

## Research Article

# Adaptive Numerical Method for Approximation of Traffic Flow Equations

Neda Najafzadeh, Saeid Abbasbandy , and Elyas Shivanian 

*Department of Applied Mathematics, Imam Khomeini International University, Qazvin 34149-16818, Iran*

Correspondence should be addressed to Saeid Abbasbandy; [abbasbandy@yahoo.com](mailto:abbasbandy@yahoo.com)

Received 18 May 2022; Revised 29 August 2022; Accepted 6 October 2022; Published 3 November 2022

Academic Editor: Jing Zhao

Copyright © 2022 Neda Najafzadeh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For a long time now, traffic equations have been considered, and different modeling has been done for it. In this article, we work on the macroscopic model, especially the most famous light model. Because these models are among the stiff and shocking problems, theoretical methods do not give good answers to these problems. This paper describes a meshless method to solve the traffic flow equation as a stiff equation. In the proposed method, we also use the exponential time differencing (ETD) method and the exponential time differencing fourth-order Runge–Kutta (ETDRK4). The purpose of this new method is to use methods of the moving least squares (MLS) method and a modified exponential time differencing fourth-order Runge–Kutta scheme. To solve these equations, we use the meshless method MLS to approximate the spatial derivatives and then use method ETDRK4 to obtain approximate solutions. In order to improve the possible instabilities of method ETDRK4, approaches have been stated. The MLS method provided good results for these equations due to its high flexibility and high accuracy and has a moving window and obtains the solution at the shock point without any false oscillations. The technique is described in detail, and a number of computational examples are presented.

## 1. Introduction

Car traffic modeling has been around since time immemorial, around the early twentieth century. Many methods have been modeled according to different cases, from microscopic methods that have been modeled individually for each vehicle to dynamic kinetic and macroscopic methods that deal with the relationship between vehicles and even driver behavior, all of which have been modeled in different ways ([1–5]).

There are many articles on traffic flow modeling. Existing traffic flow simulations are usually based on mesoscopic modeling; that is, models that are microscopic simulations is mixed with macroscopy, meaning that, in some cases, it is based on aggregation, or macroscopic models that are combined with microscopic modes, and in some cases, it is based on the individual mode of each vehicle [6–8].

There are many traffic models in these categories, such as automated cellular models, car chase models, gas kinetic models, fluid flow models, and hydrodynamics [9–11].

Microscopic models have many applications, such as traffic forecasting, accident detection, and traffic control, which are typically described using a system of nonlinear partial differential equations. Macroscopic traffic models look at traffic flow as a cumulative concept and are commonly used to analyze service level and also supply and demand, during regional planning or surface transportation networks [12]. In 1955, Lighthill and Whitham and, in 1956, Richards showed that equations expressing water flow could explain the flow of car traffic. Their idea was to assume machines to be small particles and their density to be principal quantity. They established one of the most famous models of traffic flow called model LWR. The LWR model is the first model of traffic flow in the form of equations of nonlinear hyperbolic partial derivatives [7, 13].

In any case, it makes sense to assume that maintaining the number of vehicles leads to a conservation laws. Because they show severe discontinuities in traffic jams, there is an adaptive relationship between traffic jams and shock waves

[6, 14–16]. Fluid dynamic models are often considered the best simulations for traffic flow because they seem to be better able to detect some phenomena, such as the formation and propagation of shocks on roads, and the answers to problems can create strong discontinuities, even with perfectly smooth initial conditions (see [17]). We investigate traffic simulations from the term of macroscopic models that use fluid dynamics and consist of conservation laws. In Section 2, we will explain these equations in more detail.

In this article, an adaptive numerical method is proposed based on the method the meshless method and ETD scheme. In recent years, meshless methods including MLS scheme have been used for diverse differential equations.

One of the numerical methods examined in this paper and used to discretize time is the “exponential temporal difference” method, which involves the correct integration of dominant equations and then the approximate integral of nonlinear expressions [18, 19]. In principle, this method was in the category of first-order explicit methods, which over time has been extended to explicit and explicit designs. We discuss the sustainability of these methods, which are further explored in Ref. [20]. To improve ETD methods, we consider new and more accurate ETD methods. The method combines ETD scheme with Runge–Kutta [19]. As Cox and Matthews (the originators of this method) recognized, they have a big problem with eigenvalues close to zero, especially when the matrix for the linear part (L) is not diagonal. If these problems are not solved, ETD schemes for PDE equations whose matrix is not diagonal will have many errors and then fail. In this work, we use modified ETD schemes to avoid these problems [21]. Low-order ETD methods are more suitable for application in computational electrodynamics [22]. They are often taken independently [19, 20, 23]. In his article, Iserles stated that in 1928, Filon proposed ideas related to this method in the field of ODE [24, 25]. The best solution to the problems of this method is to use the fourth-order Runge–Kutta formula (ETDRK4) with exponential time difference [19]. Cox and Matthews argued that ETD schemes perform better than implicit-explicit (IMEX) schemes, where linear expressions predominate and perform better than integration-factor (IF) schemes, where nonlinear expressions predominate. We will explain more about this method in Section 3.

In recent decades, methods for better solving equations called meshless methods have been proposed [26]. These methods are used to discretize the domain without using a predefined mesh by creating a system of algebraic equations for the entire problem domain. How this method works is that a set of nodes that gets diffused across the domain of the problem is used to indicate its domain, and a set of nodes is used at the domain boundaries to represent its boundaries. The set of these diffuse nodes is called field nodes [27]. Mesh-free methods can be divided into three groups:

- (1) Mesh-free methods based on weak forms, such as the element-free Galerkin method (EFG) [28]
- (2) Mesh-free methods based on strong forms, such as collocation method based on radial basis functions (RBFs) [29, 30]

- (3) Mesh-free methods based on a combination of weak forms and collocation method [31, 32]

In the written works, several methods without weak mesh are stated which are as follows:

- (1) Dispersed elements method (DEM) [33]
- (2) Smooth particle hydrodynamics (SPH) [34]
- (3) Reproducing kernel particle method (RKPM) [35]
- (4) Boundary node method (BNM) [36]
- (5) Partition unity finite element method (PUFEM) [37]
- (6) Finite sphere method (FSM) [38]
- (7) Boundary point interpolation method (BPIM) [39]
- (8) Border radius point interpolation method (BRPIM) [40]
- (9) Meshless local Petrov-Galerkin (MLPG) [41]
- (10) Meshless local radius point interpolation (MLRPI) [42–47]

One of the meshless methods is moving least squares (MLS) method, in which we use a local interpolation or approximation to express an experimental function with unknown values at some node points [48]. In this paper, the moving least squares (MLS) approximation is used. In Section 4, we discuss this method in detail. In Section 5, we explain about our adaptive method. In Section 6, a number of examples are solved with these methods.

## 2. The Traffic Flow Equation

Traffic modeling and simulation is the subject of research that has attracted much attention today. Travel information technology seeks to alleviate the problems of congestion and travel time that travelers face in most cities around the world. Attempts are being made to answer questions, such as where should the traffic lights or stop signs be installed, what should be the traffic light cycle, and where is the construction site of entrance, exit, and overpass?

The aim of this research is to solve the traffic equation in a way that has the least fluctuations and obtain the solution of the shock point with the least error. When solving traffic equations is done by researchers, more general goals such as increasing car traffic and reducing traffic congestion, accidents, and air pollution can be achieved. This is possible through better scheduling of traffic lights and informing drivers in choosing the route based on real and correct information of traffic situations. These issues have led to the issue of modeling and predicting traffic parameters, as one of the main components has attracted considerable scientific interest. For this reason, mathematical modeling and solving traffic flow models as one of the effective methods of traffic management is very important.

Traffic models can be found in approximately three models: (1) microscopic models, (2) mesoscopic models, and (3) macroscopic models. There are many models by researchers who study traffic from other perspectives, see Refs. [1–5]. In many cases, attention is paid to a section of road or small

sections of the urban network. One of the most widely used models of traffic flow is the macroscopic model that expresses traffic flow as one or a set of nonlinear hyperbolic partial equations. Since there are sudden changes or shocks in the answer to most of these equations, choosing the appropriate numerical method to solve this category of equations is critical.

The macro theory looks at traffic flow as a cumulative concept. This method is the most appropriate method to study the steady state of the flow phenomenon based on physical analogies such as heat flow and fluid flow, and therefore, it best explains the device's potential performance. In these models, traffic flow simulations are performed on a part of the road. The interactions of individual road users are not considered.

