Utilization Finite Element Methods for Solving the Inverse Problem of Viscous Burgers' Equation

Hassar. H. Mohameed[1], Prof. Dr. Ghassan Ezzuldin Arif[2], Dr. Mohammed Azeez Hilal[3]

1,2 Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Iraq.

3 Baquba Technical Institute, Middle Technical University, Iraq.

[1] hhisan.1989@gmail.com, [2] ghasanarif@tu.edu.iq, [3] mohammed_azeez_hilal@mtu.edu.iq.

Utilization Finite Element Methods for Solving the Inverse Problem of Viscous Burgers' Equation

Hassan H. Mohamood[1], Prof.Dr.Ghassan Ezzuldin Arif[2], Dr. Mohammed Azeez Hilal[3]

1,2 Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Iraq.

3 Baquba Technical Institute, Middle Technical University, Iraq.

[1] hhisan.1989@gmail.com, [2] ghasanarif@tu.edu.iq, [3] mohammed_azeez_hilal@mtu.edu.iq.

Abstract

In this paper, we utilize a combination of the finite element method and the finite difference method to solve the one-dimensional nonlinear Burgers equation. The discretization in time was applied to construct the method, and for each time step, a system of nonlinear equations was obtained. To solve these equations, the inverse Cole-Hopf transformation was employed. The efficiency of the proposed method was demonstrated by comparing the numerical solutions obtained for different viscosity values with the exact solutions. The results showed a remarkable agreement between them.

Keywords: Inverse problem, Burgers' equation, Finite element methods, Cole-Hopf transformation.

1. *Introduction*:

In this paper we will consider the one-dimensional viscid Burgers equation to be the nonlinear parabolic partial differential equation (*PDE*)

$$u_t + uu_x - \mu u_{xx} = 0 \qquad (1.1)$$

where $\mu > 0$ is the kinematic viscosity. This is the simplest *PDE* combining both nonlinear propagation effects and diffusive effects. When the term μu_{xx} is removed from (1.1) we obtain that called the inviscid Burgers equation which is a hyperbolic PDE.

Bateman introduced this equation in 1915 [1]. Bateman utilized this equation as a representation of the movement of a viscous fluid in situations where the viscosity approaches zero. He derived two distinct types of steady state solutions for the problem of an infinite domain. More than thirty years later, Johannes M. Burgers [3] introduced the equation (1) aiming to create a straightforward mathematical model that could capture the essential characteristics observed in turbulent hydrodynamic flows. Consequently, the equation became known as the Burgers equation due to its association with Burgers' efforts . Despite advancements in concepts and techniques for solving nonlinear partial differential

equations, obtaining Analytical Solutions for Diffuseness Problems remains challenging. Therefore, Numerical Solutions are often considered the most effective approach for studying the characteristics of these equations [11]. Therefore, it is essential to examine the characteristics of the Burger's equation in order to understand the fundamental equations of fluid mechanics.

Over the past few decades, numerous researchers have been interested in obtaining solutions for the Burger's equation through various analytical and numerical methods. These methods include the Cole-Hopf transformation introduced by Cole in 1951[5], Fletcher's in 1983 [6], the finite element method implemented by Cecchi and others in 1996[4], and the explicit finite difference method used by Kutluay and others in 1999[9], Ozis and others in 2003[13], the finite element method using various forms of cubic spline-B functions by Kutluay and others in 2004[38], and the finite difference method implemented by Hassanien and others in 2005[7], the implicit finite difference method applied by Kadalbajoo and others in 2005[8], Liao in 2008[12], Jiang and Wang in 2010 [19], the implicit logarithmic finite difference method developed by Srivastava and others in 2013[15], Bhrawy and Others in 2015 [2], and Tamsir and others in 2016[16].

This study utilized a combination of the finite element method and the finite difference method, which were applied to discretize the nonlinear Burgers equation in time with the inverse problem situation. The inverse Cole-Hopf transformation was employed to solve the equation. The obtained numerical results were compared to Cole's exact solution for $\mu = .1$ of viscosity value based on H^1 *and* L^2 errors. The comparison revealed that the numerical results were highly satisfactory.

