received: 6 November 2019; Accepted: 9 December 2019; Published: 10 December 2019



**Abstract:** In this paper, we study the meshless local Petrov–Galerkin (MLPG) method based on the moving least squares (MLS) approximation for finding a numerical solution to the Stefan free boundary problem. Approximation of this problem, due to the moving boundary, is difficult. To overcome this difficulty, the problem is converted to a fixed boundary problem in which it consists of an inverse and nonlinear problem. In other words, the aim is to determine the temperature distribution and free boundary. The MLPG method using the MLS approximation is formulated to produce the shape functions. The MLS approximation plays an important role in the convergence and stability of the method. Heaviside step function is used as the test function in each local quadrature. For the interior nodes, a meshless Galerkin weak form is used while the meshless collocation method is applied to the the boundary nodes. Since MLPG is a truly meshless method, it does not require any background integration cells. In fact, all integrations are performed locally over small sub-domains (local quadrature domains) of regular shapes, such as intervals in one dimension, circles or squares in two dimensions and spheres or cubes in three dimensions. A two-step time discretization method is used to deal with the time derivatives. It is shown that the proposed method is accurate and stable even under a large measurement noise through several numerical experiments.

**Keywords:** free boundary problems; moving boundary; weak form; meshless local Petrov–Galerkin (MLPG) method; moving least squares (MLS) approximation

**MSC:** 65N21; 80A22

## 1. Introduction

A free boundary problem (FBP) is a partial differential equation with initial and boundary conditions, in which a part of the boundary of domain, called a free boundary, is unknown at the outset of the problem. FBPs have many applications in science and engineering. FBPs usually happen in phase separation problems, which can be either stationary or moving free boundaries. The Stefan problem is one kind of the free boundary problems which describes the process of melting and solidification. In this paper, the numerical solution of these problems are considered. The determination of the temperature function and the position of the free boundary are desired [1–5]. The existence and uniqueness of the solution to these problems are investigated in References [2,3,5]. In recent years, several methods have been employed for solving the Stefan problems numerically, such as the homotopy analysis method [6,7], Lie-group shooting method [8], finite difference and finite element methods [9] and the variational iteration method [10]. Grzymkowski and Slota [11,12] applied the Adomian decomposition method (ADM) to solve one-phase Stefan problems and Slota [13] used the

homotopy perturbation method for one-phase inverse Stefan problems. In Reference [14] the method of fundamental solutions was applied to the one-dimensional Stefan problem.

For many years, the finite element method (FEM) has been considered a standard and effective technique for numerically solving many applied problems in science and engineering [15,16]. Due to several limitations, these techniques cannot solve some of the complex problems of today's world. For this reason, the development and formulation of new and effective numerical techniques in recent years has been an interesting field for some engineers and mathematicians. In recent years, meshless methods have gained considerable attention, in engineering and applied mathematics. Flexibility and simplicity are the advantages of these methods. Meshless methods overcome the shortcomings of the mesh-based techniques [17]. In these methods, a system of algebraic equations is created using a set of scattered nodes—called field nodes—within the domain and its boundary for representation (but not for discretization) the whole domain of the problem and its boundary, therefore it is not necessary to use a predefined mesh for the domain discretization.

Meshless methods are generally divided into three categories. The first category includes methods that use integration and are based on weak forms of PDEs, such as the element free Galerkin method [18–24]. The second methods are based on the strong forms of PDEs and use integration, for example, the meshless collocation method based on radial basis functions (RBFs) [25–28] are in this category. The third category is a set of methods based on the combination of weak forms and strong forms. The meshless methods based on strong form are truly meshless methods and the implementation of these methods are usually simple. They are also computationally efficient. In spite of several advantages, they also have some shortcomings, such as numerical instability and less accuracy.

The meshless weak form methods are those that use the global and local weak form. The stability and accuracy of these methods make them more attractive. In these methods, the global weak forms are used and numerical integrations are carried out on the global background cells in solving the algebraic equations. In meshless local weak form methods, it does not require any background integration cells for field nodes. The meshless local Petrov–Galerkin (MLPG) method [29–40] is based on the local weak form of PDEs. In the MLPG method, the numerical integrations are performed over a local small sub-domain defined for each node. The local sub-domains usually have a regular shape, such as interval, circle, square, sphere, cube, and so forth.

The moving least square (MLS) approximations have an important role in using the MLPG method. By considering a local sub-domain for each field node, the MLS approximates the solving function at each field node. In this paper, a kind of MLPG method using the MLS approximation is applied for numerically solving the Stefan problem.

The layout of the paper is as follows. In the next section, we give the formulation of the inverse Stefan problem. Section 3 briefly describes the MLS approximation. In Section 4 we present the time discretization of the problem. In Section 5 the local weak form formulation of the discretized problem is presented. We present the MLPG discretization of the problem in Section 6. Numerical examples are given and solved to observe the performance of the proposed method in Section 7. At last, we give a conclusion in Section 8.

## 2. Statement of the Problem

Consider the following heat conduction equation

$$\frac{\partial v(y, t)}{\partial t} = \alpha \frac{\partial^2 v(y, t)}{\partial y^2}, \quad 0 < y < s(t), \quad 0 < t < T, \quad (1)$$

subject to the initial condition:

$$v(y, 0) = f(y), \quad 0 < y < s_0, \quad s_0 = s(0), \quad (2)$$

and boundary condition

$$v(0, t) = g(t), \tag{3}$$

where  $\alpha$  denotes the thermal diffusivity and  $v(y, t)$ ,  $t$  and  $x$  denote the temperature, time and spatial location, respectively.

On the free boundary  $s(t)$  we have

$$v(s(t), t) = h(t), \tag{4}$$

$$\beta \frac{\partial v(y, t)}{\partial y} \Big|_{y=s(t)} = \kappa \frac{ds(t)}{dt}, \tag{5}$$

where  $\alpha$  is the thermal diffusivity,  $\beta$  is the thermal conductivity,  $\kappa$  is the latent heat of fusion per unit volume,  $h(t)$  is the temperature of the phase change,  $v(y, t)$  is temperature and  $t$  and  $x$  refer to time and spatial location, respectively. Equation (4) represents the continuity of temperature and Equation (5) is the Stefan condition.

In this problem, we try to find  $v(y, t)$ , the temperature distribution in given domain and  $s(t)$ , the free boundary. This is a nonlinear problem due to Stefan condition [4].

By using the change of variable, we the free boundary problem is transformed into a fixed boundary problem.

Let

$$x = \frac{y}{s(t)}, \quad u(x, t) = v(y, t). \tag{6}$$

Then Equations (1)–(5) are changed to

$$\frac{\partial u(x, t)}{\partial t} = \alpha \frac{1}{s^2(t)} \frac{\partial^2 u(x, t)}{\partial x^2} + x \frac{s'(t)}{s(t)} \frac{\partial u(x, t)}{\partial x}, \quad 0 < x < 1, \quad 0 < t < T, \tag{7}$$

$$u(x, 0) = f(s_0 x), \quad 0 < x < 1, \quad s_0 = s(0), \tag{8}$$

$$u(0, t) = g(t), \tag{9}$$

$$u(1, t) = h(t), \tag{10}$$

$$\beta \frac{\partial u(1, t)}{\partial x} = \kappa s(t) s'(t). \tag{11}$$

In this paper, an approach based on the MLPG method and MLS approximations is applied to the Equation (7), which is subjected to the initial condition (8) and over-specified boundary conditions (9)–(11).

### 3. The MLS Approximation Technique

In this section, the formulation of MLS approximation is explained. The trial functions at each node is represented by the MLS approximation. Consider the sub-domain  $\Omega_s$ , with the boundary  $\partial\Omega_s$ , of problem global domain  $\Omega$  around point  $x$ . In fact,  $\Omega_s$  is the domain of definition (or support) of the MLS approximation for the trial function at  $x$ . Let  $\mathbf{q}^t(x) = [q_1(x), q_2(x), \dots, q_m(x)]$  be a complete monomial basis in the space coordinate  $x$ . For example, the linear basis for one-dimensional is

$$\mathbf{q}^t(x) = \{1, x\}, \quad m = 2, \tag{12}$$

and the quadratic basis function is

$$\mathbf{q}^t(x) = \{1, x, x^2\}, \quad m = 3. \tag{13}$$

For all  $x$  belong to  $\Omega_s$  the MLS approximation  $u^h(x)$  of  $u$  in  $\Omega_s$ , over a set of random nodes  $x_i$  ( $i = 1, 2, \dots, n$ ) located in  $\Omega_s$ , is given as

$$u^h(x) = \mathbf{q}^t(x)\lambda(x), \quad \forall x \in \Omega_s, \tag{14}$$

where  $\lambda^t(x) = [\lambda_1(x), \lambda_2(x), \dots, \lambda_m(x)]$  is a vector of coefficients. In order to determine the unknown coefficient vector  $\lambda(x)$ , we define a function  $I(\lambda(x))$  as follows

$$I(\lambda(x)) = \sum_{i=1}^n w_i(x)[q^t(x_i)\lambda(x) - \hat{u}_i]^2 = [\mathbf{Q}\lambda(x) - \hat{\mathbf{u}}]^t \mathbf{W}(x)[\mathbf{Q}\lambda(x) - \hat{\mathbf{u}}], \tag{15}$$

where the matrices  $\mathbf{Q}$  and  $\mathbf{W}(x)$  in Equation (15) are defined as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{q}^t(x_1) \\ \mathbf{q}^t(x_2) \\ \vdots \\ \mathbf{q}^t(x_n) \end{pmatrix}_{n \times m}, \quad \mathbf{W}(x) = \begin{pmatrix} w_1(x) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n(x) \end{pmatrix}.$$

