Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 686020, 12 pages
http://dx.doi.org/10.1155/2014/686020

Research Article

Moving Least Squares Method for a One-Dimensional Parabolic Inverse Problem

Baiyu Wang

College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, China

Correspondence should be addressed to Baiyu Wang; wangbaiyu@hnu.edu.cn

Received 7 November 2013; Accepted 17 January 2014; Published 5 March 2014

Academic Editor: Jaan Janno

This paper investigates the numerical solution of a class of one-dimensional inverse parabolic problems using the moving least squares approximation; the inverse problem is the determination of an unknown source term depending on time. The collocation method is used for solving the equation; some numerical experiments are presented and discussed to illustrate the stability and high efficiency of the method.

1. Introduction

The inverse problem of parabolic equations appears in a wide variety of physical and engineering settings, such as elasticity, plasticity, hydrology, material sciences, heat transfer, combustion systems, medical imaging, transport problems, and control theory. Many researchers solved this problem using different methods [1–10].

For many decades, many numerical methods have been developed to solve partial differential equations, such as finite difference method, finite elements method, and boundary elements method. However, these above methods are mesh dependent methods, and, for some problems such as large deformation, the mesh generation is very difficult, especially with the irregular geometries. As a result, the meshless method appeared.

Compared with these mesh dependent numerical methods, in the meshless methods, mesh generation on the spatial domain of the problem is not needed; this property is the main advantage of these techniques over the mesh dependent methods, and it is a technique on scattered data; this method uses the distributed nodal points to approximate the unknown function; the distribution of nodes could be selected regularly or randomly in the analyzed domain, and there is no need for the geometry of the domain.

Many meshless methods have been discussed, such as smooth particle hydrodynamics method [11], the finite point method [12], diffuse element method [13], element-free Galerkin method [14], hp-clouds method [15], and meshless local Petrov-Galerkin method [16]. Moving least squares is very important for constructing the shape functions in meshless method, and it had been studied by a number of investigators, Lancaster and Salkauskas [17], Levin [18], Mirzaei and Dehghan [19], Armentano [20, 21], and so forth.

In this paper, a collocation meshless method based on moving least squares approximation is used for numerical solution of a class of inverse problems in parabolic equation. The rest of the paper is organized as follows. In Section 2, we will give an outline of the moving least squares. In Section 3, we solve the inverse problem using the meshless method based on the moving least squares. Numerical experiments will be given in Section 4 in order to illustrate the feasibility of the method.

2. An Outline of the Moving Least Squares (MLS)

The MLS method as an approximation and interpolating method has been introduced by Lancaster and Salkauskas.

Given data values $\mathbf{u} = \{u_j\}_{j=1}^{N}$ at nodes x_j, N is the number of scattered nodes, the MLS method produces a function u^h that approximates data $\mathbf{u}$ in a weighted square sense.

It can be defined as

$$u^h(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) = P^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \quad \forall \mathbf{x} \in \overline{\Omega}, \tag{1}$$

where m is the number of terms in the basis, $p_i(\mathbf{x})$ are the monomial basis functions, $a_i(\mathbf{x})$ are the coefficients of the basis functions, and

$$\begin{gathered} P^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x})], \\ a(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \ldots, a_m(\mathbf{x})]^T. \end{gathered} \tag{2}$$

The unknown coefficient $\mathbf{a}(\mathbf{x})$ is determined by minimizing the functional J, which is defined as

$$\begin{aligned} J &= \sum_{j=1}^{n} \omega(\mathbf{x}-\mathbf{x}_j)\left(u^h(\mathbf{x}) - u_j\right)^2 \\ &= \sum_{j=1}^{n} \omega(\mathbf{x}-\mathbf{x}_j)\left(\sum_{i=1}^{m} p_i(\mathbf{x}_j)a_i(\mathbf{x}) - u_j\right)^2 \\ &= \sum_{j=1}^{n} \omega(\mathbf{x}-\mathbf{x}_j)\left(P^T(\mathbf{x}_j)\mathbf{a}(\mathbf{x}) - u_j\right)^2, \end{aligned} \tag{3}$$

where n is the number of nodes in the support domain of the point $\mathbf{x}$, $\omega(\mathbf{x}-\mathbf{x}_j)$ is the weight function, and $\mathbf{x}_j$ are the nodes in the influence domain of $\mathbf{x}$.

Equation (3) can be rewritten in the vector form

$$J = (P\mathbf{a}-\mathbf{u})^T W (P\mathbf{a}-\mathbf{u}), \tag{4}$$

where

$$\begin{gathered} P = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \end{bmatrix}, \\ \mathbf{u} = [u_1, u_2, \ldots, u_n]^T, \\ W = \begin{bmatrix} \omega(\mathbf{x}-\mathbf{x}_1) & 0 & \cdots & 0 \\ 0 & \omega(\mathbf{x}-\mathbf{x}_2) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \omega(\mathbf{x}-\mathbf{x}_n) \end{bmatrix}. \end{gathered} \tag{5}$$

According to the above conditions (4)-(5), taking the derivative $\mathbf{a}(\mathbf{x})$ to zero, we have

$$\frac{\partial J}{\partial \mathbf{a}} = A(\mathbf{x})\mathbf{a}(\mathbf{x}) - B(\mathbf{x})\mathbf{u} = 0, \tag{6}$$

which is equivalent to

$$A(\mathbf{x})\mathbf{a}(\mathbf{x}) = B(\mathbf{x})\mathbf{u}, \tag{7}$$

where

$$\begin{gathered} A(\mathbf{x}) = P^T W P, \\ B(\mathbf{x}) = P^T W. \end{gathered} \tag{8}$$

Then, we get

$$\mathbf{a}(\mathbf{x}) = A^{-1}(\mathbf{x}) B(\mathbf{x}) u, \tag{9}$$

and, by substituting (9) into (1), we have

$$u^h(\mathbf{x}) = P^T(\mathbf{x})\mathbf{a}(\mathbf{x}) = \Phi^T(\mathbf{x})\mathbf{u} = \sum_{j=1}^{N} \phi_j(\mathbf{x}) u_j, \tag{10}$$

where

$$\Phi^T(\mathbf{x}) = P^T(\mathbf{x}) A^{-1}(\mathbf{x}) B(\mathbf{x}) \tag{11}$$

and $\phi_j(\mathbf{x})$ is called the shape function.

3. The Inverse Problem and Its Numerical Solution

In this section, the collocation meshless method based on moving least squares approximation is implemented for the numerical solution of the inverse problem of parabolic equations, and the problem is as follows.

Find $u = u(x,t)$ and $f(t)$ which satisfy

$$\begin{gathered} u_t(x,t) = u_{xx}(x,t) + f(t), \quad (x,t) \in Q_T = (0,l) \times (0,T), \\ u(x,0) = \varphi(x), \quad x \in (0,l), \\ u(0,t) = h_0(t), \quad u(l,t) = h_l(t), \quad t \in (0,T). \end{gathered} \tag{12}$$

And the additional specification is

$$u(x^*, t) = E(t), \quad x^* \in (0,l), t \in (0,T), \tag{13}$$

where $\varphi(x)$, $h_0(t)$, $h_l(t)$, and $E(t)$ are known functions and x^* is a fixed prescribed point in the admissible set.

Assume that these known functions satisfy the compatibility conditions; the unique solvability of this inverse problem is presented in [5]. However, this inverse problem is ill-posed; that is to say, the solution is unstable, and any small change in the additional specification may result in a dramatic change in the numerical solution. To obtain a stable numerical solution, some regularization strategies should be applied.

