Int. J. Nonlinear Anal. Appl. 14 (2023) 10, 257–282
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.29292.4117

Numerical solutions the nonlinear Burgers-Fisher and Burgers' equations with adaptive numerical method

Neda Najafzadeh

Department of Applied Mathematics, Payame Noor University (PNU), P.O.Box 19395-4697, Tehran, Iran

(Communicated by Saeid Abbasbandy)

Abstract

In this paper, a numerical method for finding the numerical solution of the Burgers-Fisher and Burgers' nonlinear equations is proposed. These equations are very important in many physical problems such as fluid dynamics, turbulence, sound waves and etc. We describe a meshless method to solve the nonlinear Burgers' equation as a stiff equation. In the proposed method, we also use the exponential time differencing (ETD) method. In this method, the moving least squares (MLS) method is used for the spatial part and the exponential time differencing (ETD) is used for the time part. 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. Method MLS provided good results for these equations due to its high flexibility and high accuracy and having a moving window, and obtains the solution at the shock point without any false oscillations. The method is described in detail, and a number of computational examples are presented. The accuracy of the proposed method is demonstrated by several test simulation.

Keywords: Adaptive Numerical Analysis; Burgers' nonlinear equation; Moving Least Squares(MLS); Exponential Time Differencing(ETD); Meshless Method; Burgers-Fisher equation
2020 MSC: 65M50, 35Q30

1 Introduction

In this article, we deal with the method of numerical solution of the Burgers Fisher and Burgers' equations. For the first time in 1915, Bateman [4] proposed a steady state solution to this equation, and then in 1948, the Burgers-Fisher and Burger [6, 7]used it to show, some properties of turbulent fluid in a channel due to the interaction of contrasting effects. Convection and diffusion provided this equation, because of wich, these equations were called the "Burgers' equation" which has three sentences: 1-convective, 2-viscosity and 3-time dependent. This equation is a highly nonlinear equation whose behaviors are almost similar to the one-dimensional Navier-Stokes equation but differ in one part, the pressure gradient.These equations usually have a hybrid state, meaning that whenever the viscosity expression is predominant, it is a parabolic, and otherwise a hyperbola, in which case, severe discontinuities may occur over a limited time due to the nonlinearity of the Burgers' equation. Such discontinuities cause many problems in the process of solving these equations. These features of the Burgers' equation make it a great model for testing numerical methods.

Email address: n.najafzadeh@pnu.ac.ir, mn_najafzadeh@yahoo.com (Neda Najafzadeh)

Received: December 2022 *Accepted:* January 2023

This equation has many applications in various sciences and problems, including one-dimensional turbulence, sound waves in a viscous environment, shock waves in a viscous environment, waves in liquid viscous elastic tubes, and hydrodynamic magnetic waves in an environment with limited electrical conductivity. One of the reasons why these equations are so popular in the scientific community is that they are widely used.

Solving the Burgers' equation has been considered for many years, and numerical solving of this equation is still a very active and popular issue. Many papers have been written for the numerical solution of these equations, and various methods have been proposed to solve the Burgers' equation, including automatic differentiation method [2], Galerkin finite element method [15], spectral colocation method [31], Sinc differential quadratic method [32], polynomial-based differential quadratic method [33], B-Splines quartz differential quadratic method [34], A new numerical scheme [66], an implicit fourth-order compact finite difference scheme[36], some implicit methods [62], adomin-pade technique [14], a hematopoietic analysis method [49], a differential transform method and a hematopoietic analysis method [50], semi-implicit finite differences [51], modified cubic B-splines collocation method [44], Localized Differential Quadrature method [5], Mixed Finite Volume Element Methods[67], The Lie-group method based on radial basis functions[24] and etc.

In this paper, we also numerically solve the Burgers-Fisher equation. The Burgers-Fisher equation has many applications in fluid dynamics models and is very useful for understanding the concept of physical flows. It has also been widely used in fields such as gas dynamics, number theory, heat conduction, traction, and so on.

The Burgers-Fisher equation is a very nonlinear equation,which consists of the mechanisms of reaction, convection, and diffusion. The particular Burgers Fisher equation is of special interest.

Many numerical solutions have been proposed to solve the Burgers-Fisher equations, including Adomin decomposition method [64], The tanh method [65], Spectral collocation method and spectral domain decomposition method [21], A numerical simulation and Explicit solutions [30], A restrictive Pade approximation [26], Analytic approximate solutions [8], Numerical Treatment of Burgers-Fisher equation[13] and etc., which will be explained in more detail in section 2.

This article proposes a adaptive numerical method based on the meshless method and ETD scheme. In recent years, meshless methods 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.[53, 12]

In principle, this method was in the category of first order explicit methods. Regarding the stability of these methods and more details, refer to the article [9].
To improve ETD methods, we consider new and more accurate ETD methods, combining ETD scheme with Runge-Kutta 4 [12].

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 will have many errors and fail for PDE equations whose matrix is not diagonal. In this work, we use modified ETD schemes to avoid these problems[35].

Low-order ETD methods are more suitable for application in computational electrodynamics [63]. They are often taken independently [10, 12, 43]—In his article, Iserslie states that in 1928, Filon proposed ideas related to this method in the field of ODE.[20, 27]—The best solution to the problems of this method is to use the fourth order Runge–Kutta formula (ETDRK4) with exponential time difference[12]. Cox and Matthews argue 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, meshless methods have been proposed to solve equations better[41]. 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 diffuse 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[60]. 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)[16],
2. Mesh-free methods based on strong forms, such as collocation method based on Radial Basis Functions(RBFs)[3, 59],

3. Mesh-free methods based on a combination of weak forms and collocation method.[19, 18]

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

1. Dispersed Elements Method(DEM)[47],
2. Smooth Particle Hydrodynamics(SPH)[11],
3. Reproducing Kernel Particle Method(RKPM)[42],
4. Boundary Node Method(BNM)[45],
5. Partition Unity Finite Element Method(PUFEM)[46],
6. Finite Sphere Method(FSM)[17],
7. Boundary Point Interpolation Method(BPIM)[22],
8. Border Radius Point Interpolation Method(BRPIM)[23],
9. Meshless Local Petrov Galerkin(MLPG)[61],
10. Meshless Local Radius Point Interpolation(MLRPI)[54, 55, 56, 1, 57, 58].

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[40]. In this paper, the moving least squares (MLS) approximation is used, which will be discussed in detail in section 4. In section 5, we will explain about our adaptive method. In Section 6, a number of examples are solved with these methods.

2 the Burgers-Fisher and Burgers' equations

This section provides additional explanations of the first-order Burgers' equation and the Fisher Burgers equation. Burgers' equation is a nonlinear partial differential equation widely used and known in applied mathematics and physical sciences such as gas dynamics, nonlinear acoustics and fluid mechanics. This equation was first studied by Bateman[4]in 1915 and later by Burgers[6, 7] in 1948. Since then, it has been widely considered as the Burgers' equation by the scientific community and attempts have been made to solve this equation in various ways. This equation presents well the turbulence caused by two opposite phenomena, diffusion and convection, and it also has a convection term, a viscosity term, and a time-dependent term. When the expression phrase or sticky sentence is predominant, the answer form is parabolic, or otherwise hyperbolic. If the propagation sentence is not predominant in contrast to the convection sentence, very strong shock waves or discontinuities may occur due to the nonlinearity of the Burgers' equation, even if the initial conditions are sufficiently smooth. Such discontinuities are one of the biggest problems in obtaining the answer to these equations, so these features make the Burgers' equation very suitable for testing the numerical methods used to obtain the answer to these problems. Therefore, the Burgers' equation plays a very important role among the partial differential equations with analytical solutions along with a set of basic functions and boundary constraints. Using nonlinear conversion, Hopf3 and Cole4 showed that it is possible to convert the Burgers' equation to a linear diffusion equation. Since its inception, this equation has been widely considered by researchers due to its various practical applications, such as gas dynamics, shock theory, traffic flow, viscous flow and turbulence, traffic. In this paper, we consider the one-dimensional equation of Burgers as follows:

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \qquad a \leq x \leq b \quad t \geq 0 \tag{2.1}$$

with the initial conditions: $u(x, 0) = h(x)$, $a \leq x \leq b$ and with the boundary conditions: $u(a, t) = f(x)$, $u(b, t) = g(x)$, $t \geq 0$, where α is a constant and ν is a kinematic viscosity.

Benton and Platzman investigated exact solutions to equations similar to the next Bergrick. In recent years, many attempts have been made to obtain numerical solutions with less error in the Burgers' equation for small and large quantities of kinematic viscosity, especially for small viscosities, because in this case the infinite series converges to the exact answer slowly, and the numerical solutions have many errors. Many papers have been written for the numerical solution of these equations and various methods have been proposed to solve the Burgers' equation, including automatic differentiation method [2], Galerkin finite element method [15], spectral colocation method [31], Sinc differential quadratic method [32], polynomial-based differential quadratic method [33], B-Splines quartz differential quadratic method [34], The Lie-group method based on radial basis functions[24], etc. In this paper, several examples of this type of equation with different initial and boundary conditions are solved, and the solution are compared with different viscosity coefficients.

In this work, we also deal with the Burgers-Fisher equation. This equation is very important for explaining different problems in science. Fischer first proposed the famous equation encountered in various disciplines as a model for propagating a mutated gene in which $u(x,t)$ represents the density of mean. In later years, this equation was used as the basis for a variety of models for many problems in various sciences. The most general form of the Fisher equation is called the Burgers-Fisher equation. Numerous interactions in physical problems that exist in models of different systems and structures lead to the generalized Burgers-Fisher equation. The Burgers-Fisher equation is the one whose nonlinearity is very high because it is a combination of reaction, convection, and diffusion mechanisms. This equation is called Burgers-Fisher because of the properties of the convective phenomenon, the diffusion transfer, and the type of reactions. Its high nonlinearity has attracted much attention from researchers, many of whom have made great efforts to solve the Burgers Fisher equation and obtain a numerical answer to this equation. Adomian decomposition method in [64] ,the tanh method by Wazwaz [65], spectral collocation method and spectral domain decomposition method by Javidi and Golbabai [21], a non-standard finite difference scheme by Mickens. Kaya and Sayed introduced a numerical simulation and Explicit Solutions [30], Ismail and Rabboh presented a restrictive Pade approximation[26], Numerical Treatment of Burgers-Fisher equation[13], Analytic approximate solutions [8] and etc.

The following is a generalized one dimensional Burgers Fisher equation that has been developed in various fields of science

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta}) \qquad a \leq x \leq b \quad t \geq 0 \tag{2.2}$$

Where α, β and δ are parameters. In this work, the Burgers-Fisher generalized equation is solved by the proposed method. Numerical results are compared with exact solutions to verify the current method.

3 The MLS approximation scheme

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

In this paper, among the meshless methods, we used the moving least square (MLS) because it seems to work better for shock problems. We consider a subdomain Λ_s, In the neighborhood the point $\mathbf{x}$, which exists within the principal domain of the problem Λ, and 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 Λ_s, on random points of nodes $\mathbf{x}_i, i = 1, .., N$. We illustrate the approximation of the v function by the method of moving least squares with $v^h(\mathbf{x})$ upon the domain Λ_s, and we can define $\forall \mathbf{x} \in \Lambda_s$:

$$v^h(\mathbf{x}) = \mathbf{D}^T(\mathbf{x})\mathbf{q}(\mathbf{x}) \qquad \forall \mathbf{x} \in \Lambda_s \tag{3.1}$$

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

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

Here, if we want to shift the origin to a fixed point $\mathbf{x}^e = (X_1^e, X_2^e, ..., X_n^e)^T$, $\mathbf{x}$ in $\mathbf{D}(\mathbf{x})$ must be replaced by $\mathbf{x} - \mathbf{x}^e$. Point $\mathbf{x}^e = (X_1^e, X_2^e, ..., X_n^e)^T$ is on $\Re(x)$, where $\Re(x)$ represents the influence domin of $\mathbf{x}$. Then, a linear basis by

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

and a quadratic basis by

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

The weight function used in the MLS method is as follows

$$w_i(\mathbf{x}) = \varphi\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{h}\right), \qquad i = 1, ..., N \tag{3.2}$$

where the function φ is 1.nonnegative, 2.γth times continuously differentiable, 3. its derivatives are bounded for a higher order than γ, 4.compactly supported. Many weight functions used in meshless methods have these conditions, such as Gaussian, exponential, cubic, and quartic splines. In this paper, the Gaussian weight function is used, which is defined as follows:

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

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 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 number of nodes required in the definition domain of each sample point, and on the other hand, choosing a very small r_s will cause a large error in the calculations. Care must also be taken that the r_s should be small enough, that is, not too small to preserve the local properties of the MLS approximation.

