symmetry

MDPI

Article

Explicit Runge–Kutta Numerical Manifold Method for Solving the Burgers' Equation via the Hopf–Cole Transformation

Yue Sun , Qian Chen , Tao Chen * and Longquan Yong

School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong 723001, China; sunyue@snut.edu.cn (Y.S.); chenqian@snut.edu.cn (Q.C.); yonglongquan@126.com (L.Y.)
* Correspondence: chentao@snut.edu.cn

Abstract: This paper presents an efficient numerical manifold method for solving the Burgers' equation. To improve accuracy and streamline the solution process, we apply a nonlinear function transformation technique that reformulates the original problem into a linear diffusion equation. We utilize a dual cover mesh along with an explicit multi-step time integration method for spatial and temporal discretization, respectively. Constant cover functions are employed across mathematical covers, interconnected by a linear weight function for each manifold element. The full discretization formulation is derived using the Galerkin weak form. To efficiently compute the inverse of the symmetric positive definite mass matrix, we employ the Crout algorithm. The performance and convergence of our method are rigorously evaluated through several benchmark numerical tests. Extensive comparisons with exact solutions and alternative methods demonstrate that our approach delivers an accurate, stable, and efficient computational scheme for the Burgers' equation.

Keywords: convection-dominated equation; dual cover mesh; Hopf–Cole transformation; Galerkin method; Runge–Kutta scheme

check for updates

Citation: Sun, Y.; Chen, Q.; Chen, T.; Yong, L. Explicit Runge–Kutta Numerical Manifold Method for Solving the Burgers' Equation via the Hopf–Cole Transformation. *Symmetry* **2024**, *16*, 1521. https://doi.org/10.3390/sym16111521

Academic Editor: Mariano Torrisi

Received: 7 October 2024
Revised: 3 November 2024
Accepted: 11 November 2024
Published: 13 November 2024

1. Introduction

Burgers' equation [1] is a fundamental quasi-linear partial differential equation (PDE) that appears in various fields, including gas dynamics, acoustic wave, biological process, and traffic flow modeling. It provides a simpler version of the Navier–Stokes equations, capturing the interplay between nonlinear convection and diffusion. The focus of this study is on the following problem:

$$u_t + uu_x = \nu u_{xx}, \qquad (x,t) \in (a,b) \times (T_0, \infty), \tag{1}$$

$$u(x, T_0) = u_0(x), \qquad x \in (a,b), \tag{2}$$

$$u(a,t) = 0 = u(b,t), \quad t \in [T_0, \infty), \tag{3}$$

in which $u = u(x,t)$ represents the fluid velocity, T_0 is the initial time, and ν denotes the kinematics viscosity coefficient. $1/\nu$ indicates the Reynolds number.

Despite its simple form, the Burgers' equation presents significant analytical challenges due to its nonlinearity, with exact solutions only available for specific cases [2,3]. Computationally solving Burgers' equation is also hindered by the complex interaction between the nonlinear convection term and linear dissipation. In convection-dominated scenarios, where the viscosity ν is small, resolving sharp gradients and shock waves becomes particularly difficult. These difficulties necessitate the development of stable and accurate numerical techniques.

Throughout the past several decades, different computational techniques have been introduced to solve this challenge, including the finite difference method (FDM) [4–7], finite element method (FEM) [8–10], spectral methods [11–14], and finite volume method (FVM) [15,16]. However, the performance of these approaches depends heavily on the quality of the spatial discretization. Accurately resolving steep gradients often requires fine meshes, which increase computational cost and can lead to numerical instabilities. Although the local discontinuous Galerkin method (LDG) [17–19] helps mitigate mesh dependency, it introduces additional degrees of freedom within each element, further escalating computational costs.

Alternatively, meshless methods avoid the reliance on structured grids and can achieve better flexibility in capturing complex nonlinear behavior, such as the characteristic-based element-free Galerkin method (EFCGM) [20,21], semi-Lagrangian EFG method [22], reproducing kernel particle method [23], radial basis function method [24], and finite point method [25]. The numerical manifold method (NMM) [26], a partition-of-unity mesh-free approach [27], utilizes a dual cover mesh for spatial discretization, offering greater flexibility in representing complex geometries and material domains. Compared to conventional mesh-based approaches, this method does not require the mesh to strictly conform to problem geometry shape and achieves higher accuracy with fewer degrees of freedom. Due to these advantages, the NMM has shown significant potential in solving nonlinear PDEs, particularly in solid mechanics [28–36] and fluid flow problems [37–39].

In our previous work [40], we enhanced the NMM by integrating the characteristic method to solve the Burgers' equation. The results showcased the high accuracy and stability of this approach across a wide range of benchmark examples, demonstrating its capability to yield reliable solutions. However, a notable challenge arises as the viscosity coefficient ν decreases. In this case, the method necessitates smaller time steps and finer cover sizes to accurately capture sharp gradients. This issue has a significant impact on computational efficiency.

The Hopf–Cole (H-C) transformation was independently proposed by Hopf [41] and Cole [2] in 1950s. This transformation can convert the Burgers' equation into a linear diffusion equation by removing the nonlinear convective term. In this way, the solution process of the original problem can be greatly simplified [42,43]. Since then, several studies [44–50] have applied this function transformation technique to numerically solve the Burgers' equation, and diversified numerical methods were adopted to discretize the transformed linear diffusion equation. Öziş et al. [51] employed the FEM with linear elements to solve the resulting linear diffusion equation, while Zhao et al. [52] applied an LDG method. Mukundan and Awasthi [53] also tackled the transformed equation using the method of lines combined with a backward difference scheme.

The present study develops a Runge–Kutta numerical manifold Galerkin method (RKNMGM) to tackle the Burgers' equation using the H-C transform. The problem domain is spatially discretized with a dual cover mesh. The unknowns of the transformed equation are approximated using piecewise constant cover functions combined with first-order polynomial weighting functions over manifold elements. Temporally, the time derivative term is integrated using the total variation diminishing Runge–Kutta scheme with second-order accuracy (TVD-RK2) [54]. Due to the TVD property for limiting numerical oscillations, this scheme offers enhanced efficiency and stability compared to the explicit Crank–Nicolson method adopted in our previous study [40]. The full discretization formulation the RKNMGM is derived from is based on the Galerkin weak form, resulting in a symmetric system of equations. The Crout algorithm is utilized to efficiently compute the inverse of the symmetric positive definite mass matrix. Finally, the solution to the original Burgers' equation is recovered from the results of the transformed problem by applying the H-C transformation once more.

Several benchmark numerical examples on the Burgers' equation are conducted to assess the performance of the RKNMGM. Diversified parameter settings are considered to ensure a comprehensive evaluation of the method's robustness. Our results are rigorously compared against exact solutions and alternative numerical approaches, highlighting the accuracy and efficiency of the current method. Additionally, we investigate the convergence behavior of the RKNMGM across various mesh configurations, which provides insights into its stability and reliability under different discretization strategies. Through extensive analysis and discussion, the effectiveness of this method for steep gradient problems is demonstrated.

2. H-C Transformation

By importing an auxiliary function $\theta(x,t)$, the H-C transform uses the following formulation to substitute $u(x,t)$, i.e.,

$$u \equiv -2\nu \frac{\partial}{\partial x} \ln \theta = -2\nu \frac{\theta_x}{\theta}. \tag{4}$$

With Equation (4), we can transform the original problem defined by Equations (1)–(3) into the following simpler problem:

Governing Equation: substituting Equation (4) into Equation (1) leads to

$$\theta_t = \nu\theta_{xx}, \quad \forall (x,t) \in (a,b) \times (T_0, \infty). \tag{5}$$

Initial Condition (IC): replacing Equation (2) with Equation (4) gives

$$\theta(x,0) = \exp\left[-(2\nu)^{-1} \int_0^x u_0(s)\, ds\right], \quad \forall x \in (a,b). \tag{6}$$

Boundary Condition (BC): applying the transformation upon Equation (2) results in

$$\theta_x(x,t) = 0, \quad \theta_x(b,t) = 0, \quad \forall t \in [T_0, \infty). \tag{7}$$

In the following sections, we will present the numerical algorithms developed to solve the transformed problem, followed by the procedure used to recover the solution of Equations (1)–(3) from the computed results.

3. The RKNMGM Scheme

3.1. Spatial Discretization

The RKNMGM utilizes NMM's dual cover mesh for the spatial discretization of the problem domain. Such mesh comprises two interrelated collections of overlapping regions. The first is mathematical covers (MCs), whose union forms a superset of the problem domain. The second is physical covers (PCs), in which each PC occupies the portion of an MC that intersects with the problem domain. The intersections of adjacent PCs generate manifold elements (MEs), which serve as the foundational units for discretization.

For the transformed problem modeled by Equations (5)–(7), we can build its dual cover mesh with the following procedure:

Step 1 Adopt the uniform mesh with $N+3$ nodes with $x_1 = a$ and $x_{N+1} = b$, and the MCs are $\overline{C}_k = [x_{k-1}, x_{k+1}]$, $k = 1, 2, \ldots, N+1$, which are shown as the bases of the overlapping triangles at the lowest row in Figure 1;

Step 2 Computing the intersection of each MC and the problem domain $[a,b]$ leads to the boundary PCs $C_1 = [x_1, x_2]$ and $C_N = [x_N, x_{N+1}]$, and the interior PCs $C_k = [x_{k-1}, x_{k+1}]$, $k = 2, 3, \ldots, N$, which are illustrated as the bases of the overlapping triangles in the middle row in Figure 1;

Step 3 Computing the intersections of every adjacent pair of PCs gives the MEs: $E_k = [x_k, x_{k+1}]$, $k = 1, 2, \ldots, N$, which are rendered as the triangle bases in the uppermost row in Figure 1.

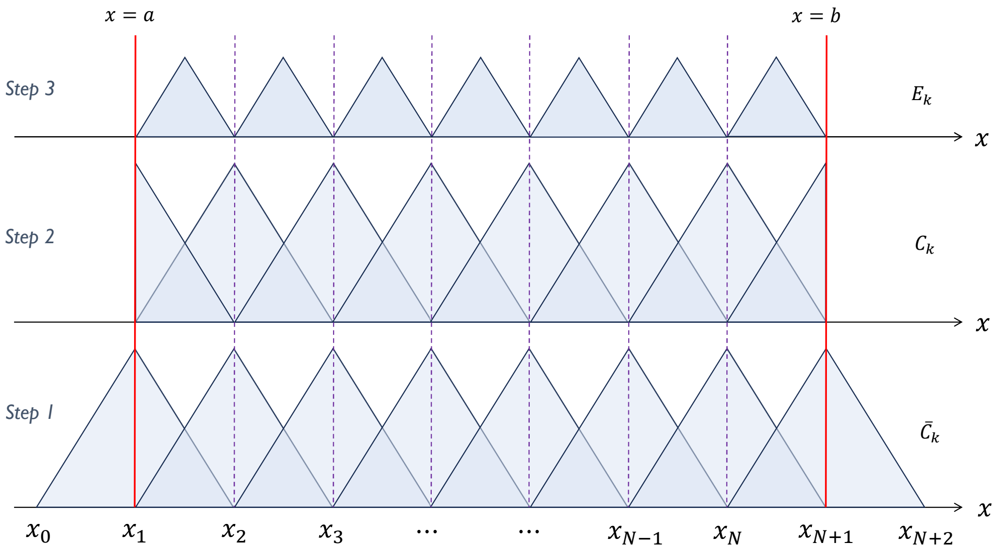

Figure 1. Procedure to build the dual cover mesh for the RKNMGM: Step 1 to generate the MC collection $\{\overline{C}_k\}_{k=1}^{N+1}$; Step 2 to create the PC set $\{C_k\}_{k=1}^{N+1}$; and Step 3 to form the ME series $\{E_k\}_{k=1}^{N}$.