In the above relations,  $w_i(x)$ ,  $i = 1, 2, \dots, n$ , is the weight function corresponding to the node  $x_i$ , so that for each  $x$  in the support of  $w_i(x)$  we have  $w_i(x) > 0$ ,  $n$  is the number of nodes in  $\Omega_s$  for which the weight functions  $w_i(x) > 0$  and  $\hat{\mathbf{u}}^t = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n]$  is the vector of fictitious nodal values. It is necessary to mention that  $\hat{u}_i$ ,  $i = 1, 2, \dots, n$ , are not equal to nodal values  $u_i$ ,  $i = 1, 2, \dots, n$ , of the unknown trial function  $u^h(x)$  in general (Figure 1). The stationarity of  $I(\lambda(x))$  in Equation (15) with respect to  $\lambda(x)$  we have

$$\mathbf{F}(x)\lambda(x) = \mathbf{G}(x)\hat{\mathbf{u}}, \tag{16}$$

where  $\mathbf{F}(x)$  and  $\mathbf{G}(x)$  are matrices defined as follows

$$\mathbf{F}(x) = \mathbf{Q}^t \mathbf{W}(x) \mathbf{Q} = \mathbf{G}(x) \mathbf{Q} = \sum_{i=1}^n w_i(x) \mathbf{q}(x_i) \mathbf{q}^t(x_i), \tag{17}$$

$$\mathbf{G}(x) = \mathbf{Q}^t \mathbf{W}(x) = [w_1(x) \mathbf{q}(x_1), w_2(x) \mathbf{q}(x_2), \dots, w_n(x) \mathbf{q}(x_n)]. \tag{18}$$

The MLS approximation is well-defined only when the matrix  $\mathbf{F}$  in Equation (16) is non-singular, that is, if and only if the rank of  $\mathbf{Q}$  equals  $m$ . A necessary condition to have a well-defined MLS approximation is that at least  $m$  weight functions are non-zero (i.e.,  $n > m$ ) for each sample point  $x \in \Omega$ . Computing  $\lambda(x)$  from Equation (16) and substituting it into Equation (14), gives

$$u^h(x) = \Psi^t(x) \cdot \hat{\mathbf{u}} = \sum_{i=1}^n \psi_i(x) \hat{u}_i, \quad x \in \Omega_s, \tag{19}$$

where

$$\Psi^t(x) = \mathbf{q}^t(x) \mathbf{F}^{-1}(x) \mathbf{G}(x) \tag{20}$$

or

$$\psi_i(x) = \sum_{j=1}^m q_j(x) [\mathbf{F}^{-1}(x) \mathbf{G}(x)]_{ji}. \tag{21}$$

The function  $\psi_i$  is usually called the shape function of the MLS approximation corresponding to nodal point  $x_i$ .

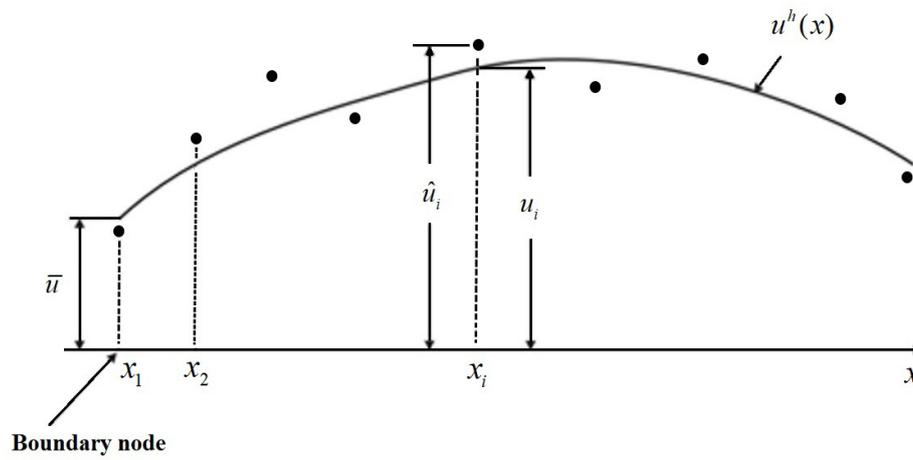
The partial derivative of  $\psi_i(x)$  with respect to  $x$  is defined as

$$\psi_{i,x} = \sum_{j=1}^m [q_{j,x} (\mathbf{F}^{-1} \mathbf{G})_{ji} + q_j (\mathbf{F}^{-1} \mathbf{G}_{,x} + \mathbf{F}_{,x}^{-1} \mathbf{G})_{ji}], \tag{22}$$

where  $(\mathbf{F}^{-1})_{,x} = \mathbf{F}_{,x}^{-1} = -\mathbf{F}^{-1}\mathbf{F}_{,x}\mathbf{F}^{-1}$  and  $(\cdot)_{,x}$  denotes the derivative with respect to  $x$ . In this paper, the following Gaussian weight function is used

$$w_i(x) = \begin{cases} \frac{\exp[-(\frac{l_i}{c_i})^2] - \exp[-(\frac{r_s}{c_i})^2]}{1 - \exp[-(\frac{r_s}{c_i})^2]}, & 0 \leq l_i \leq r_s, \\ 0, & l_i \geq r_s, \end{cases}$$

where  $l_i = \|x - x_i\|$ ,  $c_i$  is a constant controlling the shape of the weight function  $w_i(x)$  and  $r_s$  is the size of the support domain.  $r_s$  must be chosen large enough to have sufficient number of nodes covered in the domain of definition of every sample point ( $n \geq m$ ) to ensure the regularity of  $\mathbf{F}$ .



**Figure 1.** The distinction between the nodal values  $u_i$  of the trial function  $u^h(x)$ , and the undetermined fictitious nodal values  $\hat{u}_i$ .

#### 4. The Time Discretization of the Problem

We use the following finite difference approximations for the time derivative operators

$$\frac{\partial u(x, t)}{\partial t} \cong \frac{1}{\Delta t} (u^{k+1}(x) - u^k(x)), \tag{23}$$

$$s'(t) \cong \frac{1}{\Delta t} (s^{k+1} - s^k), \tag{24}$$

where  $u^j(x) = u(x, t_j)$ ,  $s^j = s(t_j)$ ,  $t_j = t_0 + j\Delta t$ ,  $j = 0, 1, \dots, M$  and  $\Delta t = \frac{T}{M}$ . Also by using the Crank–Nicolson technique, we have the following approximations:

$$\frac{1}{s^2(t)} \frac{\partial^2 u(x, t)}{\partial x^2} \cong \frac{1}{2} \left( \frac{1}{(s^{k+1})^2} u_{xx}^{k+1} + \frac{1}{(s^k)^2} u_{xx}^k \right), \tag{25}$$

$$\frac{1}{s(t)} \frac{\partial u(x, t)}{\partial x} \cong \frac{1}{2} \left( \frac{1}{s^{k+1}} u_x^{k+1} + \frac{1}{s^k} u_x^k \right). \tag{26}$$

Using the above approximations, Equations (7) and (11) can be respectively written as:

$$\frac{u^{k+1} - u^k}{\Delta t} = \frac{\alpha}{2} \left( \frac{1}{(s^{k+1})^2} u_{xx}^{k+1} + \frac{1}{(s^k)^2} u_{xx}^k \right) + \frac{s^{k+1} - s^k}{2\Delta t} x \left( \frac{1}{s^{k+1}} u_x^{k+1} + \frac{1}{s^k} u_x^k \right), \tag{27}$$

$$\frac{s^{k+1} - s^k}{\Delta t} = \frac{\beta}{\kappa} \frac{1}{s^k} u_x^k(1). \tag{28}$$

Suppose that  $\lambda = \frac{2}{\Delta t}$ , then we have

$$\lambda(u^{k+1} - u^k) = \alpha \left( \frac{1}{(s^{k+1})^2} u_{xx}^{k+1} + \frac{1}{(s^k)^2} u_{xx}^k \right) + \frac{\lambda}{2} \left( 1 - \frac{s^k}{s^{k+1}} \right) x u_x^{k+1} - \frac{\lambda}{2} \left( 1 - \frac{s^{k+1}}{s^k} \right) x u_x^k, \quad (29)$$

$$s^{k+1} = s^k + \frac{\beta}{\kappa} \Delta t \left( \frac{1}{s^k} u_x^k(1) \right). \quad (30)$$