2. *Problem Statement*:

This study will focus on considering the viscous one-dimensional Burgers equation, which is given as follows:

$$\begin{cases} u_t + uu_x - \mu u_{xx} = 0\,, & x \in \Omega,\ t \in (t_0, t_f] \\ \mathcal{R}u = g, & x \in \partial\Omega, \quad t \in (t_0, t_f] \\ u = u_0(x), & x \in \bar{\Omega}, \quad t = t_0 \end{cases} \qquad (2.1)$$

Where x belongs to the spatial domain Ω, t belongs to the time interval $(t_0, t_f]$ and $\mu = 1/R$ s.t R is the Reynold number. The equation is subject to a boundary condition $\mathcal{R}u = g$ on the boundary $\partial\Omega$ and an initial condition $u = u_0(x)$ at time $t = t_0$.

2.1 *Cole-Hopf Transformation* [5]:

An Important work was done by Cole and Hopf by finding a substitution for Burgers' equation that converts (2.1) into the linear heat equation. This transformation, known as the Cole-Hopf transformation, is expressed as follows:

$$u(x,t) = -2\mu \frac{\beta_x(x,t)}{\beta(x,t)} \qquad (2.2)$$

By applying the Cole-Hopf transformation, where $\beta(x,t)$ is a non-zero function, Burgers' equation can be transformed into the simplified form:

$$\frac{\partial \beta}{\partial t} = \mu \frac{\partial^2 \beta}{\partial x^2}$$

Nevertheless, this transformation introduces complexities in defining the boundary conditions and initial conditions, as we will soon observe. However, despite these challenges, the Cole-Hopf transformation enables us to obtain exact solutions for the one-dimensional Burgers' equation in various general scenarios.

2.2 *Exact Solution* [18]:

In this case, let's specifically consider homogeneous Dirichlet-boundary conditions as follows:

$$\begin{cases} u_t + uu_x - \mu u_{xx} = 0\,, & x \in \Omega, & t \in (t_0, t_f] \\ u = 0, & x \in \partial\Omega, & t \in (t_0, t_f] \\ u = u_0(x), & x \in \bar{\Omega}, & t = t_0 \end{cases} \qquad (2.3)$$

Through the utilization of the Cole-Hopf transformation, we obtain the following form:

$$u = -2\mu \frac{\beta_x(x,t)}{\beta(x,t)} = 0, \quad x \in \partial\Omega$$

Consequently, our own boundary conditions transformed into Neumann-conditions:

$$\beta_x(x,t) = 0\,, \quad for\ x \in \partial\Omega$$

The following are the results that we obtain after solving the separable ordinary differential equations (*ODEs*) for our initial conditions:

$$\beta(0,x) = \beta_0(x) = e^{-\frac{1}{2\mu}\int_\delta^x u_0(\theta)\,d\theta}$$

All that together give us:

$$\begin{cases} \dfrac{\partial \beta}{\partial t} - \mu \dfrac{\partial^2 \beta}{\partial x^2} = 0\,, & x \in \Omega, t \in (t_0, t_f] \\ \dfrac{\partial \beta}{\partial t} = 0, & x \in \partial\Omega, \quad t \in (t_0, t_f] \\ \beta = e^{-\frac{1}{2\mu}\int_\delta^x u_0(\theta)\,d\theta}, & x \in \bar{\Omega}, \quad t = t_0 \end{cases} \qquad (2.4)$$

in such a way δ forms the left boundary of $\partial\Omega$.

Now, we are able to derive an exact solution that can be utilized to assess and validate the accuracy of our numerical methods. Specifically, we select the spatial domain Ω to be the interval $(0,1)$ for this purpose.

Considering the boundary conditions,

$$\beta_x(0,t) = \beta_x(1,t) = 0$$

We can express,

$$\beta(x,t) = a + e^{-\pi^2 \mu t}\cos(\pi x), \quad for\ a > 1$$

By making this specific choice, the selected solution satisfies the boundary condition and provides a closed-form expression for the initial condition u_0.

We obtain the following result by utilizing of the Cole-Hopf transformation

$$u(x,t) = \frac{2\mu\pi\, e^{-\pi^2 \mu t}\sin(\pi x)}{a + e^{-\pi^2 \mu t}\cos(\pi x)} \qquad (2.5)$$

and the accompanying initial condition is:

$$u_0(x) = \frac{2\mu\pi\, \sin(\pi x)}{a + \cos(\pi x)} \qquad (2.6)$$

3. Numerical Methods:

This section will discuss the different methods used to address problem (2.1), which fall under the category of finite element approaches.