3.2. Approximation

In the RKNMGM, the approximation of unknown variables is conducted over MCs, while the integration is calculated over MEs. Below, the approximation approach and integration formulations used in our method are described.

Given the dual cover mesh $(\{\overline{C}_k\}_{k=1}^{N+1}, \{C_i\}_{k=1}^{N+1}, \{E_k\}_{k=1}^{N})$, we approximate the unknown variable θ over E_i by a linear polynomial—that is,

$$\theta_k(x) \approx \hat{\theta}_k(x) = \boldsymbol{p}(x)\boldsymbol{d}_k, \quad x \in E_k, \quad k = 1, 2, \cdots, N, \tag{8}$$

where $\boldsymbol{p}(x) = (1, x)$ represents the linear polynomial basis function and $\boldsymbol{d}_k = (d_k^1, d_k^2)^\top$ denotes the coefficient. In the RKNMGM, the cover functions θ_k are the unknown to be determined. Thus, we will represent the coefficient by these cover functions below.

Noting $x_k \in \overline{C}_k$ and $x_{k+1} \in \overline{C}_{k+1}$, it is manifest that x_k and x_{k+1}, as the two ends of E_k, satisfy Equation (8)—that is,

$$\begin{aligned} \hat{\theta}_k(x_k) &= d_k^1 + d_k^2 x_k, \quad x_k \in \overline{C}_k, & (9) \\ \hat{\theta}_{k+1}(x_{k+1}) &= d_k^1 + d_k^2 x_{k+1}, \quad x_{k+1} \in \overline{C}_{k+1}. & (10) \end{aligned}$$

In the present study, we assume the cover function keeps constant over each MC. Therefore, substituting $\hat{\theta}_k(x_k) = \theta_k$ and $\hat{\theta}_k(x_{k+1}) = \theta_{k+1}$ into Equations (9) and (10) leads to

$$\boldsymbol{d}_k = \begin{pmatrix} d_k^1 \\ d_k^2 \end{pmatrix} = \frac{1}{h} \begin{pmatrix} -1 & 1 \\ x_{k+1} & -x_k \end{pmatrix} \begin{pmatrix} \theta_k \\ \theta_{k+1} \end{pmatrix}, \tag{11}$$

where $h = x_{k+1} - x_k$ due to the adopted uniform mesh, $k = 1, 2, \ldots, N$.

Furthermore, by substituting Equation (11) into Equation (8), we can obtain the final approximation formulation,

$$\hat{\theta}_k(x) = \mathbf{N}_k(x)\boldsymbol{\theta}_k, \quad k = 1, 2, \cdots, N, \tag{12}$$

in which $\mathbf{N}_k(x)$ is E_k's linear weight function—that is,

$$\mathbf{N}_k(x) = h^{-1}(x_{k+1} - x, \quad x - x_k), \quad k = 1, 2, \cdots, N. \tag{13}$$

and $\boldsymbol{\theta}_k = (\theta_k, \theta_{k+1})^\top$ contains the cover functions, which are the unknown to be determined.

3.3. Weak Formulation

Since the system defined in Equations (5)–(7) is self-adjoint, the Galerkin weighted residual method can yield a symmetric system of equations. Consequently, we apply this method to transform Equation (5) into its weak form.

Choose the first-order Soblev space $H^1[a,b]$ as the trial and test function space. Multiplying both sides of Equation (5) by any test function $w(x)$ from $H^1[a,b]$, we obtain

$$\langle w,\ \theta_t \rangle = \langle w,\ \theta_{xx} \rangle, \tag{14}$$

where the notation $\langle \cdot, \cdot \rangle$ represents the inner product in $L_2[a,b]$.

By applying integration by parts to the above equation, we arrive at

$$\langle w,\ \theta_t \rangle = \langle w_x,\ -\nu\theta_x \rangle + w(b)\theta_x(b,t) - w(a)\theta_x(a,t). \tag{15}$$

We eliminate the last two terms on the right-hand side of Equation (15) by applying the Neumann BC in Equation (7); then, we have

$$\int_a^b w \frac{\partial \theta}{\partial t} dx = -\nu \int_a^b \frac{\partial w}{\partial x} \frac{\partial \theta}{\partial x} dx. \tag{16}$$

3.4. Semi-Discretization Scheme

Now, we derive the semi-discretization scheme from the weak form presented in Equation (16) by approximating $\theta(x,t)$ and $w(x)$ with Equation (12), i.e.,

$$\theta(x,t) = \sum_{k=1}^{N} \boldsymbol{N}_k(x)\boldsymbol{\theta}_k, \tag{17}$$

$$w(x) = \boldsymbol{N}_j(x), \quad x \in E_j,\ j = 1,2,\cdots,N. \tag{18}$$

Then, substituting the above formulations into Equation (16) leads to

$$\boldsymbol{M}_k \frac{\partial \boldsymbol{\theta}_k}{\partial t} = -\nu \boldsymbol{K}_k \boldsymbol{\theta}_k, \quad k = 1,2,\cdots,N, \tag{19}$$

where $\boldsymbol{M}_k$ and $\boldsymbol{K}_k$ are the mass matrix and stiffness matrix on E_i, respectively, taking the formulas below

$$\boldsymbol{M}_k = \int_{E_k} \boldsymbol{N}_k^{\top}(x)\boldsymbol{N}_k(x)dx, \quad \boldsymbol{K}_k = \int_{E_k} \frac{d\boldsymbol{N}_k^{\top}(x)}{dx}\frac{d\boldsymbol{N}_k(x)}{dx}dx. \tag{20}$$

For calculating the integrals over E_k in Equation (20), we substitute Equation (13) into Equation (20), and subsequently employ the simplex integration method [55,56] to obtain the analytically exact expressions of $\boldsymbol{M}_i$ and $\boldsymbol{K}_i$, i.e.,

$$\boldsymbol{M}_k = \frac{h}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \boldsymbol{K}_k = \frac{1}{h}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad k = 1,2,\cdots,N. \tag{21}$$

3.5. Time Integration Scheme

Next, we employ the TVD-RK2 [54] to solve the ODE problem of Equation (19) with the initial condition in Equation (6). As $\boldsymbol{M}_i$ is symmetric positive definite, we can rewrite the semi-discretized system Equation (19) into the following general form:

$$\frac{d\boldsymbol{\theta}_k}{dt} = -\nu \boldsymbol{M}_k^{-1}\boldsymbol{K}_k\boldsymbol{\theta}_k \equiv \mathcal{L}(\boldsymbol{\theta}_k), \tag{22}$$

where $\mathcal{L}$ denotes the spatial operator.

For solving Equation (22) for $t \in [T_0, T_0 + T]$, the TVD-RK2 scheme advances $\boldsymbol{\theta}_k$ over a time step Δt in the following two phases:

$$\begin{cases} \boldsymbol{\theta}_k^{(n+\frac{1}{2})} = \boldsymbol{\theta}_k^{(n)} + \Delta t \mathcal{L}(\boldsymbol{\theta}_k^{(n)}), \\ \boldsymbol{\theta}_k^{(n+1)} = \dfrac{1}{2}\left[\boldsymbol{\theta}_k^{(n)} + \boldsymbol{\theta}_k^{(n+\frac{1}{2})} + \Delta t \mathcal{L}(\boldsymbol{\theta}_k^{(n+\frac{1}{2})})\right], \end{cases} \quad n = 1, 2, \ldots, T/\Delta t, \tag{23}$$

where $\boldsymbol{\theta}_k^{(n)}$ represents $\boldsymbol{\theta}(x_k, T_0 + n\Delta t)$. Particularly, $\boldsymbol{\theta}_k^{(0)} = (\theta_k^{(0)}, \theta_{k+1}^{(0)})$ contains the initial cover functions over $\overline{C}_k$ and $\overline{C}_{k+1}$, respectively.

Since the cover function remains constant during a time step, we have $\theta_k^{(0)} = \theta(x, T_0)$, $\forall x \in \overline{C}_k$, and $k = 1, 2, \cdots, N+1$. Particularly, we take $\theta_k^{(0)} = \theta(x_k, T_0)$, where x_k is the midpoint of $\overline{C}_k$. According to Equation (6), we have

$$\theta_k^{(0)} = \theta(x_k, T_0) = \exp\left[-\frac{1}{2\nu}\int_0^{x_k} u_0(s)\, ds\right]. \tag{24}$$

As described in Equation (23), the TVD-RK2 scheme is an explicit time integration method. To ensure the computational stability, Δt has to satisfy the CFL condition [54]:

$$\Delta t \leq \frac{h}{2\,\|\boldsymbol{\theta}^{(n)}\|_\infty}, \tag{25}$$

in which $\boldsymbol{\theta}^{(n)}$ contains all $N+1$ cover functions, and the norm indicates

$$\|\boldsymbol{\theta}^{(n)}\|_\infty = \max\{|\theta_1^{(n)}|, |\theta_2^{(n)}|, \cdots, |\theta_{N+1}^{(n)}|\}. \tag{26}$$

3.6. Solution Restoration

Given $\{\theta_k\}_{k=1}^{N+1}$ as the solution of Equations (5)–(7) by the RKNMGM, we recover the solution of Equations (1)–(3) using the H-C transform once more.

First, use the central differential formula to compute θ_x, and Equation (4) becomes

$$u_k^{(n)} \equiv u(x_k, n\Delta t) \approx \frac{\nu\left(\theta_{k-1}^{(n)} - \theta_{k+1}^{(n)}\right)}{h\theta_k^{(n)}}, \tag{27}$$

where $k = 1, 2, \cdots, N+1$, and $n = 1, 2, \cdots, T/\Delta t$.