Macrosimulation models are commonly used to analyze service, supply, and demand levels during regional planning or large-scale transportation networks. Micro models are more detailed than macro models and can therefore be used to assess the effects of the proposed level of improvement on road facilities with a higher degree of accuracy. However, due to the nature and breadth of information that the shredding models simulate, they run slowly compared to the macro models, so they require lengthy calculations.

In this paper, we examine models from a macroscopic perspective that are inspired by fluid dynamics and composed of conservation laws. The traffic flow equation is

$$\frac{\partial \rho}{\partial t} + \frac{\partial (f)}{\partial x} = 0. \quad (1)$$

Macroscopic models present the parameters of traffic density ( $\rho$ ), flow rate ( $f$ ), and average speed ( $v$ ), and the basic relationship is  $f = \rho v(\rho(x, t))$ , where  $t$  is time and  $x$  is space. The number of vehicles per unit length is called traffic flow density ( $\rho = n/\Delta x$ ). Traffic flow rate indicates the number of vehicles passing a certain point per unit of time  $f = (n/\Delta t)$ .

In the model LWR, speed is expressed as  $v = v(\rho)$ , s.t.  $0 \leq \rho \leq 1$ . Here, it is assumed that the function  $f$  has the following conditions:

- (1) The function  $f$  is continuous until the second time
- (2) The function  $f$  is strictly concave
- (3) Let  $f(0) = f(\rho_{\max}) = 0$

When the number of vehicles gradually increases, the density and therefore the flow also increases. If the number of cars is constantly increased, we will reach a situation where other cars cannot move. In this case, the density is equal to the density of traffic jams and the flow is zero, and  $\rho_{\max}$  is the density of traffic jams.

One of the most important discussions of traffic modeling is how to define the velocity function as a function of density. There are many models for defining velocity in terms of density [49].

### 3. The MLS Approximation Scheme

Meshless methods work without using a predetermined meshing to discretize the problem domain. Among the many meshless methods, we chose method 1 because there were

good reasons for this choice; that is, the development of a smooth weight function that is a guarantee for the continuity of variable fields, very high flexibility of this method, and high accuracy of this method.

MLS combines the moving window with weight functions that have compact support. This method is very accurate even with low-order basis functions and also has very good stability because it uses a local approximation scheme. Calculations are performed in subdomains of the global domain. A very attractive feature of MLS is its flexibility. With the parameters of the weight function, it is easy to provide the appropriate tuning for the MLS and thus control and adjust the data fit.

In the traffic equations, shocks inevitably appear and the fitting method is preferred to solve these equations. The least squares (LS) method is the most used one in data fitting.

Most LS-based methods are global approximation schemes that are not suitable for large amounts of data, irregular or sparse distribution cases. Therefore, the MLS method, which is a local approximation and has high computational accuracy and stability, was proposed. The moving window introduced in MLS shows a good performance in solving these problems. The compact support weight function indicates that only partial measurement data near the unknown measurement point are involved in the calculation. Then, the moving window in MLS acts as a smooth section. Due to the fact that rigid segmentation causes problems in the way of part selection and fitting discontinuity, segment selection is avoided and thus the continuity and smoothness of the connection is guaranteed.

In this section, we explain one of the meshless methods called MLS for the approximation of the function  $v(\mathbf{x})$  in the domain  $\Lambda$ . Assume that  $\Lambda$  is expressed by computational geometry techniques. In this supposed domain, we assume the nodes  $\mathbf{x}_i$ ,  $i = 1, \dots, N$  and denote the approximate value of the related with the MLS method at the point of the assumed node  $i$  with  $v_i$ . As explained in section 1, meshless methods use local interpolation or approximation to express an experimental function that at several node points is the value of unknown variable.

Here, among the meshless methods, we used the moving least square (MLS) method because it seems to work better for shock problems. We consider a subdomain  $\Lambda_s$ . In the neighborhood, the point  $\mathbf{x}$ , which exists within the principal domain of the problem  $\Lambda$ , is represented as the support domain of the MLS approximation for the experimental function at the point  $\mathbf{x}$ .

To approximate the function  $v$  in  $\Lambda_s$ , on random points of nodes  $\mathbf{x}_i$ ,  $i = 1, \dots, N$ . We illustrate the approximation of the  $v$  function by the method of moving least squares with  $v^h(\mathbf{x})$  upon the domain  $\Lambda_s$ , and we can define  $\forall \mathbf{x} \in \Lambda_s$  as

$$v^h(\mathbf{x}) = \mathbf{D}^T(\mathbf{x})\mathbf{q}(\mathbf{x}), \quad \forall \mathbf{x} \in \Lambda_s, \quad (2)$$

where  $\mathbf{D}^T(\mathbf{x})$  is a vector of order  $m$  as  $\mathbf{D}^T(\mathbf{x}) = [d_1(\mathbf{x}), d_2(\mathbf{x}), \dots, d_m(\mathbf{x})]$ .  $\mathbf{D}^T(\mathbf{x})$  is a complete monomial basis, and  $\mathbf{q}(\mathbf{x})$  is a vector since  $\mathbf{q}(\mathbf{x}) = [q_1(\mathbf{x}), q_2(\mathbf{x}), \dots, q_m(\mathbf{x})]$ . Let  $q_j(\mathbf{x})$ ,  $j = 1, 2, \dots, m$  be a function of the coordinates of space  $x$ , and  $d_j(\mathbf{x})$  be monomial in the coordinates of space,

and  $m$  be the number of basic polynomial functions. To make  $d_j(x)$ , the Pascal triangle is used and a complete base is certainly preferred. The basic functions are ascertained by

$$\mathbf{D}^T(\mathbf{x}) = \begin{cases} 1, & m = 1, \\ \{1, \mathbf{x}\}, & m = 2, \\ \{1, \mathbf{x}, \mathbf{x}^2\}, & m = 3. \end{cases} \quad (3)$$

$$\mathbf{D} = \begin{cases} [1, x_1 - x_1^e, (x_1 - x_1^e)^2]^T, & \mathbf{x} \in \mathbb{R}, \\ [1, x_1 - x_1^e, x_2 - x_2^e, (x_1 - x_1^e)^2, (x_2 - x_2^e)^2, (x_1 - x_1^e)(x_2 - x_2^e)]^T, & \mathbf{x} \in \mathbb{R}^2, \end{cases} \quad (5)$$

for  $m = 2$ . The weight function used in the MLS method is as follows:

$$w_i(\mathbf{x}) = \varphi\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{h}\right), \quad (6)$$

$$i = 1, \dots, N,$$

where the function  $\varphi$  is non-negative,  $\gamma$  th times continuously differentiable, and its derivatives are bounded for a higher order than  $\gamma$  and compactly supported. Many weight functions used in meshless methods have these conditions, such as Gaussian, exponential, cubic, and quadratic splines. In this paper, the Gaussian weight function is used, which is defined as follows:

$$w_i(\mathbf{x}) = \begin{cases} \frac{\exp[-(d_i/c_i)^2] - \exp[-(r_s/c_i)^2]}{1 - \exp[-(r_s/c_i)^2]}, & 0 \leq d_i \leq r_s, \\ 0, & d_i \geq r_s, \end{cases} \quad (7)$$

where  $d_i = \|\mathbf{x} - \mathbf{x}_i\|$ ,  $c_i$  is a constant that controls the shape of the weight function  $w_i$  and  $r_s$  is the size of the support domain. The  $c_i$  parameter is a constant that controls the shape of the weight function, which can be dependent on the points for more flexibility. This shape parameter, in general, may vary from one subdomain to another. In addition, this shape parameter can be chosen differently in different directions. As an example, a problem with nine nodes that is uniformly distributed is presented in Figure 1(a). The most appropriate weight function for this case is shown in Figure 1(b). In Figure 2(a), the uniform distribution of nodes depends on the coordinate direction, where the grid spacing is different in the  $x$  and  $y$  directions. To obtain more accurate spatial derivatives, the weight function for this case should be different in each direction in Figure 2(b). In other words, the  $c_i$  parameter must be larger in the  $x$  direction. To read more about weight function and shape parameter, refer to [50].

The support size of the weight function  $w_i$  related to node  $i$  ( $r_s$ ) is very important. On the one hand, it should be chosen so that it is large enough to cover the required

number of nodes in the definition domain of each sample point, and on the other hand, a very small  $r_s$  can also cause a large error in calculating the entries of matrix when using Gauss numeric quadrants. Care must also be taken that  $r_s$  should be small enough, that is, not too small to preserve the local properties of the MLS approximation.

$$\mathbf{D}(\mathbf{x}) = [1, x_1 - x_1^e, x_2 - x_2^e, \dots, x_n - x_n^e]^T, \quad \mathbf{x} \in \mathbb{R}^n, \quad (4)$$

for  $m = 1$  and a quadratic basis by

number of nodes in the definition domain of each sample point, and on the other hand, a very small  $r_s$  can also cause a large error in calculating the entries of matrix when using Gauss numeric quadrants. Care must also be taken that  $r_s$  should be small enough, that is, not too small to preserve the local properties of the MLS approximation.