### 5. The Local Weak Form Formulation

Let  $\Omega_q^i$  be a sub-domain associated with the nodal point  $x_i, i = 1, 2, \dots, N$ , (called local quadrature cell) in the global domain  $\Omega$ .  $\Omega_q^i, i = 1, 2, \dots, N$ , overlap each other and union of them cover the whole global domain  $\Omega$ . In this paper  $\Omega_q^i$  are intervals centered at  $x_i$  of radius  $r_q$ . By applying the MLPG method, the local weak form is obtained over local quadrature cells  $\Omega_q^i$ . For each node  $x_i \in \Omega_q^i$  the local weak form of Equation (29) is represented as follows

$$\begin{aligned} \lambda \int_{\Omega_q^i} (u^{k+1} - u^k) v(x) dx &= \int_{\Omega_q^i} \alpha \left( \frac{1}{(s^{k+1})^2} u_{xx}^{k+1} + \frac{1}{(s^k)^2} u_{xx}^k \right) v(x) dx + \\ \frac{\lambda}{2} \int_{\Omega_q^i} \left( \left( 1 - \frac{s^k}{s^{k+1}} \right) x u_x^{k+1} \right) v(x) dx &- \frac{\lambda}{2} \int_{\Omega_q^i} \left( \left( 1 - \frac{s^{k+1}}{s^k} \right) x u_x^k \right) v(x) dx, \end{aligned} \quad (31)$$

where the Heaviside step function [41,42]

$$v(x) = \begin{cases} 1, & \mathbf{x} \in \Omega_q^i, \\ 0, & \mathbf{x} \notin \Omega_q^i, \end{cases} \quad (32)$$

is used as the test function. Applying the integration by parts in Equation (31) the following local weak form equation is obtained:

$$\begin{aligned} \lambda \int_{\Omega_q^i} u^{(k+1)} dx - \frac{\lambda}{2} \int_{\Omega_q^i} \left( \left( 1 - \frac{s^k}{s^{k+1}} \right) x u_x^{k+1} \right) dx - \frac{\alpha}{(s^{k+1})^2} u_x^{k+1} \Big|_{\partial \Omega_q^i} &= \\ \lambda \int_{\Omega_q^i} u^{(k)} dx - \frac{\lambda}{2} \int_{\Omega_q^i} \left( \left( 1 - \frac{s^{k+1}}{s^k} \right) x u_x^k \right) dx + \frac{\alpha}{(s^k)^2} u_x^k \Big|_{\partial \Omega_q^i}, \end{aligned} \quad (33)$$

where  $\partial \Omega_q^i$  is the boundary of  $\Omega_q^i$ .

### 6. MLPG Discretization

In this section, we obtain a system of algebraic equations from discretization of the Equation (33), by employing MLS approximation. Equation (33) is discretized for this purpose. Consider the  $N$  regularly points  $x_i, i = 1, 2, \dots, N$ , in the domain of the problem and its boundary such that  $x_{i+1} - x_i = h$ . Suppose that  $u(x_i, t_k)$ , is determined and  $u(x_i, t_{k+1})$ , is unknown for  $i = 1, 2, \dots, N$ . In order to determine the  $N$  unknown quantities  $u(x_i, t_{k+1})$  we need to have  $N$  equations. For interior nodes  $x_i$  of the domain  $\Omega$ , by replacing MLS approximation Formula (19) in the Equation (33), the following discrete equations are obtained

$$\begin{aligned} \lambda \sum_{j=1}^N \left( \int_{\Omega_q^i} \phi_j(x) dx \right) u_j^{(k+1)} - \frac{\lambda}{2} \left( 1 - \frac{s^k}{s^{k+1}} \right) \sum_{j=1}^N \left( \int_{\Omega_q^i} x \phi_j'(x) dx \right) u_j^{(k+1)} - \\ \frac{\alpha}{(s^{k+1})^2} \sum_{j=1}^N \phi_j'(x) u_j^{(k+1)} \Big|_{\partial \Omega_q^i} &= \lambda \sum_{j=1}^N \left( \int_{\Omega_q^i} \phi_j(x) dx \right) u_j^{(k)} - \\ \frac{\lambda}{2} \left( 1 - \frac{s^{k+1}}{s^k} \right) \sum_{j=1}^N \left( \int_{\Omega_q^i} x \phi_j'(x) dx \right) u_j^{(k)} + \frac{\alpha}{(s^k)^2} \sum_{j=1}^N \phi_j'(x) u_j^{(k)} \Big|_{\partial \Omega_q^i}. \end{aligned} \quad (34)$$

For boundary nodes  $x = 0$  and  $x = 1$  we set

$$u^{k+1}(0) = g(t_{k+1}), \tag{35}$$

$$u^{k+1}(1) = h(t_{k+1}), \tag{36}$$

together with Equation (30) for which its discrete form is written as

$$s^{k+1} = s^k + \frac{\beta}{\kappa} \frac{\Delta t}{s^k} \sum_{j=1}^N \phi'_j(1) u_j^{(k)}. \tag{37}$$

The matrix form of Equations (34)–(36) for all  $N$  nodal points can be represented as follows

$$\begin{aligned} & \left[ \lambda \sum_{j=1}^N a_{ij} - \frac{\lambda}{2} \left( 1 - \frac{s^k}{s^{k+1}} \right) \sum_{j=1}^N b_{ij} - \frac{\alpha}{(s^{k+1})^2} \sum_{j=1}^N c_{ij} \right] u_j^{(k+1)} = \\ & \left[ \lambda \sum_{j=1}^N a_{ij} - \frac{\lambda}{2} \left( 1 - \frac{s^{k+1}}{s^k} \right) \sum_{j=1}^N b_{ij} + \frac{\alpha}{(s^k)^2} \sum_{j=1}^N c_{ij} \right] u_j^{(k)}, \end{aligned} \tag{38}$$

where

$$a_{ij} = \int_{\Omega_q^i} \phi_j(x) dx, \quad b_{ij} = \int_{\Omega_q^i} x \phi'_j dx, \quad c_{ij} = \phi'_j(x)|_{\partial\Omega_q^i}. \tag{39}$$

Assuming that

$$\begin{aligned} A_{ij} &= \lambda a_{ij} - \frac{\lambda}{2} \left( 1 - \frac{s^k}{s^{k+1}} \right) b_{ij} - \frac{\alpha}{(s^{k+1})^2} c_{ij}, \\ B_{ij} &= \lambda a_{ij} - \frac{\lambda}{2} \left( 1 - \frac{s^{k+1}}{s^k} \right) b_{ij} + \frac{\alpha}{(s^k)^2} c_{ij}, \end{aligned}$$

and  $U = (u_i)_{N \times 1}$ , Equations (37) and (38) yield the following system of equations

$$\begin{cases} s^{k+1} = s^k + \frac{\beta}{\kappa} \frac{\Delta t}{s^k} \sum_{j=1}^N \phi'_j(1) u_j^{(k)}, \\ AU^{(k+1)} = BU^{(k)}. \end{cases} \tag{40}$$

According to the boundary conditions (35) and (36) we have for each step

$$A_{11} = A_{NN} = 1, \quad \forall j \neq 1 : A_{1j} = 0, \quad \forall j \neq N : A_{Nj} = 0, \tag{41}$$

$$B_{11} = B_{NN} = 1, \quad \forall j \neq 1 : B_{1j} = 0, \quad \forall j \neq N : B_{Nj} = 0, \tag{42}$$

$$u_1^{k+1} = g(t_{k+1}), \quad u_N^{k+1} = h(t_{k+1}). \tag{43}$$

At the first step, when  $k = 0$ , due to the initial conditions, we have the following assumptions

$$U^{(0)} = [f(s_0x_1), f(s_0x_2), \dots, f(s_0x_N)]^t, \tag{44}$$

$$s(0) = s_0. \tag{45}$$

### 7. Numerical Experiments

In this section, we test the described meshless method with two examples. In numerical computations, the input Cauchy data are considered with noise

$$\tilde{g}(t_i) = g(t_i)(1 + \delta R(i)), \tag{46}$$

where  $\delta$  denotes the level of noise, and  $R(i)$  are random numbers in  $[-1, 1]$ . In these examples, the domain integrals are approximated using the 4 points Gaussian quadrature rule. In order to investigate the accuracy of computed approximations and the efficiency of the presented method, the following root mean square (RMS) error and absolute error formulas are applied

$$RMS = \sqrt{\frac{\sum_{i=1}^N (u_{exact}(x_i, t_j) - u_{approx}(x_i, t_j))^2}{N}}, \tag{47}$$

$$Absolute\ error\ u = |u_{exact}(x_i, t_j) - u_{approx}(x_i, t_j)|, \tag{48}$$

$$Absolute\ error\ s = |s_{exact}(t_j) - s_{approx}(t_j)|. \tag{49}$$

In implementing the meshless local weak form, each local quadrature domain  $\Omega_q^i$  is taken as interval centered at  $x_i$  of radius  $r_q = 0.7h$  where  $h = x_{i+1} - x_i, i = 0, 1, 2, \dots, N - 1$ . Also the radius of support domain  $\Omega_s$  is  $r_s = 4r_q$  and the quadratic basis functions (13) is used in Equation (14).