Since the collection $\{u_k^{(n)}\}_{k=1}^{N+1}$ comprises the numerical prediction on the MCs for Equations (1)–(3), we can use the NMM approximation provided in Equation (12) again to compute the solution at any point within the problem domain, i.e.,

$$u(x, n\Delta t) = \sum_{k=1}^{N} \boldsymbol{N}_k(x)\begin{pmatrix} u_k^{(n)} \\ u_{k+1}^{(n)} \end{pmatrix}, \quad n = 1, 2, \cdots, T/\Delta t. \tag{28}$$

Here, it worth noting that the weighting function $\boldsymbol{N}_k(x)$ possesses the Kronecker delta property in the ME E_k.

4. Implementation Aspects

The numerical implementation of the RKNMGM for solving Equations (1)–(3) can be described in Algorithm 1.

In Line 10, Equation (23) is repeatedly executed, and it is computationally heavy to compute the inverse of the mass matrix $\boldsymbol{M}$.

As $\boldsymbol{M}$ is symmetric positive definite, we employ Crout's method, described in Algorithm 2, to compute $\boldsymbol{M}^{-1}$.

Assume u^* and $\hat{u}$ as the exact solution and the prediction obtained by the RKNMGM at (x,t), respectively. The accuracy of the RKNMGM at (x,t) can be measured by the relative error $\epsilon = |u^* - \hat{u}|/|u^*|$.

To further evaluate the global accuracy of our method, assume $\boldsymbol{u}^*$ and $\hat{\boldsymbol{u}}$ to denote the exact solution and the prediction computed by the RKNMGM on all PCs. The global errors can be computed by

$$\|\boldsymbol{u}^* - \hat{\boldsymbol{u}}\|_2 = \left[\frac{1}{N+1}\sum_{k=0}^{N}\left(u^*(x_k,t) - \hat{u}(x_k,t)\right)^2\right]^{1/2} \tag{29}$$

$$\|\boldsymbol{u}^* - \hat{\boldsymbol{u}}\|_\infty = \max_{0\le k\le N}\left|u^*(x_k,t) - \hat{u}(x_k,t)\right|. \tag{30}$$

where the operators $\|\cdot\|_2$ and $\|\cdot\|_\infty$ represent the L_2 and L_∞ norms, respectively.

Algorithm 1: The RKNMGM scheme for solving Equations (1)–(3)

```
input  : $\nu$, $[a,b]$, $T$, $u_0(x)$, $\Delta t$, $N$
output : $u(x,T)$
Build the dual cover mesh $(\{\overline{C}_i\}_{i=1}^{N+1}, \{C_i\}_{i=1}^{N+1}, \{E_i\}_{i=1}^{N})$;
// Compute $\boldsymbol{M}$ and $\boldsymbol{K}$
for $i \leftarrow 1$ to $N$ do
    Add $\boldsymbol{M}_i$ into $\boldsymbol{M}$ at the diagonal starting at $(i,i)$ ;
    Add $\boldsymbol{K}_i$ into $\boldsymbol{K}$ at the diagonal starting at $(i,i)$ ;
end
// Solving Equations (5)−(7)
Compute $\boldsymbol{\theta}_i^{(0)}, i = 1,2,\ldots,N+1$, according to Equation (24);
for $n \leftarrow 1$ to $\frac{T}{\Delta t}$ do
    Compute Equation (23) for $\theta_1^{(n)}, \theta_2^{(n)}, \ldots, \theta_{N+1}^{(n)}$;
end
// Solution restoration
for $i \leftarrow 1$ to $N+1$ do
    Compute Equation (27) for $u(x_i,T)$;
end
Compute Equation (28) for $u(x,T), \forall x \in [a,b]$;
```

Algorithm 2: Crout's Method to Solve $\boldsymbol{M}^{-1}$

```
Input: $\boldsymbol{M}$, $N$
Output: $\boldsymbol{M}^{-1}$
Decompose $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{U}$;
for $k \leftarrow 1$ to $N+1$ do
    // Forward substitution by solving $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{I}(:,k)$
    for $p \leftarrow 1$ to $N+1$ do
        $\boldsymbol{y}(p) = \left[\boldsymbol{I}(p,k) - \sum_{q=1}^{p-1}\boldsymbol{L}(p,q)\boldsymbol{y}(q)\right]\Big/\boldsymbol{L}(p,p)$;
    end
    // Backward substitution by solving $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$
    for $p \leftarrow N+1$ to $1$ do
        $\boldsymbol{x}(p) = \left[\boldsymbol{y}(p) - \sum_{j=p+1}^{N+1}\boldsymbol{U}(p,q)\boldsymbol{x}(q)\right]\Big/\boldsymbol{U}(p,p)$;
    end
    $\boldsymbol{M}^{-1}(:,k) = \boldsymbol{x}$;
end
return $\boldsymbol{M}^{-1}$;
```

5. Validation Problems

Problem 1. *Given the problem presented in Equations (1)–(3) on* $(x,t) \in [0,1] \times (0,T]$ *with*

$$u_0(x) = \sin(\pi x), \quad x \in (0,1), \tag{31}$$

Cole [2] gave the exact solution

$$u(x,t) = \frac{2\pi\nu \sum_{m=1}^{\infty} A_m \exp(-\nu m^2 \pi^2 t) m \sin(m\pi x)}{A_0 + \sum_{m=1}^{\infty} A_m \exp(-\nu m^2 \pi^2 t) m \cos(m\pi x)}, \tag{32}$$

where, using $f(s) = \exp\{(2\pi\nu)^{-1}[\cos(\pi s) - 1]\}$, *the coefficients are*

$$A_0 = \int_0^1 f(s) ds, \quad A_m = 2\int_0^1 f(s)\cos(m\pi s) ds, \quad m = 1,2,3,\ldots. \tag{33}$$

Below, we employ the RKNMGM scheme to numerically solve this problem. Using Equation (4), we transform the governing equation from Equation (1) into Equation (5), then convert the IC in Equation (31) into

$$\theta^{(0)}(x) = \exp\left[\frac{\cos(\pi x) - 1}{2\nu\pi}\right], \quad \forall x \in (0,1), \tag{34}$$

and the homogeneous BC in Equation (7) becomes

$$\theta_x\big|_{x=0} = 0 = \theta_x\big|_{x=1}, \quad \forall t \in (0,T]. \tag{35}$$

As shown in Figure 2, we evaluate the convergence characteristics of the RKNMGM for this problem with $\nu = 0.1$ when $T = 0.1$. Five different ME sizes, i.e., $h = 1/50, 1/100, 1/200, 1/400, 1/800$, are adopted. During the computations, we keep $\Delta t = 10^{-6}$ as the time step length. It can be observed from the figure that the global errors of the RKNMGM in the L_2 and L_∞ consistently decrease while refining the spatial discretization, showcasing the good stability and convergence characteristic.

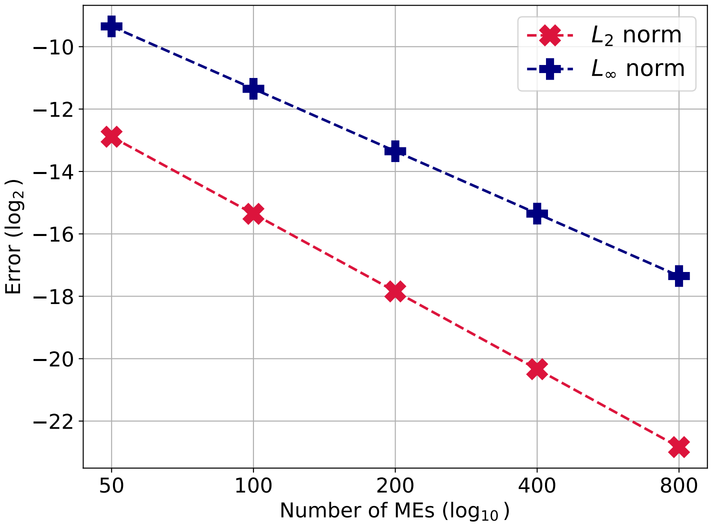

Figure 2. Convergence performance of the RKNMGM for Problem 1.

Then, we examine the numerical solution of Problem 1 for different T while using a viscosity coefficient of $\nu = 0.1$. For each case, the parameters of the RKNMGM are set to be $h = 0.05$ and $\Delta t = 0.001$. As depicted in Figure 3, we present the comparison of the solution curves produced by the RKNMGM against the exact solutions. The figure clearly indicates that the results from our method closely match the analytical formulations.

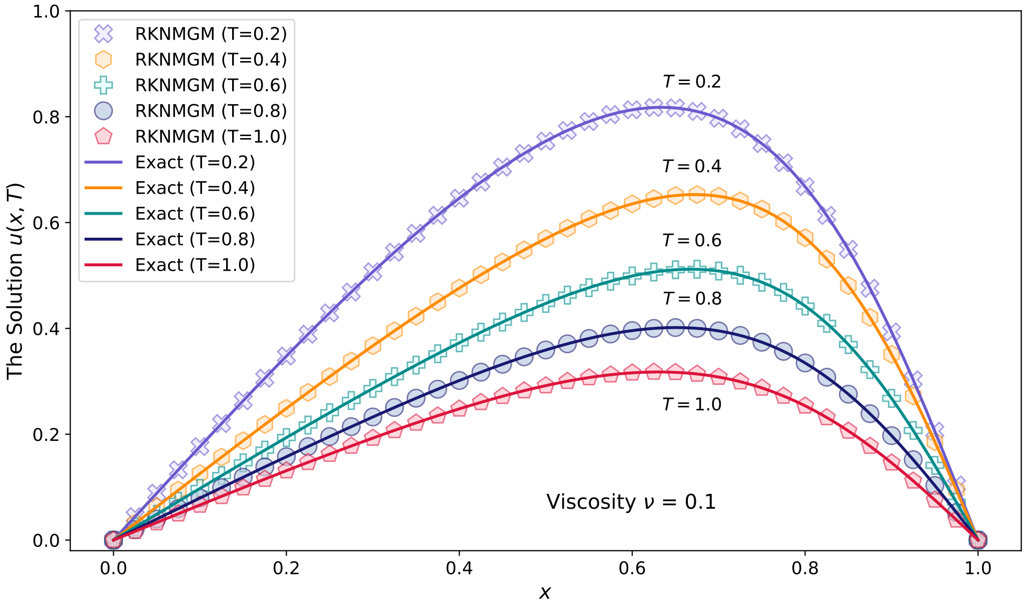

Figure 3. RKNMGM results using $\Delta t = 10^{-5}$, $h = 1/200$ for Problem 1 with $\nu = 0.1$.