According to (13), consider the following transformation [22]:

$$E'(t) = u_{xx}|_{x=x^*} + f(t), \tag{14}$$

and using (14), we get

$$f(t) = E'(t) - u_{xx}|_{x=x^*}. \tag{15}$$

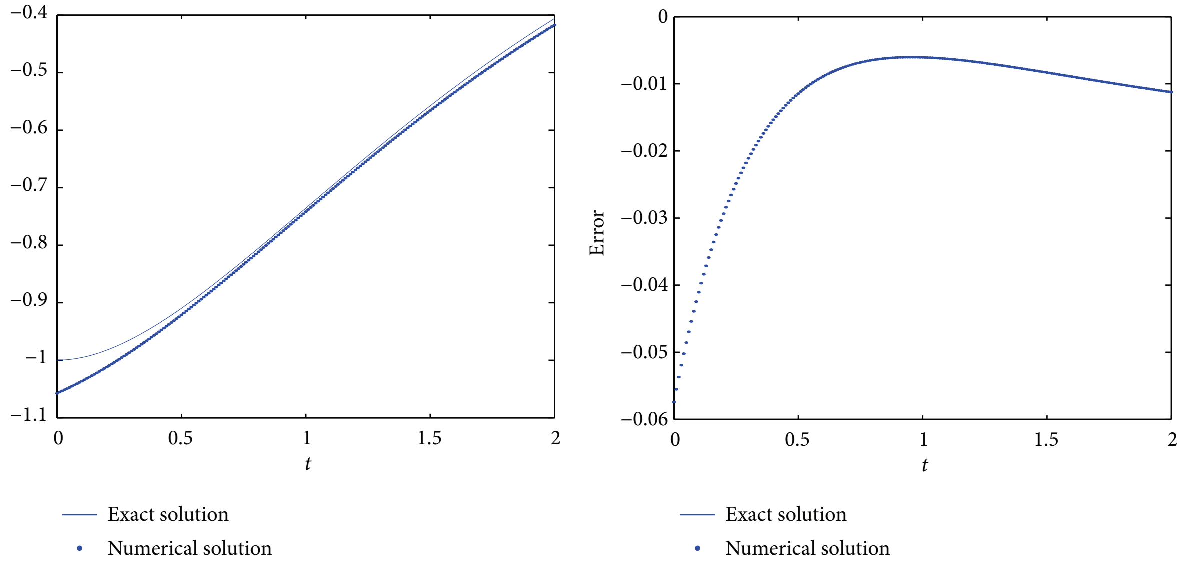
By substituting (14) into (12), (12) is transformed to the following equation:

$$\begin{gathered} u_t = u_{xx} + \left(E'(t) - u_{xx}|_{x=x^*}\right), \quad (x,t) \in Q_T, \\ u(x,0) = \varphi(x), \quad x \in (0,l), \\ u(0,t) = h_0(t), \quad u(l,t) = h_l(t), \quad t \in (0,T). \end{gathered} \tag{16}$$

So the inverse problem is transformed to a direct problem.

TABLE 1: The errors for different Δx and Δt.

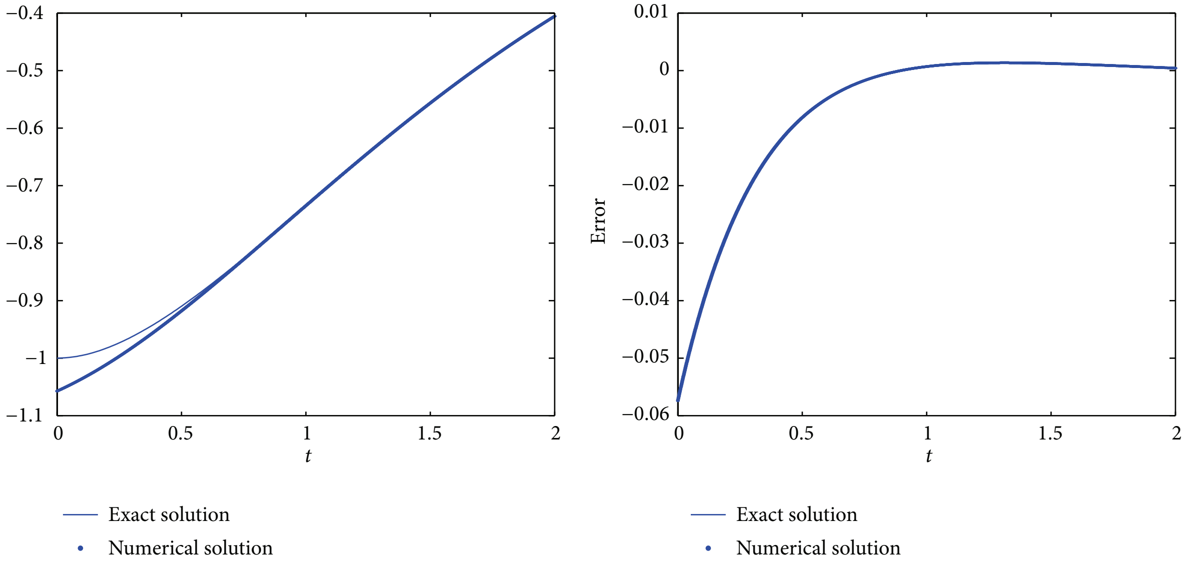
Δt	$\Delta x = 0.5$		$\Delta x = 0.1$	
	Eu	Ef	Eu	Ef
0.01	1.1428×10^{-2}	6.1902×10^{-3}	3.0506×10^{-3}	3.0599×10^{-3}
0.005	1.0569×10^{-2}	4.1804×10^{-3}	1.5394×10^{-3}	1.5832×10^{-3}
0.001	9.9744×10^{-3}	1.8376×10^{-3}	3.4687×10^{-4}	4.1021×10^{-4}
0.0005	9.9068×10^{-3}	1.2985×10^{-3}	2.1206×10^{-4}	2.5863×10^{-4}
0.0001	9.8539×10^{-3}	5.8058×10^{-4}	1.3264×10^{-4}	1.0545×10^{-4}



(a)

(b)

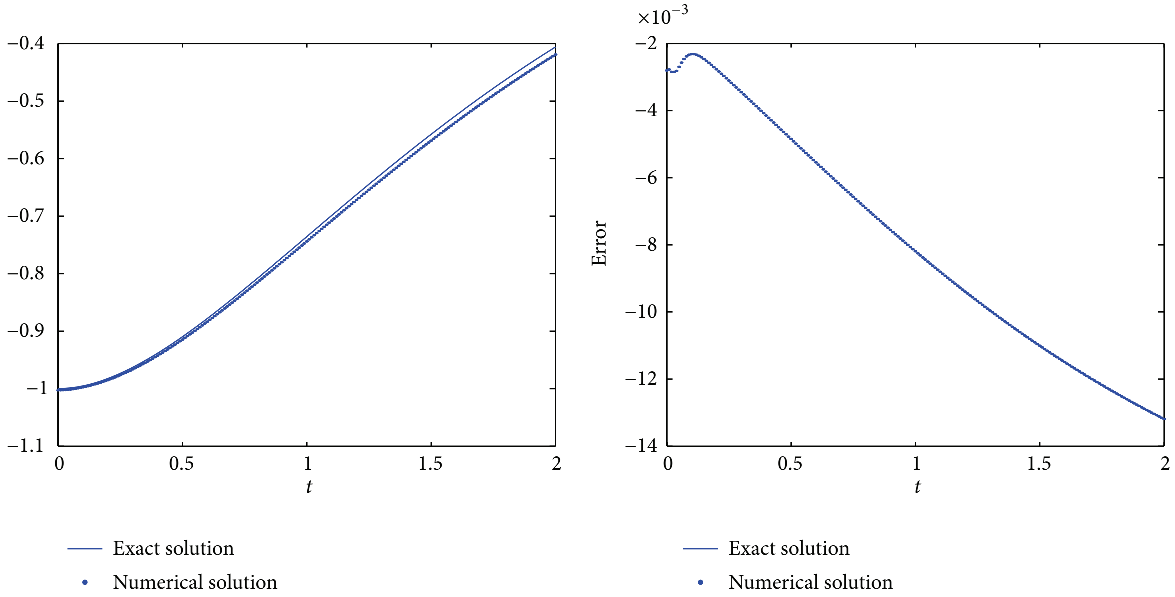
FIGURE 1: The exact solution $f(t)$ and the numerical solution of $\hat{f}(t)$ (a), the error $f(t) - \hat{f}(t)$ (b) with $\Delta x = 0.5$, $\Delta t = 0.01$.