**Example 1.** We consider the problem (1)–(5) with  $\alpha = 1, T = 1, \beta = -1, \kappa = 1$  and

$$f(y) = \exp(1 - y), \quad g(t) = \exp(1 + t), \quad h(t) = 1.$$

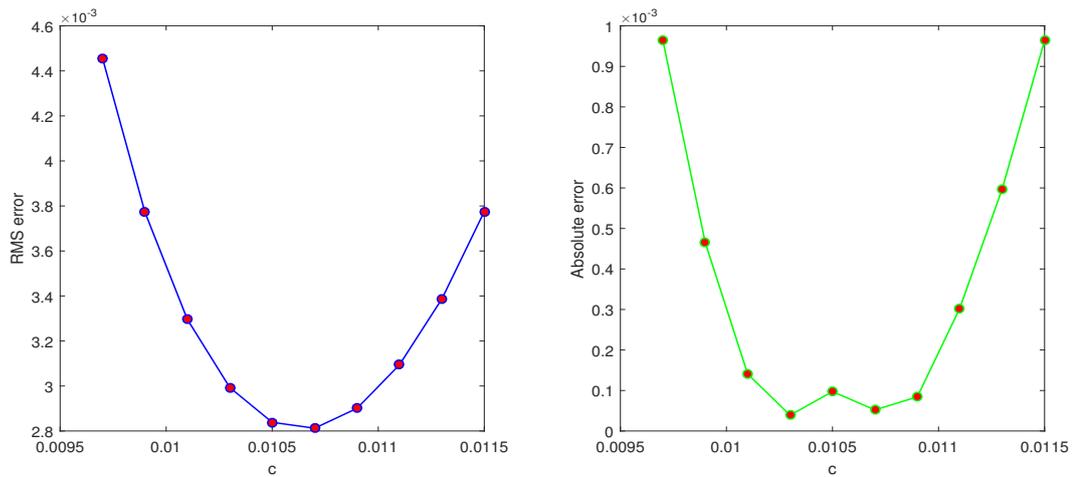
The analytical solutions of problem are

$$v(y, t) = \exp(1 - y + t), \quad s(t) = t + 1,$$

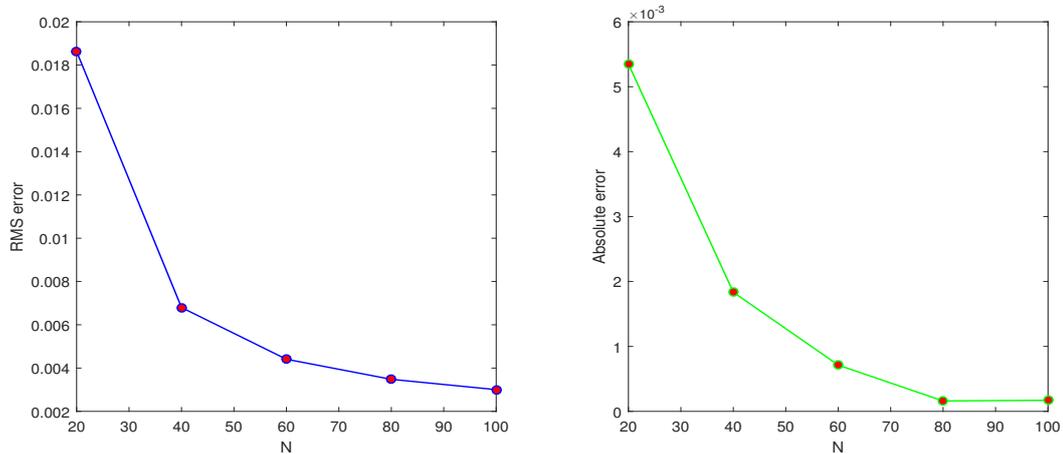
and

$$u(x, t) = v(xs(t), t) = \exp(1 - x(t + 1) + t), \quad s(t) = t + 1.$$

The results of using the proposed method are obtained with  $\Delta t = 0.01, h = 0.01$ . Figure 2 presents the RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  versus the shape parameter  $c$  at  $t = 1$ . For other values of  $t$  the results are almost the same. In this example, the interval (0.0097, 0.01155) is suggested for choosing  $c$ . It is necessary to note that, ill conditioning occurs by increasing  $c$ . Hereafter, we fix it at  $c = 1.1h = 0.011$ . In Figure 3 RMS versus  $N$  is plotted at  $t = 1$ . It can be seen that in this figure, the error values decrease by increasing  $N$ . Values of RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  with  $\delta = 0$  and  $\delta = 0.1$  are presented in Table 1. It shows that the numerical results are more accurate when there exists no noise on the input data. Under a noise level  $\delta = 0.1$ , the numerical result obtained by the MLPG method is also acceptable. Exact solution, numerical solution and Absolute errors for  $u(x, t)$  and  $s(t)$  are plotted in Figures 4–6.



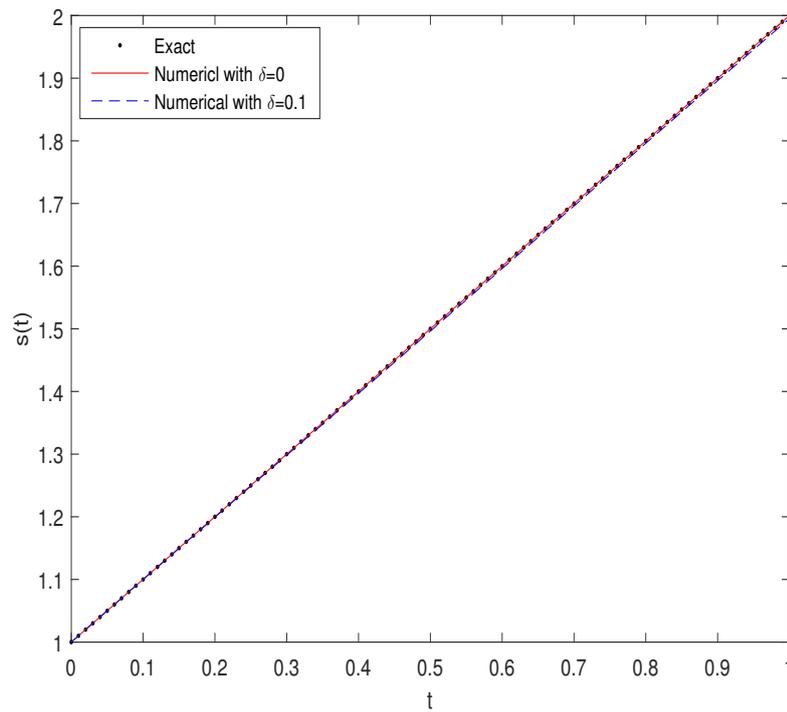
**Figure 2.** Example 1: Diagram of RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  at  $t = 1$  versus shape parameter  $c$  by the meshless local Petrov–Galerkin (MLPG) method when  $h = 0.01, \Delta t = 0.01$ .



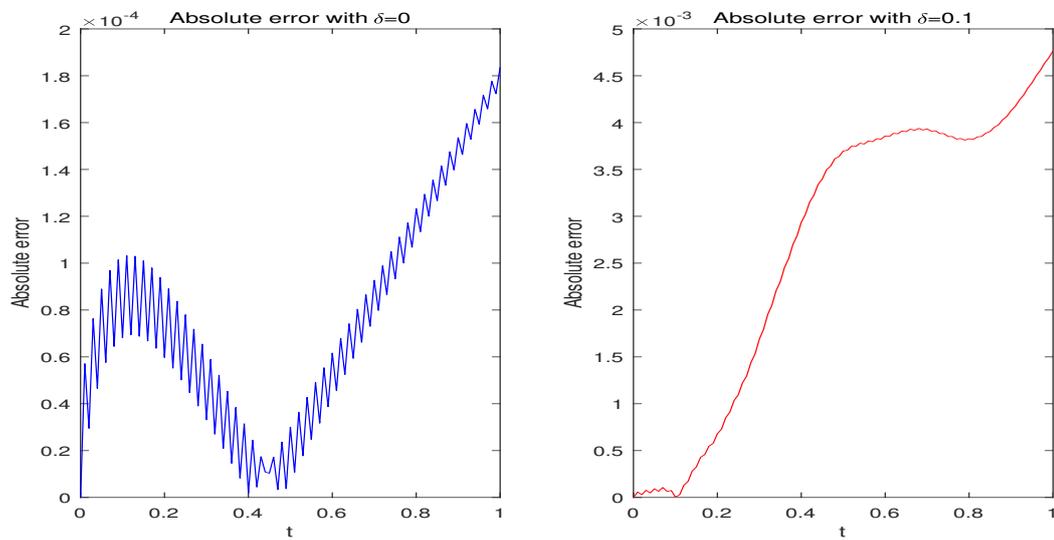
**Figure 3.** Example 1: Diagram of RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  at  $t = 1$  versus  $N$  by the MLPG method when  $\Delta t = 0.01, c = 0.011$ .