As mentioned: $v^h(\mathbf{x}) = \mathbf{D}^T(\mathbf{x})\mathbf{q}(\mathbf{x})$, which has already been explained about $\mathbf{D}^T(\mathbf{x})$, and now we will examine the coefficients of $\mathbf{q}(\mathbf{x})$ in detail. To define a coefficient vector $\mathbf{q}(\mathbf{x})$, we use the $L2$-norm in such a way that these coefficients are determined by minimizing the weighted discrete $L2$-norm. As follows:

$$\mathbf{J}(x) = \sum_{i=1}^{n} w_i(\mathbf{x})[\mathbf{D}^T(\mathbf{x}_i)\mathbf{q}(\mathbf{x}) - \hat{v}_i]^2 = [\mathbf{D}.\mathbf{q}(\mathbf{x}) - \hat{\mathbf{v}}]^T.\mathbf{W}.[\mathbf{D}.\mathbf{q}(\mathbf{x}) - \hat{\mathbf{v}}] \tag{3.3}$$

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

$$\mathbf{D} = \begin{pmatrix} \mathbf{d}^T(\mathbf{x}_1) \\ \mathbf{d}^T(\mathbf{x}_2) \\ \cdots \\ \mathbf{d}^T(\mathbf{x}_n) \end{pmatrix}, \quad \mathbf{W} = diag(w_1(\mathbf{x}), w_2(\mathbf{x}), ..., w_n(\mathbf{x})), \quad \hat{\mathbf{v}}^T = [\hat{v}_1, \hat{v}_2, ..., \hat{v}_n]. \tag{3.4}$$

Note that $\hat{v}_i, i = 1, 2, ..., n$ 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 $\hat{\mathbf{v}}$, which is expressed below. This relationship is derived from the stationarity of J in Eq.(3.1) with respect to $\mathbf{q}(\mathbf{x})$.

$$\mathbf{Q}(\mathbf{x})\mathbf{q}(\mathbf{x}) = \mathbf{S}(\mathbf{x})\hat{\mathbf{v}} \tag{3.5}$$

To define the matrices $\mathbf{Q}(\mathbf{x})$ and $\mathbf{S}(\mathbf{x})$:

$$\mathbf{Q}(\mathbf{x}) = \mathbf{D}^T.\mathbf{W}.\mathbf{D} \tag{3.6}$$
$$\mathbf{S}(\mathbf{x}) = \mathbf{D}^T.\mathbf{W}$$

Obtaining $\mathbf{q}(\mathbf{x})$ from the equation(3.5) and substituting it in Eq.(3.1) yields:

$$v^h(\mathbf{x}) = \mu^T(\mathbf{x}).\hat{\mathbf{v}} = \sum_{i=1}^{n} \mu_i(\mathbf{x})\hat{\mathbf{v}}_i, \quad \mathbf{x} \in \Lambda_s \tag{3.7}$$

$$\mu^T(\mathbf{x}) = \mathbf{D}^T(\mathbf{x})\mathbf{Q}^{-1}(\mathbf{x})\mathbf{S}(\mathbf{x}) \tag{3.8}$$

or

$$\mu_i(\mathbf{x}) = \sum_{j=1}^{m} \mathbf{d}_j(\mathbf{x})[\mathbf{Q}^{-1}(\mathbf{x})\mathbf{S}(\mathbf{x})]_{ji} \tag{3.9}$$

$\mu_i(\mathbf{x})$ is called a shape function of the MLS approximation dependent on the $\mathbf{x}_i$ node. From Eqs.(3.9) and (3.7), 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 non-zero. In fact, if $\mu_i(\mathbf{x}) = 0$, 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.

4 A modified ETD scheme

In Burgers' equation, we have both linear and non-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. But because we have a non-linear 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:

$$v_t = \mathcal{L}v + \mathcal{N}(v,t). \tag{4.1}$$

In this equation, $\mathcal{L}$ and $\mathcal{N}$ represent the linear and nonlinear operator. First, the spatial part of the PDE equation is discretized by related methods, and thus we obtain a system of ODEs.

$$v_t = \mathbb{L}v + \mathbb{N}(v,t). \tag{4.2}$$

We begin by multiplying Eq.(4.2) through by $1/e^{\mathbb{L}t}$, the term $1/e^{\mathbb{L}t}$ is the integrating factor. Then we take the integral from the equation on a single time step of length h from $t = t_n \quad to \quad t = t_{n+1} = t_n + h$ to give:

$$v_{n+1} = e^{\mathbb{L}h}v_n + e^{\mathbb{L}h}\int_0^h e^{-\mathbb{L}\zeta}\mathbb{N}(v(t_n+\zeta), t_n+\zeta)\,d\zeta. \tag{4.3}$$

In this equation, different orders of ETD schemes can be obtained from how the integrals are approximated. A sequence of recursive formulas are presented that provide higher order approximations of a multistep type. Here is a production formula:

$$v_{n+1} = e^{\mathbb{L}h}v_n + h\sum_{m=0}^{s-1}\eta_m\sum_{k=0}^{m}(-1)^k\binom{m}{k}\mathbb{N}_{n-k} \tag{4.4}$$

wherein s reperesent the order of the method. The coefficients η_m can be obtained from the following recursive relation

$$\mathbb{L}h\eta_0 = e^{\mathbb{L}h} - \mathbf{I}$$
$$\mathbb{L}h\eta_{m+1} + \mathbf{I} = \eta_m + \frac{1}{2}\eta_{m-1} + \frac{1}{3}\eta_{m-2}... + \frac{1}{m+1}\eta_0 \tag{4.5}$$

whenever we use Rung Kuta methods in the set of ETD methods and derive its time step with Rung Kutta, the method is called ETDRK. In this paper, we use only the fourth order method and derive it with ETD methods. In this case, the method is called ETDRK4. This is not entirely clear and needs to be explained in the form of a symbolic manipulation system. Here are the formulas for the ETDRK4 method:

$$\begin{aligned}
a_n &= e^{\mathbb{L}h/2}v_n + \frac{(e^{\mathbb{L}h/2}-\mathbf{I})}{\mathbb{L}}\mathbb{N}(v_n,t_n),\\
b_n &= e^{\mathbb{L}h/2}v_n + \frac{(e^{\mathbb{L}h/2}-\mathbf{I})}{\mathbb{L}}\mathbb{N}(a_n,t_n+h/2),\\
c_n &= e^{\mathbb{L}h/2}a_n + \frac{(e^{\mathbb{L}h/2}-\mathbf{I})}{\mathbb{L}}(2\mathbb{N}(b_n,t_n+h/2)-\mathbb{N}(v_n,t_n)),\\
\varpi &= -4-\mathbb{L}h+e^{\mathbb{L}h}(4-3\mathbb{L}h+(\mathbb{L}h)^2\\
\varrho &= 2+\mathbb{L}h+e^{\mathbb{L}h}(-2+\mathbb{L}h)\\
\varsigma &= -4-3\mathbb{L}h-(\mathbb{L}h)^2+e^{\mathbb{L}h}(4-\mathbb{L}h)\\
v_{n+1} &= e^{\mathbb{L}h}v_n + \frac{\varpi\mathbb{N}(v_n,t_n)+2\varrho(\mathbb{N}(a_n,t_n+h/2)+\mathbb{N}(b_n,t_n+h/2))+\varsigma\mathbb{N}(c_n,t_n+h)}{h^2\mathbb{L}^3}
\end{aligned} \tag{4.6}$$

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

$$\alpha = \frac{\varpi}{h^2\mathbb{L}^3}, \quad \beta = \frac{\varrho}{h^2\mathbb{L}^3}, \quad \gamma = \frac{\varsigma}{h^2\mathbb{L}^3}. \tag{4.7}$$

As mentioned before, we tried to solve the Burgers' equations with the methods described, and examples of these equations are solved in the next section.

5 Adaptive method

In this section, an attempt has been made to solve the Burgers-Fisher and Burgers' equations using the methods described in Sections 3 and 4. The first part provides explanations about solving the Burgers' equation and then the second part provides explanations about solving the Burgers-Fisher equation with the mentioned method are given.

5.1 Adaptive method for Burgers' equations

Consider the one-dimensional Burgers' equation:

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \qquad a \le x \le b \quad t \ge 0 \tag{5.1}$$

with the initial conditions: $u(x,0) = h(x)$, $a \le x \le b$ and with the boundary conditions: $u(a,t) = f(x)$, $u(b,t) = g(x)$, $t \ge 0$ where α is a constant and ν is a kinematic viscosity.

In this section, we explain how to solve this equation according to MLS and ETDRK4. To solve the equation, we first use MLS to approximate spatial derivatives and then ETDRK4 to approximate time derivatives.

We divide the main domain of the problem (Λ) into N parts so that the distance between the nodes is constant, and for each point we consider a neighborhood (Λ_s)that is inside the main domain. To approximate the function u, by MLS method, in Λ_s, on random points of nodes $x_i, i = 1,..,n$. From relation 3.5,3.7:

$$u^h(x,t) = \sum_{i=1}^{n} \mu_i(x)\hat{u}_i(t) \tag{5.2}$$

By replacing in the equation we have:

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial(\sum_{i=1}^{n} \mu_i(x)\hat{u}_i(t))}{\partial x} = \nu \frac{\partial^2(\sum_{i=1}^{n} \mu_i(x)\hat{u}_i(t))}{\partial x^2}. \tag{5.3}$$

Note that $\hat{u}_i(t) = \hat{u}(x_i, t)$ in equation 5.3 represent 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 qth differentiable on Λ. If $w_i(x) \in C^q(\Lambda)$ and $d_j(x) \in C^s(\Lambda), i = 1,2,...,n, j = 1,2,...,m$, then $\mu_i(x) \in C^r(\Lambda)$ with $r = min(q,s)$. The k derivatives of $\mu_i(x)$ as:

$$\mu_{i,k} = \sum_{j=1}^{m} [d_{j,k}(Q^{-1}S)_{ji} + d_j(Q^{-1}S_{,k} + Q^{-1}_{,k}S)_{ji}] \tag{5.4}$$

where we show the derivative inverse of the matrix Q, which depends on x_k, with the symbols $Q^{-1}_{,k} = (Q^{-1})_{,k}$, obtained by the formula $Q^{-1}_{,k} = -Q^{-1}Q_{,k}Q^{-1}$. Now, considering the spatial discretization and the use of the MLS method, we have:

$$\frac{\partial u(x,t)}{\partial t} = -\alpha u(x,t)(\sum_{j=1}^{n} \mu'_j(x)\hat{u}_j(t)) + \nu(\sum_{j=1}^{n} \mu''_j(x)\hat{u}_j(t)). \tag{5.5}$$

For $x_i \in \Lambda$ (Λ is the global domain of problem):

$$\dot{u}_i(t) = -\alpha u_i(t)(\sum_{j=1}^{n} \mu'_j(x)\hat{u}_j(t)) + \nu(\sum_{j=1}^{n} \mu''_j(x)\hat{u}_j(t)). \tag{5.6}$$

Note that the derivative of u is relative to time, and the derivative of μ is defined in 5.4. When we disassemble the PDE space partition, we get a system of ODEs:

$$\begin{bmatrix}\dot{u}_1(t)\\ \dot{u}_2(t)\\ \vdots\\ \dot{u}_n(t)\end{bmatrix} = -\alpha \begin{bmatrix}u_1(t)\\ u_2(t)\\ \vdots\\ u_n(t)\end{bmatrix} \circ \begin{bmatrix}\mu_1'(x_1) & \mu_2'(x_1) & \cdots & \mu_n'(x_1)\\ \mu_1'(x_2) & \mu_2'(x_2) & \cdots & \mu_n'(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1'(x_n) & \mu_2'(x_n) & \cdots & \mu_n'(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix} + \nu \begin{bmatrix}\mu_1''(x_1) & \mu_2''(x_1) & \cdots & \mu_n''(x_1)\\ \mu_1''(x_2) & \mu_2''(x_2) & \cdots & \mu_n''(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1''(x_n) & \mu_2''(x_n) & \cdots & \mu_n''(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix}$$

where $\circ$ is Hadamard product.