(a)

(b)

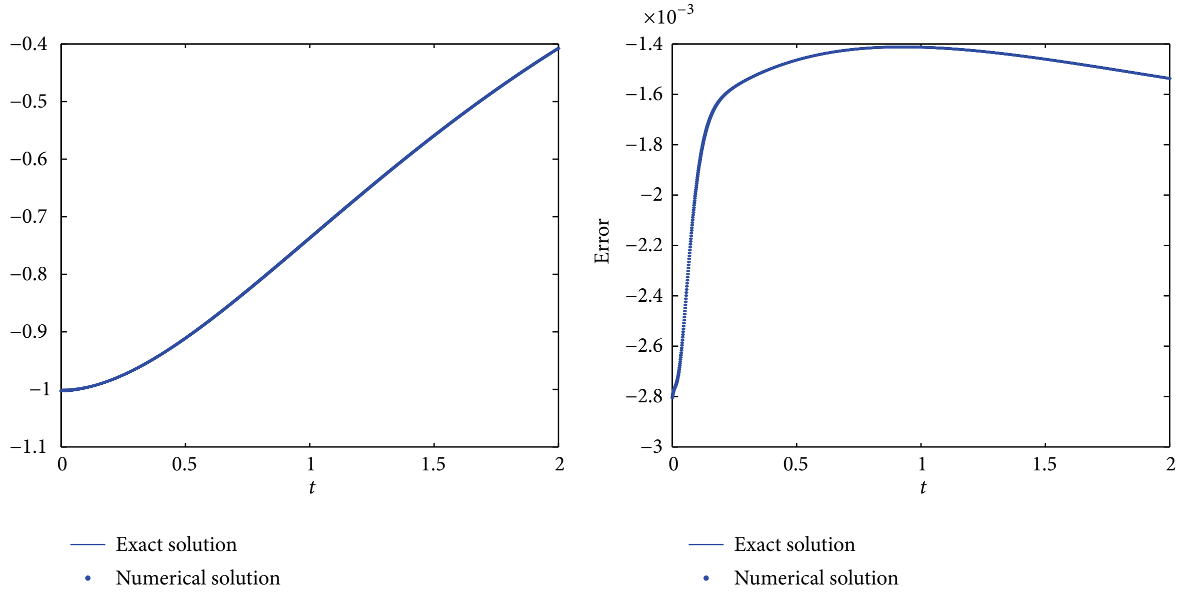
FIGURE 2: The exact solution $f(t)$ and the numerical solution of $\hat{f}(t)$ (a), the error $f(t) - \hat{f}(t)$ (b) with $\Delta x = 0.5$, $\Delta t = 0.001$.



(a)

(b)

Figure 3: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ (a), the error $f(t) - \widehat{f}(t)$ (b) with $\Delta x = 0.1$, $\Delta t = 0.01$.



(a)

(b)

Figure 4: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ (a), the error $f(t) - \widehat{f}(t)$ (b) with $\Delta x = 0.1$, $\Delta t = 0.001$.

Table 2: The errors for different γ.

γ	Eu	Ef
0.01	1.0799×10^{-2}	2.0392×10^{-3}
0.005	5.5640×10^{-3}	1.4692×10^{-3}
0.001	1.3792×10^{-3}	7.4892×10^{-4}

Table 3: The errors for different q.

q	Eu	Ef
0.01	3.4147×10^{-4}	8.7343×10^{-4}
0.005	3.8703×10^{-4}	6.4143×10^{-4}
0.001	3.2703×10^{-4}	4.1964×10^{-4}

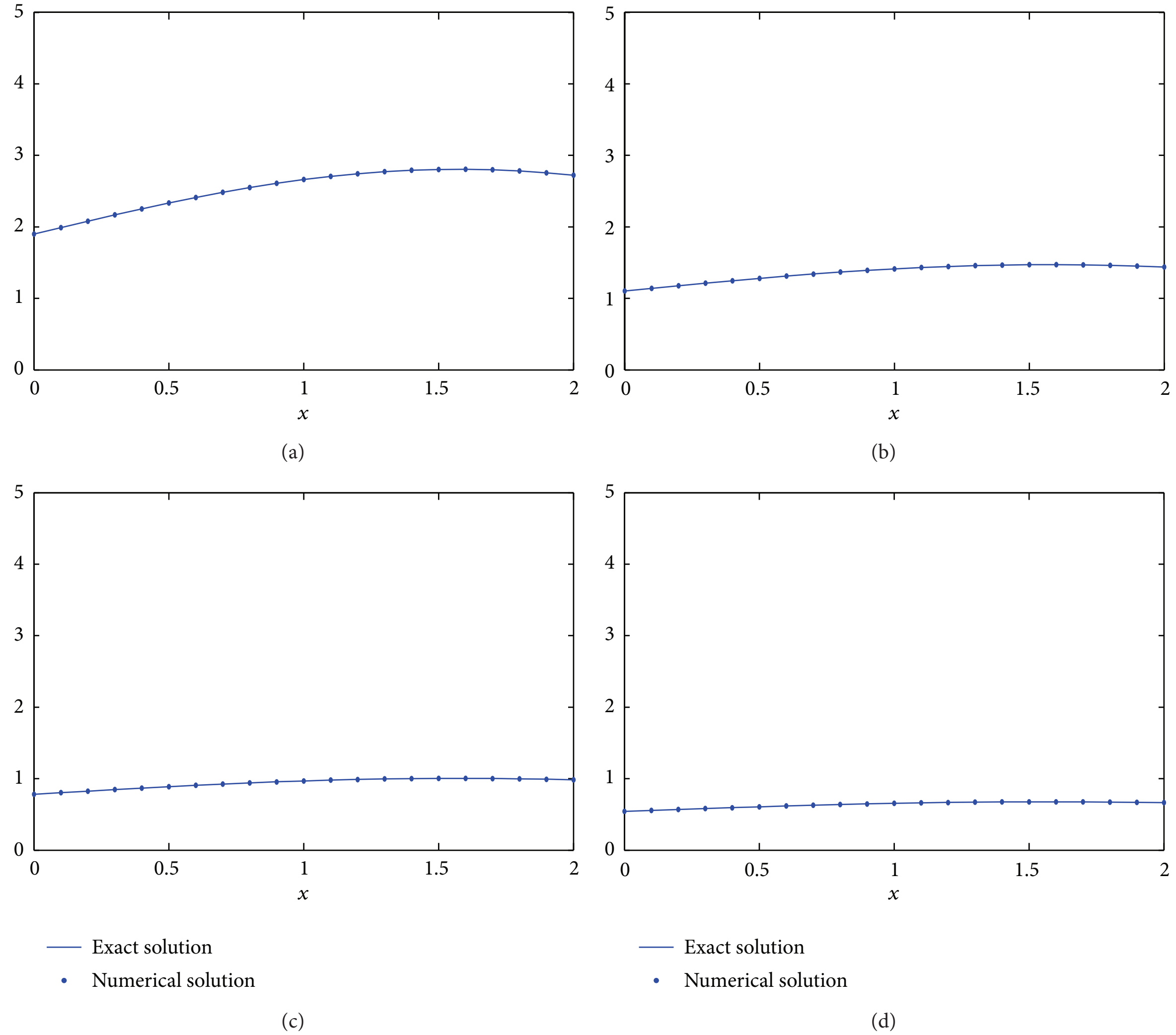


FIGURE 5: The exact solution $u(x, t)$ and numerical solution $\widehat{u}(x, t)$ at $t = 0.1$ (a), 1 (b), 1.5 (c), and 2 (d).