**Table 1.** Example 1: RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  without noise and with level noise  $\delta = 0.1$  for different time levels when  $h = 0.01, \Delta t = 0.01, c = 0.011$ .

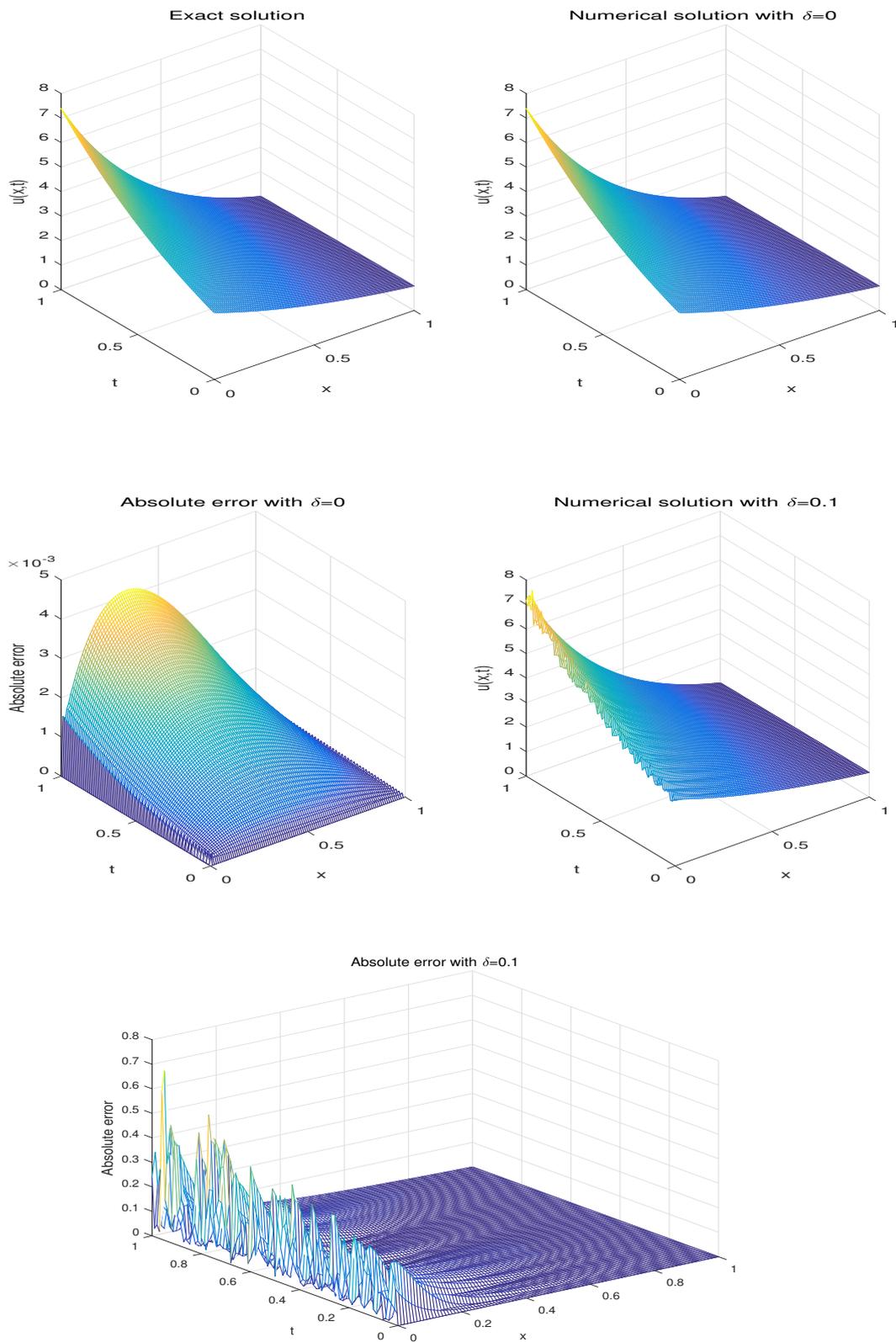
$t$	RMS Error for $u(x, t)$ with $\delta = 0$	RMS Error for $u(x, t)$ with $\delta = 0.1$	Absolute Error for $s(t)$ with $\delta = 0$	Absolute Error for $s(t)$ with $\delta = 0.1$
0	0	0	0	0
0.1	$5.0608 \times 10^{-4}$	$2.184549 \times 10^{-2}$	$6.815 \times 10^{-5}$	$4.45 \times 10^{-6}$
0.2	$7.6318 \times 10^{-4}$	$3.766709 \times 10^{-2}$	$5.977 \times 10^{-5}$	$6.7943 \times 10^{-4}$
0.3	$9.7185 \times 10^{-4}$	$5.135603 \times 10^{-2}$	$3.313 \times 10^{-5}$	$1.68936 \times 10^{-3}$
0.4	$1.18114 \times 10^{-3}$	$2.942010 \times 10^{-2}$	$1.92 \times 10^{-6}$	$2.93583 \times 10^{-3}$
0.5	$1.40740 \times 10^{-3}$	$3.451082 \times 10^{-2}$	$3.002 \times 10^{-5}$	$3.69751 \times 10^{-3}$
0.6	$1.65791 \times 10^{-3}$	$8.867908 \times 10^{-2}$	$6.163 \times 10^{-5}$	$3.85729 \times 10^{-3}$
0.7	$1.93721 \times 10^{-3}$	$6.922609 \times 10^{-2}$	$9.272 \times 10^{-5}$	$3.93321 \times 10^{-3}$
0.8	$2.24902 \times 10^{-3}$	$4.306378 \times 10^{-2}$	$1.2333 \times 10^{-4}$	$3.82775 \times 10^{-3}$
0.9	$2.59697 \times 10^{-3}$	$4.763296 \times 10^{-2}$	$1.5360 \times 10^{-4}$	$4.12823 \times 10^{-3}$
1.0	$2.98488 \times 10^{-3}$	$5.332613 \times 10^{-2}$	$1.8367 \times 10^{-4}$	$4.76457 \times 10^{-3}$



**Figure 4.** Example 1: Comparing the approximation solutions with the exact solution  $s(t)$  when  $h = 0.01, \Delta t = 0.01, c = 0.011$ .



**Figure 5.** Example 1: Absolute error for  $s(t)$  without noise and with level noise  $\delta = 0.1$  when  $h = 0.01, \Delta t = 0.01, c = 0.011$ .



**Figure 6.** Example 1: Exact solution, numerical solution and absolute error for  $u(x,t)$  without noise and with level noise  $\delta = 0.1$  when  $h = 0.01$ ,  $\Delta t = 0.01$ ,  $c = 0.011$ .

**Example 2.** In this example, we consider the problem (1)–(5) with  $\alpha = 1, \beta = -1, \kappa = 1, T = 1$  and

$$f(y) = \exp\left(1 - \frac{\sqrt{5}}{5}(1 + y)\right) - 1, \quad g(t) = \exp\left(1 - \frac{\sqrt{5}}{5} + \frac{t}{5}\right) - 1, \quad h(t) = 0.$$

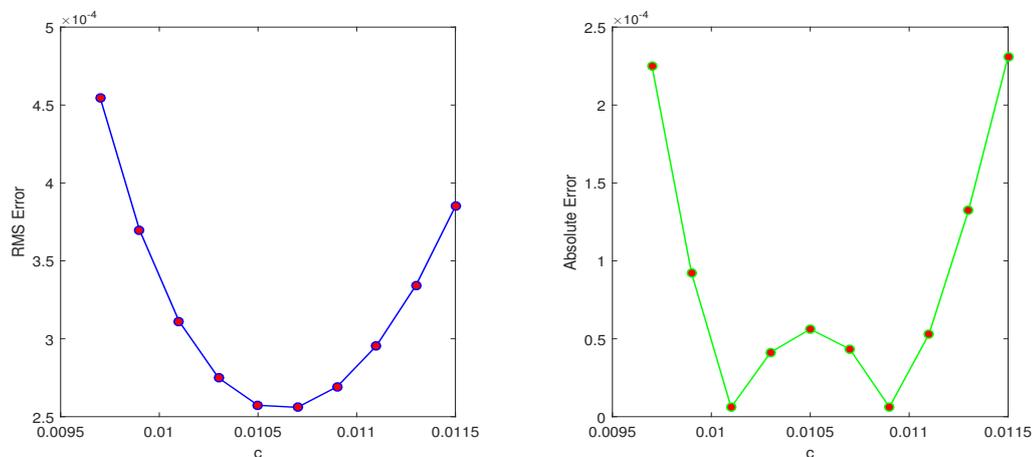
The analytical solutions of problem are given by

$$v(y, t) = \exp\left(1 - \frac{\sqrt{5}}{5}(1 + y) + \frac{t}{5}\right) - 1, \quad s(t) = \frac{t}{\sqrt{5}} + \sqrt{5} - 1,$$

