NUMERICAL STUDY OF BURGERS' EQUATION USING FEM

ME5204 Finite Element Analysis (course project)

Ashmit Sinha
me21b037@smail.iitm.ac.in

Koneru Nitish
me21b090@smail.iitm.ac.in

Department of Mechanical Engineering,
Indian Institute of Technology Madras, Chennai - 600036, India.

Abstract

This paper presents a numerical study on solving the one-dimensional Burgers' equation, a fundamental nonlinear partial differential equation, using the Finite Element Method. The Burgers' equation is solved under three different sets of initial and boundary conditions to demonstrate the versatility and robustness of the FEM approach. The weak form of the equation is derived, and spatial discretization is performed using linear basis functions. Time-stepping schemes are employed to capture the transient behavior of the system, with the nonlinearity addressed using the Picard iteration method for efficient linearization. The numerical results are validated against exact or analytical solutions, showcasing the accuracy of the method. The study highlights the effectiveness of FEM in handling nonlinear problems and provides insights into the dynamics of viscous fluid systems under varying conditions.

1 Introduction

The Burgers' equation describes the dynamics of viscous fluid flow in one spatial dimension, incorporating both nonlinear convective effects and viscous diffusion. This equation has applications in fluid dynamics, traffic flow, and gas dynamics, serving as a simplified model for more complex systems (see [1]). The kinematic viscosity term governs the balance between convection and diffusion, making the equation a fundamental benchmark for testing numerical methods.

In this study, the Burgers' equation is solved using the Finite Element Method (FEM) under three distinct sets of boundary and initial conditions. These cases are designed to assess the effectiveness of the FEM in handling transient, nonlinear dynamics across varying conditions. The nonlinear term is addressed using Picard iteration, ensuring numerical stability and convergence.

Key aspects include:

- Discretization of the spatial domain using linear basis functions.
- Temporal evolution captured through a time-stepping scheme.
- Comparison of numerical results with analytical or exact solutions to validate accuracy.

This work provides insights into the behavior of the velocity field under different scenarios, emphasizing the robustness of FEM for solving nonlinear partial differential equations.

For this project, we refer to the article by Bilge et.al[2], make use of the original problem statement and achieve promising results utilizing FEM.

2 Governing equations and weak form

The Burgers' equation governing the dynamics of a viscous fluid in one spatial dimension is given by:

$$\frac{\partial u(x,t)}{\partial t} + u(x,t)\frac{\partial u(x,t)}{\partial x} = \nu \frac{\partial^2 u(x,t)}{\partial x^2}, \tag{1}$$

where $u(x,t)$ is the velocity field, ν is the kinematic viscosity, x is the spatial coordinate, and t is the time.

The term $u(x,t)\frac{\partial u(x,t)}{\partial x}$ represents the nonlinear convective effects, while $\nu\frac{\partial^2 u(x,t)}{\partial x^2}$ models the diffusive effects due to viscosity.

2.1 Derivation of the Weak Form

To derive the weak form:

1. Multiply the equation by a test function $v \in H^1(\Omega)$:

$$v\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2}\right) = 0.$$

2. Integrate over the domain Ω:

$$\int_\Omega v\frac{\partial u}{\partial t}\,dx + \int_\Omega vu\frac{\partial u}{\partial x}\,dx - \nu\int_\Omega v\frac{\partial^2 u}{\partial x^2}\,dx = 0.$$

3. Apply integration by parts to the diffusion term:

$$-\nu\int_\Omega v\frac{\partial^2 u}{\partial x^2}\,dx = \nu\int_\Omega \frac{\partial v}{\partial x}\frac{\partial u}{\partial x}\,dx - \nu\int_{\partial\Omega} v\frac{\partial u}{\partial x}\,ds.$$

Since the problems we solve specifically have only dirichlet boundary conditions :

$$\int_\Omega v\frac{\partial u}{\partial t}\,dx + \int_\Omega vu\frac{\partial u}{\partial x}\,dx + \nu\int_\Omega \frac{\partial v}{\partial x}\frac{\partial u}{\partial x}\,dx = 0.$$

2.2 Picard Linearization

To handle the nonlinear term $u\frac{\partial u}{\partial x}$, Picard linearization is used:

1. Assume a known solution u^k from the previous iteration.

2. Replace $u\frac{\partial u}{\partial x}$ with $u^k\frac{\partial u}{\partial x}$, resulting in a linearized weak form:

$$\int_\Omega v\frac{\partial u}{\partial t}\,dx + \int_\Omega vu^k\frac{\partial u}{\partial x}\,dx + \nu\int_\Omega \frac{\partial v}{\partial x}\frac{\partial u}{\partial x}\,dx = 0.$$

3 Discretization approach and Procedure

- Discuss the element type used, give its shape functions. Give complete derivation of all the matrices, boundary terms for an element. - Discuss the meshing procedure - if you are solving transient, discuss the temporal scheme here. details of that should be given

3.1 Time Discretization

Using the implicit Euler method, approximate the time derivative as:

$$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^n}{\Delta t},$$

where u^{n+1} is the unknown solution at the next time step and u^n is the known solution. Substituting into the weak form gives:

$$\int_\Omega v \frac{u^{n+1} - u^n}{\Delta t} \, dx + \int_\Omega v u^k \frac{\partial u^{n+1}}{\partial x} \, dx + \nu \int_\Omega \frac{\partial v}{\partial x} \frac{\partial u^{n+1}}{\partial x} \, dx = 0.$$

3.2 Spatial Discretization Using Finite Elements

1. Approximate u and v using finite element basis functions:

$$u(x,t) \approx \sum_j u_j(t)\phi_j(x), \quad v(x) = \sum v_i \phi_i(x),$$

   where $\phi_j(x)$ are the finite element basis functions.

2. Substituting into the weak form yields a system of equations for the unknown coefficients u_j^{n+1}:

$$\mathbf{M}\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{C}(\mathbf{u}^k)\mathbf{u}^{n+1} + \nu \mathbf{K}\mathbf{u}^{n+1} = 0,$$

   where:

   - $\mathbf{M}$ is the mass matrix,
   - $\mathbf{C}(\mathbf{u}^k)$ is the convection matrix (linearized with u^k),
   - $\mathbf{K}$ is the stiffness matrix.

3.3 Shape Function and Numerical Integration

In the 1D case, linear basis functions ($\phi_j(x)$) are utilized. The shape function ξ for an element in the reference domain $[-1, 1]$ is expressed as:

$$\phi_1(\xi) = \frac{1-\xi}{2}, \quad \phi_2(\xi) = \frac{1+\xi}{2}.$$

The derivatives of the shape functions with respect to ξ are:

$$\frac{d\phi_1}{d\xi} = -\frac{1}{2}, \quad \frac{d\phi_2}{d\xi} = \frac{1}{2}.$$

For numerical integration, Gaussian quadrature is employed. The integral over the physical domain Ω is mapped to the reference element as:

$$\int_\Omega f(x)\, dx \approx \sum_{i=1}^{n_q} w_i f(x_i),$$

where x_i are the Gauss points, and w_i are the corresponding weights. The mapping from the reference element to the physical domain is given by:

$$x(\xi) = \frac{x_1(1-\xi)}{2} + \frac{x_2(1+\xi)}{2},$$

and the Jacobian of the transformation is:

$$J = \frac{x_2 - x_1}{2}.$$

3.4 Solution Procedure

1. Start with an initial guess u^0. This is generally specified in the boundary conditions.
2. At each time step, solve the linearized system for u^{n+1} using the Picard update u^k.
3. Update $u^k \rightarrow u^{n+1}$ and repeat until convergence. Convergence criteria in the case is the difference in norm of u^{n+1} and u^k tending to a tolerance of your choice. We chose it as 10^{-10}.

3.5 Post-Processing

After solving, We compute physical quantities such as velocity profiles and error norms. We then perform a mesh convergence study to validate the solution.

3.6 Boundary Conditions

We solve it for three different boundary conditions. They are listed below:

- **Problem 1:** This involves solving the equation with sinusoidal initial conditions and homogeneous Dirichlet boundary conditions. The exact solution is used for comparison.
- **Problem 2:** This focuses on quadratic initial conditions and homogeneous Dirichlet boundary conditions, with exact coefficients used for validation.
- **Problem 3:** This addresses a more complex scenario with an exponential-based initial condition and analytical boundary conditions on a custom domain.