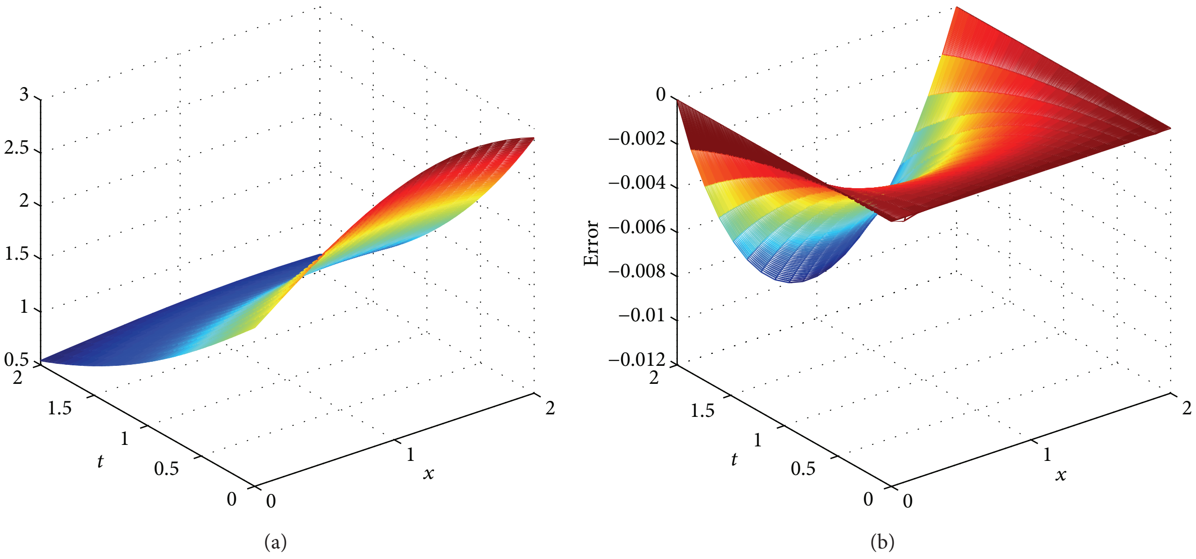
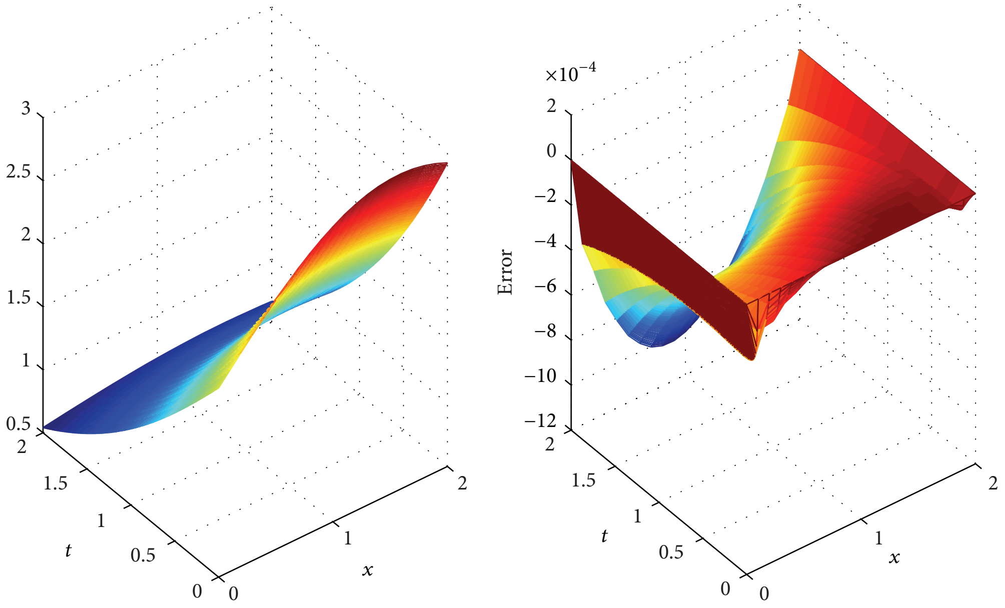


FIGURE 6: The numerical solution of $\widehat{u}(x, t)$ (a), the error $u(x, t) - \widehat{u}(x, t)$ (b) with $\Delta x = 0.1$, $\Delta t = 0.01$.

Table 4: The errors for different Δt.

	$\gamma = 0.01$	$\gamma = 0.001$	$q = 0.01$	$q = 0.001$
$\Delta t = 0.01$	7.0116×10^{-3}	3.6493×10^{-3}	3.6401×10^{-3}	3.0238×10^{-3}
$\Delta t = 0.001$	2.0392×10^{-3}	7.4892×10^{-4}	8.5168×10^{-4}	4.2884×10^{-4}
$\Delta t = 0.0001$	6.3895×10^{-4}	2.2068×10^{-4}	2.7201×10^{-4}	1.1466×10^{-4}



(a)

(b)

Figure 7: The numerical solution of $\hat{u}(x,t)$ (a), the error $u(x,t) - \hat{u}(x,t)$ (b) with $\Delta x = 0.1$, $\Delta t = 0.001$.

Table 5: The errors for different Δx, Δt and noisy parameter.

	FDM	MLS
$\Delta x = 0.5, \Delta t = 0.01, \gamma = 0$	3.4720×10^{-2}	9.0561×10^{-3}
$\Delta x = 0.5, \Delta t = 0.001, \gamma = 0$	3.4877×10^{-3}	9.0857×10^{-4}
$\Delta x = 0.1, \Delta t = 0.01, \gamma = 0$	3.4720×10^{-2}	6.4498×10^{-3}
$\Delta x = 0.1, \Delta t = 0.001, \gamma = 0$	3.4877×10^{-3}	6.4677×10^{-4}
$\Delta x = 0.5, \Delta t = 0.01, \gamma = 0.001$	3.5700×10^{-2}	9.5460×10^{-3}
$\Delta x = 0.5, \Delta t = 0.001, \gamma = 0.001$	7.3814×10^{-3}	1.4210×10^{-3}
$\Delta x = 0.1, \Delta t = 0.01, \gamma = 0.001$	3.5691×10^{-2}	6.7775×10^{-3}
$\Delta x = 0.1, \Delta t = 0.001, \gamma = 0.001$	7.2994×10^{-3}	9.9116×10^{-4}
$\Delta x = 0.5, \Delta t = 0.01, q = 0.001$	2.5285×10^{-1}	8.9877×10^{-3}
$\Delta x = 0.5, \Delta t = 0.001, q = 0.001$	2.4914×10^{0}	9.7315×10^{-4}
$\Delta x = 0.1, \Delta t = 0.01, q = 0.001$	2.5354×10^{-1}	6.4462×10^{-3}
$\Delta x = 0.1, \Delta t = 0.001, q = 0.001$	2.5664×10^{0}	6.7999×10^{-4}