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

$$\mathbb{L} = \nu \begin{bmatrix}\mu_1''(x_1) & \mu_2''(x_1) & \cdots & \mu_n''(x_1)\\ \mu_1''(x_2) & \mu_2''(x_2) & \cdots & \mu_n''(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1''(x_n) & \mu_2''(x_n) & \cdots & \mu_n''(x_n)\end{bmatrix}$$

and

$$\mathbb{N}(u,t) = -\alpha \begin{bmatrix}u_1(t)\\ u_2(t)\\ \vdots\\ u_n(t)\end{bmatrix} \circ \begin{bmatrix}\mu_1'(x_1) & \mu_2'(x_1) & \cdots & \mu_n'(x_1)\\ \mu_1'(x_2) & \mu_2'(x_2) & \cdots & \mu_n'(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1'(x_n) & \mu_2'(x_n) & \cdots & \mu_n'(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix}$$

For time discretization, ETD, as previously described, is used.

Assuming that we divide the main domain into n parts with equal distances, $u(xi, kt), i = 1, 2, ..., n$ is specified, and our goal is to calculate $u(xi, (k+1)t), i = 1, 2, ..., n$. Multiply the equation by $e^{-\mathbb{L}t}$, then integrate the equation into a single time step of length h, so:

$$u_i^{k+1} = e^{\mathbb{L}h} u_i^k + e^{\mathbb{L}h} \int_0^h e^{-\mathbb{L}\zeta} (\alpha u_i(t_k+\zeta) \sum_{j=1}^{n} \mu_j'(x_i) \hat{u}_j(t_k+\zeta))\, d\zeta \tag{5.7}$$

where $u_i^k = u(x_i, t_k)$, and then we solve it using the formulas of ETDRK4 experessed in 4.6.

5.2 Adaptive method for Burgers-fisher equations

Consider the one-dimensional Burgers-fisher equation:

$$\frac{\partial u}{\partial t} + \alpha u^\delta \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^\delta) \qquad a \le x \le b \quad t \ge 0 \tag{5.8}$$

where α, β, and δ are parameters. In this part, as in the previous part, we solve the equation according to methods MLS and ETDRK4. As explained in the previous section, we first use MLS to approximate spatial derivatives and then ETDRK4 to approximate time derivatives. We divide the main domain of the problem (Λ) into N parts so that the distance between the nodes is constant, and for each point, we consider a neighborhood (Λ_s)that is inside the main domain. To approximate the function u, by MLS, in Λ_s, on random points of nodes $x_i, i = 1, .., n$. From relation 5.2 and replacing in the burgers-fisher equation, we have:

$$\frac{\partial u}{\partial t} + \alpha u^\delta \frac{\partial (\sum_{i=1}^n \mu_i(x)\hat{u}_i(t))}{\partial x} = \frac{\partial^2 (\sum_{i=1}^n \mu_i(x)\hat{u}_i(t))}{\partial x^2} + \beta (\sum_{i=1}^{n} \mu_i(x)\hat{u}_i(t))(1-u^\delta) \tag{5.9}$$

Note that $\hat{u}_i(t) = \hat{u}(x_i, t)$ in Equation 5.9 represent node values that are fictitious. Now, considering the spatial discretization and the use of the MLS method, for $x_i \in \Lambda$ (Λ is the global domain of problem):

$$\dot{u}_i(t) = -\alpha u_i^\delta(t) (\sum_{j=1}^{n} \mu_j'(x)\hat{u}_j(t)) + \sum_{j=1}^{n} \mu_j''(x)\hat{u}_j(t) + \beta (\sum_{i=1}^{n} \mu_i(x)\hat{u}_i(t))(1-u_i^\delta(t)). \tag{5.10}$$

Note that the derivative of u is relative to time, and the derivative of μ is defined in 5.4. When we disassemble the PDE space partition, we get a system of ODEs:

$$\begin{bmatrix}\dot{u}_1(t)\\ \dot{u}_2(t)\\ \vdots\\ \dot{u}_n(t)\end{bmatrix} = -\alpha \begin{bmatrix}u_1^{\delta}(t)\\ u_2^{\delta}(t)\\ \vdots\\ u_n^{\delta}(t)\end{bmatrix} \circ \begin{bmatrix}\mu_1'(x_1) & \mu_2'(x_1) & \cdots & \mu_n'(x_1)\\ \mu_1'(x_2) & \mu_2'(x_2) & \cdots & \mu_n'(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1'(x_n) & \mu_2'(x_n) & \cdots & \mu_n'(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix} + \begin{bmatrix}\mu_1''(x_1) & \mu_2''(x_1) & \cdots & \mu_n''(x_1)\\ \mu_1''(x_2) & \mu_2''(x_2) & \cdots & \mu_n''(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1''(x_n) & \mu_2''(x_n) & \cdots & \mu_n''(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix}$$

$$+\beta \begin{bmatrix}\mu_1(x_1) & \mu_2(x_1) & \cdots & \mu_n(x_1)\\ \mu_1(x_2) & \mu_2(x_2) & \cdots & \mu_n(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1(x_n) & \mu_2(x_n) & \cdots & \mu_n(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix} - \beta \begin{bmatrix}u_1^{\delta+1}(t)\\ u_2^{\delta+1}(t)\\ \vdots\\ u_n^{\delta+1}(t)\end{bmatrix}$$

where $\circ$ is Hadamard product. In this step, to approximate time derivatives by ETDRK4, we have to divide the flux function into two parts, linear and non-linear:

$$\mathbb{L} = \begin{bmatrix}\mu_1''(x_1) & \mu_2''(x_1) & \cdots & \mu_n''(x_1)\\ \mu_1''(x_2) & \mu_2''(x_2) & \cdots & \mu_n''(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1''(x_n) & \mu_2''(x_n) & \cdots & \mu_n''(x_n)\end{bmatrix}$$

and

$$\mathbb{N}(u,t) = -\alpha \begin{bmatrix}u_1^{\delta}(t)\\ u_2^{\delta}(t)\\ \vdots\\ u_n^{\delta}(t)\end{bmatrix} \circ \begin{bmatrix}\mu_1'(x_1) & \mu_2'(x_1) & \cdots & \mu_n'(x_1)\\ \mu_1'(x_2) & \mu_2'(x_2) & \cdots & \mu_n'(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1'(x_n) & \mu_2'(x_n) & \cdots & \mu_n'(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix}$$

$$+\beta \begin{bmatrix}\mu_1(x_1) & \mu_2(x_1) & \cdots & \mu_n(x_1)\\ \mu_1(x_2) & \mu_2(x_2) & \cdots & \mu_n(x_2)\\ \vdots & \vdots & \vdots & \vdots\\ \mu_1(x_n) & \mu_2(x_n) & \cdots & \mu_n(x_n)\end{bmatrix} \begin{bmatrix}\hat{u}_1(t)\\ \hat{u}_2(t)\\ \vdots\\ \hat{u}_n(t)\end{bmatrix} - \beta \begin{bmatrix}u_1^{\delta+1}(t)\\ u_2^{\delta+1}(t)\\ \vdots\\ u_n^{\delta+1}(t)\end{bmatrix}$$

For time discretization, ETD, as previously described, is used. Assuming that we divide the main domain into n parts with equal distances, it is assumed that $u(xi, kt), i = 1, 2, ..., n$ is specified, and our goal is to calculate $u(xi, (k+1)t), i = 1, 2, ..., n$. Multiply the equation by $e^{-\mathbb{L}t}$, and then integrate the equation into a single time step of length h, so :

$$u_i^{k+1} = e^{\mathbb{L}h}u_i^k + e^{\mathbb{L}h}\int_0^h e^{-\mathbb{L}\zeta}\left(\alpha u_i^{\delta}(t_k+\zeta)\sum_{j=1}^{n}\mu_j'(x_i)\hat{u}_j(t_k+\zeta) + \beta\sum_{j=1}^{n}\mu_j(x_i)\hat{u}_j(t_k+\zeta) - \beta u_i^{\delta+1}(t_k+\zeta)\right)d\zeta \quad (5.11)$$

where $u_i^k = u(x_i, t_k)$, and then we solve it using the formulas of ETDRK4 experessed in 4.6. The problem that we have wich with the ETDRK4 method for these equations (Burgers and Burgers-fisher) is that it becomes unstable for some points. One way to solve this problem is to use a Taylor expansion with a cut-off point for small eigenvalues so if the matrix is linear diagonal operator the Taylor expansion is used for the diagonal points the below cut-off. But one of the serious problems of this method is that it is not possible to generalize this method to non-diagonal problems.

Another method that can be used to solve this problem is the contour integral of the idea of complex analysis. In fact, we have a function to evaluate that has a singularity point and is analytical elsewhere. Near this singular point, the function have numerically error. The suggested solution is to use a contour integral on a complex plane that includes z and is separated from $z = 0$, by

$$f(z) = (2\pi i)^{(-1)}\int_{\Gamma} f(s)(s-z)^{(-1)}ds. \quad (5.12)$$

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

$$f(\mathbb{L}) = (2\pi i)^{(-1)}\int_{\Gamma} f(s)(sI-\mathbb{L})^{-1}ds. \quad (5.13)$$

where the contour must contain the eigenvalues of $\mathbb{L}$. 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 Γ. When $\mathbb{L}$ is real, we use only the upper half of the circle for the contour area whose center is the real axis, and then we get the real part of the result. The contour integral can be approximated by the mean $f(s)$ on Γ, so that we obtain the mean at points of equal distance along Γ, or again consider only the upper half of Γ, and then consider only that real part.

To show the result in Table 1, we have calculated the coefficients inside the bracket by the Taylor expansion of order 5 and the contour integral then their error compared to the real answer is presented in Table .1

Table 1: *Calculate $\alpha(z), \beta(z), \gamma(z)$ of the contour integral with the average of more than 32 points on a semicircle in the upper half of the plane and by Taylor of order 5. As is clear, the contour integral provides perfect accuracy for all of these z values.*

coefficient	z	error for contour integral	error for 5-term Taylor
	1	0.000000000000021	0.001047783902218
	$1/e$	0.000000000000013	0.000006367522118
	$1/e^2$	0.000000000000233	0.000000041380440
α	$1/e^3$	0.000000000000486	0.000000002747111
	$1/e^4$	0.000000000020571	0.000000000224171
	1	0.000000000000003	0.000170552493335
	$1/e$	0.000000000000000	0.000001052574737
	$1/e^2$	0.000000000000084	0.000000068764081
β	$1/e^3$	0.000000000001371	0.000000000471871
	$1/e^4$	0.000000000047107	0.000000000467991
	1	0.000000000000005	0.000114832083180
	$1/e$	0.000000000000004	0.000000704169700
	$1/e^2$	0.000000000000054	0.000000045899611
γ	$1/e^3$	0.000000000002312	0.000000000328701
	$1/e^4$	0.000000000057995	0.000000000582001

As shown in the table, calculating the coefficients inside the bracket (i.e. α, β, γ in 4.7) with the contour integral method is much less error. In the next section, examples are solved with this new method that acceptable answers are obtained.