3.1 *Finite Element Methods*:

On our linearized equation (2.4), we are currently in the stage of using the *FEM*, with the methodology that was offered by Öziş [13]. This strategy is not only very easy to put into practice, but it also avoids the requirement for utilizing any further linearization methods for (2.3). Yet, since we are going to apply the Cole-Hopf transformation (2.2) to determine $u^h(x,t)$ even after solving the approximation $\beta^h(x,t)$, in case of, this strategy going to typically miss one-degree of precision owing to the derivative-term (2.2). As a consequence of this, higher order *FEM* approaches, such as those addressed in this work, are required. According to the author's best knowledge and understanding, a novel approach for solving (2.3) may be found by first solving (2.4) with higher order techniques, and then applying (2.2) thereafter.

3.1.1 *Weak Form*:

Considering any random test function, such as $v(x) \in H(0,1)$ multiplied by (2.4), such that $H(0,1)$ indicates the whole of the space occupied by functions that are (piecewise continuous) along with no less than a single (piecewise continuous) derivative over the (0, 1) domain. Our *BCs* were more natural than necessary, because we are dealing with homogeneous Neumann-conditions. Which it gives

$$\int_0^1 v\frac{\partial \beta}{\partial t} - v\mu\frac{\partial^2 \beta}{\partial x^2}dx = 0$$

By part integration, we have

$$\int_0^1 v\frac{\partial \beta}{\partial t} + \mu\frac{\partial \beta}{\partial x}\frac{dv}{\partial x}dx = \mu\left[v(1)\frac{\partial \beta}{\partial x}(1,x) - v(0)\frac{\partial \beta}{\partial x}(0,t)\right]$$

For the *BC* $\frac{\partial \beta}{\partial x}(1,t) - \frac{\partial \beta}{\partial x}(0,t) - 0$, we are having

$$\int_0^1 v\frac{\partial \beta}{\partial t} + \mu\frac{\partial \beta}{\partial x}\frac{dv}{\partial x}dx = 0 \qquad (2.7)$$

(2.7) indicates a weak solution to (2.4). To put it another way, a classical solution to equation (2.4) will fulfil the requirements of equation (2.7), but a solution to equation (2.7) will only fulfil the requirements of equation (2.4) when it is continuous.

3.1.2 *Discretization*:

Our domain is discretized using *N* equidistant intervals. Next, we proceed by inserting $\mathcal{P}-1$ equidistant points into the middle of every interval, resulting in the formation of the points x_i, where i ranges from 0 to $N_{\mathcal{P}}$. It's important to note that $\mathcal{P}$ represents the degree of the polynomial space employed in our finite element approximation. Now, we proceed by selecting a limited collection of elements from $z^h \subset H(0,1)$ denoted as a finite-dimensional subset, then we opt for a set of piecewise polynomials with a degree of $\mathcal{P}$ for $z^h$to serve as our basis. We represent our basis-functions as $\{\varphi_i\}_{i=0,\dots,N_{\mathcal{P}}}$ where each φ_i corresponds to a point x_i and spans over one or two intervals. Specifically, the basis-functions that correspond to the endpoints of the initial *N*-intervals extend their influence over both neighboring intervals, except for $\varphi_0$and $\varphi_{N_{\mathcal{P}}}$, which do not extend beyond the domain (0,1). On the other hand, each basis-function φ_i associated with an interior point only affects the original interval in which x_i is located.

Moreover, we specify that our basis is nodal, meaning that $\varphi_j(x_i) = 1\ for\ i = j\ \ and\ \ \varphi_j(x_i) = 0\ \ for\ \ i \neq j\ \ for\ \ x_i$ to the support of φ_j. Therefore, we could write

$$\beta^h(x,t) = \sum_{i=0}^{N_\mathcal{P}} e_i(t)\varphi_i(x)$$

in fact, each coefficient $e_i(t)$ corresponds to our model's approximation at x_i during time t is an obvious advantage of the above formulation, $i.e.$

$$e_i(t) = u^h(x_i,t) \qquad (2.8)$$

3.1.3 *Matrix Form*:

Now, the formulation represented by equation (2.7) is now transformed into,

$$\int_0^1 \left[v \sum_{i=0}^{N_\mathcal{P}} e_i'(t)\varphi_i(x) + \mu \sum_{i=0}^{N_\mathcal{P}} e_i(t)\varphi_i'(x)v'(x) \right] dx = 0$$