As it was said  $v^h(\mathbf{x}) = \mathbf{D}^T(\mathbf{x})\mathbf{q}(\mathbf{x})$  that we talked about  $\mathbf{D}^T(\mathbf{x})$ , now we are talking about the coefficients of  $\mathbf{q}(\mathbf{x})$  in detail.

To define a coefficient vector  $\mathbf{q}(\mathbf{x})$ ,  $\mathbf{q}(\mathbf{x})$ , we use the  $L2$ -norm in such a way that these coefficients are determined by minimizing the weighted discrete  $L2$ -norm:

$$\mathbf{J}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) [\mathbf{D}^T(\mathbf{x}_i)\mathbf{q}(\mathbf{x}) - \hat{v}_i]^2 \quad (8)$$

$$= [\mathbf{D}\cdot\mathbf{q}(\mathbf{x}) - \hat{\mathbf{v}}]^T \cdot \mathbf{W} \cdot [\mathbf{D}\cdot\mathbf{q}(\mathbf{x}) - \hat{\mathbf{v}}].$$

Here,  $w_i(\mathbf{x})$  is defined as the weight function for node  $i$ . Note that for all points in the support domain of  $w_i(\mathbf{x})$ ,  $w_i(\mathbf{x})$ , the value of the weight function at each point, i.e.,  $w_i(\mathbf{x})$ ,  $w_i(\mathbf{x})$  is positive.  $\mathbf{x}_i$  represents the value of  $\mathbf{x}$  in node  $i$ , and  $n$  is the number of nodes in  $\Lambda_s$ , for each weight function is  $w_i(\mathbf{x}) > 0$ . The matrices  $\mathbf{D}$  and  $\mathbf{W}$  are defined as

$$\mathbf{D} = \begin{pmatrix} \mathbf{d}^T(\mathbf{x}_1) \\ \mathbf{d}^T(\mathbf{x}_2) \\ \dots \\ \mathbf{d}^T(\mathbf{x}_n) \end{pmatrix}, \quad (9)$$

$$\mathbf{W} = \text{diag}(w_1(\mathbf{x}), \dots, w_n(\mathbf{x})),$$

$$\hat{\mathbf{v}}^T = [\hat{v}_1, \dots, \hat{v}_n].$$

Note that  $\hat{v}_i$ ,  $i = 1, 2, \dots, n$  are not node values of the unknown trial function  $v^h(\mathbf{x})$  but are artificial node values. There is a linear relationship between  $\mathbf{q}(\mathbf{x})$  and  $\hat{\mathbf{v}}$ . This relationship is derived from the stationarity of  $J$  in (2) with respect to  $\mathbf{q}(\mathbf{x})$ , i.e.,

$$\mathbf{Q}(\mathbf{x})\mathbf{q}(\mathbf{x}) = \mathbf{S}(\mathbf{x})\hat{\mathbf{v}}. \quad (10)$$

Define the matrices  $\mathbf{Q}(\mathbf{x})$  and  $\mathbf{S}(\mathbf{x})$  as

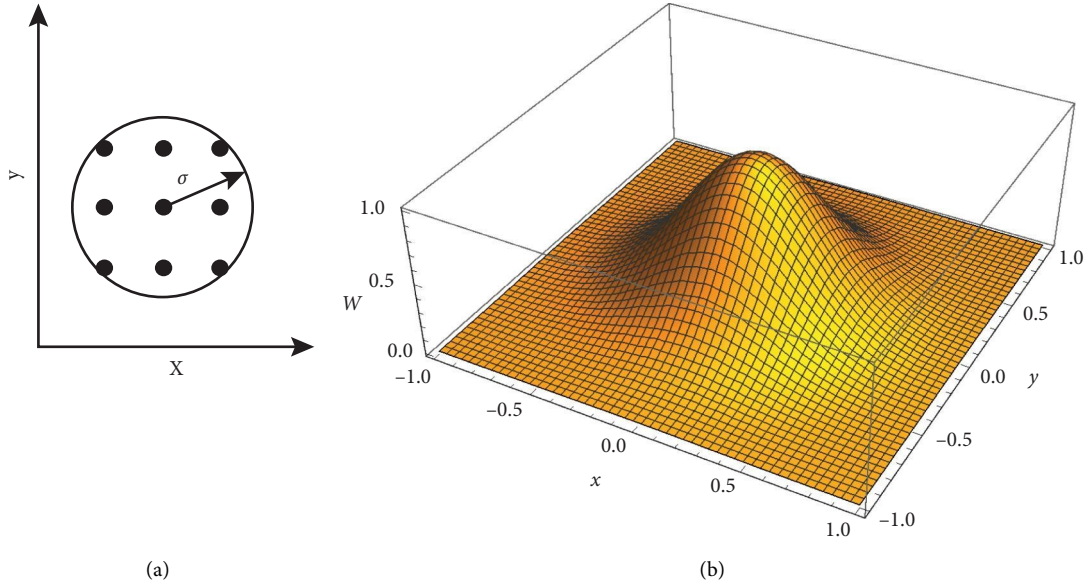


FIGURE 1: (a) Nodes. (b) Weight function.

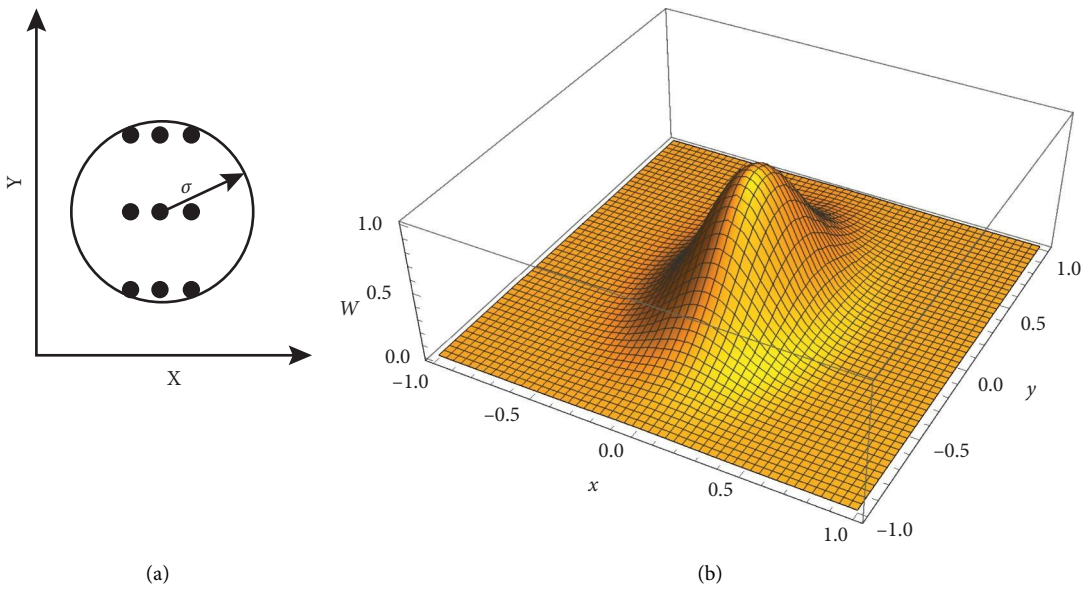


FIGURE 2: (a) Nodes. (b) Weight function.

$$\begin{aligned} \mathbf{Q}(\mathbf{x}) &= \mathbf{D}^T \cdot \mathbf{W} \cdot \mathbf{D}, \\ \mathbf{S}(\mathbf{x}) &= \mathbf{D}^T \cdot \mathbf{W}. \end{aligned} \tag{11}$$

Obtaining  $\mathbf{q}(\mathbf{x})$  from the equation (10) and substituting it in (2) yields

$$\begin{aligned} v^h(\mathbf{x}) &= \boldsymbol{\mu}^T(\mathbf{x}) \cdot \hat{\mathbf{v}} \\ &= \sum_{i=1}^n \mu_i(\mathbf{x}) \hat{v}_i, \quad \mathbf{x} \in \Lambda_s, \end{aligned} \tag{12}$$

$$\boldsymbol{\mu}^T(\mathbf{x}) = \mathbf{D}^T(\mathbf{x}) \mathbf{Q}^{-1}(\mathbf{x}) \mathbf{S}(\mathbf{x}), \tag{13}$$

or

$$\mu_i(\mathbf{x}) = \sum_{j=1}^m \mathbf{d}_j(\mathbf{x}) [\mathbf{Q}^{-1}(\mathbf{x}) \mathbf{S}(\mathbf{x})]_{ji}, \tag{14}$$

where  $\mu_i(\mathbf{x})$  is called a shape function of the MLS approximation dependent on the  $\mathbf{x}_i$  node. From (13) and (11), it can sometimes be concluded that  $\mu_i(\mathbf{x}) = 0$  when  $w_i(\mathbf{x}) = 0$ . In practical applications, usually  $w_i(\mathbf{x})$  is selected such that over the support of  $\mathbf{x}_i$  node is nonzero. In fact,  $\mu_i(\mathbf{x}) = 0$ , for  $\mathbf{x}$  that is not in the support area of the node  $\mathbf{x}_i$ , is to maintain the local character of the MLS approximation.