6 stability of method

As explained, a new method was used in this article to solve the Burgers-Fisher and Burgers equations, which is a combination of methods 1 and 2. To check the stability of the method of this article, we have to check the stability of the two stated methods. 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 6.1. 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 determinate of $\mathbf{Q}(\mathbf{x})$:

$$det(\mathbf{Q}(\mathbf{x})) = C_d(\mathbf{x}, n, m)h^{2\sum_{i=1}^{m} i}, \quad \forall \mathbf{x} \in \Lambda. \tag{6.1}$$

In addition, there is a constant $h_0 > 0$, for $h \leq h_0$, such that a bounded and computable number $C_c(\mathbf{x}, n, m)$ which is independent of h that the norm2 condition number of $\mathbf{Q}(\mathbf{x})$

$$cond(\mathbf{Q}(\mathbf{x})) = C_c(\mathbf{x}, n, m)h^{-2m}, \quad \forall \mathbf{x} \in \Lambda. \tag{6.2}$$

Proof . Let $x_e \in \Re(x)$ be a fixed point. We assume bounded constant vectors r_i such that $\forall x_i \in \Re(x) \quad x_i = x_e + r_i h$. So, there are bounded constants such that the basis functions $d_i(x)$ apply in:

$$d_i(x_j) = r_{ij}h^i \quad i = 1, 2, ..., m.$$

then:

$$[\mathbf{Q}(\mathbf{x})]_{ik} = \sum_{j=1}^{m} w_j(x)d_i(x_j)d_k(x_j) = h^{i+k}q_{ik}(x), \quad i,k = 1,2,...,m \tag{6.3}$$

where $q_{ik}(x) = \sum_{j=1}^{m} w_j(x)r_{ij}r_{kj}$. Due to the dagree of $[\mathbf{Q}(\mathbf{x})]_{ik}$, we can verify that $\mathbf{Q}(\mathbf{x})$ can be diagonalize:

$$\mathbf{QD}(\mathbf{x}) = diag[\mathbf{M}_0(\mathbf{x}), h^2\mathbf{M}_1(\mathbf{x}), h^4\mathbf{M}_2(\mathbf{x}), ..., h^{2m}\mathbf{M}_m(\mathbf{x})]. \tag{6.4}$$

where $\mathbf{M}_i(\mathbf{x}), \quad i = 0,1,...,m$ are diagonal matrices of order $\binom{i+n-1}{n-1}$ and are independent of h thus the lth element of $\mathbf{M}_i(\mathbf{x})$ on diagonal be d_{i_l} where $i_l = \sum_{k=1}^{i-1}\binom{k+n-1}{n-1} + l, \quad l = 1,2,...,\binom{i+n-1}{n-1}$. So the iith element of $\mathbf{QD}(\mathbf{x})$ is $h^{2i}d_i, i = 1,2,...,m$. Hence:

$$det(\mathbf{Q}(\mathbf{x})) = det(\mathbf{QD}(\mathbf{x})) = \prod_{i=1}^{m} h^{2i}d_i(\mathbf{x},) = C_d(\mathbf{x},n,m)h^{2\sum_{i=1}^{m} i}$$

where $C_d(\mathbf{x},n,m) = \prod_{i=1}^{m} d_i(\mathbf{x})$ is a bounded number. The Maximum value on the diagonal elements of matrix $\mathbf{M}_0(\mathbf{x})$ is called $d_{max}(\mathbf{x})$ and the minimum value on the diagonal of matrix $\mathbf{M}_m(\mathbf{x})$ is denominated $d_{min}(\mathbf{x})$. Now there is $h_0 > 0$ that for all $h \leq h_0$, the maximum and minimum eigenvalues of the matrix $\mathbf{QD}(\mathbf{x})$ are $d_{max}(\mathbf{x})$ and $d_{min}(\mathbf{x})h^{2m}$, respectively. On the other hand, according to (3.6), the matrix $\mathbf{Q}(\mathbf{x})$ is symmetric and positive definite. Therefore, the 2-norm condition number $\mathbf{Q}(\mathbf{x})$ can be written as follows:

$$cond(\mathbf{Q}(\mathbf{x})) = cond(\mathbf{QD}(\mathbf{x})) = \frac{|\ d_{max}(\mathbf{x})\ |}{|\ d_{min}(\mathbf{x})h^{2m}\ |} = C_c(\mathbf{x},n,m)h^{-2m}$$

where $C_c(\mathbf{x},n,m) = \frac{|d_{max}(\mathbf{x})|}{|d_{min}(\mathbf{x})|}$ is a bounded number. . □

Eq.6.1 states that

$$det(\mathbf{Q}(\mathbf{x})) \to 0 \quad as \quad h \to 0.$$

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, which 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 Eq.(6.2), we find that the condition number $\mathbf{Q}(\mathbf{x})$ has an inverse relationship with factor h^{2m}. That is, as h^{2m} decreases, the condition number increases, and as a result, the rounding error increases, 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 condition number tends to infinity. Therefore, we conclude that the instability of the MLS method depends on h and m, so that by reducing the value of h too much and increasing the value of m too much, the instability of the MLS method increases. Thes, we conclude that the MLS method achieves good and low-error solutions to solve problems when matrix $\mathbf{Q}$ in (3.5) is non-singular, and this happens when the rank $\mathbf{D}$ is equal to m.

In order not to have the problem of instability and MLS method is a well-defined method, a necessary condition is that at least m weight functions are non-zero (ie $n > m$) for each point $\mathbf{x} \in \Lambda$ and the points in Λ_s do not follow from a regular and special pattern.

Assumption1:there are positive numbers c_1 and c_2 independent of h such that $m \leq c_1 \leq n \leq c_2$.

Assumption2:the rank $\mathbf{D}$ given by (3.4) is equal to m.

Assumption3:there is bounded numbers $C_{w_i}(\mathbf{x})$ independent of h such that $d^{\nu}w_i = C_{w_i}(\mathbf{x})h^{-|\nu|}$, where $0 \leq |\ \nu\ | \leq \gamma$.

Note that the weight function is ν-th times continuously differentiable, i.e. $w_i(\mathbf{x}) \in C^{\nu}(\Lambda)$

Lemma 6.1. There are bounded numbers $c_{ik}(\mathbf{x})$ independent of h such that

$$[\mathbf{Q}^{-1}(\mathbf{x})]_{ik} = c_{ik}(\mathbf{x})h^{-i-k}, \quad i,k = 1,2,...,m.$$

Lemma 6.2. For any $\mathbf{x} \in \Lambda$, there are $C_{\mu_i}(\lambda,\mathbf{x})$ independent of h such that

$$d^{\lambda}\mu_i(\mathbf{x}) = C_{\mu_i}(\lambda,\mathbf{x})h^{-|\lambda|}, \quad 0 \leq |\lambda| \leq \nu, \ \ i = 1,2,\ldots,n.$$

Besides, there are constants C_{μ_1} and C_{μ_2} independent of h such that

$$C_{\mu_1} h^{-|\lambda|} \leq \|d^\lambda \mu_i(\mathbf{x})\|_{L^\infty(\Lambda)} \leq C_{\mu_2} h^{-|\lambda|}, \quad 0 \leqslant |\lambda| \leqslant \gamma, \ \ i = 1, 2, \ldots, n.$$

Proof . by (34) and Assumption 3:

$$d^\lambda [\mathbf{Q}(\mathbf{x})]_{jk} = \sum_{i=1}^{n} d^\lambda w_i(\mathbf{x}) r_{ji} h^j r_{ki} h^k = h^{j+k-|\lambda|} \sum_{i=1}^{n} C_{w_i}(\mathbf{x}) r_{ji} r_{ki}.$$

Applying $d_i(x_j) = r_{ij} h^i \quad i = 1, 2, ..., m.$ and Eq.(3.6) yields

$$[\mathbf{S}(\mathbf{x})]_{ki} = w_i(\mathbf{x}) d_k(\mathbf{x}_i) = w_i(\mathbf{x}) r_{ki} h^k.$$

Then using agin Assumption 3,

$$d^\lambda [\mathbf{S}(\mathbf{x})]_{ki} = d^\lambda w_i(\mathbf{x}) r_{ki} h^k = h^{k-|\lambda|} C_{wi}(\mathbf{x}) r_{ki}.$$

Let

$$\mathbf{F}(\mathbf{x}) = \mathbf{Q}^{-1}(\mathbf{x}) \mathbf{S}(\mathbf{x}).$$

From Leibniz's formula and Lemma6.1:

$$\begin{aligned}
d^\lambda [\mathbf{F}(\mathbf{x})]_{ji} &= \sum_{k=1}^{m} \left[\mathbf{Q}^{-1}(\mathbf{x})\right]_{jk} \left\{ d^\lambda [\mathbf{S}(\mathbf{x})]_{ki} - \sum_{l=1}^{m} \sum_{\alpha \leq \lambda, \alpha \neq \mathbf{0}} \binom{\lambda}{\alpha} d^\alpha [\mathbf{Q}(\mathbf{x})]_{kl} d^{\lambda-\alpha} [\mathbf{F}(\mathbf{x})]_{li} \right\} \\
&= \sum_{k=1}^{m} h^{-j-k} c_{jk}(\mathbf{x}) \times \\
&\left\{ h^{k-|\lambda|} C_{w_i}(\mathbf{x}) r_{ki} - \sum_{l=1}^{m} \sum_{\alpha \leq \lambda, \alpha \neq \mathbf{0}} \binom{\lambda}{\alpha} h^{k+l-|\alpha|} \sum_{I=1}^{n} C_{w_I}(\mathbf{x}) r_{kI} r_{lI} d^{\lambda-\alpha} [\mathbf{F}(\mathbf{x})]_{li} \right\} \\
&= b_j^0(\mathbf{x}) h^{-j-|\lambda|} - \sum_{l=1}^{m} \sum_{\alpha \leq \lambda, \alpha \neq \mathbf{0}} \binom{\lambda}{\alpha} m_{jl}(\mathbf{x}) h^{-j+l-|\alpha|} d^{\lambda-\alpha} [\mathbf{F}(\mathbf{x})]_{li}
\end{aligned}$$

where

$$b_j^0(\mathbf{x}) = \sum_{k=1}^{m} c_{jk}(\mathbf{x}) r_{ki} C_{w_i}(\mathbf{x})$$

$$m_{jl}(\mathbf{x}) = \sum_{k=1}^{m} c_{jk}(\mathbf{x}) \sum_{I=1}^{n} C_{w_I}(\mathbf{x}) r_{kI} r_{lI} = \sum_{I=1}^{n} C_{w_I}(\mathbf{x}) r_{lI} \sum_{k=1}^{m} r_{kI} c_{jk}(\mathbf{x}).$$

So

$$d^\lambda [\mathbf{F}(\mathbf{x})]_{ji} = b_j^\lambda(\mathbf{x}) h^{-j-|\lambda|},$$

where

$$b_j^\lambda(\mathbf{x}) = b_j^0(\mathbf{x}) - \sum_{l=1}^{m} \sum_{\alpha \leq \lambda, \alpha \neq 0} \binom{\lambda}{\alpha} m_{jl}(\mathbf{x}) b_l^{\lambda-\alpha}(\mathbf{x}).$$

Inasmuch as $\mathbf{x}_e$ is an evaluation point fixed on the influence domin of $\mathbf{x}$, there is a constant vector $\mathbf{r}_e$ such that $\mathbf{x} - \mathbf{x}_e = \mathbf{r}_e h$. Then we have bounded constants r_j^α such that

$$d^\alpha d_j(\mathbf{x}) = r_j^\alpha h^{j-|\alpha|}, \quad j = 1, 2, \cdots, m,$$

for $j \geq |\alpha|$. If $j < |\alpha|$, it is obvious that $d^\alpha d_j(\mathbf{x}) = 0$. Hence, there exist $C_{\mu_i}(\lambda, \mathbf{x})$ independent of h such that

$$\begin{aligned}
d^\lambda \mu_i(\mathbf{x}) &= \sum_{j=1}^{m} \sum_{\alpha \leq \lambda, \alpha \neq \mathbf{0}} \binom{\lambda}{\alpha} d^\alpha d_j(\mathbf{x}) d^{\lambda-\alpha} [\mathbf{F}(\mathbf{x})]_{ji} \\
&= \sum_{j=1}^{m} \sum_{\alpha \leq \lambda, \alpha \neq \mathbf{0}} \binom{\lambda}{\alpha} r_j^\alpha h^{j-|\alpha|} b_j^{\lambda-\alpha}(\mathbf{x}) h^{-j-|\lambda|+|\alpha|} \\
&= \left[\sum_{\alpha \leq \lambda, \alpha \neq 0} \binom{\lambda}{\alpha} \sum_{j=1}^{m} r_j^\alpha b_j^{\lambda-\alpha}(\mathbf{x}) \right] h^{-|\lambda|} \\
&:= C_{\mu_i}(\lambda, \mathbf{x}) h^{-|\lambda|}
\end{aligned}$$

Now, we prove the second part of the lemma. Inasmuch as $a_{jk}(\mathbf{x}), C_{w_I}(\mathbf{x})$ and r_{ki} are bounded, we have $b_j^0(\mathbf{x})$ and $m_{jl}(\mathbf{x})$ are bounded. As a result, $b_j^\lambda(\mathbf{x})$ are bounded. It can be concluded from Assumption 1 that: r_j^α is bounded and independent of h. Therefore, there are real numbers $C_{\mu 1}$ and $C_{\mu 2}$ independent of h such that

$$C_{\mu 1} h^{-|\lambda|} \leq \left|d^\lambda \mu_i(\mathbf{x})\right| \leq C_{\mu 2} h^{-|\lambda|}, \quad \forall \mathbf{x} \in \Lambda.$$

□

For $m \geq 0$ and $n \in [1, \infty]$, assume $W^{m,n}(\Lambda)$ and $H^m(\Lambda) \triangleq W^{m,2}(\Lambda)$ be the usual Sobolev space .