With rewriting, we get

$$\sum_{i=0}^{N_\mathcal{P}} e_i'(t) \int_0^1 v(x)\varphi_i(x)dx + \mu \sum_{i=0}^{N_\mathcal{P}} e_i(t) \int_0^1 \varphi_i'(x)\frac{dv}{dx}dx = 0$$

Now, if we consider v as a basis-function $\varphi_j(x)$, it gives us

$$\sum_{i=0}^{N_\mathcal{P}} e_i'(t) \int_0^1 \varphi_i(x)\varphi_j(x)dx + \mu \sum_{i=0}^{N_\mathcal{P}} e_i(t) \int_0^1 \varphi_i'(x)\varphi_j'(x)dx = 0$$

Moreover, in case we substitute $v = \varphi_j(x)\ for\ j = 0, \dots, N_\mathcal{P}$, we can derive the following set of equations:

$$Qe'(t) = -We(t), \quad Q_{ij} = \int_a^b \varphi_i(x)\varphi_j(x)dx, \quad W_{ij} = \int_a^b \varphi_i'(x)\varphi_j'(x)dx$$

Therefore, the solution to issue (2.7) is obtained via solving of an ordinary differential equation. Where Q is the $(mass\ matrix)$ and W is the $(stifness\ matrix)$. Through the selection of our basis, it can be demonstrated that the mass and stiffness matrices we utilize exhibit sparsity, positive definiteness, and symmetry [13]. We employ Gaussian-quadrature with an order of $(2\mathcal{P}-1)$ in solving the mentioned integrals to accurately calculate these matrices [14].

3.2.4 Methods for Cole-Hopf Transformation:

After obtaining $\beta^h(x,t)$ by solving equation (2.4), it is necessary to utilize the inverse Cole-Hopf transformation to determine $u^h(x,t)$ using the equation

$$u^h = -2\mu\frac{\beta_x}{\beta}$$

Öziş [13] proposes an approximation for β_x using the central difference method, which is known for its second-order accuracy.

$$\beta_x(x_i,t_j) \approx \frac{\beta_{i+1}^j - \beta_{i-1}^j}{2\Delta x}$$

When considering $p = 1$, specifically when dealing with piecewise linear functions, this option is highly commendable. However, when $p > 1$, selecting this option imposes a restriction on our accuracy, limiting it to the second order within the spatial domain. Consequently, this nullifies the benefits gained from utilizing polynomials of higher orders. Instead, in this investigation, for $p > 1$, we approximate $\beta_x(x,t)$ by utilizing the finite element derivative

$$\beta_x(x_i,t_j) = \sum_{i=1}^{N_P} e_i(t)\varphi_i'(x)$$

The level of approximation is accurate until the order p. Hence, we anticipate achieving an overall accuracy of p order in the L^2 norm and $p-1$ order in the H^1 norm.

3. Numerical Implementation:

Our analysis will primarily concentrate on employing the exact solution (2.5) and the initial condition (2.6) to assess and examine various numerical methods, which we will discuss in greater detail below. We will then present results that summarize our comparison of these methods at $\mu = .1$, equivalent to a Reynolds number of 10. This setting is potentially more challenging numerically compared to $\mu = 1$..

For the different methods on the spatial grid spacing Δx and the temporal grid spacing Δt over the time required to identify the solution, the H^1 *and* L^2 errors are given as follows:

$$H^1 \text{ error: } \left\| u^h - u \right\|_1 = \int_0^1 u^h + \frac{du^h}{dx} - u - \frac{du}{dx}\, dx$$

$$L^2\ error:\quad \left\|u^h - u\right\|_0 = \int_0^1 u^h - u\, dx$$

Here u^h is the approximate solution and u the exact solution at the final-time t_f, and considering the grid points x_i. Gaussian quadrature of degree $2p$ is employed as a method to integrate and compute the H^1 and L^2 errors over each interval, to measure the accuracy of our solutions and ensure reliable and precise results for each of our methods.

3.1 *Discrete in Time Method*:

In order to solve this Ordinary Differential Equation (*ODE*), we are going to investigate two different approaches. One approach is to divide the problem into discrete intervals in both time and space.