Table 1 presents the comparison of our results with exact solutions and those provided by the EFCGM [20] and the ENMCGM [40] at specific points and different times. Here, the EFCGM adopts 20 nodes with $\Delta t = 10^{-3}$, the ENMCGM uses $h = 1/2000$ with $\Delta t = 10^{-7}$, while the RKNMGM employs $h = 1/200$ with $\Delta t = 10^{-5}$. Compared with the EFCGM, our method needs finer discretization to achieve similar accuracy. However, the RKNMGM can strictly satisfy the homogeneous BC, which is challenging for the EFCGM due to using the MLS approximation. Besides, the EFCGM adopts the cubic spline weight function and utilizes Gaussian quadrature to calculate integrals, while our method employ a linear weight function and directly uses exact integration formulations. Compared with the ENMCGM, the present method requires much less MEs and smaller time step size, and the accuracy is better. These comparisons demonstrate that the efficiency and effectiveness of our method is on par with the EFCGM as well as the ENMCGM.

Table 1. Numerical results and relative errors ϵ of different methods for Problem 1 ($\nu = 0.1$).

T	x	Exact	EFCGM [20]	ϵ	ENMCGM [40]	ϵ	Present	ϵ
0.4	0.25	0.308894	0.308892	7.21×10^{-6}	0.308901	2.19×10^{-5}	0.308859	1.14×10^{-4}
	0.50	0.569632	0.569629	6.06×10^{-6}	0.569705	1.27×10^{-4}	0.569603	5.08×10^{-5}
	0.75	0.625438	0.625446	1.30×10^{-5}	0.625473	5.61×10^{-5}	0.625460	3.61×10^{-5}
0.6	0.25	0.240739	0.240747	3.31×10^{-5}	0.240737	8.40×10^{-6}	0.240718	8.61×10^{-5}
	0.50	0.447206	0.447214	1.90×10^{-5}	0.447261	1.24×10^{-4}	0.447184	4.73×10^{-5}
	0.75	0.487215	0.487214	2.00×10^{-6}	0.487215	5.16×10^{-8}	0.487219	9.21×10^{-6}
0.8	0.25	0.195676	0.195679	1.75×10^{-5}	0.195665	5.40×10^{-5}	0.195659	8.56×10^{-5}
	0.50	0.359236	0.359241	1.38×10^{-5}	0.359269	9.17×10^{-5}	0.359214	6.16×10^{-5}
	0.75	0.373922	0.373923	3.33×10^{-6}	0.373893	7.69×10^{-5}	0.373911	3.00×10^{-5}
1.0	0.25	0.162565	0.162560	2.99×10^{-5}	0.162547	1.10×10^{-4}	0.162554	6.92×10^{-5}
	0.50	0.291916	0.291919	1.04×10^{-5}	0.291928	4.13×10^{-5}	0.291900	5.38×10^{-5}
	0.75	0.287474	0.287472	8.37×10^{-6}	0.287426	1.68×10^{-4}	0.287464	3.45×10^{-5}

We then address this problem with a viscosity coefficient of $\nu = 0.01$ using the RKNMGM, employing $h = 0.02$ and $\Delta t = 10^{-6}$. Figure 4 illustrates the comparison between our results and the exact solutions, indicating that the RKNMGM yields an outstanding agreement.

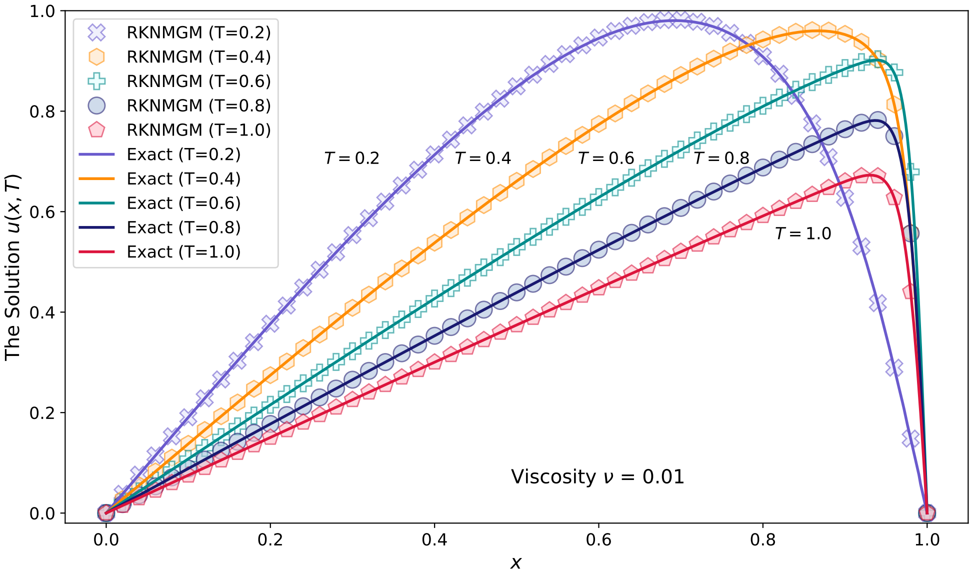

Figure 4. RKNMGM results using $\Delta t = 10^{-6}$, $h = 1/500$ for Problem 1 with $\nu = 0.01$.

Table 2 provides a comparison of our results and those obtained from various numerical methods with the exact solution across different final times. In this test, the EFCGM adopts 50 nodes with $\Delta t = 10^{-3}$, the ENMCGM uses $h = 1/2000$ with $\Delta t = 10^{-7}$, while the RKNMGM employs $h = 1/500$ with $\Delta t = 10^{-6}$. The RKNMGM achieves similar accuracy to the EFCGM [20] while strictly enforcing the homogeneous BC, a challenge for the latter. Additionally, our method proves more accurate than the ENMCGM [40], even though the ENMCGM uses a finer spatial discretization and smaller time steps.

Table 2. Numerical results and relative errors ϵ of different methods for Problem 1 ($\nu = 0.01$).

T	x	Exact	EFCGM [20]	ϵ_1	Ref. [40]	ϵ_3	Present	ϵ
0.4	0.25	0.341915	0.341914	2.73×10^{-6}	0.342154	6.99×10^{-4}	0.341947	9.38×10^{-5}
	0.50	0.660711	0.660712	1.56×10^{-6}	0.660903	2.91×10^{-4}	0.661311	9.08×10^{-4}
	0.75	0.910265	0.910256	9.39×10^{-6}	0.910337	7.96×10^{-5}	0.911848	1.74×10^{-3}
0.6	0.25	0.268965	0.268958	2.55×10^{-5}	0.269190	8.37×10^{-4}	0.268971	2.30×10^{-5}
	0.50	0.529418	0.529426	1.46×10^{-5}	0.529653	4.43×10^{-4}	0.529732	5.92×10^{-4}
	0.75	0.767243	0.767247	4.85×10^{-6}	0.767409	2.16×10^{-4}	0.768302	1.38×10^{-3}
0.8	0.25	0.221482	0.221491	4.10×10^{-5}	0.221680	8.94×10^{-4}	0.221479	1.09×10^{-5}
	0.50	0.439138	0.439142	8.54×10^{-6}	0.439380	5.51×10^{-4}	0.439314	3.99×10^{-4}
	0.75	0.647395	0.647438	6.61×10^{-5}	0.647604	3.22×10^{-4}	0.648058	1.02×10^{-3}
1.0	0.25	0.188194	0.188193	5.11×10^{-6}	0.188367	9.19×10^{-4}	0.188189	2.81×10^{-5}
	0.50	0.374420	0.374423	7.91×10^{-6}	0.374653	6.22×10^{-4}	0.374524	2.79×10^{-4}
	0.75	0.556051	0.556051	5.31×10^{-7}	0.556273	4.00×10^{-4}	0.556476	7.64×10^{-4}

Problem 2. *For Equations (1)–(3) on* $(x,t) \in (0,1) \times (0,T]$ *with*

$$u_0(x) = \frac{2\nu\pi\sin(\pi x)}{a+\cos(\pi x)}, \quad x \in (0,1),\ a > 0, \tag{36}$$

Wood [3] has provided the exact solution

$$u(x,t) = \frac{2\nu\pi\sin(\pi x)e^{-\pi^2\nu t}}{a+\cos(\pi x)e^{-\pi^2\nu t}}, \quad a > 1. \tag{37}$$

Before performing Algorithm 1 of the RKNMGM for the numerical analysis on this problem, we use Equation (4) to transform Equation (1) into Equation (5). Taking the parameter $a = 2$, we use the H-C transform to turn the IC in Equation (36) into

$$\theta^{(0)}(x) = \frac{2+\cos(\pi x)}{3}, \quad \forall x \in (0,1), \tag{38}$$

and adapt the homogeneous BC in Equation (7) into the Neumann BC in Equation (35).

As shown in Figure 5, the numerical results of the RKNMGM for Problem 2 are compared against the analytical solutions when $T = 1.0$. Diversified viscosity coefficients are considered, i.e., $\nu = 0.1, 0.01, 0.001$, and smaller ν leads to a more smoothly changing curve. During the computation, the time step size and ME size are set as $\Delta t = 10^{-4}$ and $h = 1/200$, respectively. A close agreement with the exact solutions can be found from Figure 5. It is indicated that our method is effective and stable.

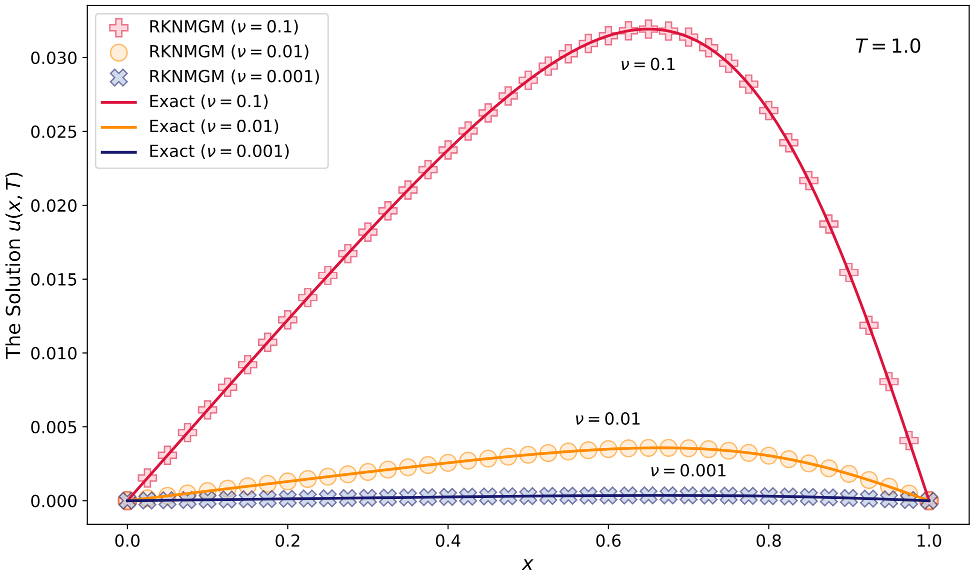

Figure 5. The results of the RKNMGM against the exact solutions for Problem 2 when $T = 1.0$.