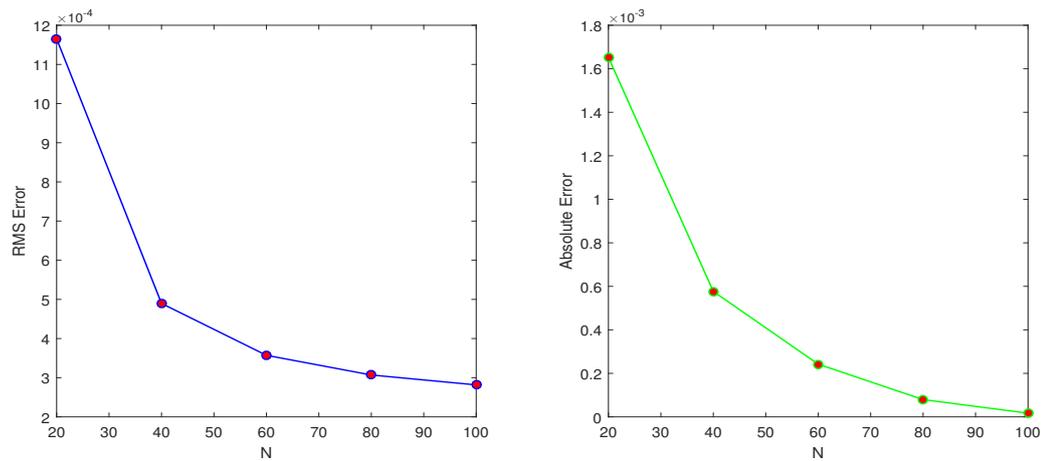
and

$$u(x, t) = v(xs(t), t) = \exp\left(1 - \frac{\sqrt{5}}{5}\left(1 + x\left(\frac{t}{\sqrt{5}} + \sqrt{5} - 1\right)\right) + \frac{t}{5}\right) - 1, \quad s(t) = \frac{t}{\sqrt{5}} + \sqrt{5} - 1.$$

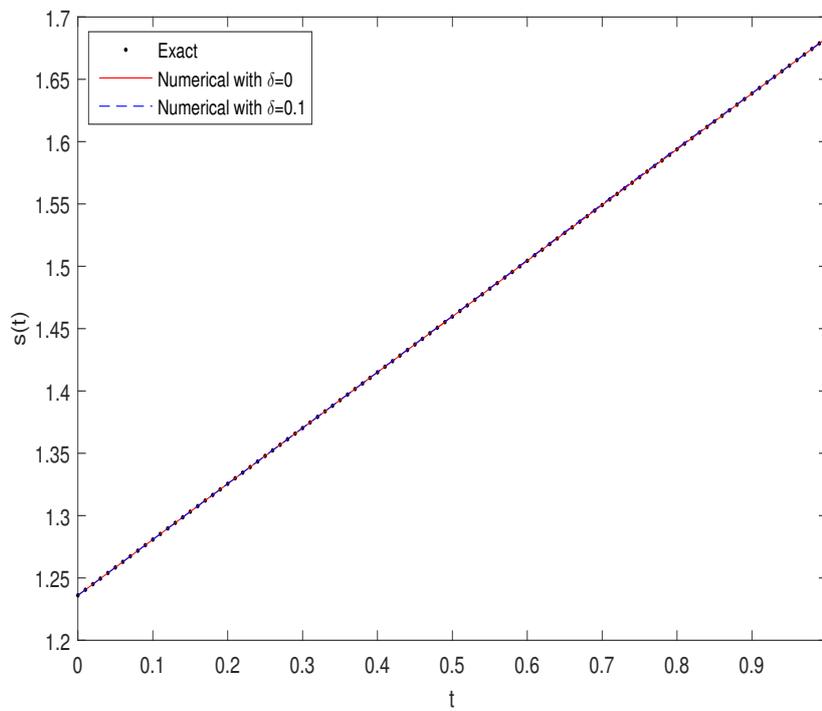
The approximation results of using the proposed method are obtained with  $\Delta t = 0.01, h = 0.01$ . Figure 7 presents the RMS error for  $u(x, t)$  and absolute error for  $s(t)$  versus the shape parameter  $c$  at  $t = 1$ . For other values of  $t$  the results are similar. In this example, the interval  $(0.0097, 0.01155)$  is also suggested for choosing  $c$ . It is necessary to note that ill conditioning occurs by increasing  $c$ . Hereafter we fix it at  $c = 1.1h = 0.011$ . In Figure 8 RMS versus  $N$  is plotted at  $t = 1$ . We observe that in this figure, the error values decrease by increasing  $N$ . Values of RMS error for  $u(x, t)$  and absolute error for  $s(t)$  with  $\delta = 0$  and  $\delta = 0.1$  are presented in Table 2. We see that the numerical results are more accurate when there exist no noise on the input data. Under a noise level  $\delta = 0.1$ , the numerical result obtained by the MLPG method is also acceptable. The exact solution, numerical solution and absolute errors for  $u(x, t)$  and  $s(t)$  are plotted in Figures 9–11, respectively.



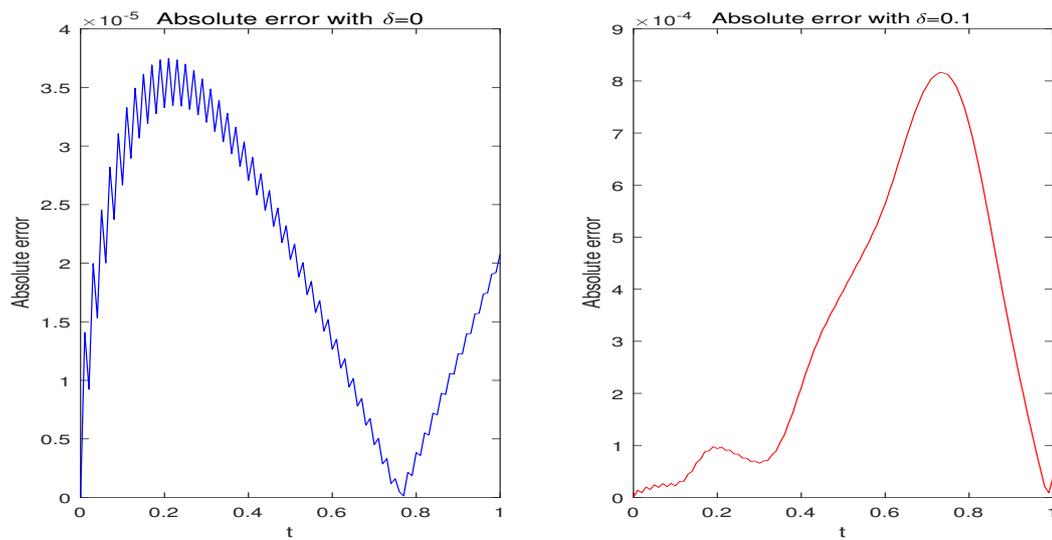
**Figure 7.** Example 2: Diagram of RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  at  $t = 1$  versus shape parameter  $c$  by the MLPG method when  $h = 0.01, \Delta t = 0.01$ .



**Figure 8.** Example 2: Diagram of *RMS* error for  $u(x, t)$  and absolute error for  $s(t)$  at  $t = 1$  versus  $N$  by the MLPG method when  $\Delta t = 0.01, c = 0.011$ .