Therefore, we will approximation $e(t)$ and $e^j \approx e(j\Delta t)\ and\ e'(t)$ with finite-difference $e'(t) \approx \frac{1}{\Delta t}(e^{j+1} - e^j)$ We have

$$\frac{1}{\Delta t} Q\left(e^{j+1} - e^j\right) + \mu W\left(\lambda e^{j+1} + (1-\lambda)e^j\right) = 0$$

Such that $0 \leq \lambda \leq 1$ is the factor of weight that representing:

- $\lambda = 0$, the explicit 1th-order forward-Euler method with T is stable conditionally.
- $\lambda = 0.5$, the implicit 1th-order with dissipative backward-Euler method in time is stable unconditionally.
- $\lambda = 1$, the implicit 2th-order Crank-Nicolson method with T is stable unconditionally.

We have chosen to exclusively employ the Crank-Nicolson method in this study due to our aim of achieving a significant level of precision.

At this point, we are able to write down our equation.

$De^{j+1} = Ee^j$ such that $D = Q + \mu\lambda\Delta tW$ and $E = Q - \mu(1-\lambda)\Delta tW$

Hence, by calculating $D\ and\ E$ only once, we can progress our system in time by solving the equation $Ae^{j+1} = b$, where $A = D\ and\ b = Ee^j$. Due to the sparsity, symmetry, and positive definiteness of matrix D, we can employ a specialized Cholesky solver designed for sparse matrices to effectively calculate the desired solution. The chol(A, 'lower') command in $MATLAB^{®}$ offers an effective solver for this objective. The following tables show the results obtained by this method:

Δx	Δt	H^1 error	H^1conv.	L^2 error	L^2conv.	T
1.25E-01	1.25E-01	3.90E-02	—	6.66E-04	—	308.6
6.25E-02	6.25E-02	1.98E-02	0.974	1.66E-04	2.004	507.9
3.12E-02	3.12E-02	1.00E-02	0.989	4.15E-05	1.999	897.7
1.56E-02	1.56E-02	5.01E-03	0.996	1.04E-05	1.999	1777.7
7.81E-03	7.81E-03	2.51E-03	0.998	2.60E-06	1.999	3748
3.90E-03	3.90E-03	1.25E-03	0.999	6.50E-07	1.999	8191.1
1.95E-03	1.95E-03	6.28E-04	0.999	1.62E-07	1.999	23711.6
9.76E-04	9.76E-04	3.14E-04	0.999	4.06E-08	1.999	109360.9

Table 3.1: H^1 and L^2 error and convergence rates for the quadratic DFEM method

Δx	Δt	H^1 error	H^1conv.	L^2 error	L^2conv.	T
5.00E-01	3.00E-02	7.80E-04	—	2.54E-05	—	303.1
2.50E-01	3.00E-03	3.74E-05	4.38	6.56E-07	5.278	482.7
1.25E-01	9.00E-04	1.82E-06	4.36	2.05E-08	4.998	927.5
6.25E-02	1.00E-04	1.10E-07	4.044	5.64E-10	5.186	2648
3.12E-02	3.00E-05	6.93E-09	3.997	1.96E-11	4.847	19683.9

Table 3.2: H^1 and L^2 error and convergence rates for the quintic DFEM method

Δx	Δt	H^1 error	H^1conv.	L^2 error	L^2conv.	T
1.00E+00	3.12E-02	5.55E-04	—	2.10E-05	—	489.1
5.00E-01	1.95E-03	1.06E-05	5.707	1.82E-07	6.846	777.8
2.50E-01	1.22E-04	1.92E-08	9.107	2.25E-10	9.66	1405.8
1.25E-01	1.52E-05	2.78E-10	6.116	9.34E-12	4.594	3854.5
6.25E-02	6.10E-05	1.58E-10	0.816	3.65E-11	-1.968	10065.6

Table 3.3: H^1 and L^2 error and convergence rates for the octic DFEM method

It is observed that the *DFEM* linear approach attains rates of $p + 1$ in the L^2 norm and p in the H^1 norm. This outcome can be attributed to the utilization of a second-order accurate approximation of θx in the inverse Cole-Hopf transform discussed in section 3.2.4.

3.2 *Continuous in Time Method*:

Our second approach involves treating our ordinary differential equation *ODE* to be a continuous problem. Instead of discretizing our problem, we are able to solve the follows *ODE*

$$Qe'(t) = -Ve(t)$$

directly by utilizing an *ODE* solver to find a solution for $e(t)$. Our solver, out of necessity, will approach the *ODE* discretely. However, unlike the finite difference in time method, we will employ adaptive step sizes, eliminating the need to directly construct a time grid. In doing so, we enable greater flexibility in our approach to solving the *ODE*. To accomplish this, we make use of the *ODE* solvers provided in $MATLAB^{®}$:

- The Explicit Runge-Kutta (4,5) Dormand-Prince, non-stiff ode45,

- The Explicit Runge-Kutta (2,3) Bogacki and Shampine, non-stiff ode23,
- The Adams-Bashforth-Moulton, non-stiff ode113,
- The Numerical Differentiation Formula, stiff ode15s,
- The Modified Rosenbrock formula of order 2, stiff ode23s.