Then, we evaluate the convergence performance of the RKNMGM for Problem 2 in the case of $T = 0.1$. In Table 3, the global errors of the RKNMGM are demonstrated for different ME numbers and viscosity coefficients. During the computations, set $\Delta t = 10^{-6}$, and diversified viscosity coefficients are adopted. It can be observed from the table that refining the dual cover mesh can give more accurate results for all of the cases. Besides, we can also find the phenomena that when the viscosity coefficient ν is reduced, the errors in both the L_2 and L_∞ norm decrease.

Next, we evaluate the results from the current method against those obtained from the ENMCGM [40] for Problem 2, focusing on two cases: $\nu = 0.01$ and $\nu = 0.001$. During the test, both methods utilize 2000 MEs. The RKNMGM employs a time step of $\Delta t = 10^{-6}$, while the ENMCGM uses $\Delta t = 10^{-7}$. Table 4 presents the numerical results and the relative errors with respect to the analytical solution in Equation (37). The comparison reveals that the present method demonstrates greater accuracy and stability for both $\nu = 0.01$ and $\nu = 0.001$. Further, the RKNMGM can use a larger time step size while achieving better computational performance.

Table 3. Convergence behavior of the RKNMGM using $\Delta t = 10^{-6}$ and various ME numbers for Problem 2 with $T = 0.1$.

N	$\nu = 0.01$		$\nu = 0.001$		$\nu = 0.0001$	
	$\|\cdot\|_2$	$\|\cdot\|_\infty$	$\|\cdot\|_2$	$\|\cdot\|_\infty$	$\|\cdot\|_2$	$\|\cdot\|_\infty$
50	1.62×10^{-5}	1.62×10^{-6}	1.63×10^{-7}	2.39×10^{-7}	1.63×10^{-7}	2.39×10^{-7}
100	4.04×10^{-6}	4.06×10^{-7}	4.06×10^{-8}	5.97×10^{-8}	4.06×10^{-8}	5.97×10^{-8}
200	1.01×10^{-6}	1.02×10^{-7}	1.02×10^{-8}	1.49×10^{-8}	1.02×10^{-8}	1.49×10^{-8}
400	2.53×10^{-7}	2.54×10^{-8}	2.54×10^{-9}	3.73×10^{-9}	2.54×10^{-9}	3.73×10^{-9}
800	6.31×10^{-8}	6.35×10^{-9}	6.35×10^{-10}	9.32×10^{-10}	6.35×10^{-10}	9.32×10^{-10}
1600	1.58×10^{-8}	1.59×10^{-9}	1.59×10^{-10}	2.33×10^{-10}	1.59×10^{-10}	2.33×10^{-10}

Furthermore, the accuracy of the RKNMGM is compared with higher-order methods, such as the fifth-order FVM (FVCW) [15] and the collocation methods proposed by Ganaie [13] and Mittal [12]. In this test, these three references adopt the parameters $\Delta t = 10^{-4}$ and $h = 1/40$, while we choose $\Delta t = 10^{-6}$ and $h = 1/1200$. Table 5 provides the exact solutions and numerical results on specific points. The data indicate that the RKNMGM demonstrates good accuracy, while it is slightly lower than that of the FVCW scheme. The latter benefits from the six-order Padé-based approximation. Our method outperforms the other two higher-order collocation methods in terms of accuracy. When using the same computing parameters as adopted in the other three methods, i.e., $\Delta t = 10^{-6}$ and $h = 1/140$, the relative error of our method is 1.03×10^{-4} at all of the measured points. Although higher-order approaches exhibit impressive results, they may also demonstrate increased computational complexity. In contrast, the RKNMGM is more simple and easy to implement owing to using constant cover functions and linear weight functions. These comparisons highlight the effectiveness and efficiency of the RKNMGM, showcasing its competitive performance among various methods using higher-order approximation.

From Tables 4 and 5, it is noteworthy that the present method exhibits almost uniform relative errors across different points, whereas the the other four methods do not possess this characteristic. This can be attributed to the time integration scheme adopted in this work, which maintains the TVD property, effectively suppressing numerical oscillations near sharp gradients.

Table 4. Computing results and their relative errors obtained from different numerical methods for Problem 2 when $T = 0.1$.

x	$\nu = 0.01$				$\nu = 0.001$			
	Ref. [40]	ϵ_1	**Present**	ϵ	**Ref. [40]**	ϵ_2	**Present**	ϵ
0.1	0.00653527	2.68×10^{-5}	0.00653544	4.13×10^{-7}	0.00065750	2.14×10^{-6}	0.00065750	4.11×10^{-7}
0.2	0.01305544	8.02×10^{-6}	0.01305533	4.13×10^{-7}	0.00131383	4.91×10^{-7}	0.00131383	4.11×10^{-7}
0.3	0.01949374	5.35×10^{-6}	0.01949363	4.13×10^{-7}	0.00196281	6.73×10^{-7}	0.00196281	4.11×10^{-7}
0.4	0.02565934	3.54×10^{-6}	0.02565924	4.13×10^{-7}	0.00258576	2.41×10^{-7}	0.00258576	4.11×10^{-7}
0.5	0.03110745	1.97×10^{-6}	0.03110738	4.13×10^{-7}	0.00313849	1.42×10^{-7}	0.00313849	4.11×10^{-7}
0.6	0.03492868	6.54×10^{-7}	0.03492864	4.14×10^{-7}	0.00352972	5.91×10^{-8}	0.00352972	4.11×10^{-7}
0.7	0.03549597	5.27×10^{-7}	0.03549594	4.14×10^{-7}	0.00359443	1.13×10^{-7}	0.00359443	4.12×10^{-7}
0.8	0.03050154	6.40×10^{-6}	0.03050133	4.15×10^{-7}	0.00309581	6.95×10^{-7}	0.00309580	4.12×10^{-7}
0.9	0.01816643	9.56×10^{-6}	0.01816660	4.15×10^{-7}	0.00184754	3.56×10^{-6}	0.00184754	4.12×10^{-7}

Table 5. The results and relative errors of different numerical approaches for Problem 2 ($\nu = 0.1$) when $T = 0.001$.

x	**Exact**	**FVCW [15]**	ϵ	**Ganaie [13]**	ϵ	**Mittal [12]**	ϵ	**Present**	ϵ
0.1	0.06574976	0.06574976	2.78×10^{-10}	0.065750	3.66×10^{-6}	0.065750	3.66×10^{-6}	0.06574973	4.11×10^{-7}
0.2	0.13138294	0.13138294	8.94×10^{-11}	0.131383	4.91×10^{-7}	0.131383	4.91×10^{-7}	0.13138288	4.11×10^{-7}
0.3	0.19628087	0.19628087	1.37×10^{-10}	0.196281	6.73×10^{-7}	0.196281	6.73×10^{-7}	0.19628079	4.11×10^{-7}
0.4	0.25857574	0.25857574	1.21×10^{-9}	0.258576	1.01×10^{-6}	0.258576	1.01×10^{-6}	0.25857563	4.11×10^{-7}
0.5	0.31384936	0.31384935	$1.07{\times}10^{-8}$	0.313848	4.32×10^{-6}	0.313850	2.05×10^{-6}	0.31384923	4.11×10^{-7}
0.6	0.35297182	0.35297179	1.00×10^{-7}	0.352972	5.07×10^{-7}	0.352972	5.07×10^{-7}	0.35297168	4.11×10^{-7}
0.7	0.35944286	0.35944291	1.33×10^{-7}	0.359443	3.91×10^{-7}	0.359443	3.91×10^{-7}	0.35944271	4.12×10^{-7}
0.8	0.30958038	0.30958039	9.47×10^{-9}	0.309580	1.24×10^{-6}	0.309579	4.47×10^{-6}	0.30958026	4.12×10^{-7}
0.9	0.18475374	0.18475374	3.45×10^{-8}	0.184752	9.43×10^{-6}	0.184751	1.48×10^{-5}	0.18475367	4.12×10^{-7}

Problem 3. *Next, consider Equations (1)–(3) on $(x, t) \in (0, 1) \times (0, T]$ with*

$$u_0(x) = 4x(1-x), \quad \forall x \in (0,1). \tag{39}$$

The exact solution of this problem is the same as Equation (32), but the coefficients A_0 and A_m take the following formulation [4]:

$$A_0 = \int_0^1 f(s)ds, \quad A_m = 2\int_0^1 f(s)\cos(m\pi s)ds, \quad m = 1, 2, 3, \ldots. \tag{40}$$

where $f(s) = \exp\left[(3\nu)^{-1}s^2(2s-3)\right]$. Using Equation (4), Equation (1) is transformed into Equation (5), the IC in Equation (36) becomes $\theta^{(0)}(x) = f(x)$, $\forall x \in (0,1)$, and the BC in Equation (7) turns to be Equation (35).

For this example, we first assess the convergence performance of the RKNMGM using the time step size $\Delta t = 10^{-5}$ and various ME sizes—specifically, $h = 1/40$, $1/80$, $1/120$, $1/160$, and $1/200$. Here, we use a viscosity coefficient of $\nu = 0.1$ and set the final time as $T = 0.1$. As illustrated in Figure 6, the global errors are presented in both L_2 and L_∞ norms. The figure clearly demonstrates that our method converges as the discretization is refined.

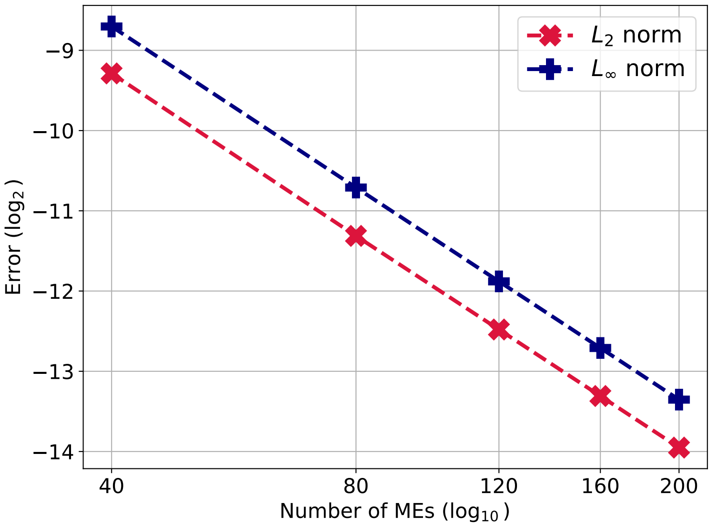

Figure 6. Convergence study of the RKNMGM using $\Delta t = 10^{-5}$ and different ME numbers for Problem 3 in the case of $\nu = 0.1$ and $T = 0.1$.

As shown in Figures 7 and 8, the results of the RKNMGM align well with the exact solutions, corresponding to the diffusive coefficients $\nu = 0.1$ and $\nu = 0.01$. In these computations, we adopt $\Delta t = 10^{-5}$ and $h = 1/200$. Notably, these curves exhibit similarities to those observed in Problem 1 owing to their similar exact solution formulations.