The approximate function $\hat{u}(x,t)$ of $u(x,t)$ can be represented as

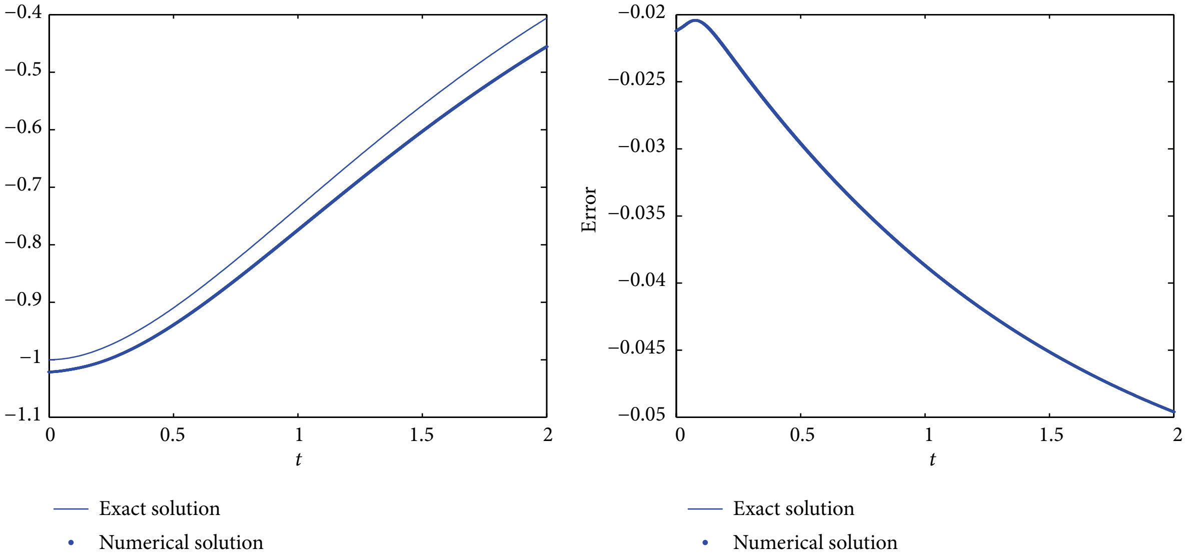
$$\hat{u}(x,t) = \sum_{j=1}^{n} \lambda_j(t)\,\phi_j(x), \tag{17}$$

where $\phi_j(x)$ is the shape function described in Section 2.

By substituting (17) into (16), the system (16) can be rewritten as

$$\begin{gathered}
\sum_{j=1}^{n} \lambda_j'(t)\,\phi_j(x) = \sum_{j=1}^{n} \lambda_j(t)\,\phi_j''(x) \\
+ \left(E'(t) - \sum_{j=1}^{n} \lambda_j(t)\,\phi_j''(x^*)\right), \\
(x,t) \in Q_T, \\
\sum_{j=1}^{n} \lambda_j(0)\,\phi_j(x) = \varphi(x), \quad x \in (0,l), \\
\sum_{j=1}^{n} \lambda_j(t)\,\phi_j(0) = h_0(t), \qquad \sum_{j=1}^{n} \lambda_j(t)\,\phi_j(l) = h_l(t), \\
t \in (0,T).
\end{gathered} \tag{18}$$

It is well known that the step lengths in the standard numerical method have regularization effect; hence, we can choose appropriate step lengths to compute the derivative



(a)

(b)

FIGURE 8: The exact solution $f(t)$ and the numerical solution of $\hat{f}(t)$ (a), the error $f(t)-\hat{f}(t)$ (b) with $\gamma = 0.01$.

of t. We apply a simple one-step forward difference formula, and we get

$$\begin{gathered}\sum_{j=1}^{n}\frac{\lambda_j(t_{m+1})-\lambda_j(t_m)}{\Delta t}\phi_j(x)\\ =\sum_{j=1}^{n}\lambda_j(t_m)\phi_j''(x)\\ +\left(\frac{E(t_{m+1})-E(t_m)}{\Delta t}-\sum_{j=1}^{n}\lambda_j(t_m)\phi_j''(x^*)\right),\\ (x,t)\in Q_T,\\ \sum_{j=1}^{n}\lambda_j(0)\phi_j(x)=\varphi(x),\quad x\in(0,l),\\ \sum_{j=1}^{n}\lambda_j(t_m)\phi_j(0)=h_0(t_m),\qquad \sum_{j=1}^{n}\lambda_j(t_m)\phi_j(l)=h_l(t_m),\\ t\in(0,T).\end{gathered}\tag{19}$$

Then, from the first formula of (19), we obtain

$$\begin{gathered}\sum_{j=1}^{n}\lambda_j(t_{m+1})\phi_j(x)\\ =\sum_{j=1}^{n}\lambda_j(t_m)\phi_j(x)\\ +\Delta t\left[\sum_{j=1}^{n}\lambda_j(t_m)\phi_j''(x)\right.\\ \left.+\left(\frac{E(t_{m+1})-E(t_m)}{\Delta t}-\sum_{j=1}^{n}\lambda_j(t_m)\phi_j''(x^*)\right)\right];\\ (x,t)\in Q_T,\end{gathered}\tag{20}$$

by substituting each x_k for x in (20), we get

$$\begin{gathered}\sum_{j=1}^{n}\lambda_j(t_{m+1})\phi_j(x_k)\\ =\sum_{j=1}^{n}\lambda_j(t_m)\phi_j(x_k)\\ +\Delta t\left[\sum_{j=1}^{n}\lambda_j(t_m)\phi_j''(x_k)\right.\\ \left.+\left(\frac{E(t_{m+1})-E(t_m)}{\Delta t}-\sum_{j=1}^{n}\lambda_j(t_m)\phi_j''(x^*)\right)\right].\\ (x,t)\in Q_T.\end{gathered}\tag{21}$$

Let

$$\hat{u}(x_k,t_m)=\sum_{j=1}^{n}\lambda_j(t_m)\phi_j(x_k);\tag{22}$$

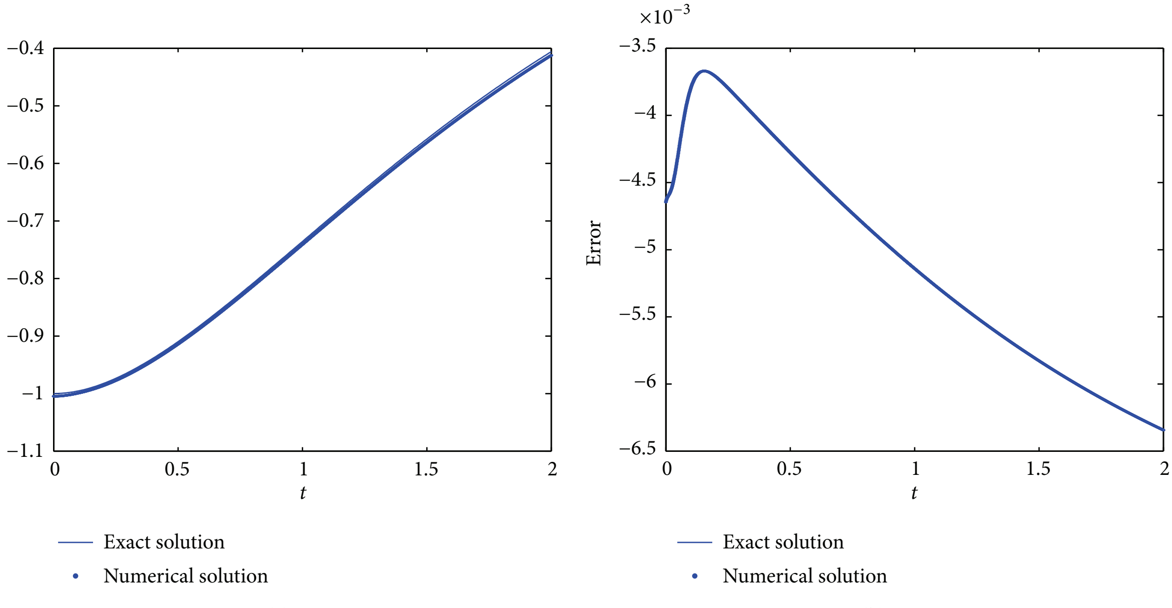


FIGURE 9: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ (a), the error $f(t) - \widehat{f}(t)$ (b) with $\gamma = 0.001$.