Theorem 6.2. Let $v^h(\mathbf{x})$ be the MLS approximation of $v(\mathbf{x})$ and $v(\mathbf{x}) \in W^{p+1,q}(\Lambda)$ with $p+1 > n/q$. then there is a constant C independent of h such that

$$\|v(\mathbf{x}) - v^h(\mathbf{x})\|_{W^{k,q}(\Lambda)} \leq C h^{\tilde{p}-k} \|v(\mathbf{x})\|_{W^{\tilde{p},q}(\Lambda)},$$
$$k = 0, 1, \cdots, \min\{\tilde{p}, \gamma\}, \quad \tilde{p} = \min\{p+1, m+1\}.$$

When $v(\mathbf{x}) \in W^{m+1,q}(\Lambda)$, we have

$$\|v(\mathbf{x}) - v^h(\mathbf{x})\|_{W^{k,q}(\Lambda)} \leq C h^{m+1-k} \|v(\mathbf{x})\|_{W^{m+1,q}(\Lambda)},$$
$$k = 0, 1, \cdots, \min\{m+1, \gamma\}.$$

When $v(\mathbf{x}) \in H^{m+1}(\Lambda)$, we have

$$\|v(\mathbf{x}) - v^h(\mathbf{x})\|_{H^k(\Lambda)} \leq C h^{m+1-k} \|v(\mathbf{x})\|_{H^{m+1}(\Lambda)},$$
$$0 \leq k \leq \min\{m+1, \gamma\}.$$

Proof . see [38] □

Now we investigate the stability of ETDRK4 scheme. By MLS, we made the system of ODE, the stability of this method was checked so we examine the stability of the method that used to solve this system, i.e. method ETDRK4. by linearizing the nonlinear autonomous system

$$\frac{dv(t)}{dt} = Lv(t) + N(v(t)) \tag{6.5}$$

with $N(v(t))$ the nonlinear part. We suppose that there exists a fixed point v_0 such that $Lv_0 + N(v_0) = 0$. By linearizing about this fixed point, we obtain

$$\frac{dv(t)}{dt} = Lv(t) + \eta v(t) \tag{6.6}$$

where $v(t)$ is now the perturbation of u_0 and $\eta = N'(v_0)$ is a diagonal matrix and the eigenvalues of N are placed in the diagonal of this matrix. For stablity, we require that $\operatorname{Re}(L+\eta) < 0$, for all η. the application of ETDRK4 method to the linearized problem (6.6) leads to a recurrence relation

$$r = \frac{v_{n+1}}{v_n} = L_0 + L_1 x + L_2 x^2 + L_3 x^3 + L_4 x^4, \tag{6.7}$$

where

$$\begin{aligned}
L_0 =& e^y \\
L_1 =& -\frac{4}{y^3} + \frac{8e^{y/2}}{y^3} - \frac{8e^{3y/2}}{y^3} + \frac{4e^{2y}}{y^3} - \frac{1}{y^2} + \frac{4e^{y/2}}{y^2} - \frac{6e^y}{y^2} + \frac{4e^{3y/2}}{y^2} - \frac{e^{2y}}{y^2} \\
L_2 =& -\frac{8}{y^4} + \frac{16e^{y/2}}{y^4} - \frac{16e^{3y/2}}{y^4} + \frac{8e^{2y}}{y^4} - \frac{5}{y^3} + \frac{12e^{y/2}}{y^3} - \frac{10e^y}{y^3} + \frac{4e^{3y/2}}{y^3} - \frac{e^{2y}}{y^3} - \frac{1}{y^2} + \frac{4e^{y/2}}{y^2} - \frac{e^{y/2}}{y^2},
\end{aligned}$$

$$L_3 = \frac{4}{y^5} - \frac{16e^{y/2}}{y^5} + \frac{16e^{y}}{y^5} + \frac{8e^{3y/2}}{y^5} - \frac{20e^{2y}}{y^5} + \frac{8e^{5y/2}}{y^5} + \frac{2}{y^4} - \frac{10e^{y/2}}{y^4} + \frac{16e^{y}}{y^4} - \frac{12e^{3y/2}}{y^4} + \frac{6e^{2y}}{y^4} - \frac{2e^{5y/2}}{y^4} - \frac{2e^{y/2}}{y^3} + \frac{4e^{y}}{y^3} - \frac{2e^{3y/2}}{y^3}$$

$$L_4 = \frac{8}{y^6} - \frac{24e^{y/2}}{y^6} + \frac{16e^{y}}{y^6} + \frac{16e^{3y/2}}{y^6} - \frac{24e^{2y}}{y^6} + \frac{8e^{5y/2}}{y^6} + \frac{6}{y^5} - \frac{18e^{y/2}}{y^5} + \frac{20e^{y}}{y^5} - \frac{12e^{3y/2}}{y^5} + \frac{6e^{2y}}{y^5} - \frac{2e^{5y/2}}{y^5} + \frac{4}{y^4} - \frac{6e^{y/2}}{y^4} + \frac{6e^{y}}{y^4} - \frac{2e^{3y/2}}{y^4},$$

where $x = \eta h, y = Lh$. The amplification factor is defined for ETDRK4, $r(x, y)$ for $y > 0$. If $y = 0$, the amplification factor becomes $1 - x + x^2/2 - x^3/6 + x^4/24$ that the stability of ETDRK4 at $y = 0$ correspond with that of the classical fourth-order Runge-Kutta method. On the other hand $\lim_{x,y\to 0} \partial_x r(x, y) = -1$ and $\lim_{x,y\to 0} \partial_y r(x, y) = -1$ thus the absolute value of the amplification factor is given as $|r(x, y)| \leq 1$. In the areas where this method is unstable, according to what was explained at the end of section 5 of this article, we use the contour integral of the idea of complex analysis.

7 Numerical example

We have described the method implemented in this article in detail in the previous sections. In this section, we will solve several examples and compare their numerical answer form with other methods.

7.1 Example1

In this example, we consider the Burgers' equation 2.1 with $\alpha = 1$:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \qquad 0 \leq x \leq 1 \quad t \geq 0$$

With the initial conditions:

$$u(x, 0) = sin(2\pi x) + 1/2sin(\pi x), \qquad 0 \leq x \leq 1$$

and with the boundary conditions:

$$u(0, t) = 0, \qquad u(1, t) = 0, \qquad t \geq 0$$

see [25]. In this subsection,we obtain the answer to this equation with the method described in this article. In Figures 1,2 and 3, the solution of this equation can be seen with $N = 40$ and with $\nu = 1$,$\nu = 0.01$,$\nu = 0.0001$. Figures a, b show the physical behavior of the answer in 3D and Figure c shows the physical behavior of the answer in the contour, which is similar to 2D.

7.2 Example2

In this example, we consider the Burgers' equation 2.1 with $\alpha = 1$:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \qquad 0 \leq x \leq 1 \quad t \geq 0$$

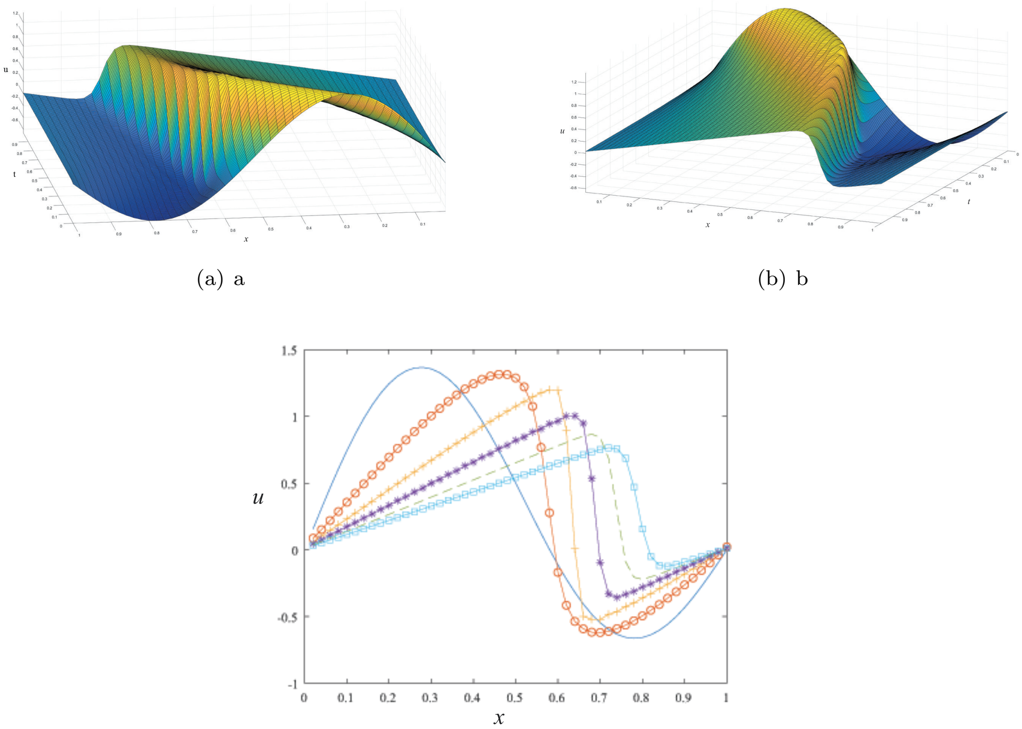
with the initial conditions: $u(x, 0) = sin(\pi x)$, $0 \leq x \leq 1$ and with the boundary conditions: $u(0, t) = 0$, $u(1, t) = 0$, $t \geq 0$. The exact solution of equation is

$$u(x, t) = \frac{2\pi\nu \sum_{i=1}^{\infty} nA_n exp(-n^2\pi^2\nu t)sin(n\pi x)}{A_0 + \sum_{i=1}^{\infty} A_n exp(-n^2\pi^2\nu t)cos(n\pi x)}$$

where

$$A_0 = \int_{i=0}^{1} exp\left(\frac{-1}{2\pi\nu}(1 - cos(\pi x))\right) dx, \quad A_n = 2\int_{i=0}^{1} exp\left(\frac{-1}{2\pi\nu}(1 - cos(\pi x))\right) cos(n\pi x)dx,$$

The numerical solutions of this equation are presented for $\nu = 0.1, 0.01$, $\nu = 0.001$ with $N = 60$ in Tables 2 and Figs. 4,5 and 6. We compare the results of solving this equation using the method proposed in this paper with the results obtained by other methods, and as showed in Table 2, better results were obtained. Figures 4, 5 and 6 show the physical behavior of the answer up to $t = 1$ (it is three-dimensional (a,b), and the contour is similar to two-dimensional shapes(c)).