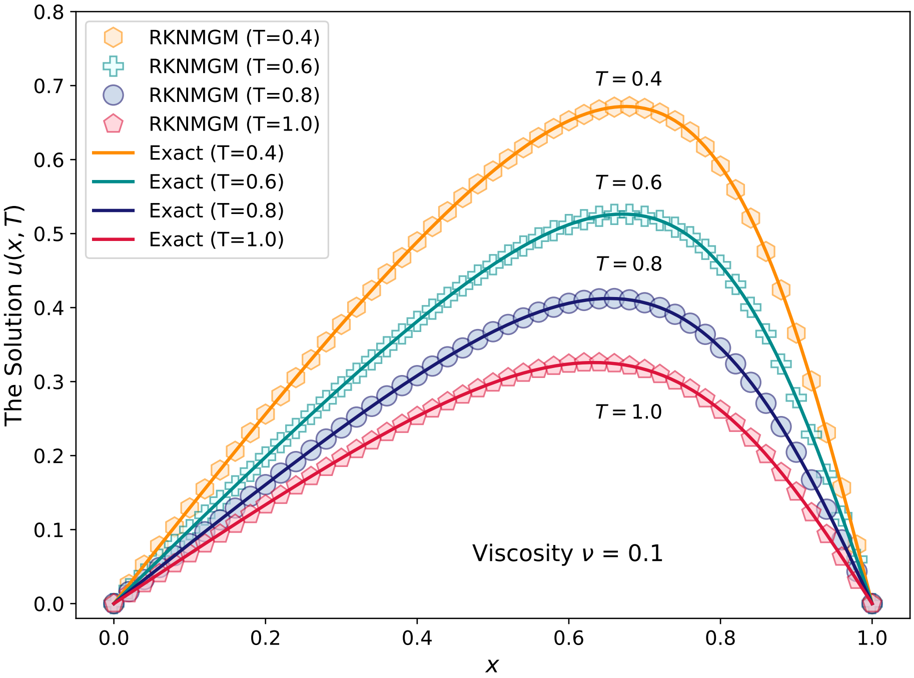

Figure 7. RKNMGM results using $\Delta t = 10^{-5}$, $h = 1/200$ for Problem 3 with $\nu = 0.1$.

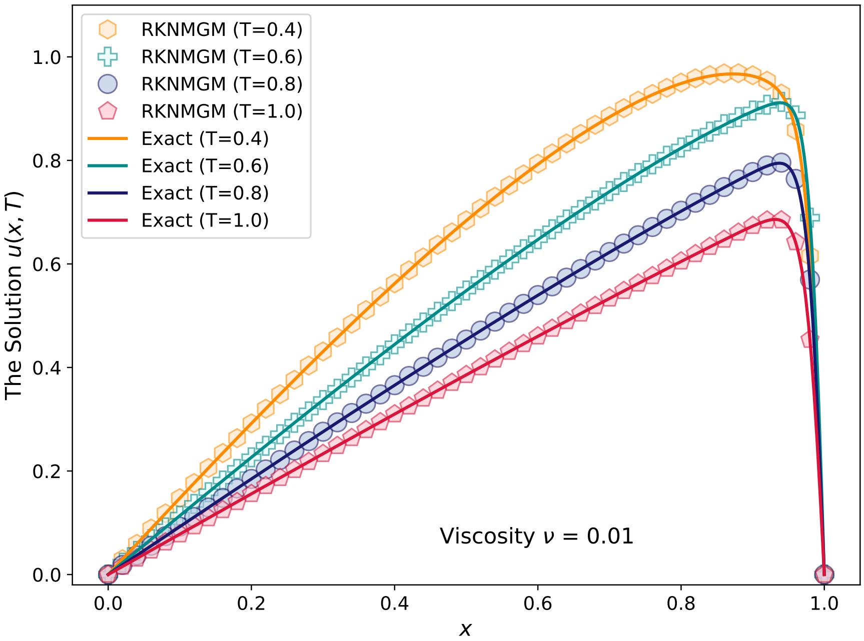

Figure 8. RKNMGM results using $\Delta t = 10^{-5}$, $h = 1/200$ for Problem 3 with $\nu = 0.01$.

We further assess the accuracy of the RKNMGM at specific points for this example. For this test, the profile of the problem includes $\nu = 0.1$ and $T = 0.4, 0.6, 0.8, 1.0$. Table 6 presents a comparison of the numerical results obtained from our method with those solved by the fourth-order FDM [4] and the ENMCGM [40]. In the tests, the FDM uses $\Delta t = 10^{-3}$ with $h = 1/80$, the ENMCGM employs $\Delta t = 10^{-7}$ with $h = 1/2000$, and we adopt $\Delta t = 10^{-6}$ with $h = 1/500$. It is evident that the RKNMGM can provide higher accuracy than the other two methods. Although our approach requires a finer mesh than the fourth-order FDM, it utilizes only constant cover functions and linear approximations, significantly reducing computational cost. Moreover, compared to the ENMCGM, which uses 2000 MEs, our method achieves greater accuracy with just 500 MEs, demonstrating the remarkable efficiency of the current approach.

Table 6. Comparison of the results and relative errors obtained from different numerical methods for Problem 3 ($\nu = 0.1$).

T	x	Exact	Hassanien [4]	ϵ_1	Sun [40]	ϵ_2	Present	ϵ
0.4	0.25	0.3175229	0.31752	9.07×10^{-6}	0.3175292	1.99×10^{-5}	0.3175169	1.89×10^{-5}
	0.5	0.5845373	0.58454	4.69×10^{-6}	0.5846054	1.17×10^{-4}	0.5845327	7.86×10^{-6}
	0.75	0.6456155	0.64562	6.96×10^{-6}	0.6456419	4.09×10^{-5}	0.6456196	6.41×10^{-6}
0.6	0.25	0.2461385	0.24614	6.27×10^{-6}	0.2461348	1.49×10^{-5}	0.2461345	1.60×10^{-5}
	0.5	0.4579764	0.45798	7.85×10^{-6}	0.4580298	1.17×10^{-4}	0.4579721	9.37×10^{-6}
	0.75	0.5026758	0.50268	8.45×10^{-6}	0.5026707	1.00×10^{-5}	0.5026757	1.87×10^{-7}
0.8	0.25	0.1995553	0.19956	2.35×10^{-5}	0.1995441	5.62×10^{-5}	0.1995525	1.39×10^{-5}
	0.5	0.3673982	0.3674	4.92×10^{-6}	0.3674304	8.77×10^{-5}	0.3673946	9.88×10^{-6}
	0.75	0.3853355	0.38534	1.16×10^{-5}	0.385303	8.44×10^{-5}	0.3853338	4.54×10^{-6}
1.0	0.25	0.1655986	0.1656	8.26×10^{-6}	0.1655804	1.10×10^{-4}	0.1655965	1.27×10^{-5}
	0.5	0.2983431	0.29834	1.04×10^{-5}	0.2983543	3.75×10^{-5}	0.2983400	1.03×10^{-5}
	0.75	0.2958567	0.29586	1.12×10^{-5}	0.2958057	1.72×10^{-4}	0.2958545	7.45×10^{-6}

Problem 4. *There exist the following specific solutions [8] for the problem presented in Equations (1)–(3)*

$$u(x,t) = \frac{x/t}{1 + \sqrt{\alpha t}\exp(\beta x^2/t)}, \quad (x,t) \in (0, 1.2) \times [1, \infty), \tag{41}$$

where $\alpha = \exp(-1/8\nu)$, $\beta = 1/4\nu$.

Consider the Burgers' equation on $(x,t) \in (0, 1.2) \times (1, T]$, with Equation (42) at $t = 1.0$ as the IC, i.e.,

$$u_0(x) = \frac{x}{1 + \sqrt{\alpha}\exp(\beta x^2)}, \quad x \in (0, 1.2). \tag{42}$$

In this case, the BC in Equation (3) becomes

$$u(0,t) = u(1.2,t) = 0, \forall t \in [0, \infty). \tag{43}$$

Equation (41) provides the exact solution for this problem.

To numerically solve this problem with the RKNMGM, we use Equation (4) to convert Equation (1) into Equation (5) and transform the IC and BC into the following forms:

$$\theta^{(0)}(x) = \frac{\exp(-\beta x^2) + \sqrt{\alpha}}{1 + \sqrt{\alpha}}, \quad \forall x \in (0, 1.2) \tag{44}$$

$$\theta_x\big|_{x=0} = 0 = \theta_x\big|_{x=1.2}, \quad \forall t \in (1, T]. \tag{45}$$

Figure 9 shows the convergence performance of the RKNMGM for Problem 4 with $\nu = 0.005$. Various spatial discretizations are considered, including cases with 50, 100, 200, 400, 800, and 1600 MEs. A consistent time step of $\Delta t = 10^{-5}$ was used throughout the computations. As depicted in the figure, the global errors of the RKNMGM decrease with increasing ME numbers, demonstrating clear convergence.

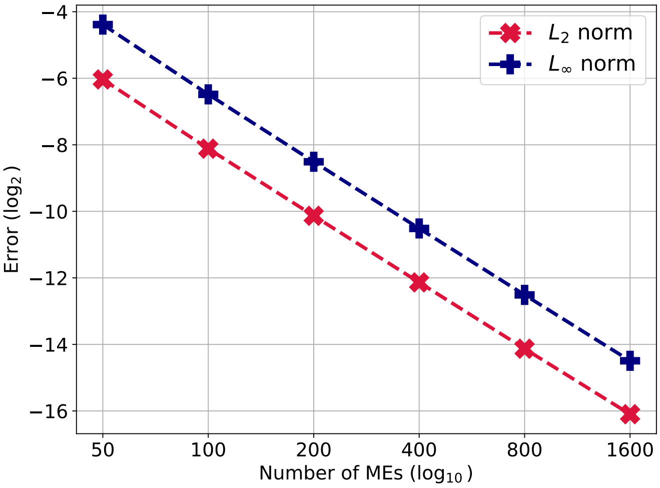

Figure 9. The convergence behavior of the RKNMGM using $\Delta t = 10^{-5}$ and various spatial discretizations for Problem 4 in the case of $\nu = 0.005$ and $T = 1.5$.

Table 7 compares the accuracy of the RKNMGM with other numerical methods, including the FEM [8] and our previously proposed ENMCGM scheme [40]. The ENMCGM utilizes 2000 MEs and employs $\Delta t = 10^{-7}$ for both of the cases $\nu = 0.005$ and $\nu = 0.001$, while the present method utilizes $\Delta t = 10^{-5}$ and uses 400 MEs and 2000 MEs for $\nu = 0.005$ and $\nu = 0.001$, respectively. Here, the computing settings used in the FEM have not been given by the author. The accuracy is evaluated in terms of global errors in both L_2 and L_∞ norms for two cases of viscosity, $\nu = 0.005$ and $\nu = 0.001$, with results presented for three different final times T. The RKNMGM demonstrates superior accuracy compared to the other two methods. This comparison shows the significant advantage of the RKNMGM in computational efficiency.