Note that the sustainability of the linear system is effectively dependent on conditioning of the coefficients matrix. The conditioning of the coefficients matrix can be measured using the condition number.

**Theorem 1.** *If we consider the  $\mathbf{Q}(\mathbf{x})$  matrix that is produced by the basis  $\mathbf{D}(\mathbf{x})$  (i.e.,  $\mathbf{Q}(\mathbf{x}) = \mathbf{D}^T \mathbf{W}(\mathbf{x}) \mathbf{D}$ ), then there is a bounded and countable number  $C_d(\mathbf{x}, n, m)$ , which is independent of  $h$  and such that for the determinate of  $\mathbf{Q}(\mathbf{x})$ , we get*

$$\det(\mathbf{Q}(\mathbf{x})) = C_d(\mathbf{x}, n, m) h^2 \sum_{i=1}^m \hat{i}, \quad \forall \mathbf{x} \in \Lambda. \quad (15)$$

In addition, there is a constant  $h_0 > 0$ , for  $h \leq h_0$ , such that a bounded and computable number  $C_c(\mathbf{x}, n, m)$  is independent of  $h$  that the norm 2 condition number of  $\mathbf{Q}(\mathbf{x})$  is

$$\text{cond}(\mathbf{Q}(\mathbf{x})) = C_c(\mathbf{x}, n, m) h^{-2\hat{m}}, \quad \forall \mathbf{x} \in \Lambda. \quad (16)$$

*Proof.* A proof is described in [51].

Equation (15) states that

$$\det(\mathbf{Q}(\mathbf{x})) \longrightarrow 0 \text{ as } h \longrightarrow 0. \quad (17)$$

Note that from the above discussion we conclude that whenever a sufficiently small nodal distance  $h$  is selected, the instantaneous matrix  $\mathbf{Q}(\mathbf{x})$  obtained in MLS approximation is approximately a single matrix. This causes the errors to increase; that is, when the instantaneous matrix  $\mathbf{Q}(\mathbf{x})$  has ill conditions or is singular, the answers obtained for the vector of coefficients  $\mathbf{q}(\mathbf{x})$  have a large error, and therefore, the function of the shape  $\mu_i(\mathbf{x})$  will have a large rounding error.

Referring to (16), we find that the condition number  $\mathbf{Q}(\mathbf{x})$ ,  $\mathbf{Q}(\mathbf{x})$  has an inverse relationship with factor  $h^{2\hat{m}}$ . That is, as  $h^{2\hat{m}}$  decreases, the condition number increases and as a result the rounding error increases because it is very clear that when the matrix conditioning number increases, the degree of ill conditioning of the matrix increases to the point that for singular matrices, the situation worsens and the condition number infinitely desires. Therefore, we conclude that the instability of the MLS method depends on  $h$  and  $\hat{m}$  so that by reducing the value of  $h$  and increasing the value of  $\hat{m}$  too much, the instability of the MLS method increases. So we conclude that the MLS method achieves well and low-error solutions to solve problems when matrix  $\mathbf{Q}$  in (10) is nonsingular. This happens when the rank  $\mathbf{D}$  is equal to  $m$ .

In order not to have the problem of instability and MLS method being a well-defined method, a necessary condition is that there must be the least at least  $m$  nonzero weight functions (i.e.,  $n > m$ ) for each point  $\mathbf{x} \in \Lambda$  and the points determined in  $\Lambda_s$  do not follow from a regular and special pattern.  $\square$

#### 4. A Modified ETD Scheme

The traffic flow equation combines nonlinear terms with linear terms. To solve these problems and obtain low-error

numerical answers to these problems, high-order approximations for place and time must be used. Because we have a nonlinear and stiff combination in these problems, it has not been used much until the second order. The general form of the stiff equations is as follows:

$$v_t = \mathcal{L}v + \mathcal{N}(v, t). \quad (18)$$

In this equation,  $\mathcal{L}$  represents the linear operator and  $\mathcal{N}$  represents the nonlinear operator. First, the spatial part of the PDE equation is discretized by the method of the previous section, and thus we obtain a system of ODEs as

$$v_t = \mathbf{L}v + \mathbf{N}(v, t), \quad (19)$$

where  $\mathbf{L}, \mathbf{N}$  are the matrices obtained after using method MLS and approximating spatial derivatives on linear and nonlinear operators,  $\mathcal{L}$  and  $\mathcal{N}$ . For this special traffic equation, both matrices are written in Section 5. We begin by multiplying equation (19) through by  $e^{-\mathbf{L}t}$ , where the term  $e^{-\mathbf{L}t}$  is the integrating factor, then we take the integral from the equation on a single time step of length  $h$  from  $t = t_n$  to  $t = t_{n+1} = t_n + h$  to give

$$v_{n+1} = e^{\mathbf{L}h} v_n + e^{\mathbf{L}h} \int_0^h e^{-\mathbf{L}\zeta} \mathbf{N}(v(t_n + \zeta), t_n + \zeta) d\zeta. \quad (20)$$

In this equation, different orders of ETD schemes can be obtained from how the integrals are approximated. A sequence of recursive formulas is presented that obtains higher order approximations than the multistage type. And the generating formula is

$$v_{n+1} = e^{\mathbf{L}h} v_n + h \sum_{m=0}^{s-1} \eta_m \sum_{k=0}^m (-1)^k \binom{m}{k} \mathbf{N}_{n-k}, \quad (21)$$

wherein  $s$  represents the order of the method. The coefficients  $\eta_m$  can be obtained from the following recursive relation:

$$\begin{aligned} \mathbf{L}h\eta_0 &= e^{\mathbf{L}h} - \mathbf{I}, \\ \mathbf{L}h\eta_{m+1} + \mathbf{I} &= \eta_m + \frac{1}{2}\eta_{m-1} + \frac{1}{3}\eta_{m-2} \cdots + \frac{1}{m+1}\eta_0. \end{aligned} \quad (22)$$

Whenever we use Runge–Kutta methods in the set of ETD methods and derive its time step with Runge–Kutta, then the method is called ETD RK. In this paper, we use only the fourth-order method and derive it with ETD methods. In this case, the method is called ETD RK4. This is not entirely clear and needs to be explained in the form of a symbolic manipulation system. Here are the formulas for the ETD RK4 method:

$$\begin{aligned}
a_n &= e^{\mathbf{L}h/2} v_n + \frac{(e^{\mathbf{L}h/2} - \mathbf{I})}{\mathbf{L}} \mathbf{N}(v_n, t_n), \\
b_n &= e^{\mathbf{L}h/2} v_n + \frac{(e^{\mathbf{L}h/2} - \mathbf{I})}{\mathbf{L}} \mathbf{N}(a_n, t_n + h/2), \\
c_n &= e^{\mathbf{L}h/2} a_n + \frac{(e^{\mathbf{L}h/2} - \mathbf{I})}{\mathbf{L}} (2\mathbf{N}(b_n, t_n + h/2) - \mathbf{N}(v_n, t_n)), \\
\omega &= -4 - \mathbf{L}h + e^{\mathbf{L}h} (4 - 3\mathbf{L}h + (\mathbf{L}h)^2), \varrho = 2 + \mathbf{L}h + e^{\mathbf{L}h} (-2 + \mathbf{L}h), \varsigma = -4 - 3\mathbf{L}h - (\mathbf{L}h)^2 + e^{\mathbf{L}h} (4 - \mathbf{L}h), v_{n+1} \\
&= e^{\mathbf{L}h} v_n + \frac{\omega \mathbf{N}(v_n, t_n) + 2\varrho (\mathbf{N}(a_n, t_n + h/2) + \mathbf{N}(b_n, t_n + h/2)) + \varsigma \mathbf{N}(c_n, t_n + h)}{h^2 \mathbf{L}^3}.
\end{aligned} \tag{23}$$

The coefficients inside the ETDRK4 formula can be rewritten as follows:

$$\begin{aligned}
\alpha &= \frac{\omega}{h^2 \mathbf{L}^3}, \\
\beta &= \frac{\varrho}{h^2 \mathbf{L}^3}, \\
\gamma &= \frac{\varsigma}{h^2 \mathbf{L}^3}.
\end{aligned} \tag{24}$$

As mentioned before, we tried to solve the traffic equations with the methods described, and an example of these traffic equations is solved in the next section.