4 Numerical Example

4.1 Problem 1

4.1.1 Boundary Conditions

We solve the Burgers Equation for the initial condition

$$u(x,0) = sin(\pi x), \quad 0 < x < 1 \tag{2}$$

and the boundary conditions:

$$u(0,t) = u(1,t) = 0, \quad t > 0 \tag{3}$$

4.1.2 Calculation of the Analytical solutions

The exact solution is given by:

$$u(x,t) = 2\pi v \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2 vt} n \sin(n\pi x) / \left[a_0 + \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2 vt} \cos(n\pi x) \right]$$

The coefficients are given by:

$$a_0 = \int_0^1 e^{-(2\pi v)^{-1}[1-\cos(\pi x)]}\, dx$$

$$a_n = 2\int_0^1 e^{-(2\pi v)^{-1}[1-\cos(\pi x)]} \cos(n\pi x)\, dx, \quad n = 1, 2, 3, \ldots$$

These integrals are computed using Numerical Integration. Since the n values tend from 0 to infinity, the cosine function within the integral will tend to oscillate with high n values, causing inaccurate approximations for smaller number of gauss points. We hence invoke a numpy function which will give us the points and weights for a large degree approximation, to the degree of 50 points to get accurate results for the coefficient. It then turns out that the first 10 coefficients are more than enough, with its value largely dimnishing later on.

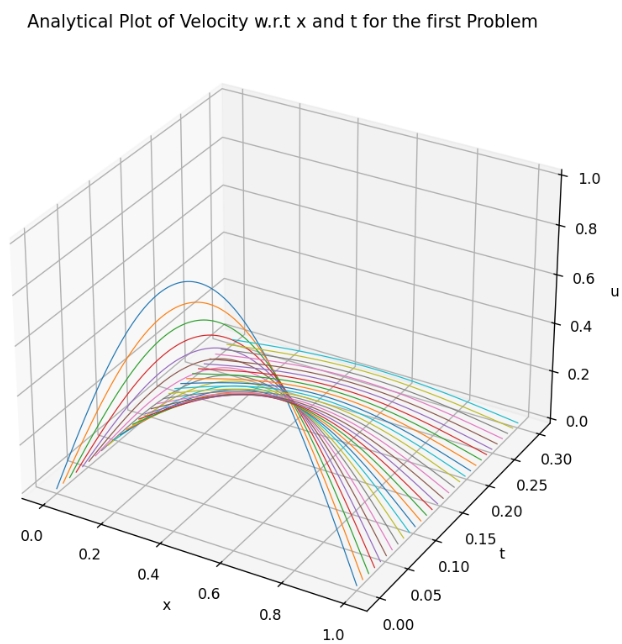

Figure 1: Analytical Solution for Problem 1

4.1.3 Calculation of the FEM solutions

The Solutions are calculated using the implementation shown in the previous section. We also have to ensure convergence w.r.t time and spacial terms (only x in this case). Numerical Integration in this case is simple enough, and will require only two integration points.

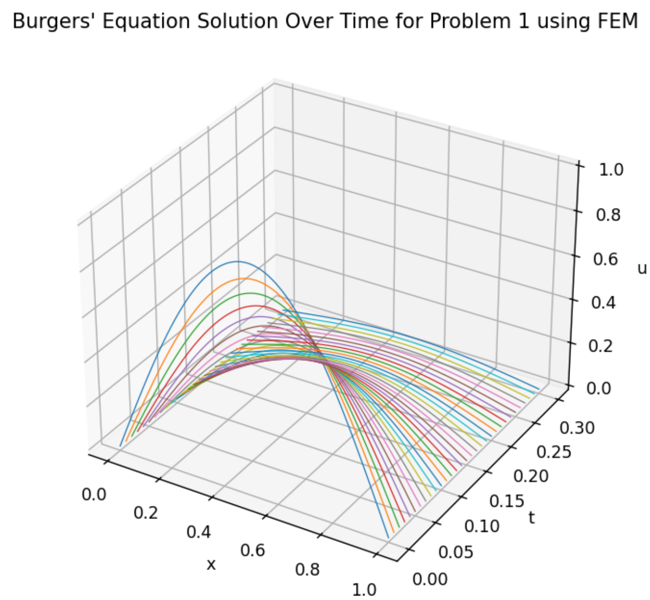

Figure 2: FEM Solution for Problem 1

Table 1: Comparison of the numerical solution with the exact solution at different times for $\nu = 1.0$, $h = 0.01$ and $k = 5 * 10^{-3}$.