Table 7. The global errors of the results computed by the RKNMGM compared with other numerical methods for Problem 4 with diversified cases of ν and T.

ν	T	FEM [8]		ENMCGM [40]		Present	
		$\|\|\cdot\|\|_2$	$\|\|\cdot\|\|_\infty$	$\|\|\cdot\|\|_2$	$\|\|\cdot\|\|_\infty$	$\|\|\cdot\|\|_2$	$\|\|\cdot\|\|_\infty$
0.005	1.7	2.107×10^{-3}	8.099×10^{-3}	5.7479×10^{-4}	2.4558×10^{-3}	1.9582×10^{-4}	5.8596×10^{-4}
	2.4	3.345×10^{-3}	1.165×10^{-2}	7.8821×10^{-4}	3.0273×10^{-3}	1.3592×10^{-4}	3.7217×10^{-4}
	3.1	4.820×10^{-3}	1.587×10^{-2}	8.9660×10^{-4}	3.2024×10^{-3}	1.0157×10^{-4}	2.5996×10^{-4}
0.001	1.7	4.123×10^{-3}	3.6675×10^{-2}	7.2690×10^{-3}	6.7517×10^{-2}	3.5527×10^{-4}	2.0779×10^{-3}
	2.4	1.432×10^{-3}	1.0812×10^{-2}	1.0116×10^{-2}	8.5248×10^{-2}	2.8907×10^{-4}	1.8220×10^{-3}
	3.1	5.761×10^{-3}	4.0855×10^{-2}	1.1546×10^{-2}	9.0540×10^{-2}	2.3257×10^{-4}	1.4406×10^{-3}

Figure 10 displays our results for Problem 4 with a viscosity coefficient of $\nu = 0.01$, compared to the exact solution. We use a time step size of $\Delta t = 10^{-5}$ and 200 MEs. As the final time T varies from 1.0 to 3.0, the curve becomes progressively smoother. The figure clearly demonstrates a close correspondence between our results and the exact solutions.

Additionally, Figure 11 presents our findings alongside the exact solution for this example with $\nu = 0.001$. In this case, we maintain a time step of $\Delta t = 10^{-5}$ while refining the spatial discretization to 2000 MEs. Noticeable steep gradients are evident in the curves for $T = 1.0, 1.5, 2.0, 2.5$, and 3.0. Overall, it is evident that the results from the RKNMGM closely match the exact solutions.

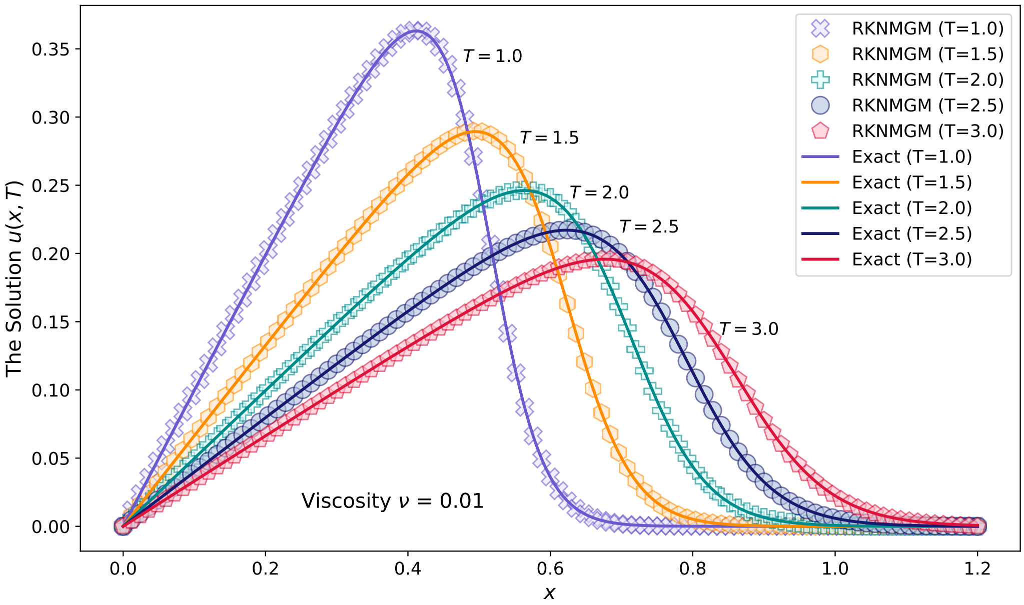

Figure 10. RKNMGM results using $\Delta t = 10^{-5}$ and 200 MEs for Problem 4 with $\nu = 0.01$ and different T against the exact solutions.

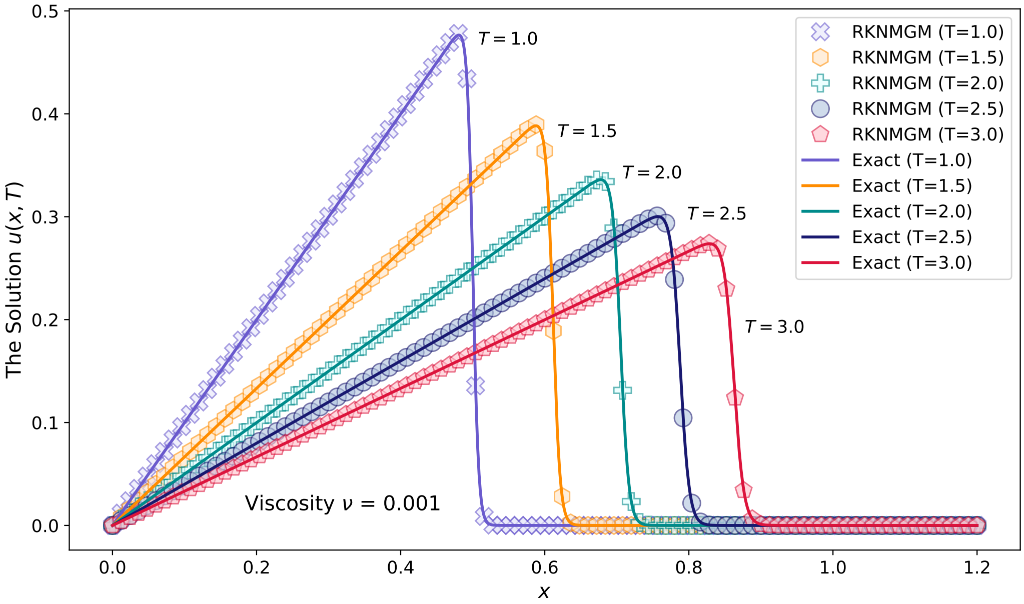

Figure 11. RKNMGM results using $\Delta t = 10^{-5}$ and 2000 MEs for Problem 4 with $\nu = 0.001$.

6. Conclusions

The problem of the Burgers' equation, as depicted in Equations (1)–(3), acts as a crucial framework for analyzing intricate nonlinear behaviors in fluid dynamics. In this paper, we proposed a Runge–Kutta numerical manifold Galerkin method (RKNMGM) specifically designed to solve this problem. This approach utilizes a dual cover mesh and PU-like weighted functions over manifold elements to enhance accuracy. To address the nonlinear advective term, we employed the H-C transform, shifting the original problem into a more manageable linear PDE. We conducted several benchmark numerical tests to evaluate the performance of the RKNMGM. Based on the numerical results, several notable advantages of the RKNMGM are worth highlighting, as follows:

(1) The H-C transform simplifies the Burgers' equation, significantly reducing the complexity of solving the original equation while preserving essential physical characteristics. The solution to the original problem can be conveniently recovered from that of the transformed problem using the H-C transform once again.

(2) By adopting the dual cover approximation with constant cover functions and linear weight functions, computational costs are reduced while Dirichlet boundary condition can be accurately imposed. Additionally, integrals over manifold elements are computed using simplex integration formulas instead of Gaussian quadrature, further enhancing accuracy and efficiency.

(3) We introduced the fully explicit TVD-RK2 scheme for temporal discretization in the NMM. Although both the TVD-RK2 and the Crank–Nicolson method have second-order accuracy, the TVD property possessed in the former one brings our present method an advantage in suppressing numerical oscillations, as evidenced in the numerical experiments.

(4) The final computational formulation, derived from the Galerkin weak form, results in a symmetric mass matrix, whose inverse can be efficiently obtained using the Crout method. This facilitates implementation and parallelization of the RKNMGM.

(5) Our numerical results for the Burgers' equation under diversified initial condition values were thoroughly compared with exact solutions and other numerical methods. These comparisons demonstrate that our method effectively captures steep gradients, even for very small viscosity coefficients.

It is important to acknowledge that this study concentrates exclusively on the one-dimensional Burgers' equation. In our future research, we look forward to extending our exploration to high-dimensional cases, which will broaden the applicability of the findings presented here. This investigation will enable us to address more complex PDEs that are significant in the realm of fluid dynamics.

Author Contributions: Conceptualization and Methodology, Y.S.; Software and Original draft preparation, Q.C.; Validation and Data Analysis, T.C.; Visualization, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of Shaanxi Province of China (Grant No. 2024JC-YBMS-014) and the Research Foundation for the Doctoral Program of Shaanxi University of Technology (Grant No. SLGQD2017-08).