(a) a (b) b

(c) c

Figure 1: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.1*** *by the new method expressed in this article with* $N = 40$ *and* $\boldsymbol{\nu = 1}$.
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*

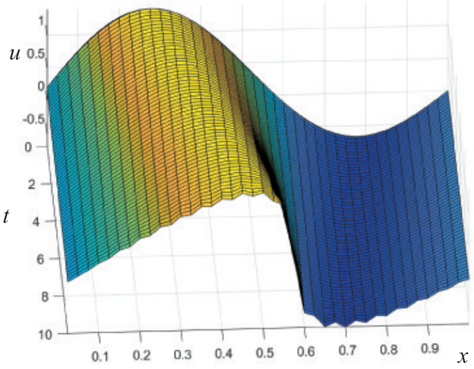
Table 2: *A comparison of the solution of equation of* ***Example 7.2*** *using the article method with exact answer and other numerical methods.* $\boldsymbol{\nu = 0.01}$

x	t	[29]$\Delta t = 0.01$	[2]$\Delta t = 0.0001$	[28]$\Delta t = 0.001$	[5] $\Delta t = 0.0001$	new method $\Delta t = 1/20$	exact
0.25	0.4	0.34229	0.34193	0.34184	0.34191	0.34191	0.34191
	0.6	0.26902	0.26898	0.26891	0.26896	0.26897	0.26896
	0.8	....	0.22149	0.22143	0.22148	0.22148	0.22148
	1	0.18817	0.18820	0.18815	0.18819	0.18819	0.18819
0.5	0.4	0.66797	0.66076	0.66060	0.66071	0.66071	0.66071
	0.6	0.53211	0.52945	0.52931	0.52941	0.52942	0.52942
	0.8	....	0.43916	0.43905	0.43913	0.43914	0.43914
	1	0.37500	0.37443	0.37436	0.37442	0.37442	0.37442
0.75	0.4	0.93680	0.91046	0.91026	0.91026	0.91025	0.91026
	0.6	0.77724	0.76733	0.76719	0.76724	0.76723	0.76724
	0.8	....	0.64744	0.64745	0.64739	0.64744	0.64740
	1	0.55833	0.55608	0.55608	0.55605	0.55605	0.55605

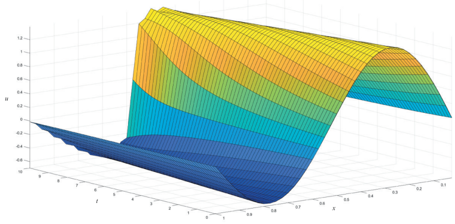
7.3 Example3

In this example, we consider the Burgers' equation 2.1 with $\alpha = 1$:

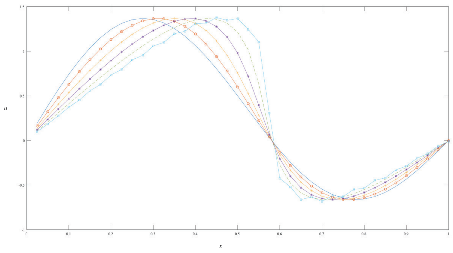
$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \qquad 0 \le x \le 1 \quad t \ge 0$$



(a) a



(b) b



(c) c

Figure 2: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.1*** *by the new method expressed in this article with* $N = 40$ *and* $\boldsymbol{\nu = 0.01}$ *to time* $t = 10$.
In Figure(c), the answer is shown at times $t = 0, 2, 4, 6, 8, 10$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*

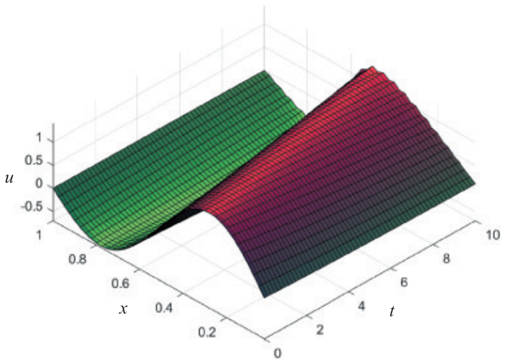
with the initial conditions: $u(x, 0) = 4x(1 - x)$, $0 \leq x \leq 1$ and with the boundary conditions: $u(0, t) = 0$, $u(1, t) = 0$, $t \geq 0$. The exact solution of equation is :

$$u(x,t) = \frac{2\pi\nu \sum_{i=1}^{\infty} nA_n exp(-n^2\pi^2\nu t) sin(n\pi x)}{A_0 + \sum_{i=1}^{\infty} A_n exp(-n^2\pi^2\nu t) cos(n\pi x)}$$

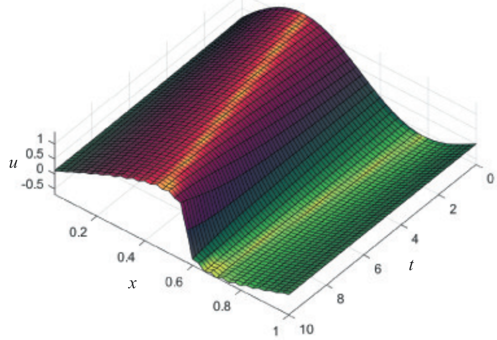
where

$$A_0 = \int_{i=0}^{1} exp\left(\frac{-1}{3\nu}(3x^2 - 2x^3)\right) dx, \qquad A_n = 2\int_{i=0}^{1} exp\left(\frac{-1}{2\pi\nu}(3x^2 - 2x^3)\right) cos(n\pi x) dx.$$

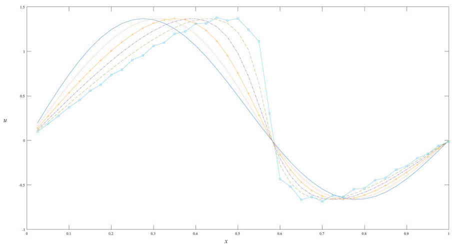
The numerical solutions of this equation are presented for $\nu = 0.1, 0.01, 0.001$ with $N = 60$ in Tables 3 and Figs. 7,8 and 9. We compare the results of solving this equation using the method proposed in this paper with the results obtained by another methods. As showed in Table 3 that better results were obtained. Figures 7,8,9 show the physical behavior of the answer up to $t = 1$ (it is three-dimensional (a,b), and the contour is similar to two-dimensional shapes(c)).



(a) a



(b) b



(c) c

Figure 3: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.1*** *by the new method expressed in this article with* $N = 40$ *and* $\boldsymbol{\nu = 0.0001}$ *to time* $t = 10$.
In Figure(c), the answer is shown at times $t = 0, 2, 4, 6, 8, 10$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*

Table 3: *A comparison of the solution of equation of* ***Example 7.3*** *using the article method with exact answer and other numerical methods.* $\boldsymbol{\nu = 0.01}$

x	t	[29]$\Delta t = 0.01$	[28]$\Delta t = 0.001$	[48]$N = 180$	new$\Delta t = 1/20, N = 60$	exact
0.25	0.4	0.36273	0.36217	0.36226	0.36226	0.36226
	0.6	0.28212	0.28197	0.28204	0.28205	0.28204
	0.8	....	0.23040	0.23045	0.23045	0.23045
	1	0.19467	0.19465	0.19469	0.19469	0.19469
0.5	0.4	0.69186	0.68357	0.68368	0.68368	0.68368
	0.6	0.55125	0.54822	0.54832	0.54832	0.54832
	0.8	....	0.45363	0.45370	0.45371	0.45371
	1	0.38627	0.38561	0.38568	0.38568	0.38568
0.75	0.4	0.94940	0.92050	0.92051	0.92050	0.92050
	0.6	0.79399	0.78293	0.78302	0.78299	0.78299
	0.8	....	0.66264	0.66272	0.66272	0.66272
	1	0.57170	0.56924	0.56932	0.56932	0.56932

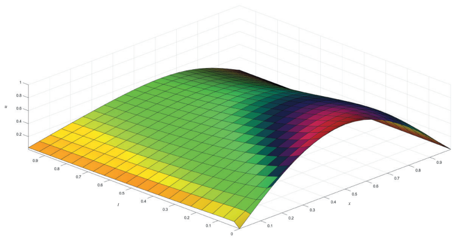
7.4 Example4

In this example, we consider the Burgers-fisher equation 2.2 with $\delta = 1$:

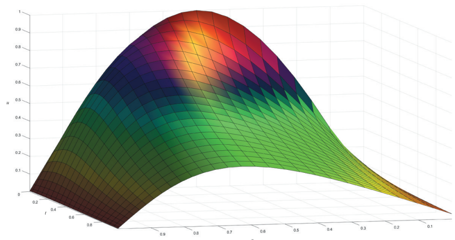
$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u) \qquad 0 \le x \le 1 \quad t \ge 0$$

where $\alpha = a, \beta = \frac{2ac-a^2}{4}$, with the initial conditions:

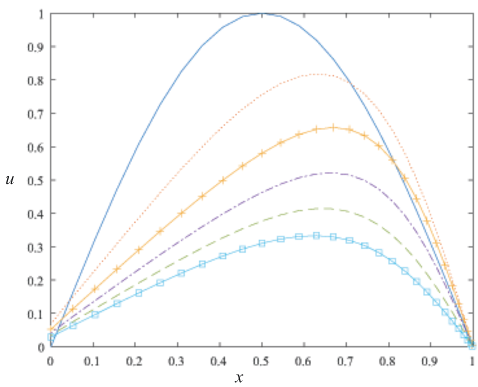
$$u(x,0) = \frac{1}{2} - \frac{1}{2} tanh(\frac{a}{4}x), \qquad 0 \le x \le 1$$



(a) a



(b) b



(c) c

Figure 4: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.2*** *by the new method expressed in this article with* $N = 60$ *and* $\boldsymbol{\nu = 0.1}$.
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*

and with the boundary conditions:

$$u(0,t) = \frac{1}{2} - \frac{1}{2} tanh(\frac{a}{4}(-ct)), \qquad u(1,t) = \frac{1}{2} - \frac{1}{2} tanh(\frac{a}{4}(1-ct)), \qquad t \geq 0$$

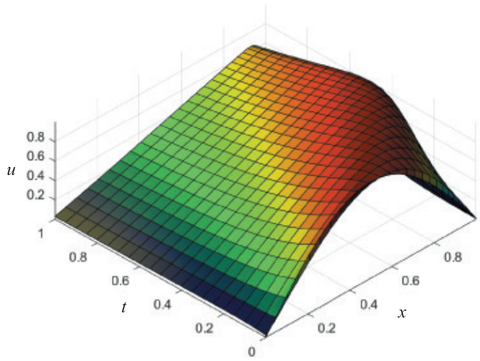
The exact solution of this equation:

$$u(x,t) = \frac{1}{2} - \frac{1}{2} tanh(\frac{a}{4}(x-ct)), \qquad 0 \leq x \leq 1$$

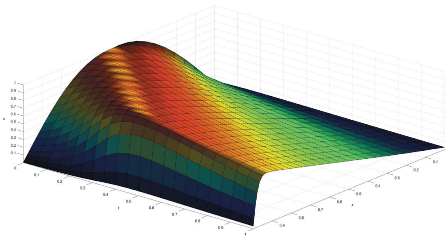
The numerical solutions of this equation are presented for $a = 24, c = 8$ with $N = 40$ in Tables 4 and Figs. 10. We compare the results of solving this equation using the method proposed in this paper with the results obtained by other methods, and as shown in Table 4 that better results were obtained. Figures 7,8,9 show the physical behavior of the answer up to $t = 0.5$ (it is three-dimensional (a,b)and the contour is similar to two-dimensional shapes(c)).

Table 4: *A comparison of L2-error for the solution of equation of* ***Example 7.4*** *using the article method with L2-error of another method*

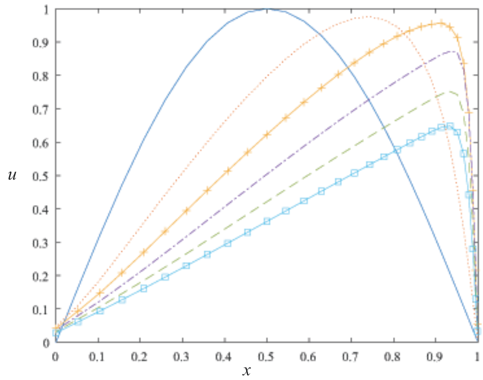
t	new method$N = 60$	[52]$N = 60$
0.01	16.7e-006	....
0.02	44.4e-006	....
0.03	121e-006	2.5e-003
0.04	324e-006	2.4e-003
0.05	853e-006	2.4e-003



(a) a



(b) b



(c) c

Figure 5: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.2*** *by the new method expressed in this article with* $N = 60$ *and* $\boldsymbol{\nu} = \mathbf{0.01}$ *to time* $t = 1$.
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*

7.5 Example5

In this example, we consider the Burgers-fisher equation 2.2 with $\delta = 1$:

$$\frac{\partial u}{\partial t} + \alpha u \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u) \qquad 0 \le x \le 1 \quad t \ge 0$$