In the written codes, for all these formulas, division by matrix  $\mathbf{L}$  means the inverse of the matrix  $\mathbf{L}$  is multiplied in the desired expression and to calculate  $e^{\mathbf{L}}$ , commands `exp` and `expm` of MATLAB software are used, if  $\mathbf{L}$  is diagonal, command `exp` is used, and if it is a nondiagonal matrix, command `expm`, `expm` is used.

## 5. Adaptive Method

In this article, we have tried to solve equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial f(\rho(x, t))}{\partial x} = 0, \tag{25}$$

by a adaptive method. In this section, we explain how to solve this equation according to methods MLS and ETDRK4. To solve equation, we first use the MLS method to approximate spatial derivatives and then use the ETDRK4 method to approximate time derivatives. We divide the main domain of the problem ( $\Lambda$ ) into  $N$  parts so that the distance between the nodes is constant, and for each point, we consider a neighborhood ( $\Lambda_s$ ) that is inside the main domain. To approximate the function  $\rho$ , by MLS method, in  $\Lambda_s$ , on random points of nodes  $x_i$ ,  $i = 1, \dots, n$ , from (12) and (14), we get

$$\rho^h(x, t) = \sum_{i=1}^n \mu_i(x) \widehat{\rho}_i(t). \tag{26}$$

By replacing the equation, we have

$$\frac{\partial \rho}{\partial t} + \frac{\partial f(\sum_{i=1}^n \mu_i(x) \widehat{\rho}_i(t))}{\partial x} = 0. \tag{27}$$

Note that  $\widehat{\rho}_i(t) = \widehat{\rho}(x_i, t)$  in (27) are node values that are fictitious. On the other hand, for the derivative  $\mu_i(x)$ , it can be explained as follows.

Let  $C^q(\Lambda)$  be the space of functions that are continuously  $q$ th differentiable on  $\Lambda$ . If  $w_i(x) \in C^q(\Lambda)$  and  $d_j(x) \in C^s(\Lambda)$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ , then  $\mu_i(x) \in C^r(\Lambda)$  with  $r = \min(q, s)$ . The  $k$  derivatives of  $\mu_i(x)$  becomes

$$\mu_{i,k} = \sum_{j=1}^m \left[ d_{j,k} (Q^{-1}S)_{ji} + d_j (Q^{-1}S_{,k} + Q_{,k}^{-1}S)_{ji} \right], \tag{28}$$

where we show the derivative inverse of the matrix  $Q$ , which depends on  $x_k$ , with the symbols  $Q_{,k}^{-1} = (Q^{-1})_{,k}$ , which can be obtained by the formula  $Q_{,k}^{-1} = -Q^{-1}Q_{,k}Q^{-1}$ . As mentioned in Section 2, the density function is  $f = \rho V(\rho)$ . In the first example  $V(\rho) = v_{\max}(1 - \rho)$ , where  $v_{\max} = 1$ , then we can write

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho}{\partial x} + 2\rho \frac{\partial \rho}{\partial x}. \tag{29}$$

Now, considering the spatial discretization and using the MLS method, we have

$$\frac{\partial \rho(x, t)}{\partial t} = -\sum_{j=1}^n \mu'_j(x) \widehat{\rho}_j(t) + 2\rho(x, t) \sum_{j=1}^n \mu'_j(x) \widehat{\rho}_j(t). \tag{30}$$

For  $x_i \in \Lambda$  ( $\Lambda$  is the global domain of problem), we have

$$\rho'_i(t) = -\sum_{j=1}^n \mu'_j(x_i) \widehat{\rho}_j(t) + 2\rho_i(t) \sum_{j=1}^n \mu'_j(x_i) \widehat{\rho}_j(t). \tag{31}$$

Note that the derivative of  $\rho$  is relative to time, and the derivative of  $\phi$  is defined in 20. When we disassemble the PDE space partition, we get a system of ODEs as

$$\begin{bmatrix} \rho_1'(t) \\ \rho_2'(t) \\ \vdots \\ \rho_n'(t) \end{bmatrix} = - \begin{bmatrix} \mu_1'(x_1) & \mu_2'(x_1) & \dots & \mu_n'(x_1) \\ \mu_1'(x_2) & \mu_2'(x_2) & \dots & \mu_n'(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1'(x_n) & \mu_2'(x_n) & \dots & \mu_n'(x_n) \end{bmatrix} \begin{bmatrix} \widehat{\rho}_1(t) \\ \widehat{\rho}_2(t) \\ \vdots \\ \widehat{\rho}_n(t) \end{bmatrix} + 2 \begin{bmatrix} \rho_1(t) \\ \rho_2(t) \\ \vdots \\ \rho_n(t) \end{bmatrix} \circ \begin{bmatrix} \mu_1'(x_1) & \mu_2'(x_1) & \dots & \mu_n'(x_1) \\ \mu_1'(x_2) & \mu_2'(x_2) & \dots & \mu_n'(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1'(x_n) & \mu_2'(x_n) & \dots & \mu_n'(x_n) \end{bmatrix} \begin{bmatrix} \widehat{\rho}_1(t) \\ \widehat{\rho}_2(t) \\ \vdots \\ \widehat{\rho}_n(t) \end{bmatrix}, \quad (32)$$

where  $\circ$  is the Hadamard product.

In this step, to approximate time derivatives by method ETDRK4, we have to divide the flux function into two parts, linear and nonlinear:

$$\mathbf{L} = - \begin{bmatrix} \mu_1'(x_1) & \mu_2'(x_1) & \dots & \mu_n'(x_1) \\ \mu_1'(x_2) & \mu_2'(x_2) & \dots & \mu_n'(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1'(x_n) & \mu_2'(x_n) & \dots & \mu_n'(x_n) \end{bmatrix}, \quad (33)$$

and

$$\mathbf{N}(\rho, t) = 2 \begin{bmatrix} \rho_1(t) \\ \rho_2(t) \\ \vdots \\ \rho_n(t) \end{bmatrix} \circ \begin{bmatrix} \mu_1'(x_1) & \mu_2'(x_1) & \dots & \mu_n'(x_1) \\ \mu_1'(x_2) & \mu_2'(x_2) & \dots & \mu_n'(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1'(x_n) & \mu_2'(x_n) & \dots & \mu_n'(x_n) \end{bmatrix} \begin{bmatrix} \widehat{\rho}_1(t) \\ \widehat{\rho}_2(t) \\ \vdots \\ \widehat{\rho}_n(t) \end{bmatrix}. \quad (34)$$

For time discretization, the method ETD, which is described in section, is used. Assuming that we divide the main domain into  $n$  parts with equal distances, it is assumed that  $\rho(x_i, kt)$ ,  $i = 1, 2, \dots, n$  is specified, and our goal is to calculate  $\rho(x_i, (k + 1)t)$ ,  $i = 1, 2, \dots, n$ . Multiply the equation by  $e^{-Lt}$  and then integrate the equation into a single time step of length  $h$ , so we get

$$\rho_i^{k+1} = e^{Lh} \rho_i^k + e^{Lh} \int_0^h e^{-L\zeta} 2\rho_i(t_k + \zeta) \sum_{j=1}^n \mu_j'(x_i) \widehat{\rho}_j(t_k + \zeta) d\zeta, \quad (35)$$

where  $\rho_i^k = \rho(x_i, t_k)$  and then we solve it using the formulas of method ETDRK4 that is expressed in (15), but we have problems with the ETDRK4 method and that it becomes unstable for some points. One way to solve this problem is to use a Taylor expansion point with a cut-off point for small eigenvalues so that if the matrix is a linear diagonal operator, the Taylor expansion is used for the diagonal points below the cut, but one of the serious problems with this method is that it possible to extend this method to nondiagonal problems? That is why we went the other way. Another way that can be used is to use the contour integral of the idea of complex analysis. In fact, we have a function to evaluate that has a single point and is analytical elsewhere. Near this singular point, the function has a numerical error. The suggested solution is to use a contour integral on a complex plane that includes  $z$  and is separated from  $z = 0$  by

$$f(z) = (2\pi i)^{(-1)} \int_{\Gamma} f(s) (s - z)^{(-1)} ds. \quad (36)$$

Now instead of  $z$ -scalar we use the matrix  $\mathbf{L}$  and apply the same method again, only the expression  $1/(s - z)$  becomes the matrix  $(sI - \mathbf{L})^{-1}$  as

$$f(\mathbf{L}) = (2\pi i)^{(-1)} \int_{\Gamma} f(s) (sI - \mathbf{L})^{-1} ds, \quad (37)$$

where the contour must be such that it contains the eigenvalues of  $\mathbf{L}$ . The result of contour integrals of functions in the complex plane is obtained using the trapezoidal rule, which converges exponentially.