Our experiments with numerical investigations demonstrate that when addressing this problem, ode15s yields the most optimal results.

The following tables show the results obtained by this method:

Δx	Δt	H^1 error	H^1conv.	L^2 error	L^2conv.	T
1.25E-01	1.00E-04	3.89E-02	—	6.44E-04	—	198.8
6.25E-02	1.00E-04	1.98E-02	—	1.61E-04	1.999	325.1
3.12E-02	1.00E-05	1.00E-02	0.973	4.04E-05	1.995	582.9
1.56E-02	1.00E-05	5.01E-03	0.989	1.04E-05	1.956	1131.3
7.81E-03	1.00E-06	2.51E-03	0.995	2.57E-06	2.017	2261.3
3.90E-03	1.00E-07	1.25E-03	0.998	6.36E-07	2.016	5180.5
1.95E-03	1.00E-07	6.28E-04	0.999	1.69E-07	1.909	13948.4
9.76E-04	1.00E-08	3.14E-04	0.999	3.97E-08	2.094	65447.1

Table 3.4: H^1 and L^2 error and convergence rates for the quadratic CFEM method

Δx	Δt	H^1 error	H^1conv.	L^2 error	L^2conv.	T
1.00E+00	1.00E-04	6.98E-03	—	4.13E-04	—	177.2
5.00E-01	1.00E-05	7.79E-04	3.164	2.26E-05	4.189	234
2.50E-01	1.00E-07	3.74E-05	4.379	6.26E-07	5.177	377.6
1.25E-01	1.00E-08	1.82E-06	4.36	1.74E-08	5.166	692.8
6.25E-02	1.00E-10	1.10E-07	4.044	5.36E-10	5.024	1271.4
3.12E-02	1.00E-11	6.93E-09	3.996	3.77E-11	3.83	2434.3
1.56E-02	1.00E-12	4.41E-10	3.973	3.92E-12	3.265	5273.7
7.81E-03	1.00E-13	2.03E-10	1.119	2.10E-12	0.901	18951.3

Table 3.5: H^1 and L^2 error and convergence rates for the quintic CFEM method

Δx	Δt	H^1 error	H^1conv.	L^2 error	L^2conv.	T
1.00E+00	1.00E-05	5.54E-04	—	1.88E-05	—	364.5
5.00E-01	1.00E-07	1.06E-05	5.704	1.85E-07	6.662	556.5
2.50E-01	1.00E-10	1.92E-08	9.108	1.88E-10	9.946	967.5
1.25E-01	1.00E-13	1.79E-10	6.75	1.08E-12	7.443	1862.4
6.25E-02	1.00E-13	8.50E-11	1.074	7.23E-13	0.583	3657.1
3.12E-02	1.00E-13	3.17E-10	-1.902	4.85E-12	-2.746	7668.8

Table 3.6: H^1 and L^2 error and convergence rates for the octic CFEM method

The following figure presents the exact solution for equation (2.5).

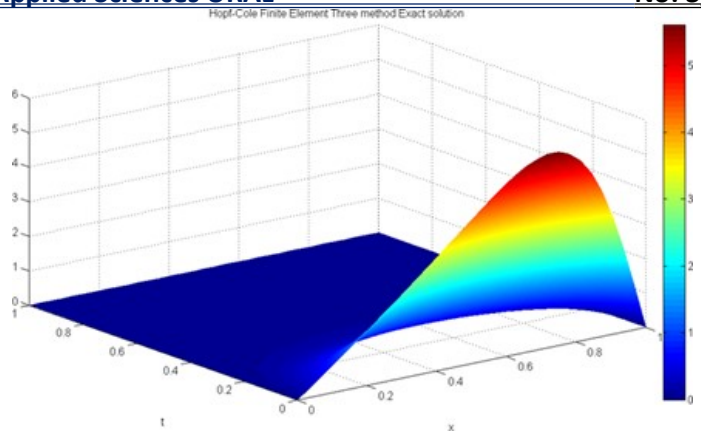

Figure 3.1: Exact solution of Eq. (2.5)