With the initial conditions:

$$u(x,0) = \frac{1}{2} + \frac{1}{2} tanh(a_1 x), \qquad 0 \le x \le 1$$

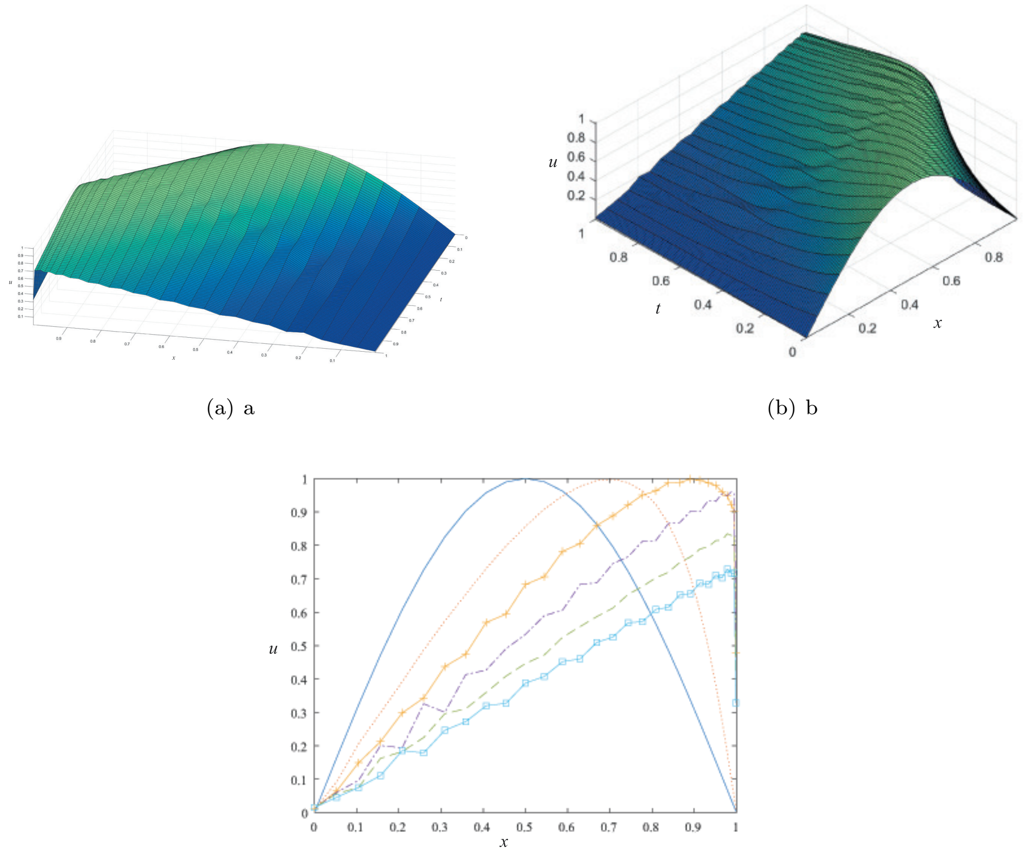
and with the boundary conditions:

$$u(0,t) = \frac{1}{2} - \frac{1}{2} tanh(a_1 a_2 t), \qquad u(1,t) = \frac{1}{2} + \frac{1}{2} tanh(a_1(1 - a_2 t)), \qquad t \ge 0$$

The exact solution of this equation:

$$u(x,t) = \frac{1}{2} - \frac{1}{2} tanh(a_1(x - a_2 t)), \qquad 0 \le x \le 1$$

where $a_1 = \frac{-\alpha\delta}{2(1+\delta)}$ and $a_1 = \frac{\alpha}{1+\delta} + \frac{\beta(1+\delta)}{\alpha}$. The numerical solutions of this equation are presented for $\alpha = 1, \beta = 1$ with $N = 40$ in Tables 5,6 and Figs. 11. We compare the results of solving this equation using the method proposed in this paper with exact solution. Figure11 shows the physical behavior of the answer up to $t = 0.5$



(a) a

(b) b

(c) c

Figure 6: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.2*** *by the new method expressed in this article with* $N = 60$ *and* $\boldsymbol{\nu = 0.001}$ *to time* $t = 1$.
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*

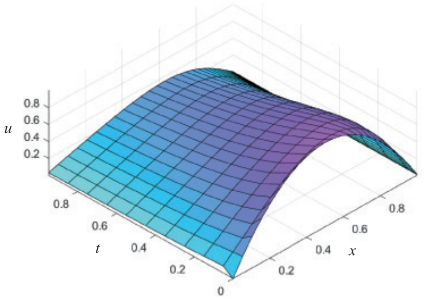
Table 5: *L2-error for the solution of equation of* ***Example 7.5*** *by the article method*

t	new method $N = 60$
0.02	3.3e-003
0.04	6.7e-003
0.06	10.1e-003
0.08	13.5e-003
0.1	661.4e-003

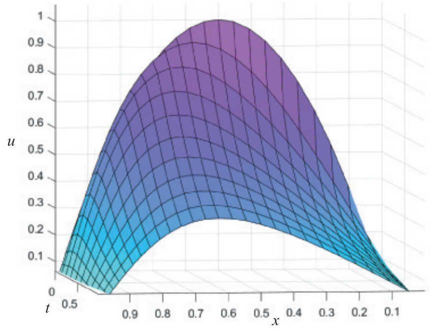
8 Conclusion

The current paper presented an efficient numerical method for solving the nonlinear Burgers-fishe and Burgers' equations. The paper also presented a new method for identifying impact points and finding a better response without vibration in these points. Accordingly, we first solved these equations from the meshless method and then using 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.

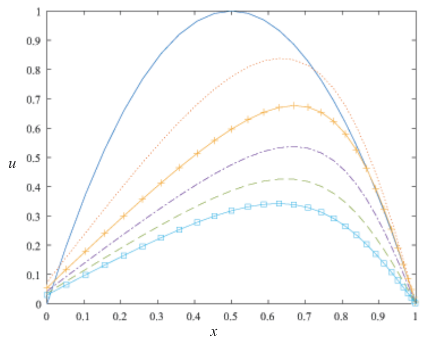
Due to the high flexibility and accuracy of MLS, it is considered to solve these equations. Because they discretize the problem domain without using a predetermined grid, it seems to work well for equations that have shocks at points in the domain. On the other hand, contour integral and other suitable 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,



(a) a



(b) b



(c) c

Figure 7: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of **example 7.3** by the new method expressed in this article with $N = 60$ and $\boldsymbol{\nu = 0.1}$.*
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ with the symbols $-, o, +, *, --, \square$ respectively.

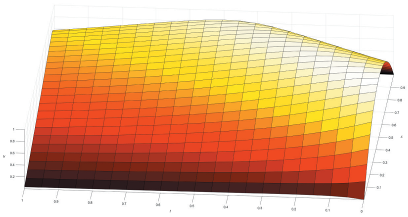
Table 6: *A comparison between the solution of equation of **Example 7.5** using the article method with exact answer and other numerical methods*

x	exact	new method$N = 60$	[13]$N = 100$
0.1	0.490626	0.490633	0.490633
0.2	0.478138	0.478146	0.478148
0.3	0.465679	0.465685	0.465690
0.4	0.453261	0.453268	0.453273
0.5	0.440902	0.440907	0.440914
0.6	0.428616	0.428620	0.428627
0.7	0.416416	0.416419	0.416428
0.8	0.404319	0.404320	0.404329
0.9	0.392336	0.392337	0.392344

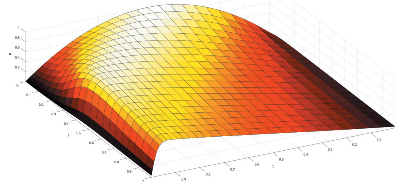
but contour integrals are not the only solutions offered for this problem, but the contour integral method attracts us due to its generality for dealing with arbitrary functions. There is considerable flexibility with this procedure.

The method mentioned in this paper has been tested on 5 examples at the end, leading to quite satisfactory resultsy. The numerical results obtained after solving the Burgers-Fisher and Burgers' equations are compared with the existing numerical solutions as well as with the exact answers to the problems. It is concluded that the method described in this article offers better accuracy than other existing numerical techniques.

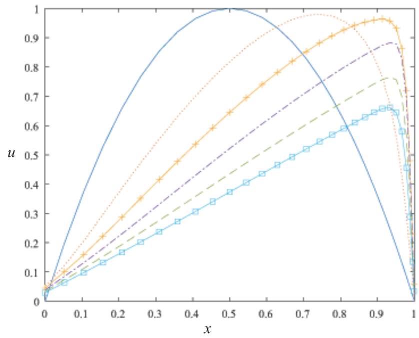
The main advantage of the design is that it can depict the behavior of numerical solutions at a small kinematic viscosity coefficient $\nu = 0.1, 0.01, 0.001, 0.0001$, up to the times when most numerical methods fail. It is possible that this method could also be used to solve model equations, including more mechanical, physical, or biophysical effects, such as nonlinear convection, reaction, linear propagation, and scattering.



(a) a

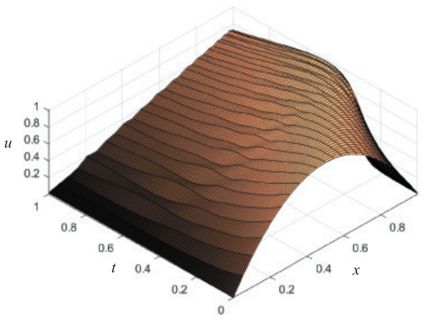


(b) b

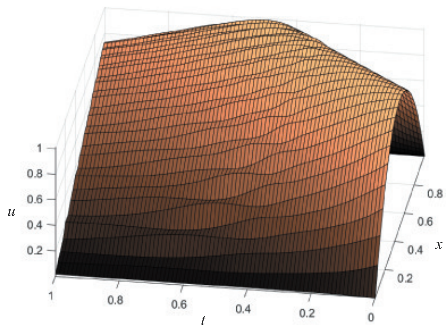


(c) c

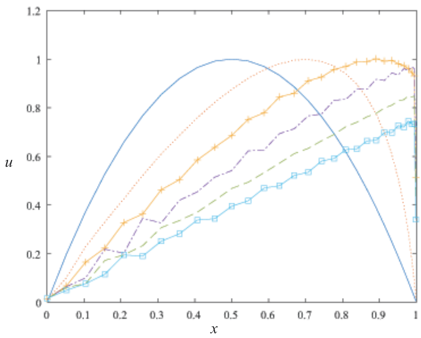
Figure 8: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.3*** *by the new method expressed in this article with* $N = 60$ *and* $\boldsymbol{\nu = 0.01}$ *to time* $t = 1$.
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*



(a) a

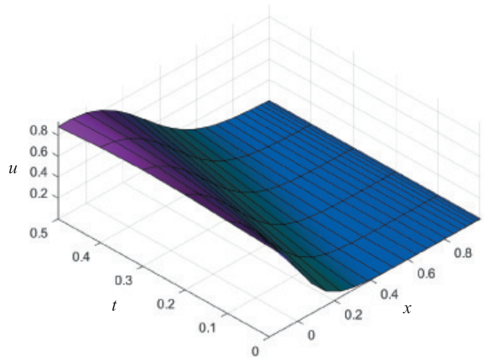


(b) b

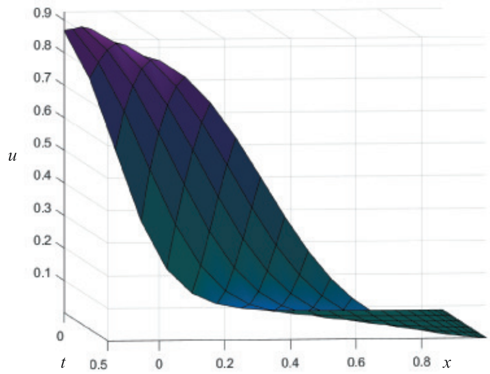


(c) c

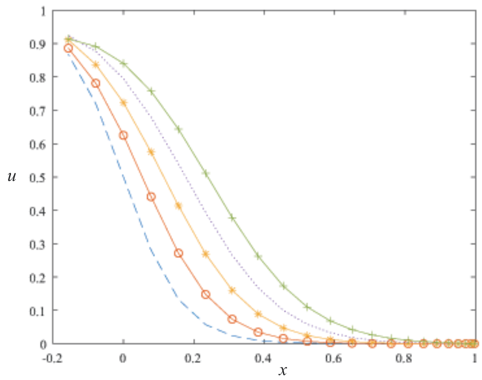
Figure 9: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.3*** *by the new method expressed in this article with* $N = 60$ *and* $\boldsymbol{\nu = 0.001}$ *to time* $t = 1$.
In Figure(c), the answer is shown at times $t = 0, 0.2, 0.4, 0.6, 0.8, 1$ *with the symbols* $-, o, +, *, --, \square$ *respectively.*



(a) a

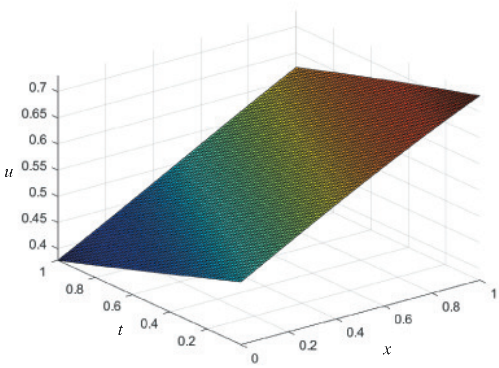


(b) b

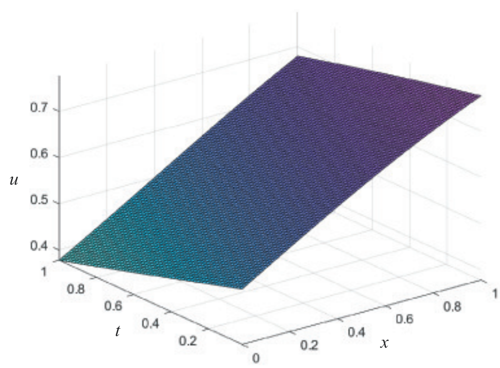


(c) c

Figure 10: *Figures (a),(b),(c) show different views of the answer to the Burgers' equation of* ***example 7.4*** *by the new method expressed in this article with* $N = 60$.
In Figure(c), the answer is shown at times $t = 0, 0.1, 0.2, 0.3, 0.4$ *with the symbols* $--, o, *, .., +, \square$ *respectively.*



(a) a



(b) b

Figure 11: *Figure (a) shows the answer to the Burgers' equation of* ***example 7.5*** *by the new method expressed in this article with* $N = 40, \alpha = 1, \beta = 1$.
Figure(b) shows the exact solution of ***example 7.5***

References

[1] S. Abbasbandy, *The effects of MHD flow of third grade fluid by means of meshless local radial point interpolation (MLRPI)*, Int. J. Ind. Math. **7** (2015), no. 1, 1—11.