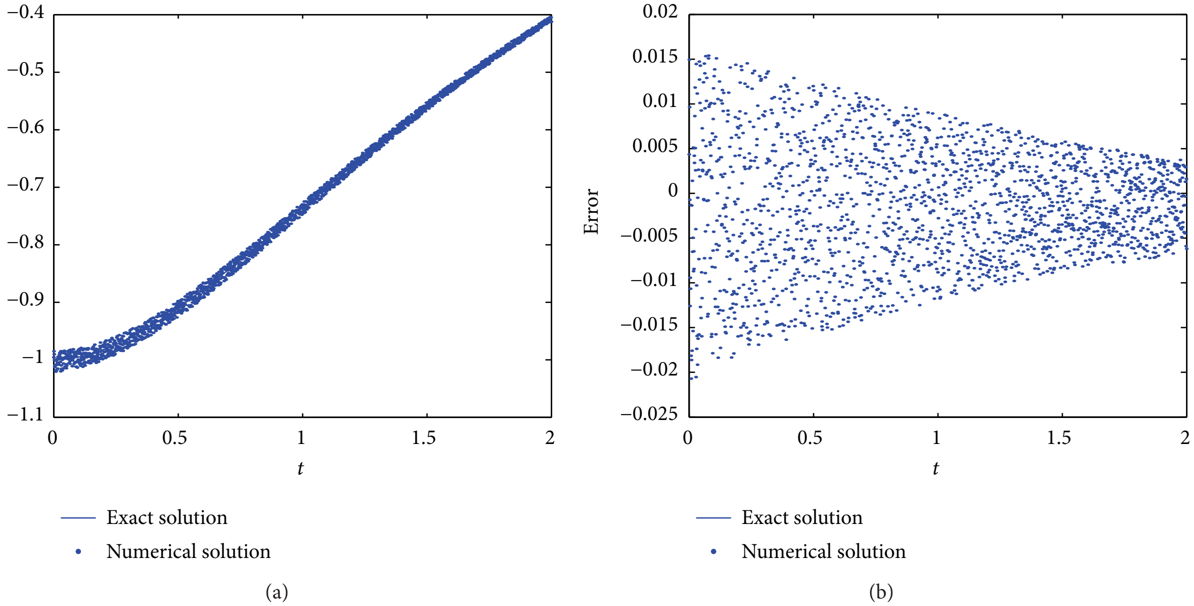


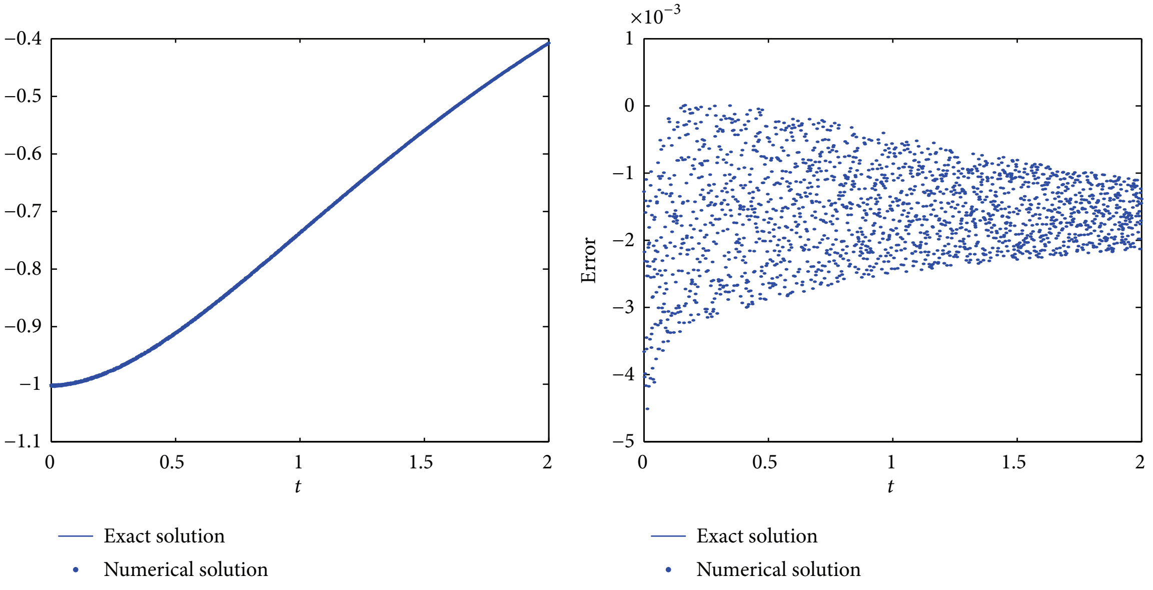
FIGURE 10: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ (a), the error $f(t) - \widehat{f}(t)$ (b) with $q = 0.01$.

then (21) can be rewritten as follows:

$$\begin{aligned}\widehat{u}\left(x_k, t_{m+1}\right) &= \sum_{j=1}^{n} \lambda_j\left(t_{m+1}\right)\phi_j\left(x_k\right)\\ &= \widehat{u}\left(x_k, t_m\right) + \Delta t \sum_{j=1}^{n} \lambda_j\left(t_m\right)\phi_j''\left(x_k\right)\\ &\quad + E\left(t_{m+1}\right) - E\left(t_m\right) - \Delta t \sum_{j=1}^{n} \lambda_j\left(t_m\right)\phi_j''\left(x^*\right);\end{aligned} \tag{23}$$

using the iterative method, by (19) and (22), we can obtain $\lambda_j(t_m)$, $j = 1, 2, \ldots, n$, $m = 1, 2, \ldots, M$, and the numerical solution $\widehat{u}(x_k, t_m)$; then we get the numerical solution

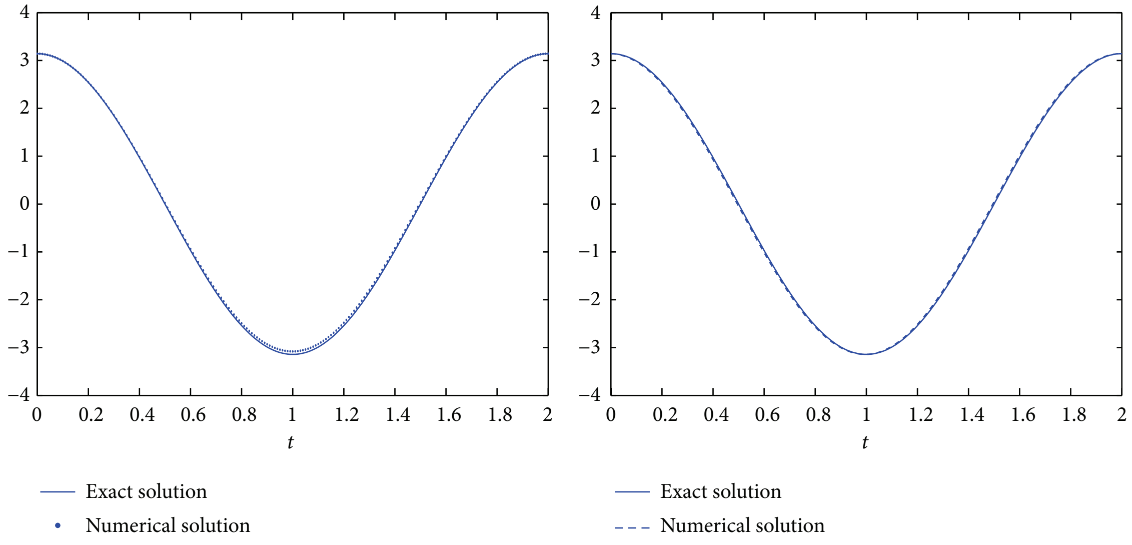
$$\widehat{f}\left(t_{m+1}\right) = \frac{E\left(t_{m+1}\right) - E\left(t_m\right)}{\Delta t} - \sum_{j=1}^{n} \lambda_j\left(t_{m+1}\right)\phi_j''\left(x^*\right). \tag{24}$$



(a)

(b)

FIGURE 11: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ (a), the error $f(t) - \widehat{f}(t)$ (b) with $q = 0.001$.