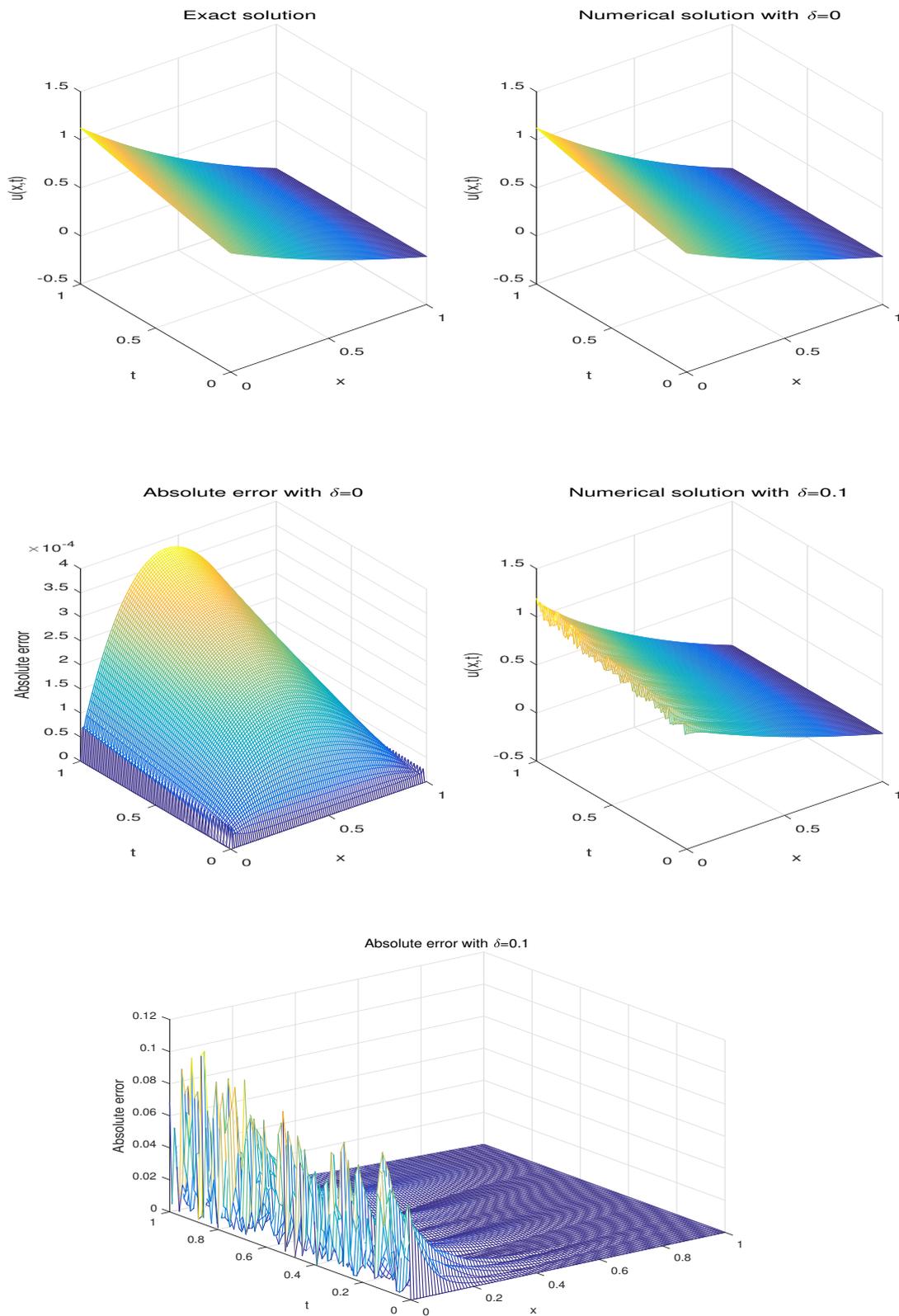
**Figure 9.** Example 2: Comparing the approximation solutions with the exact solution  $s(t)$  when  $h = 0.01, \Delta t = 0.01, c = 0.011$ .



**Figure 10.** Example 2: Absolute error for  $s(t)$  without noise and with level noise  $\delta = 0.1$  when  $h = 0.01$ ,  $\Delta t = 0.01$ ,  $c = 0.011$ .

**Table 2.** Example 2: RMS error for  $u(x, t)$  and Absolute error for  $s(t)$  without noise and with level noise  $\delta = 0.1$  for different time levels when  $h = 0.01$ ,  $\Delta t = 0.01$ ,  $c = 0.011$ .

$t$	RMS Error for $u(x, t)$ with $\delta = 0$	RMS Error for $u(x, t)$ with $\delta = 0.1$	Absolute Error for $s(t)$ with $\delta = 0$	Absolute Error for $s(t)$ with $\delta = 0.1$
0	0	0	0	0
0.1	$8.614 \times 10^{-5}$	$1.535648 \times 10^{-2}$	$2.668 \times 10^{-5}$	$2.225 \times 10^{-5}$
0.2	$1.3203 \times 10^{-4}$	$3.35379 \times 10^{-3}$	$3.327 \times 10^{-5}$	$9.390 \times 10^{-5}$
0.3	$1.6115 \times 10^{-4}$	$9.07394 \times 10^{-3}$	$3.203 \times 10^{-5}$	$6.566 \times 10^{-5}$
0.4	$1.8259 \times 10^{-4}$	$5.16000 \times 10^{-3}$	$2.708 \times 10^{-5}$	$2.1054 \times 10^{-4}$
0.5	$2.0051 \times 10^{-4}$	$4.94580 \times 10^{-3}$	$2.031 \times 10^{-5}$	$3.9630 \times 10^{-4}$
0.6	$2.1688 \times 10^{-4}$	$6.29044 \times 10^{-3}$	$1.264 \times 10^{-5}$	$5.6382 \times 10^{-4}$
0.7	$2.3270 \times 10^{-4}$	$2.182342 \times 10^{-2}$	$4.51 \times 10^{-6}$	$7.9115 \times 10^{-4}$
0.8	$2.4845 \times 10^{-4}$	$8.48735 \times 10^{-3}$	$3.83 \times 10^{-6}$	$7.1690 \times 10^{-4}$
0.9	$2.6441 \times 10^{-4}$	$5.47081 \times 10^{-3}$	$1.228 \times 10^{-5}$	$3.1084 \times 10^{-4}$
1.0	$2.8074 \times 10^{-4}$	$1.056699 \times 10^{-2}$	$2.077 \times 10^{-5}$	$3.917 \times 10^{-5}$



**Figure 11.** Example 2: Exact solution, numerical solution and absolute error for  $u(x, t)$  without noise and with level noise  $\delta = 0.1$  when  $h = 0.01, \Delta t = 0.01, c = 0.011$ .

**Example 3.** With an example, we compare the average values of the absolute errors between the present method and the Adomian decomposition method and the fourth-order Runge–Kutta method obtained in Reference [43]. We consider the problem (1)–(5) with  $\alpha = 0.1, \kappa = 10, \beta = -1, T = 0.5$  and

$$f(y) = \exp(1 - y), \quad g(t) = \exp(1 + 0.1t), \quad h(t) = 1.$$

The analytical solutions of problem are

$$v(y, t) = \exp(1 - y + 0.1t), \quad s(t) = 0.1t + 1,$$

and

$$u(x, t) = v(xs(t), t) = \exp(1 - x(0.1t + 1) + t), \quad s(t) = 0.1t + 1.$$

Tables 3 and 4 show the results of calculations related to the reconstruction of the moving boundary and the temperature distribution using the present method, the Adomian decomposition method and the fourth-order Runge–Kutta method. It can be seen that the presented procedure in this paper is useful and efficient in finding solutions of the considered problem.

**Table 3.** Example 3: Average values of the error in the reconstruction of the position of the moving boundary for  $N = 7$  (ADM, Adomian decomposition method; RKM, fourth-order Runge–Kutta method; PM, present method).

	ADM		RKM	PM
$m = 1$	0.002545	$\Delta t = 0.05$	0.002376	0.000188
$m = 3$	0.002324	$\Delta t = 0.025$	0.002372	0.000177
$m = 5$	0.002200	$\Delta t = 0.01$	0.002370	0.000171
$m = 7$	0.002095	$\Delta t = 0.005$	0.002369	0.000169

**Table 4.** Example 3: Average values of the absolute errors in the reconstruction of the temperature distribution ( $\Delta u_i$ , average value of the  $u(x_i, t_j), j = 0, 1, 2, \dots, M$ ; ADM, Adomian decomposition method; RKM, fourth-order Runge–Kutta method; PM, present method).

	ADM ( $m = 5$ )	RKM ( $M = 50$ )	PM ( $M = 50$ )
$N = 5$			
$\Delta u_1$	0.004737	0.571854	0.001150
$\Delta u_2$	0.006836	0.1038585	0.000806
$\Delta u_3$	0.005531	0.012567	0.001228
$N = 6$			
$\Delta u_1$	0.004478	0.766475	0.000886
$\Delta u_2$	0.014252	0.198930	0.000641
$\Delta u_3$	0.003705	0.043138	0.000113
$\Delta u_4$	0.005107	0.003309	0.000807
$N = 7$			
$\Delta u_1$	0.020102	0.934229	0.000576
$\Delta u_2$	0.051464	0.310715	0.000327
$\Delta u_3$	0.017199	0.089096	0.000128
$\Delta u_4$	0.011347	0.021850	0.000115
$\Delta u_5$	0.003379	0.001768	0.000459

### 8. Conclusions

In this paper, a kind of MLPG method using the MLS approximation to represent the trial function at each field node, is applied for numerically solving a nonlinear one-phase Stefan problem. Nonlinearity of this problem is due to the Stefan condition. The free boundary problem is transformed into a fixed boundary by the change of variable. In the presented method, all integrations are

performed over small local quadrature domains so it does not require using any background integration cells. In the proposed method, the shape functions are produced by the MLS approximation technique. A two-step time discretization method is used to approximate the time derivatives operators. The Heaviside step function was used as the test function in the local weak form method in MLPG. Numerical results show that the proposed method is accurate and stable, although under a large measurement noise.