[2] A. Asaithambi, *Numerical solution of the Burgers' equation by automatic differentiation*, Appl. Math. Comput. **216** (2010), 2700-–2708.

[3] M. Aslefallah and E. Shivanian, *Nonlinear fractional integrodifferential reaction–diffusion equation via radial basis functions*, Eur. Phys. J. Plus **130** (2015), no. 47, 1-–9.

[4] H. Bateman, *Some recent researches on the motion of fluids*, Mon. Weather Rev. **43** (1915), 163-–170.

[5] A. Babu, B. Han, and N. Asharaf, *Numerical solution of the viscous Burgers' equation using Localized Differential Quadrature method*, Partial Differ. Equ. Appl. Math. **4** (2021), 100044.

[6] J.M. Burgers, *Mathematical Examples Illustrating Relations Occurring in the Theory of Turbulent Fluid Motion*, North-Holland Pub. Co., 1939.

[7] J.M. Burgers, *A mathematical model illustrating the theory of turbulence*, Adv. Appl. Mech. **1** (1948), 171-–199

[8] E. Babolian and J. Saeidian, *Analytic approximate solutions to Burgers, Fisher, Huxley equations and two combined forms of these equations*, Commun. Nonlinear Sci. Numer. Simul. **14** (2009), 1984–1992.

[9] G. Beylkin, J. M. Keiser, and L. Vozovoi, *A new class of time discretization schemes for the solution of nonlinear PDEs*, J. Comput. Phys. **147** (1998), 362–374.

[10] G. Beylkin, J. M. Keiser, and L. Vozovoi, *A new class of time discretization schemes for the solution of nonlinear PDEs*, J. Comput. Phys. **147** (1998), 362—381.

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

[12] S.M. Cox and P.C. Matthews, *Exponential Time Differencing for Stiff Systems*, J. Comput. Phys. **176** (2002), 430—455.

[13] V. Chandraker, A. Awasthi, and S. Jayaraj, *Numerical Treatment of Burger-Fisher equation*, Proc. Technol. **25** (2016), 1217-–1225

[14] M. Dehghan, A. Hamidi, and M. Shakourifar, *The solution of coupled Burgers equations using Adomian–Pade technique*, Appl. Math. Comput. **189** (2007), 1034—1047.

[15] A. Dogan, *A Galerkin finite element approach to Burgers' equation*, Appl. Math. Comput. **157** (2004), 331-–346.

[16] M. Dehghan and A. Ghesmati, *Combination of meshless local weak and strong (mlws) forms to solve the two dimensional hyperbolic telegraph equation*, Eng. Anal. Bound. Elem. **34** (2010), no. 4, 324—336.

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

[18] F. Fakhar-Izadi and M. Dehghan, *Space-time spectral method for a weakly singular parabolic partial integro-differential equation on irregular domains*, Comput. Math. Appl. **67** (2014), no. 10, 1884—1904

[19] H. Fatahi, J. Saberi-Nadjafi, E. Shivanian, *A new spectral meshless radial point interpolation (SMRPI) method for the twodimensional Fredholm integral equations on general domains with error analysis*, J. Comput. Appl. Math. **294** (2016), 196—209

[20] L.N.G. Filon, *On a quadrature formula for trigonometric integrals*, Proc. Roy. Soc. Edin. Sect. A **49** (1928–1929), 38-–47.

[21] A. Golbabai and M. Javidi, *A spectral domain decomposition approach for the generalized Burgers-Fisher equation*, Chaos Solitons Fract. **39** (2009), 385–392.

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

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

[24] M. Hajiketabi, S. Abbasbandy, and F. Casas, *The Lie-group method based on radial basis functions for solving nonlinear high dimensional generalized Benjamin–Bona–Mahony–Burgers equation in arbitrary domains*, Appl. Math. Comput. **321** (2018), 223—243

[25] W. Huang and R.D. Russell, *Adaptive Moving Mesh Methods*, Springer, New York, 2011.

[26] H.N.A. Ismail and A.A.A. Rabboh, *A restrictive Pade approximation for the solution of the generalized Fisher and Burger-Fisher equation*, Appl. Math. Comput. **154** (2004), 203–210.

[27] A. Iserles, *On the numerical quadrature of highly oscillating integrals I: Fourier transforms*, IMA J. Numer. Anal. **24** (2004), 365-–391.

[28] R. Jiwari, *A Haar wavelet quasilinearization approach for numerical simulation of Burgers' equation*, Comput. Phys. Commun. **183** (2012), 2413-–2423

[29] M.K. Kadalbajoo, K.K. Sharma, and A. Awasthi, *A parameter-uniform implicit difference scheme for solving time dependent Burgers' equation*, Appl. Math. Comput. **170** (2005), 1365—1393.

[30] D. Kaya and S.M.El. Sayed, *A numerical simulation and explicit solutions of the generalized Burger-Fisher equation*, Appl. Math. Comput. **152** (2004), 403–413.

[31] A.K. Khalifa, Khalida Inayat Noor, and Muhammad Aslam Noor, *Some numerical methods for solving Burgers equation*, Int. J. Phys. Sci. **6** (2011), no. 7, 1702—1710.

[32] A. Korkmaz, *Shock wave simulations using sinc differential quadrature method*, Eng. Comput. **28** (2011), no. 6, 654-–674.

[33] A. Korkmaz and Idris Dag, *Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation*, J. Franklin Inst. **348** (2011), no. 10, 2863-–2875.

[34] A. Korkmaz, A. Murat Aksoy, and I. Dag, *Quartic B-spline differential quadrature method*, Int. J. Nonlinear Sci. **11** (2011), no. 4, 403-–411.

[35] A. Kassam and L.N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, Siam J. Sci. Comput, Soc. Ind. Appl. Math. **26** (2005), no. 4, 1214-–1233.

[36] W. Liao, *An implicit fourth-order compact finite difference scheme for onedimensional Burgers' equation*, Appl. Math. Comput. **206** (2008), 755—764.

[37] R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2004.

[38] X.L. Li, *Error estimates for the moving least-square approximation and the element-free Galerkin method in n-dimensional spaces*, Appl. Numer. Math. **99** (2016), 77-–97.

[39] X.L. Li and Sh. Li,*On the stability of the moving least squares approximation and the element-free Galerkin method*, Comput. Math. Appl. **72** (2016), 1515–1531.

[40] Z. Lei, G. Tianqi, Z. Ji, J. Shijun, S. Qingzhou, and H. Ming, *An adaptive moving total least squares method for curve fitting*, Measurement **49** (2014), 107—112.

[41] G. Liu and Y. Gu, *An Introduction to Meshfree Methods and Their Programming*, Springer, Berlin, 2005.

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

[43] D.R. Mott, E.S. Oran, and B. van Leer, *A quasi-steady state solver for the stiff ordinary differential equations of reaction kinetics*, J. Comput. Phys. **164** (2000), 407-–428.

[44] R.C. Mittal and R.K. Jain, *Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method*, Appl. Math. Comput. **218** (2012), 7839—7855

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

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

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

[48] H. Nojavan, S. Abbasbandy, and M. Mohammadi, *Local variably scaled Newton basis function collocation method for solving Burgers' equation*, Appl. Math. Comput. **330** (2018), 23—41

[49] M.M. Rashidi, G. Domairry, and S. Dinarvand, *Approximate solutions for the Burgers' and regularized long wave equations by means of the homotopy analysis method*, Commun. Nonlinear Sci. Numer. Simul. **14** (2009), 708—717.

[50] M.M. Rashidi and E. Erfani, *New analytic method for solving Burgers' and nonlinear heat transfer equations and comparison with HAM*, Comput. Phys. Comm. 180 (2009) 1539–1544.

[51] K. Rahman, N. Helil, and R. Yimin, *Some new semiimplicit finite difference schemes for numerical solution of Burgers equation*, Int. Conf. Comput. Appl. Syst. Model., IEEE, 2010, V14—451.

[52] A.R. Soheili, A. Kerayechian, and N. Davoodi, *Adaptive numerical method for Burgers-type nonlinear equations*, Appl. Math. Comput. **219** (2012), 3486—3495

[53] C. Schuster, A. Christ, and W. Fichtner, *Review of FDTD time-stepping for efficient simulation of electric conductive media*, Microwave Optic. Technol. Lett. **25** (2000), 16–21.

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

[55] E. Shivanian, *A new spectral meshless radial point interpolation (SMRPI) method: a well-behaved alternative to the meshless weak forms*, Eng. Anal. Bound. Elem. **54** (2015), 1—12.

[56] E. Shivanian, *On the convergence analysis, stability, and implementation of meshless local radial point interpolation on a class of three-dimensional wave equations*, Int. J. Numer. Meth. Eng. May. **105** (2015), no. 2, 83–110.

[57] E. Shivanian, A. Rahimi, and M. Hosseini, *Meshless local radial point interpolation to three-dimensional wave equation with Neumann's boundary conditions*, Int. J. Comput. Math. **93** (2016), 2124–2140.

[58] E. Shivanian and H.R. Khodabandehlo, *Application of meshless local radial point interpolation (MLRPI) on a one-dimensional inverse heat conduction problem*, Ain. Shams. Eng. J. **7** (2016), no. 3, 993–1000

[59] 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*, Z. Naturf. A **70** (2015), no. 8, 673-–682

[60] E. Shivanian, *Local integration of population dynamics via moving least squares approximation*, Engin. Comput. **32** (2016), no. 2, 331—342.

[61] E. Shivanian, *Local Petrov–Galerkin (MLPG) method for three-dimensional nonlinear wave equations via moving least squares approximation*, Eng. Anal. Bound. Element. **50** (2015), 249—257

[62] A.H.A.E. Tabatabaei, E. Shakour, M. Dehghan, *Some implicit methods for the numerical solution of Burgers equation*, Appl. Math. Comput. **191** (2007), 560—570.

[63] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House, Boston, 1995.

[64] A.M. Wazwaz and A. Gorguis, *An analytic study of Fisher's equation by using Adomian decomposition method*, Appl. Math. Comput. **154** (2004), 609–620.

[65] A.M. Wazwaz, *The Tanh method for generalized forms of nonlinear heat conduction and Burgers-Fisher equations*, Appl. Math. Comput. **169** (2005), 321–338.

[66] M. Xu, R. Wang, J. Zhang, and Q. Fang, *A novel numerical scheme for solving Burgers' equation*, Appl. Math. Comput. **217** (2011), 4473-–4482.

[67] J. Zhao, H. Li, Z. Fang, and X. Bai, *Numerical Solution of Burgers' Equation Based on Mixed Finite Volume Element Methods*, Discrete Dyn. Nature Soc. **2020** (2020), Article ID 6321209, 13 pages.