(a)

(b)

FIGURE 12: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ with no noisy data, (a) MLS and (b) FDM.

4. Numerical Experiments and Discussions

In this section, we give two examples to illustrate the correctness of the theoretical result and the feasibility of the method.

Example 1. Consider (12)-(13), with the conditions

$$
\begin{gathered}
u(x, 0) = \varphi(x) = 2 + \sin x, \\
u(0, t) = h_0(t) = (2 + t) e^{-t}, \\
u(l, t) = h_l(t) = (2 + t + \sin l) e^{-t}, \\
u(x^*, t) = E(t) = (2 + t + \sin x^*) e^{-t},
\end{gathered}
\tag{25}
$$

and we let $l = 2$, $T = 2$, and $x^* = 1$.

The exact solutions are

$$
\begin{gathered}
u(x, t) = (2 + t + \sin x) e^{-t}, \\
f(t) = -(1 + t) e^{-t}.
\end{gathered}
\tag{26}
$$

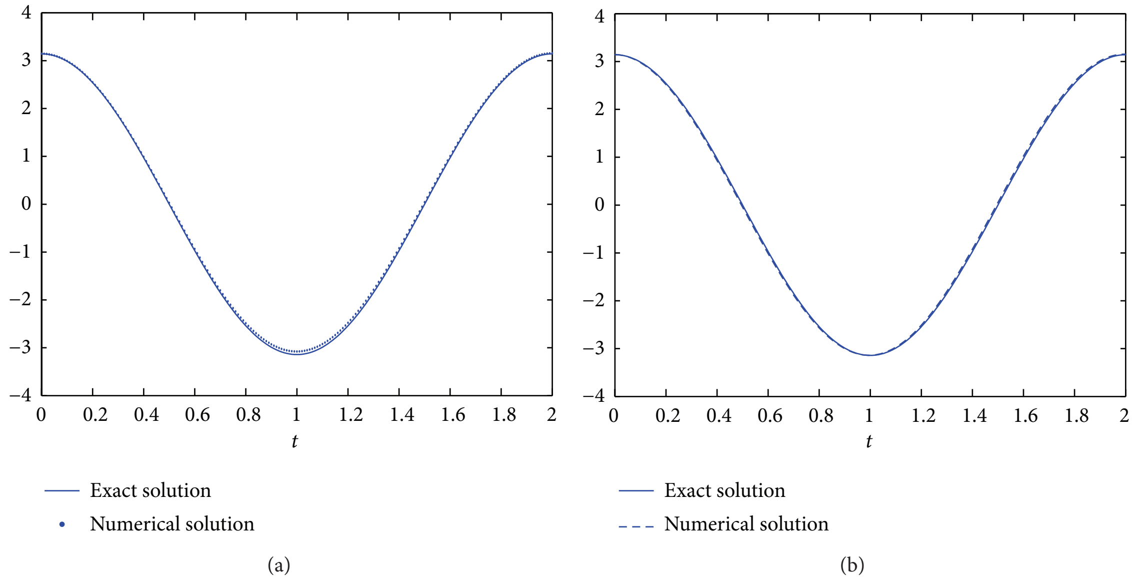


FIGURE 13: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ with $\gamma = 0.001$, (a) MLS and (b) FDM.

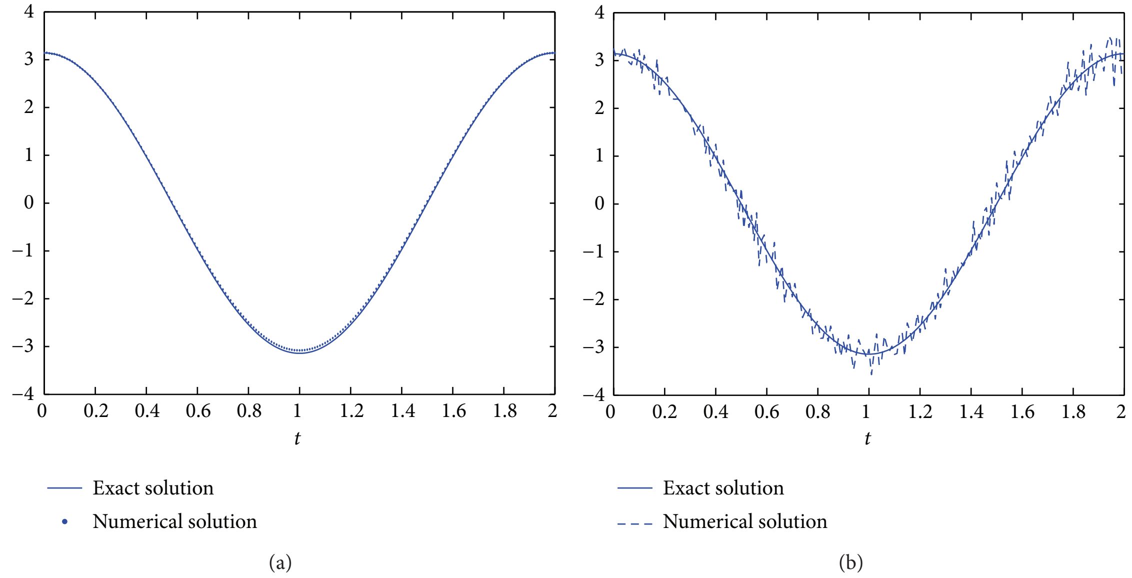


FIGURE 14: The exact solution $f(t)$ and the numerical solution of $\widehat{f}(t)$ with $q = 0.001$, (a) MLS and (b) FDM.

At first, in order to get the approximation effect of $\widehat{f}(t)$ according to different nodes of x and t, we let $\Delta x = 0.5$, $\Delta t = 0.01, 0.001$ and $\Delta x = 0.1$, $\Delta t = 0.01, 0.001$, respectively; the exact solution $f(t)$, the numerical solution $\widehat{f}(t)$, and the error $f(t) - \widehat{f}(t)$ are plotted in Figures 1–4.

As seen from Figures 1 and 2 if the Δx is fixed, the error decreases as Δt decreases, but around the initial point, the error changes a little. When we decrease Δx, from Figures 3 and 4, it is clear that the error around the initial point decreases obviously, and with the decrease of Δt, the total error decreases too.

In Figure 5, we plot the exact solution u and the numerical solutions $\widehat{u}$ at $t = 0.1, 1, 1.5, 2$, respectively, with $\Delta x = 0.1$ and $\Delta t = 0.001$.

Figure 5 shows clearly that the approximate solution $\widehat{u}$ is well for different value of t.

Figures 6 and 7 give the numerical solution $\widehat{u}$ and the error $\widehat{u} - u$ with $\Delta x = 0.1$, $\Delta t = 0.01, 0.001$, respectively.

As seen from Figures 6 and 7 the effect of different Δt to the approximate solution $\widehat{u}$ is the same to $\widehat{f}$; that is, the error decreases according to decreasing Δt.

Next, in order to illustrate the effect of the noisy data on numerical solution, we define the artificial error function as follows:

$$E_{\gamma}(t) = E(t)(1+\gamma), \tag{27}$$

where γ represents the noisy parameter, and we consider two cases in this paper as follows:

(i) γ is const;

(ii) γ is random uniformly distributed on $(-q, q)$.

Let $\gamma = 0.01, 0.001$, respectively; in case (i), we plot the exact solution $f(t)$, the numerical solution $\widehat{f}(t)$, and the error $f(t) - \widehat{f}(t)$ with $\Delta x = 0.1$, $\Delta t = 0.001$, in Figures 8 and 9.