The contour area can be considered as a circle, and we usually consider 32 points with equal distance in this circle. There are many different choices for the contour region that work well, but the important thing is that the eigenvalues are really restricted by  $\Gamma$ . When  $\mathbf{L}$  is real, we use only the upper half of the circle for the contour area whose center is the real axis, then we get the real part of the result. The contour integral can be approximated by the mean  $f(s)$  on  $\Gamma$  so that we obtain the mean at points of equal distance along  $\Gamma$ , or again consider only the upper half of  $\Gamma$ , and then consider only that real part.

## 6. Numerical Example

We have described the method implemented in this article in detail in the previous sections. In this section, we will solve two examples of traffic equations and compare their numerical answer form with other methods. As explained, because these equations are in the category of stiff equations, it is very important to display the numerical answer obtained with the least vibrations at the moment of shock and after.

6.1. Example 1. Consider the following traffic flow equation:

$$\frac{\partial \rho(x, t)}{\partial t} + \frac{\partial F(\rho(x, t))}{\partial x} = 0. \quad (38)$$

The density function in this model is expressed as follows:

$$F(\rho(x, t)) = \rho V(\rho), \quad (39)$$

where  $V$  is

$$V(\rho) = v_{\max}(1 - \rho), \quad 0 \leq \rho \leq 1. \quad (40)$$

Here, we assume that  $v_{\max} = 1$ , and the initial conditions are as follows:

$$\rho(x, 0) = 0.25 + 0.7e^{-\beta x^2}, \quad (41)$$

where  $\beta = 0.01$ .

The density bulge (or density shock) changes over time. After a while, the shock wave simulation shows that the density of the traffic flow increases and the speed decreases very quickly (drivers who were speeding in light traffic suddenly had to brake). As vehicles exit the shock zone, they gradually accelerate, resulting in a gradual decrease in



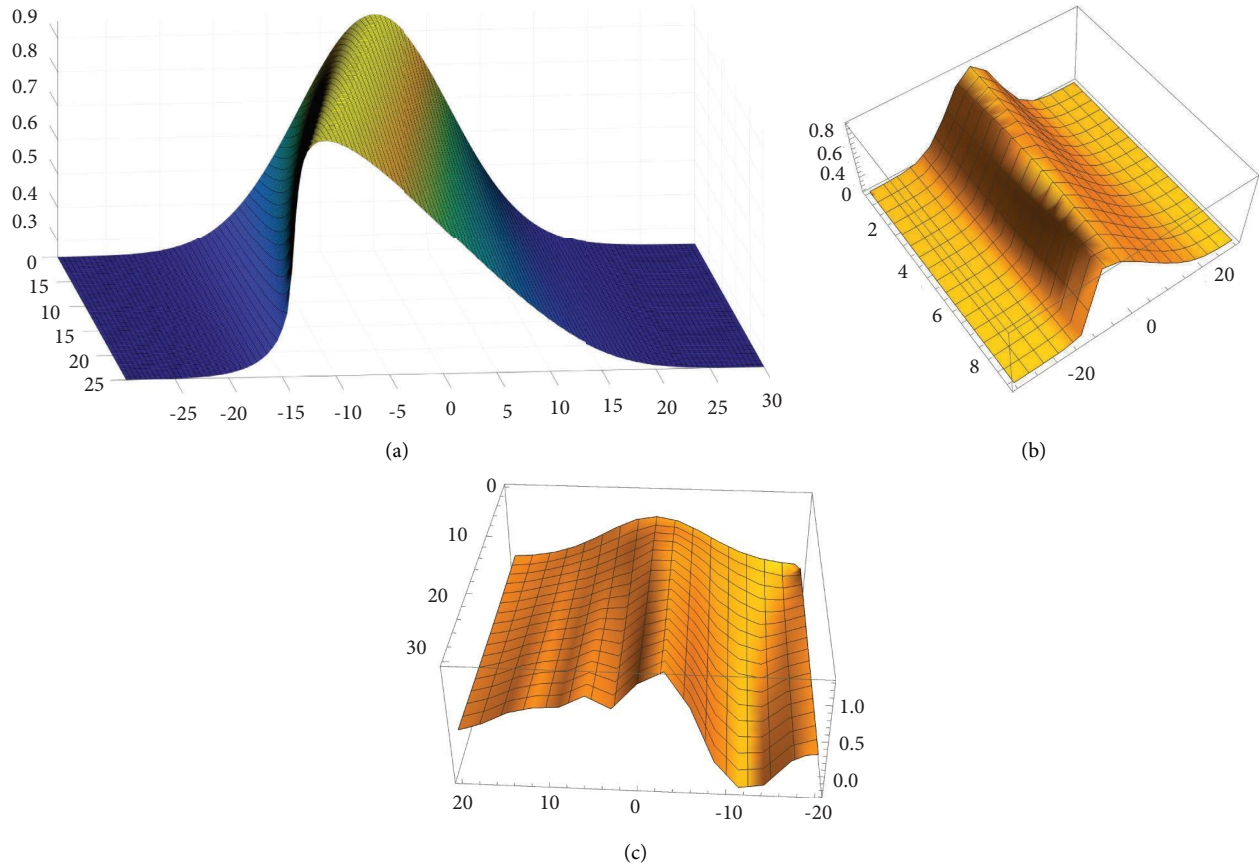


FIGURE 3: A comparison of the solution of equation of Example 1 using the present method: (a) with solution by Mathematica Software and (b) solution by the method of reproduction kernel.

density. Figure 1(a) shows the numerical answer to this problem in the method presented in this paper and rejects the shape of the shock part well. The figure is drawn to  $t = 30$ .

Because these problems do not have an exact answer and they are very difficult to solve, we tried to solve this problem with different methods, and the best answer is obtained from the method mentioned in the article. It is important that the movement of the shock point in the figures after the occurrence of traffic is completely clear, but in some methods, since the occurrence of the shock, the error is very high and does not show the movement of the shock. However, in the method mentioned in the article, as shown in Figure 3(a), it shows the shock point well.

Compared to Figure 3(b), which was obtained by Mathematica Software and did not pass from time 8 due to shock, but in our calculation method, it was obtained up to time 30 and the figure is acceptable. In Figure 3(c), an attempt has been made to solve the equation by the method of reproduction kernel, where unfortunately the error has increased so much after the shock time, which is quite clear in the figure. Also, in Figure 4, we compare the moment  $t = 25$ , which is the time after the shock, with the figure

shown in Ref. [52], which shows the routes of individual vehicles and the density observed by each driver. A comparison of the obtained solution using the present method with MLS and without MLS is shown in Figure 5. Table 1 shows a comparison of the obtained solution using the article method with other numerical methods.

6.2. *Example 2.* In the second example, we consider another model of the traffic problem in which the density function is the same as in the previous example, but the initial data are piecewise function:

$$\rho(x, 0) = \begin{cases} 1, & 4 \leq x \leq 6, \\ 0.5, & \text{otherwise.} \end{cases} \quad (42)$$

This can be interpreted as cars approaching a congested traffic or encountering a line of cars behind a red light waiting to turn green. This example describes the traffic situation at the traffic light installation site. Here, the traffic light is in position  $x = 6$ , and at moment  $t = 0$ , it has changed from red to green. Figure 6 shows the numerical solutions up to time  $t = 30$  computed in the new proposed scheme.

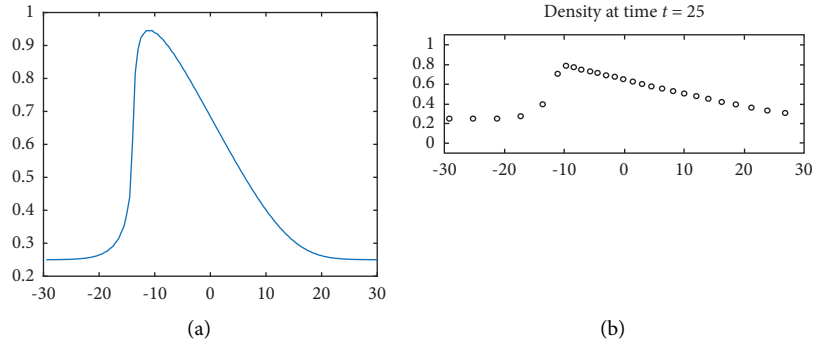


FIGURE 4: A comparison of the solution of equation of example 1 using the present method with the solution in Ref. [52].