Data Availability Statement: Profiles presented in the study are included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Burgers, J.M. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.* **1948**, *1*, 171–199.
2. Cole, J.D. On a quasi-linear parabolic equation occurring in aerodynamics. *Quart. Appl. Math.* **1951**, *9*, 225–236. [CrossRef]
3. Wood, W. An exact solution for Burger's equation. *Commun. Numer. Meth. Eng.* **2006**, *22*, 797–798. [CrossRef]
4. Hassanien, I.; Salama, A.; Hosham, H.A. Fourth-order finite difference method for solving Burgers' equation. *Appl. Math. Comput.* **2005**, *170*, 781–800. [CrossRef]
5. Yağmurlu, M.; Gagir, A. A finite difference approximation for numerical simulation of 2D viscous coupled burgers equations. *Math. Sci. Appl. E-Notes* **2022**, *10*, 146–158. [CrossRef]
6. Kaur, R.; Kukreja, V.; Parumasur, N.; Singh, P. Two different temporal domain integration schemes combined with compact finite difference method to solve modified Burgers' equation. *Ain Shams Eng. J.* **2022**, *13*, 101507. [CrossRef]
7. Vandandoo, U.; Zhanlav, T.; Chuluunbaatar, O.; Gusev, A.; Vinitsky, S.; Chuluunbaatar, G. Higher-Order Finite-Difference Methods for the Burgers' Equations. In *High-Order Finite Difference and Finite Element Methods for Solving Some Partial Differential Equations*; Springer Nature: Cham, Switzerland, 2024; pp. 35–67.
8. Dogan, A. A Galerkin finite element approach to Burgers' equation. *Appl. Math. Comput.* **2004**, *157*, 331–346. [CrossRef]
9. Chen, H.; Jiang, Z. A characteristics-mixed finite element method for Burgers' equation. *J. Appl. Math. Comput.* **2004**, *15*, 29–51. [CrossRef]
10. Saka, B.; Dağ, İ. A numerical study of the Burgers' equation. *J. Franklin Inst.* **2008**, *345*, 328–348. [CrossRef]
11. Dağ, İ.; Irk, D.; Saka, B. A numerical solution of the Burgers' equation using cubic B-splines. *Appl. Math. Comput.* **2005**, *163*, 199–211. [CrossRef]
12. Mittal, R.; Jain, R. Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. *Appl. Math. Comput.* **2012**, *218*, 7839–7855. [CrossRef]

13. Ganaie, I.A.; Kukreja, V.K. Numerical solution of Burgers' equation by cubic Hermite collocation method. *Appl. Math. Comput.* **2014**, *237*, 571–581. [CrossRef]
14. Arar, N.; Deghdough, B.; Dekkiche, S.; Torch, Z.; Nagy, A. Numerical Solution of the Burgers' Equation Using Chelyshkov Polynomials. *Int. J. Appl. Comput. Math.* **2024**, *10*, 33. [CrossRef]
15. Guo, Y.; Shi, Y.f.; Li, Y.m. A fifth-order finite volume weighted compact scheme for solving one-dimensional Burgers' equation. *Appl. Math. Comput.* **2016**, *281*, 172–185. [CrossRef]
16. Shahabi, A.; Ghiassi, R. A robust second-order godunov-type method for Burgers' equation. *Int. J. Appl. Comput. Math.* **2022**, *8*, 82. [CrossRef]
17. Xin, J.; Flaherty, J. Implicit time integration of hyperbolic conservation laws via discontinuous Galerkin methods. *Int. J. Numer. Methods Biomed. Eng.* **2011**, *27*, 711–721. [CrossRef]
18. Zhang, R.P.; Yu, X.J.; Zhao, G.Z. Modified Burgers' equation by the local discontinuous Galerkin method. *Chin. Phys. B* **2013**, *22*, 030210.
19. Hutridurga, H.; Kumar, K.; Pani, A.K. Discontinuous Galerkin methods with generalized numerical fluxes for the Vlasov-viscous Burgers' system. *J. Sci. Comput.* **2023**, *96*, 7.
20. Zhang, X.H.; Ouyang, J.; Zhang, L. Element-free characteristic Galerkin method for Burgers' equation. *Eng. Anal. Boundary Elem.* **2009**, *33*, 356–362. [CrossRef]
21. Zhang, L.; Ouyang, J.; Wang, X.; Zhang, X. Variational multiscale element-free Galerkin method for 2D Burgers' equation. *J. Comput. Phys.* **2010**, *229*, 7147–7161. [CrossRef]
22. Ma, L.; Zhao, L.; Wang, X. An iteration-free semi-Lagrangian meshless method for Burgers' equations. *Eng. Anal. Boundary Elem.* **2023**, *150*, 482–491. [CrossRef]
23. Hashemian, A.; Shodja, H.M. A meshless approach for solution of Burgers' equation. *J. Comput. Appl. Math.* **2008**, *220*, 226–239. [CrossRef]
24. Xie, H.; Li, D. A meshless method for Burgers' equation using MQ-RBF and high-order temporal approximation. *Appl. Math. Modell.* **2013**, *37*, 9215–9222. [CrossRef]
25. Shyaman, V.; Sreelakshmi, A.; Awasthi, A. A higher order implicit adaptive finite point method for the Burgers' equation. *J. Differ. Equ. Appl.* **2021**, *29*, 235–269. [CrossRef]
26. Shi, G.H. Manifold method of material analysis. In Proceedings of the Transactions of the 9th Army Conference on Applied Mathematics and Computing, US Army Research Office Minneapolis,Minneapolis, MN, USA, 21–25 October 1991; Volume 92.
27. Lin, J.S. A mesh-based partition of unity method for discontinuity modeling. *Comput. Methods Appl. Mech. Eng.* **2003**, *192*, 1515–1532. [CrossRef]
28. Zheng, H.; Liu, Z.; Ge, X. Numerical manifold space of Hermitian form and application to Kirchhoff's thin plate problems. *Int. J. Numer. Meth. Eng.* **2013**, *95*, 721–739. [CrossRef]
29. Zheng, H.; Wang, F. The numerical manifold method for exterior problems. *Comput. Methods Appl. Mech. Eng.* **2020**, *364*, 112968. [CrossRef]
30. Guo, H.; Cao, X.; Liang, Z.; Lin, S.; Zheng, H.; Cui, H. Hermitian numerical manifold method for large deflection of irregular Föppl-von Kármán plates. *Eng. Anal. Bound. Elem.* **2023**, *153*, 25–38. [CrossRef]
31. Tan, F.; Tong, D.; Liang, J.; Yi, X.; Jiao, Y.Y.; Lv, J. Two-dimensional numerical manifold method for heat conduction problems. *Eng. Anal. Bound. Elem.* **2022**, *137*, 119–138. [CrossRef]
32. Ji, X.; Zhang, H.; Han, S. Transient heat conduction modeling in continuous and discontinuous anisotropic materials with the numerical manifold method. *Eng. Anal. Bound. Elem.* **2023**, *155*, 518–527. [CrossRef]
33. Ning, Y.; Lu, Q.; Liu, X. Fracturing failure simulations of rock discs with pre-existing cracks by numerical manifold method. *Eng. Anal. Bound. Elem.* **2023**, *148*, 389–400. [CrossRef]
34. Zhang, N.; Zheng, H.; Li, X.; Wu, W. On hp refinements of independent cover numerical manifold method—some strategies and observations. *Sci. China Technol. Sci.* **2023**, *66*, 1335–1351. [CrossRef]
35. Zhang, L.; Kong, H.; Zheng, H. Numerical manifold method for steady-state nonlinear heat conduction using Kirchhoff transformation. *Sci. China Technol. Sci.* **2024**, *67*, 992–1006. [CrossRef]
36. Tong, D.; Yi, X.; Tan, F.; Jiao, Y.; Liang, J. Three-dimensional numerical manifold method for heat conduction problems with a simplex integral on the boundary. *Sci. China Technol. Sci.* **2024**, *67*, 1007–1022. [CrossRef]
37. Zhang, Z.; Zhang, X.; Yan, J. Manifold method coupled velocity and pressure for Navier–Stokes equations and direct numerical solution of unsteady incompressible viscous flow. *Comput. Fluids* **2010**, *39*, 1353–1365. [CrossRef]
38. Lin, S.; Cao, X.; Zheng, H.; Li, Y.; Li, W. An improved meshless numerical manifold method for simulating complex boundary seepage problems. *Comput. Geotech.* **2023**, *155*, 105211. [CrossRef]
39. Su, H.; Lin, S.; Xie, Z.; Gong, Y.; Qi, Y. Fundamentals and progress of the manifold method based on independent covers. *Sci. China Technol. Sci.* **2024**, *67*, 966–991. [CrossRef]
40. Sun, Y.; Chen, Q.; Chen, T.; Yong, L. Explicit Numerical Manifold Characteristic Galerkin Method for Solving Burgers' Equation. *Axioms* **2024**, *13*, 343. [CrossRef]
41. Hopf, E. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Commun. Pure Appl. Math.* **1950**, *3*, 201–230. [CrossRef]
42. Fahmy, E.; Raslan, K.R.; Abdusalam, H. On the exact and numerical solution of the time-delayed Burgers equation. *Int. J. Comput. Math.* **2008**, *85*, 1637–1648. [CrossRef]

43. Kudryashov, N.A.; Sinelshchikov, D.I. Exact solutions of equations for the Burgers hierarchy. *Appl. Math. Comput.* **2009**, *215*, 1293–1300. [CrossRef]
44. Ohwada, T. Cole-Hopf transformation as numerical tool for the Burgers equation. *Appl. Comput. Math* **2009**, *8*, 107–113.
45. Cordero, A.; Franques, A.; Torregrosa, J.R. Numerical solution of turbulence problems by solving Burgers' equation. *Algorithms* **2015**, *8*, 224–233. [CrossRef]
46. Uddin, M.; Ali, H. The space–time kernel-based numerical method for Burgers' equations. *Mathematics* **2018**, *6*, 212. [CrossRef]
47. Al-shimmary, A.F.; Kareem, S.; Hussain, A.K. A new approach to solve Burges' equation using Runge-Kutta 6th order method based on Cole-Hopf transformation. *J. Eng. Appl. Sci.* **2020**, *15*, 2362–2369.
48. Abd-Elhameed, W.M. Novel expressions for the derivatives of sixth kind Chebyshev polynomials: Spectral solution of the non-linear one-dimensional Burgers' equation. *Fractal Fract.* **2021**, *5*, 53. [CrossRef]
49. Savović, S.; Ivanović, M.; Min, R. A Comparative Study of the Explicit Finite Difference Method and Physics-Informed Neural Networks for Solving the Burgers' Equation. *Axioms* **2023**, *12*, 982. [CrossRef]
50. Singh, G.; Singh, I.; AlDerea, A.M.; Alanzi, A.M.; Khalifa, H.A.E.W. Solutions of (2+1)-D & (3+1)-D Burgers Equations by New Laplace Variational Iteration Technique. *Axioms* **2023**, *12*, 647. [CrossRef]
51. Öziş, T.; Aksan, E.; Özdeş, A. A finite element approach for solution of Burgers' equation. *Appl. Math. Comput.* **2003**, *139*, 417–428. [CrossRef]
52. Guo-Zhong, Z.; Xi-Jun, Y.; Di, W. Numerical solution of the Burgers' equation by local discontinuous Galerkin method. *Appl. Math. Comput.* **2010**, *216*, 3671–3679. [CrossRef]
53. Mukundan, V.; Awasthi, A. Efficient numerical techniques for Burgers' equation. *Appl. Math. Comput.* **2015**, *262*, 282–297. [CrossRef]
54. Shu, C.W. Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.* **1988**, *9*, 1073–1084. [CrossRef]
55. Shi, G.H. Modeling dynamic rock failure by discontinuous deformation analysis with simplex integrations. In Proceedings of the ARMA North America Rock Mechanics Symposium, Austin, TX, USA, 1–3 June 1994.
56. Lin, S.; Xie, Z. A new recursive formula for integration of polynomial over simplex. *Appl. Math. Comput.* **2020**, *376*, 125140. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.