x	$t = 0.125$		$t = 0.14$	
	FEM	Exact	FEM	Exact
0	0	0	0	0
0.1	0.0865	0.08607	0.07496	0.07454
0.2	0.1655	0.16467	0.14332	0.14252
0.3	0.22987	0.22871	0.19886	0.19774
0.4	0.27337	0.27198	0.23619	0.23485
0.5	0.29120	0.28971	0.25121	0.249778
0.6	0.28062	0.27917	0.24171	0.24032
0.7	0.24160	0.24035	0.20781	0.20660
0.8	0.17724	0.17631	0.15227	0.15138
0.9	0.09377	0.09327	0.08050	0.08002
1	0	0	0	0

The below table shows that the model is performing increasingly better upon making the elements smaller. With time steps $\approx 10^{-5}$ the solutions will take a lot of time, but will present more accurate solutions.

Table 2: Comparison of the numerical solution with the exact solution at different h for $\nu = 1.0$, and $k = 5 * 10^{-4}$.

x	$h = 0.1$ FEM	$h = 0.05$ FEM	$h = 0.02$ FEM	Exact
0	0	0	0	0
0.1	0.10894	0.10949	0.10964	0.10962
0.2	0.20861	0.20968	0.20998	0.20996
0.3	0.29018	0.29173	0.29216	0.29213
0.4	0.34575	0.34767	0.34824	0.34821
0.5	0.36908	0.37129	0.37191	0.37188
0.6	0.35646	0.35873	0.35937	0.35934
0.7	0.30753	0.30960	0.31018	0.31016
0.8	0.22598	0.22757	0.22801	0.22801
0.9	0.11968	0.12054	0.12079	0.12079
1	0	0	0	0
L2	0.1187	0.02496	0.001488	

Table 3: Comparison of the numerical solution with the exact solution at different times for $\nu = 1.0$, $\nu = 0.01$, $h = 0.0125$ and $k = 5 * 10^{-4}$.

x	t	$\nu = 1.0$		$\nu = 0.01$	
		FEM	Exact	FEM	Exact
0.25	0.10	0.254237	0.253638	0.566419	0.566328
	0.15	0.157136	0.156601	0.512271	0.512148
	0.20	0.096881	0.096442	0.466726	0.466583
	0.25	0.059557	0.059218	0.428148	0.427995
0.50	0.10	0.372442	0.371577	0.947270	0.947414
	0.15	0.227613	0.226824	0.899984	0.900098
	0.20	0.139115	0.138473	0.848315	0.848365
	0.25	0.085027	0.084538	0.796779	0.796762
0.75	0.10	0.273205	0.272582	0.860064	0.860134
	0.15	0.164952	0.164369	0.922505	0.922756
	0.20	0.099904	0.099435	0.961452	0.961891
	0.25	0.060700	0.060347	0.974168	0.974689

The above table showcases the convergence of our code for lower viscosities, showcasing that it is capable of handling these problems without having convergence issues.

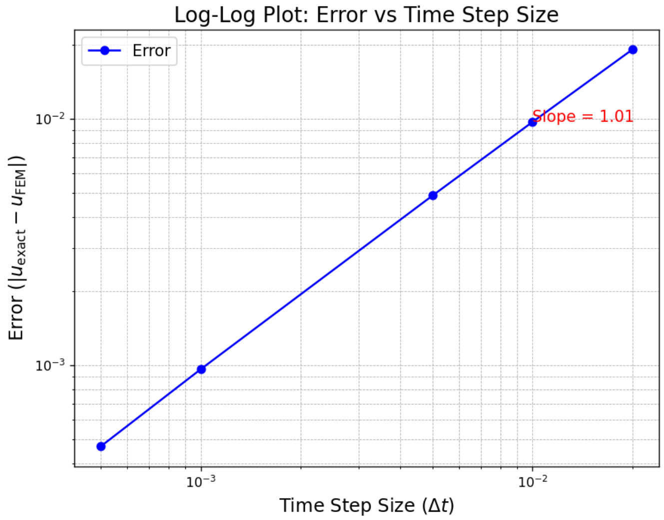

Figure 3: Log error vs Log Δt plot for Problem 1

The Log error vs Log δt plot gives a slope of 1, which is near ideal, since we are using first order accuracy w.r.t time.