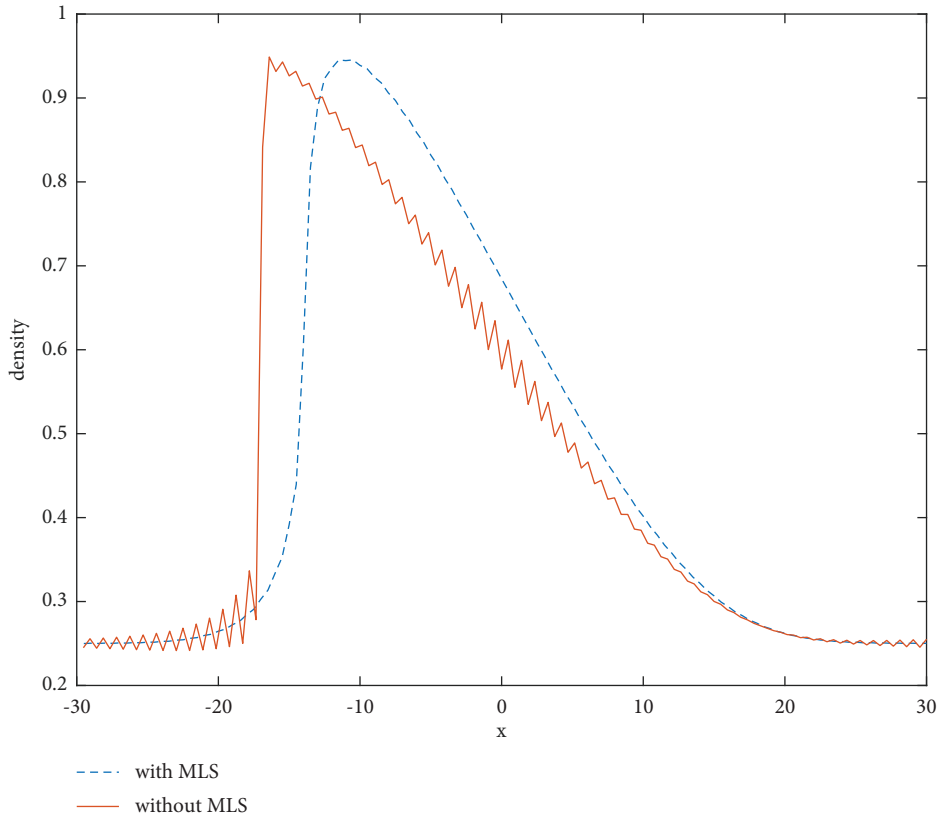


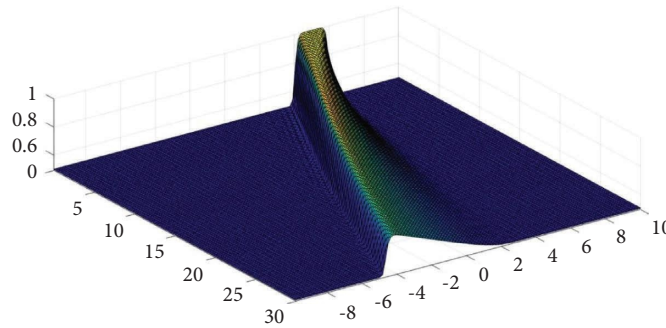
FIGURE 5: A comparison of the solution of equation of example 1 using the present method with the solution without MLS.

TABLE 1: A comparison of the solution of equation of example 1 using the present method with other numerical methods.

| $t$ | $x$ | Without MLS | Mathematica | New method |
|-----|-----|-------------|-------------|------------|
| 5   | -20 | 0.2624      | 0.2640      | 0.2631     |
|     | -10 | 0.6136      | 0.5600      | 0.5613     |
|     | 0   | 0.8872      | 0.9089      | 0.9184     |
|     | 10  | 0.4653      | 0.4695      | 0.4726     |
|     | 20  | 0.2618      | 0.2624      | 0.2626     |
| 10  | -20 | 0.2628      | .....       | 0.2633     |
|     | -10 | 0.7288      | .....       | 0.6537     |
|     | 0   | 0.8877      | .....       | 0.8535     |
|     | 10  | 0.4334      | .....       | 0.4476     |
|     | 20  | 0.2616      | .....       | 0.2625     |

TABLE 1: Continued.

| $t$ | $x$ | Without MLS | Mathematica | New method |
|-----|-----|-------------|-------------|------------|
| 15  | -20 | 0.2632      | .....       | 0.2636     |
|     | -10 | 0.8444      | .....       | 0.8151     |
|     | 0   | 0.7369      | .....       | 0.7877     |
|     | 10  | 0.4112      | .....       | 0.4285     |
|     | 20  | 0.2615      | .....       | 0.2623     |
| 20  | -20 | 0.2599      | .....       | 0.2639     |
|     | -10 | 0.9019      | .....       | 0.9347     |
|     | 0   | 0.7410      | .....       | 0.7313     |
|     | 10  | 0.3976      | .....       | 0.4134     |
|     | 20  | 0.2621      | .....       | 0.2622     |
| 25  | -20 | 0.2538      | .....       | 0.2646     |
|     | -10 | 0.8550      | .....       | 0.9388     |
|     | 0   | 0.5769      | .....       | 0.6844     |
|     | 10  | 0.3849      | .....       | 0.4016     |
|     | 20  | 0.2606      | .....       | 0.2620     |

FIGURE 6: Example 2: for  $N = 120$  and  $t = 30$ .

## 7. Conclusion

Traffic issues are very important, and on the other hand, they are very difficult to solve due to problems such as having a shock, and it is very important to choose a good numerical method for traffic flow models.

In the present article, a new method has been proposed to identify shock points well and to find a better answer without vibrations in these points. In the method described in the article, first we solved such problems from meshless methods and then with the ETDRK4 method. We chose the MLS method from among the meshless methods. In this method, we used quadratic basic functions. Then, we used the ETDRK4 method to approximate the time derivative. Due to the high flexibility and accuracy of MLS, it is considered to solve these traffic equations, which have shocks. Because they discretize the problem domain without using a predetermined grid, it seems to work well for equations that have shocks at points in the domain. On the other hand, contour integral and other suitable approaches are used for controlling the instability and challenges of ETDRK4. Now, by combining these methods, we get good results for solving these equations. In MLS, parameter  $c$  can be changed, and with it, the shape of the weight function can be controlled,

and the type of weight function can be different. In ETDRK4, as mentioned, we used contour integral to control instability, but contour integrals are not the only solutions offered for this problem, but the contour integral method attracts us due to its generality for dealing with arbitrary functions. There is considerable flexibility with this procedure. Finally, two examples of traffic equations presented in this way were solved. The results showed that this method can successfully solve traffic problems.

## Data Availability

No data were used to support the study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] C. Daganzo, *Fundamental of Transportation and Traffic Operations*, New York, NY, USA, 1997.
- [2] S. Hoogendoorn, *Continuum Modelling of Multilane Traffic Flow*, Delft University Press, Delft, the Netherlands, 1999.