From Figures 8 and 9, we can see that when the noisy data exist in additional specification, the effect of numerical solution is worse than when there is no noisy data. However, the effect of approximation becomes better with the noisy parameter γ decrease; this suggests that the method in this paper is stable.

In case (ii), we let the parameter $q = 0.01$ and 0.001, respectively, and plot the exact solution $f(t)$, the numerical solution $\widehat{f}(t)$, and the error $f(t) - \widehat{f}(t)$ with $\Delta x = 0.1$, $\Delta t = 0.001$, and the results are shown in Figures 10 and 11.

Figures 10 and 11 show that when the noisy parameter is random, the result of numerical solution is also bad, but it is the same to the case that γ is const.; with the parameter q decrease, the effect of approximation becomes better. Moreover, comparing with the fact that γ is const, when the noisy parameter is random, although the result fluctuation is very big, they can be controlled in a certain range.

At last, in order to illustrate the approximate effect, we define the root mean square error of the functions u and f as follows:

$$\begin{gathered} Eu = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(u(x_i,t_j) - \widehat{u}(x_i,t_j)\right)^2}{MN}}, \\ Ef = \sqrt{\frac{\sum_{j=1}^{N}\left(f(t_j) - \widehat{f}(t_j)\right)^2}{N}}, \end{gathered} \tag{28}$$

where $u(x_i, t_j)$ and $f(t_j)$ are the exact solution and $\widehat{u}(x_i, t_j)$ and $\widehat{f}(t_j)$ are the approximate numerical solution. The error results for different Δx and Δt with no noisy data are listed in Table 1.

We will give the error results for different noisy parameter with $\Delta x = 0.1$, $\Delta t = 0.001$ in Tables 2 and 3 and for different Δt with the same noisy parameter in Table 4.

From Tables 2 and 3, we see that the errors vary with the noisy parameter. In Table 4, we find that the approximation effect varies with Δt for the same noisy parameter, and these show that the length of Δt has the effect of regularization, and our method in this paper is stable.

In the next example, we will compare the numerical solution using the finite difference method (FDM) and the meshless collocation method based on moving least squares (MLS), respectively.

Example 2. Consider the conditions

$$\begin{gathered} u(x,0) = \varphi(x) = x^2, \\ u(0,t) = h_0(t) = 2t + \sin(\pi t), \\ u(l,t) = h_l(t) = 4 + 2t + \sin(\pi t), \\ u(x^*,t) = E(t) = 1 + 2t + \sin(\pi t), \end{gathered} \tag{29}$$

and we let $l = 2$, $T = 2$, and $x^* = 1$.

The exact solutions are

$$\begin{gathered} u(x,t) = x^2 + 2t + \sin(\pi t), \\ f(t) = \pi\cos(\pi t). \end{gathered} \tag{30}$$

We plot the exact solution and numerical solution of $f(t)$ in three different cases using FDM and MLS, respectively.

Let $\Delta x = 0.1$, $\Delta t = 0.01$; first, we plot the result in Figure 12 with no noisy data.

In Figures 13 and 14, we let $\gamma = 0.001$ and $q = 0.001$, respectively.

From Figures 12, 13, and 14, we can see that when there is no noisy data, the two methods are stable relatively, and, when there exist noisy data, the method FDM is unstable, especially in the case that the noisy parameter is random; but the method MLS is always stable.

Finally, we give the errors of numerical solution with different cases using these two methods in Table 5.

From Table 5, we obtain that the method MLS is more stable and efficient than the method FDM in any case.

5. Conclusion

In this paper, the meshless method based on the moving least squares is used for solving the inverse problem of parabolic equation with the time-dependent source term. We use some transformations to change the problem, and, in order to overcome the ill-posedness, we use the appropriate length of time. From the experiments, we find that the numerical solution is stable. Moreover, compared with the method FDM, we see that our method is stable and highly efficient.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Hunan Provincial Innovation Foundation for Postgraduate (CX2011B156) and NNSF of China (nos. 11271117 and 11201136).

References

[1] Z. H. Liu and B. Y. Wang, "Coefficient identification in parabolic equations," *Applied Mathematics and Computation*, vol. 209, no. 2, pp. 379–390, 2009.

[2] J. B. Liu, B. Y. Wang, and Z. H. Liu, "Determination of a source term in a heat equation," *International Journal of Computer Mathematics*, vol. 87, no. 5, pp. 969–975, 2010.

[3] A. Hasanov and Z. H. Liu, "An inverse coefficient problem for a nonlinear parabolic variational inequality," *Applied Mathematics Letters*, vol. 21, no. 6, pp. 563–570, 2008.

[4] Z. H. Liu and S. Tatar, "Analytical solutions of a class of inverse coefficient problems," *Applied Mathematics Letters*, vol. 25, no. 12, pp. 2391–2395, 2012.

[5] A. G. Fatullayev and E. Can, "Numerical procedures for determining unknown source parameter in parabolic equations," *Mathematics and Computers in Simulation*, vol. 54, no. 1–3, pp. 159–167, 2000.

[6] A. G. Fatullayev, "Numerical solution of the inverse problem of determining an unknown source term in a heat equation," *Mathematics and Computers in Simulation*, vol. 58, no. 3, pp. 247–253, 2002.

[7] A. Farcas and D. Lesnic, "The boundary-element method for the determination of a heat source dependent on one variable," *Journal of Engineering Mathematics*, vol. 54, no. 4, pp. 375–388, 2006.

[8] A. E. Badia and T. Ha-Duong, "On an inverse source problem for the heat equation. Application to a pollution detection problem," *Journal of Inverse and Ill-Posed Problems*, vol. 10, no. 6, pp. 585–599, 2002.

[9] J. R. Cannon, "Determination of an unknown heat source from overspecified boundary data," *SIAM Journal on Numerical Analysis*, vol. 5, pp. 275–286, 1968.

[10] M. Choulli and M. Yamamoto, "Conditional stability in determining a heat source," *Journal of Inverse and Ill-Posed Problems*, vol. 12, no. 3, pp. 233–243, 2004.

[11] J. J. Monaghan, "An introduction to SPH," *Computer Physics Communications*, vol. 48, no. 1, pp. 89–96, 1988.

[12] E. Oate, F. Perazzo, and J. Miquel, "A finite point method for elasticity problems," *Computers and Structures*, vol. 79, no. 22–25, pp. 2151–2163, 2001.

[13] B. Nayroles, G. Touzot, and P. Villon, "Generalizing the finite element method: diffuse approximation and diffuse elements," *Computational Mechanics*, vol. 10, no. 5, pp. 307–318, 1992.

[14] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-free Galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994.

[15] C. A. Duarte and J. T. Oden, "Hp clouds—an meshless method to solve boundary-value problems," Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1995.

[16] S. N. Atluri and S. Shen, *The Meshless Local Petro-Galerkin (MLPG) Method*, Tech Science Press, Stuttgart, Germany, 2002.

[17] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," *Mathematics of Computation*, vol. 37, no. 155, pp. 141–158, 1981.

[18] D. Levin, "The approximation power of moving least-squares," *Mathematics of Computation*, vol. 67, no. 224, pp. 1517–1531, 1998.

[19] D. Mirzaei and M. Dehghan, "A meshless based method for solution of integral equations," *Applied Numerical Mathematics*, vol. 60, no. 3, pp. 245–262, 2010.

[20] M. G. Armentano, "Error estimates in Sobolev spaces for moving least square approximations," *SIAM Journal on Numerical Analysis*, vol. 39, no. 1, pp. 38–51, 2001.

[21] M. G. Armentano and R. G. Durán, "Error estimates for moving least square approximations," *Applied Numerical Mathematics*, vol. 37, no. 3, pp. 397–416, 2001.

[22] D. Colton, R. Ewing, and W. Rundell, *Inverse Problems in Partial Differential Equation*, SIAM, Philadelphia, Pa, USA, 1990.