**Author Contributions:** All authors contributed equally to this work.

**Acknowledgments:** We thank the anonymous reviewers for helpful comments, which lead to definite improvement in the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Crank, J. *Free and Moving Boundary Problems*; Clarendon Press: Oxford, UK, 1984.
2. Gupta, S. *The Classical Stefan Problem. Basic Concepts, Modelling and Analysis*; Elsevier: Amsterdam, The Netherlands, 2003.
3. Meirmanov, A. *The Stefan Problem*; Walter de Gruyter: Berlin, Germany, 1992.
4. Ozisik, M. *Heat Conduction*; Wiley: New York, NY, USA, 1980.
5. Rubinstein, L. *The Stefan Problem*; AMS: Providence, RI, USA, 1971.
6. Hetmaniok, E.; Slota, D.; Wituła, R.; Zielonka, A. Solution of the One-Phase Inverse Stefan Problem by Using the Homotopy Analysis Method. *Appl. Math. Model.* **2015**, *39*, 6793–6805. [[CrossRef](#)]
7. Onyejekwe, O.N. The Solution of One-Phase Inverse Stefan Problem by Homotopy Analysis Method. *Appl. Math. Sci.* **2014**, *8*, 2635–2644. [[CrossRef](#)]
8. Liu, C.S. Solving two typical inverse Stefan problems by using the Lie-group shooting method. *Int. J. Heat Mass Transf.* **2011**, *54*, 1941–1949. [[CrossRef](#)]
9. Zabarav, N.; Ruan, Y. A deforming finite element method analysis of inverse Stefan problems. *Int. J. Numer. Methods Eng.* **1989**, *28*, 295–313. [[CrossRef](#)]
10. Slota, D. Direct and inverse one-phase Stefan problem solved by the variational iteration method. *Comput. Math. Appl.* **2007**, *54*, 1139–1146. [[CrossRef](#)]
11. Grzymkowski, R.; Slota, D. Stefan problem solved by Adomian decomposition method. *Int. J. Comput. Math.* **2005**, *82*, 851–856. [[CrossRef](#)]
12. Grzymkowski, R.; Slota, D. One-phase inverse Stefan problem solved by Adomian decomposition method. *Comput. Math. Appl.* **2006**, *51*, 33–40. [[CrossRef](#)]
13. Slota, D. The application of the homotopy perturbation method to one-phase inverse Stefan problem. *Int. J. Comput. Math.* **2010**, *37*, 587–592. [[CrossRef](#)]
14. Johansson, B.T.; Lesnic, D.; Reeve, T. A method of fundamental solutions for the one-dimensional inverse Stefan problem. *Appl. Math. Model.* **2011**, *35*, 4367–4378. [[CrossRef](#)]
15. Asfirane, S.; Hlioui, S.; Amara, Y.; Gabsi, M. Study of a hybrid excitation synchronous machine: Modeling and experimental validation. *Math. Comput. Appl.* **2019**, *24*, 34. [[CrossRef](#)]
16. Bacciocchi, M.; Tarantino, A. Natural frequency analysis of functionally graded orthotropic cross-ply plates based on the finite element method. *Math. Comput. Appl.* **2019**, *24*, 52. [[CrossRef](#)]
17. Liu, G.; Gu, Y. *An Introduction to Meshfree Methods and Their Programming*; Springer: Berlin, Germany, 2005.
18. Belytschko, T.; Lu, Y.Y.; Gu, L. Element-free Galerkin methods. *Int. J. Numer. Methods Eng.* **1994**, *37*, 229–256. [[CrossRef](#)]
19. Belytschko, T.; Lu, Y.Y.; Gu, L. Element free Galerkin methods for static and dynamic fracture. *Int. J. Solids Struct.* **1995**, *32*, 2547–2570. [[CrossRef](#)]
20. Peng, M.; Cheng, Y. A boundary element-free method (BEFM) for two-dimensional potential problems. *Eng. Anal. Boundary Elem.* **2009**, *33*, 77–82. [[CrossRef](#)]
21. Peng, M.; Li, D.; Cheng, Y. The complex variable element-free Galerkin (CVEFG) method for elasto-plasticity problems. *Eng. Struct.* **2011**, *33*, 127–135. [[CrossRef](#)]
22. Peng, M.; Liu, P.; Cheng, Y. The complex variable element-free Galerkin (CVEFG) method for two-dimensional elasticity problems. *Int. J. Appl. Mech.* **2009**, *1*, 367–385. [[CrossRef](#)]

23. Ren, H.; Cheng, Y. The interpolating element-free Galerkin (IEFG) method for two-dimensional potential problems. *Eng. Anal. Boundary Elem.* **2012**, *36*, 873–880. [[CrossRef](#)]
24. Zhang, Z.; Hao, S.; Liew, K.; Cheng, Y. The improved element-free Galerkin method for two-dimensional elastodynamics problems. *Eng. Anal. Boundary Elem.* **2013**, *37*, 1576–1584. [[CrossRef](#)]
25. Uddin, M.; Ali, H.; Taufiq, M. On the Approximation of a nonlinear biological population model using localized radial basis function method. *Math. Comput. Appl.* **2019**, *24*, 54. [[CrossRef](#)]
26. Kansa, E. Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates. *Comput. Math. Appl.* **1990**, *19*, 127–145. [[CrossRef](#)]
27. Dehghan, M.; Shokri, A. A numerical method for solution of the two dimensional sine-Gordon equation using the radial basis functions. *Math. Comput. Simul.* **2008**, *79*, 700–715. [[CrossRef](#)]
28. Abbasbandy, S.; Ghehsareh, H.R.; Hashim, I. Numerical analysis of a mathematical model for capillary formation in tumor angiogenesis using a meshfree method based on the radial basis function. *Eng. Anal. Boundary Elem.* **2012**, *36*, 1811–1818. [[CrossRef](#)]
29. Abbasbandy, S.; Shirzadi, A. MLPG method for two-dimensional diffusion equation with Neumann's and non-classical boundary conditions. *Appl. Numer. Math.* **2011**, *61*, 170–180. [[CrossRef](#)]
30. Atluri, S.; Zhu, T. A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Comput. Mech.* **1998**, *22*, 117–127. [[CrossRef](#)]
31. Atluri, S.; Zhu, T. A new meshless local Petrov–Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. *Comput. Model. Simul. Eng.* **1998**, *3*, 187–196.
32. Atluri, S.; Zhu, T. New concepts in meshless methods. *Int. J. Numer. Meth. Eng.* **2000**, *13*, 537–556. [[CrossRef](#)]
33. Atluri, S.; Zhu, T. The meshless local Petrov–Galerkin (MLPG) approach for solving problems in elasto-statics. *Comput. Mech.* **2000**, *25*, 169–179. [[CrossRef](#)]
34. Dai, B.; Cheng, Y. An improved local boundary integral equation method for two-dimensional potential problems. *Int. J. Appl. Mech.* **2010**, *2*, 421–436. [[CrossRef](#)]
35. Dehghan, M.; Mirzaei, D. The meshless local Petrov–Galerkin (MLPG) method for the generalized two-dimensional non-linear Schrödinger equation. *Eng. Anal. Boundary Elem.* **2008**, *32*, 747–756. [[CrossRef](#)]
36. Dehghan, M.; Mirzaei, D. Meshless local Petrov–Galerkin (MLPG) method for the unsteady magnetohydrodynamic (MHD) flow through pipe with arbitrary wall conductivity. *Appl. Numer. Math.* **2009**, *59*, 1043–1058. [[CrossRef](#)]
37. Gu, Y.; Liu, G. A meshless local Petrov–Galerkin (MLPG) method for free and forced vibration analyses for solids. *Comput. Mech.* **2001**, *27*, 188–198. [[CrossRef](#)]
38. Lancaster, P.; Salkauskas, K. Surface generated by moving least squares methods. *Math. Comput.* **1981**, *37*, 141–158. [[CrossRef](#)]
39. Shirzadi, A.; Ling, L.; Abbasbandy, S. Meshless simulations of the two-dimensional fractional-time convection-diffusion-reaction equations. *Eng. Anal. Boundary Elem.* **2012**, *36*, 1522–1527. [[CrossRef](#)]
40. Shirzadi, A.; Sladek, V.; Sladek, J. A local integral equation formulation to solve coupled nonlinear reaction-diffusion equations by using moving least square approximation. *Eng. Anal. Boundary Elem.* **2013**, *37*, 8–14. [[CrossRef](#)]
41. Hu, D.; Long, S.; Liu, K.; Li, G. A modified meshless local Petrov–Galerkin method to elasticity problems in computer modeling and simulation. *Eng. Anal. Boundary Elem.* **2006**, *30*, 399–404. [[CrossRef](#)]
42. Liu, K.; Long, S.; Li, G. A simple and less-costly meshless local Petrov–Galerkin (MLPG) method for the dynamic fracture problem. *Eng. Anal. Boundary Elem.* **2006**, *30*, 72–76.
43. Grzymkowski, R.; Pleszczyński, M.; Słota, D. Comparing the Adomian decomposition method and Runge–Kutta method for solutions of the Stefan problem. *Int. J. Comput. Math.* **2006**, *83*, 409–417. [[CrossRef](#)]