The following figure illustrates an approximate solution using the CFEM method, with $p = 8,\ \Delta t = 10^{-13}\ \ and\ \Delta x = \frac{1}{4}$:

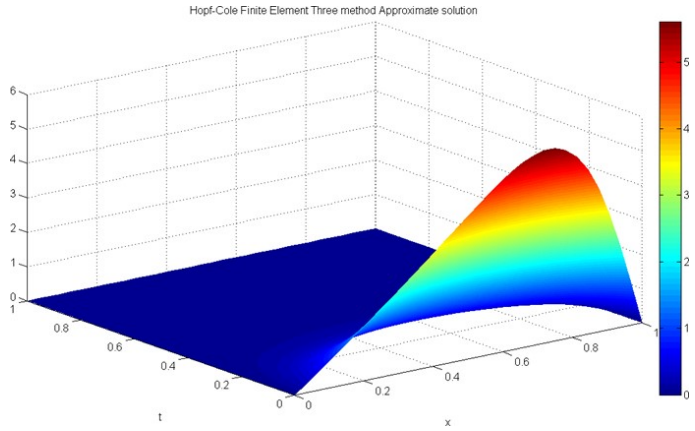

Figure 3.2: Approximate solution of Eq. (2.5)

Due to the similarity in appearance between the exact solution and the approximate solution, as demonstrated in figures 3.1 and 3.2, the discrepancy is illustrated in figure 3.3 using the CFEM method, with $p = 8,\ \Delta t = 10^{-13}\ \ and\ \Delta x = \frac{1}{4}$:

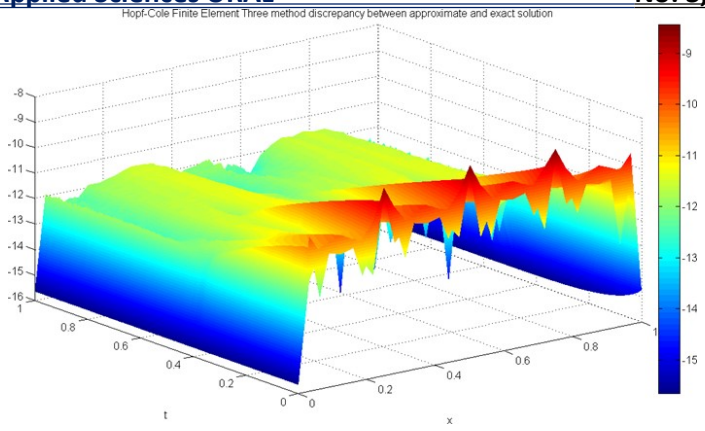

Figure 3.3: Log error of approximation versus exact solution for Eq. (2.5) with $\Delta x = 1/4$

For each of the methods, there comes a time when using a larger (Δx or Δt) the accuracy diminishes due to the effects of numerical round-off. Figure 3.4 illustrates this scenario, where the scenario remains consistent with Figure 3.3 but with a $\Delta x = \frac{1}{8}$. It is important to observe how the presence of numerical noise becomes more significant in this case.

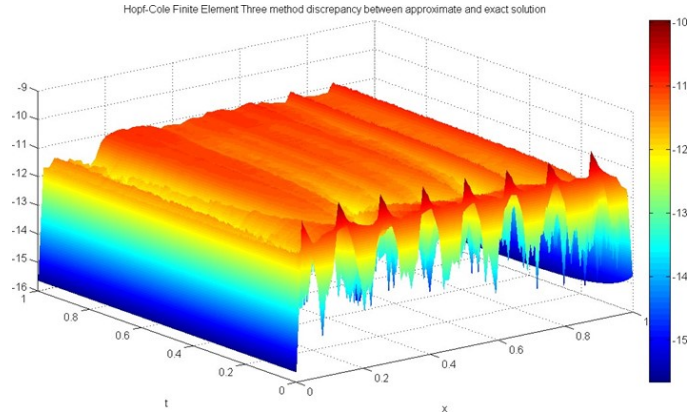

Figure 3.4: Log error of approximation vs exact solution for Eq. (2.5) with $\Delta x = 1/8$

Please keep in mind that only the most optimal solution in terms of time and numerical accuracy is presented for each method. Conversely, the Finite Element Methods (FEM) are reaching a convergence rate close to or at p in the L^2 norm and $p - 1$ in the H^1 norm. This demonstrates a more precise balance between accuracy and the required time for solving the model. Furthermore, it is evident that we are achieving convergence rates that align with the expected outcomes specified for each of the aforementioned methods.