- [3] A. D. May, *Traffic Flow Fundamentals*, Prentice-Hall, Hoboken, NJ, USA, 1990.
- [4] M. Papageorgiou, *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control*, Springer-Verlag, Secaucus, NJ, USA, 1983.
- [5] R. Vaughan, *Urban Spatial Traffic Patterns*, Pion, London, UK, 1987.
- [6] A. Sopasakis and M. A. Katsoulakis, "Stochastic modeling and simulation of traffic flow: asymmetric single exclusion process with arrhenius look-ahead dynamics," *SIAM Journal on Applied Mathematics*, vol. 66, no. 3, pp. 921–944, 2006.
- [7] M. J. Lighthill and G. B. Whitham, "On kinematic waves: II. a theory of traffic flow on long crowded roads," *Proceedings Royal Society, London, Series A*, vol. 229, no. 1178, pp. 317–345, 1955.
- [8] A. Sopasakis, "Stochastic noise approach to traffic flow modeling," *Physica A: Statistical Mechanics and its Applications*, vol. 342, no. 3–4, pp. 741–754, 2004.
- [9] C. F. Daganzo, "In traffic flow, cellular automata=kinematic waves," *Transportation Research Part B: Methodological*, vol. 40, no. 5, pp. 396–403, 2006.
- [10] D. Helbing, A. Hennecke, V. Shvetsov, M. Treiber, and M. Treiber, "Master: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model," *Transportation Research Part B: Methodological*, vol. 35, no. 2, pp. 183–211, 2001.
- [11] H. M. Zhang and T. Kim, "A car-following theory for multiphase vehicular traffic flow," *Transportation Research Part B: Methodological*, vol. 39, no. 5, pp. 385–399, 2005.
- [12] A. R. Soheili, A. Kerayechian, H. R. Tareghian, and N. Davoodi, "Adaptive numerical simulation of traffic flow density," *Computers & Mathematics with Applications*, vol. 66, no. 3, pp. 227–237, 2013.
- [13] P. I. Richards, "Shock waves on the highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.
- [14] G. Bretti, R. Natalini, and B. Piccoli, "Numerical approximations of a traffic flow model on networks," *Networks and Heterogeneous Media*, vol. 1, no. 1, pp. 57–84, 2006.
- [15] M. Garavello and B. Piccoli, "Traffic flow on a road network using the aw-rasclé model," *Communications in Partial Differential Equations*, vol. 31, no. 2, pp. 243–275, 2006.
- [16] M. Garavello and B. Piccoli, *Traffic Flow on Networks*, AIMS Series on Applied Mathematics, Pasadena, CA, USA, 2006.
- [17] A. Bressan, *Hyperbolic Systems of Conservation Laws- the One-Dimensional Cauchy Problem*, Oxford Univ. Press, Oxford, UK, 2000.
- [18] C. Schuster, A. Christ, and W. Fichtner, "Review of FDTD time-stepping schemes for efficient simulation of electric conductive media," *Microwave and Optical Technology Letters*, vol. 25, no. 1, pp. 16–21, 2000.
- [19] S. M. Cox and P. C. Matthews, "Exponential time differencing for stiff systems," *Journal of Computational Physics*, vol. 176, no. 2, pp. 430–455, 2002.
- [20] G. Beylkin, J. M. Keiser, and L. Vozovoi, "A new class of time discretization schemes for the solution of nonlinear PDEs," *Journal of Computational Physics*, vol. 147, no. 2, pp. 362–387, 1998.
- [21] A. K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff PDEs," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1214–1233, 2005.
- [22] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Boston, MA, USA, 1995.
- [23] D. R. Mott, E. S. Oran, and B. van Leer, "A quasi-steady state solver for the stiff ordinary differential equations of reaction kinetics," *Journal of Computational Physics*, vol. 164, no. 2, pp. 407–428, 2000.
- [24] L. N. G. Filon, "III.—on a quadrature formula for trigonometric integrals," *Proceedings of the Royal Society of Edinburgh*, vol. 49, pp. 38–47, 1930.
- [25] A. Iserles, "On the numerical quadrature of highly oscillating integrals I: fourier transforms," *IMA Journal of Numerical Analysis*, vol. 24, no. 3, pp. 365–391, 2004.
- [26] G. Liu and Y. Gu, *An Introduction to Meshfree Methods and Their Programming*, Springer, Berlin, Germany, 2005.
- [27] E. Shivanian, "Local integration of population dynamics via moving least squares approximation," *Engineering with Computers*, vol. 32, no. 2, pp. 331–342, 2016.
- [28] M. Dehghan and A. Ghesmati, "Combination of meshless local weak and strong (mlws) forms to solve the two dimensional hyperbolic telegraph equation," *Engineering Analysis with Boundary Elements*, vol. 34, no. 4, pp. 324–336, 2010.
- [29] M. Aslefallah and E. Shivanian, "Nonlinear fractional integrodifferential reaction–diffusion equation via radial basis functions," *Eur Phys J Plus*, vol. 130, no. 3, pp. 47–49, 2015.
- [30] E. Shivanian and H. R. Khodabandelo, "A meshless method based on radial basis and spline interpolation for 2-D and 3-D inhomogeneous biharmonic BVPs," *Zeitschrift für Naturforschung A*, vol. 70, no. 8, pp. 673–682, 2015.
- [31] F. Fakhar-Izadi and M. Dehghan, "Space-time spectral method for a weakly singular parabolic partial integro-differential equation on irregular domains," *Computers & Mathematics with Applications*, vol. 67, no. 10, pp. 1884–1904, 2014.
- [32] H. Fatahi, J. Saberi-Nadjafi, and E. Shivanian, "A new spectral meshless radial point interpolation (SMRPI) method for the two-dimensional fredholm integral equations on general domains with error analysis," *Journal of Computational and Applied Mathematics*, vol. 294, pp. 196–209, 2016.
- [33] 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.
- [34] A. G. Bratsos, "An improved numerical scheme for the sine-Gordon equation in 2 + 1 dimensions," *International Journal for Numerical Methods in Engineering*, vol. 75, no. 7, pp. 787–799, 2008.
- [35] W. K. Liu, S. Jun, and Y. F. Zhang, "Reproducing kernel particle methods," *International Journal for Numerical Methods in Fluids*, vol. 20, no. 8–9, pp. 1081–1106, 1995.
- [36] Y. X. Mukherjee and S. Mukherjee, "Boundary node method for potential problems," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 5, pp. 797–815, 1997.
- [37] J. Melenk and I. Babuska, "The partition of unity finite element method: basic theory and applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, no. 1–4, pp. 289–314, 1996.
- [38] S. De and K. J. Bathe, "The method of finite spheres," *Computational Mechanics*, vol. 25, no. 4, pp. 329–345, 2000.
- [39] Y. T. Gu and G. R. Liu, "A boundary point interpolation method for stress analysis of solids," *Computational Mechanics*, vol. 28, no. 1, pp. 47–54, 2002.
- [40] Y. Gu and G. Liu, "A boundary radial point interpolation method (BRPIM) for 2-d structural analyses," *Structural Engineering & Mechanics*, vol. 15, no. 5, pp. 535–550, 2003.
- [41] E. Shivanian, "Meshless local Petrov–Galerkin (MLPG) method for three-dimensional nonlinear wave equations via

- moving least squares approximation,” *Engineering Analysis with Boundary Elements*, vol. 50, pp. 249–257, 2015.
- [42] E. Shivanian, “Analysis of meshless local and spectral meshless radial point interpolation (MLRPI and SMRPI) on 3-D nonlinear wave equations,” *Ocean Engineering*, vol. 89, pp. 173–188, 2014.
- [43] E. Shivanian, “A new spectral meshless radial point interpolation (SMRPI) method: a well-behaved alternative to the meshless weak forms,” *Engineering Analysis with Boundary Elements*, vol. 54, pp. 1–12, 2015.
- [44] E. Shivanian, “On the convergence analysis, stability, and implementation of meshless local radial point interpolation on a class of three-dimensional wave equations,” *International Journal for Numerical Methods in Engineering*, vol. 105, no. 2, pp. 83–110, 2015.
- [45] S. Abbasbandy, “The effects of MHD flow of third grade fluid by means of meshless local radial point interpolation (MLRPI),” *Int J Ind Math*, vol. 7, no. 1, pp. 1–11, 2015.
- [46] E. Shivanian, A. Rahimi, and M. Hosseini, “Meshless local radial point interpolation to three-dimensional wave equation with neumann’s boundary conditions,” *International Journal of Computer Mathematics*, vol. 93, no. 12, pp. 2124–2140, 2015.
- [47] E. Shivanian and H. R. Khodabandehlo, “Application of meshless local radial point interpolation (MLRPI) on a one-dimensional inverse heat conduction problem,” *Ain Shams Engineering Journal*, vol. 7, no. 3, pp. 993–1000, 2016.
- [48] Z. Lei, G. Tianqi, Z. Ji, J. Shijun, S. Qingzhou, and H. Ming, “An adaptive moving total least squares method for curve fitting,” *Measurement*, vol. 49, pp. 107–112, 2014.
- [49] M. Carey and M. Bowers, “A review of properties of flow-density functions,” *Transport Reviews*, vol. 32, no. 1, pp. 49–73, 2012.
- [50] G. R. Liu and Y. T. Gu, *An Introduction to Meshfree Methods and Their Programming*, Springer-Verlag, Berlin/Heidelberg, Germany, 2005.
- [51] X. Li and S. Li, “On the stability of the moving least squares approximation and the element-free Galerkin method,” *Computers & Mathematics with Applications*, vol. 72, no. 6, pp. 1515–1531, 2016.
- [52] R. J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK, 2004.