Conclusion

in this work, we focused on studying controlling inverse problem of a viscous fluid flow through the implementation of the simple nonlinear 1D viscous Burgers' equation. we have developed and compared different methods, specifically finite element techniques along with finite difference methods prior to implementing the inverse Cole-Hopf transformation (2.2). Based on our results, it is evident that our finite element methods, particularly those based on C-FEM, exhibit an exceedingly high level of accuracy in a relatively short period of time.

It is important to note that there are a couple of downsides associated with our methodology:

- there is a slight loss of one degree of accuracy. However, this drawback is negligible when dealing with polynomials of very high degrees, as the impact on the overall precision is minimal.
- the treatment of boundary and initial conditions becomes more complicated.

it is certainly worthwhile to dedicate future research to exploring the other techniques commonly used for regularization in inverse problems.

References:

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

[2] Bhrawy, A. H., & Zaky, M. (2016). Numerical algorithm for the variable-order Caputo fractional functional differential equation. *Nonlinear Dynamics*, *85*, 1815-1823.

[3] Burgers, J. M. (1995). *Mathematical examples illustrating relations occurring in the theory of turbulent fluid motion* (pp. 281-334). Springer Netherlands.

[4] Cecchi, M. M., Nociforo, R., & Grego, P. P. (1996). Space-time finite elements numerical solutions of Burgers Problems. *Le Matematiche*, *51*(1), 43-57.

[5] Cole, J. D. (1951). On a linear quasilinear parabolic equation in aerodynamics. Q. *Appl. Math*, *9*, 225-236.

[6] Fletcher, C. A. (1983). Generating exact solutions of the two-dimensional Burgers' equations. *International Journal for Numerical Methods in Fluids*, *3*, 213-216.

[7] Hassanien, I. A., Salama, A. A., & Hosham, H. A. (2005). Fourth-order finite difference method for solving Burgers' equation. *Applied Mathematics and Computation*, *170*(2), 781-800.

[8] Kadalbajoo, M. K., Sharma, K. K., & Awasthi, A. (2005). A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equations. *Applied mathematics and computation*, *170*(2), 1365-1393.

[9] Kutluay, S. E. L. Ç. U. K., Bahadir, A. R., & Özdeş, A. (1999). Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods. *Journal of computational and applied mathematics*, *103*(2), 251-261.

[10] Kutluay, S. E. L. Ç. U. K., Esen, A. L. A. A. T. T. İ. N., & Dag, I. (2004). Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method. *Journal of computational and Applied Mathematics*, *167*(1), 21-33.

[11] Leonenko, N. & Meluikovn, 2001, "Renarmahzation and Homogenization of the solutions of heat equation with linear Potential and related Burger's equation with random data", Theory probab & Math. Statistics 1, PP.27-64

[12] Liao, W. (2008). An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation. *Applied Mathematics and Computation*, *206*(2), 755-764.

[13] Öziş, T., Aksan, E. N., & Özdeş, A. (2003). A finite element approach for solution of Burgers' equation. *Applied Mathematics and Computation*, *139*(2-3), 417-428..

[14] Pressa, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. (2007). Numerical recipes 3rd edition: The art of scientific computing.

[15] Srivastava, V. K., Awasthi, M. K., & Singh, S. (2013). An implicit logarithmic finite-difference technique for two dimensional coupled viscous Burgers' equation. *Aip Advances*, *3*(12), 122105.

[16] Tamsir M and Srivastava V.K adn Mishra P.D. Numerical simulation of three dimensional advection-diffusion equations by using modified cubic b-spline differential quadrature method. APJEST, 2016.

[17] The MathWorks. MATLAB R optimization toolbox v4.1 - documentation. http://www.mathworks.com/access/helpdesk/help/toolbox/optim/, 2008.

[18] Wood, W. L. (2006). An exact solution for Burger's equation. *Communications in numerical methods in engineering*, *22*(7), 797-798.

[19] Zhu, C. G., & Wang, R. H. (2009). Numerical solution of Burgers' equation by cubic B-spline quasi-interpolation. *Applied Mathematics and Computation*, *208*(1), 260-272.