Table 4: Convergence Results for Problem 1 at $t = 0.2$ and $x = 0.75$

Time Step (Δt)	**Numerical Solution (u)**	**Error ($\lvert u - u_{\mathbf{exact}} \rvert$)**
2×10^{-2}	0.118577	0.019142
1×10^{-2}	0.109161	0.009726
5×10^{-3}	0.104326	0.004891
1×10^{-3}	0.100399	0.000964
5×10^{-4}	0.099904	0.000469
Exact Solution	0.099435	0.000000

4.2 Problem 2

4.2.1 Boundary Conditions

We solve the Burgers Equation for the initial condition

$$u(x,0) = 4x(1-x), \quad 0 < x < 1 \tag{4}$$

and the boundary conditions:

$$u(0,t) = u(1,t) = 0, \quad t > 0 \tag{5}$$

4.2.2 Calculation of the Analytical solutions

The exact solution is given by:

$$u(x,t) = 2\pi v \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2 vt} n \sin(n\pi x) / \left[a_0 + \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2 vt} \cos(n\pi x) \right]$$

This is similar to that of the first problem. The coefficients are given by:

$$a_0 = \int_0^1 e^{-(3v)^{-1}(3-2x)x^2} dx$$

$$a_n = 2 \int_0^1 e^{-(3v)^{-1}(3-2x)x^2} \cos(n\pi x)\, dx, \quad n = 1, 2, 3, \ldots$$

These integrals are computed using Numerical Integration. Similar to problem 1, the cos(nπx) term will oscillate, hence requiring higher number of integration points. We take 50 points.

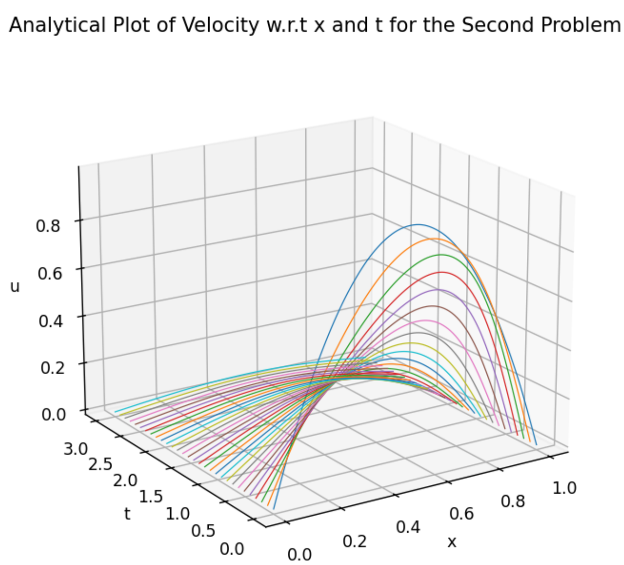

Figure 4: Analytical Solution for Problem 2

4.2.3 Calculation of the FEM solutions

The Solutions are calculated using the implementation shown in the previous section. We also have to ensure convergence w.r.t time and spacial terms (only x in this case). Numerical Integration in this case is simple enough, and will require only two integration points.

Now, repeating the same for this problem,

Table 5: Comparison of the numerical solution with the exact solution at different times for $\nu = 0.1$, $h = 0.01$ and $k = 5 * 10^{-3}$.

x	$t = 1.3$		$t = 1.4$	
	FEM	Exact	FEM	Exact
0	0	0	0	0
0.1	0.05248	0.05245	0.04834	0.04831
0.2	0.10333	0.10327	0.095002	0.09494
0.3	0.15052	0.15041	0.13790	0.13781
0.4	0.19107	0.19094	0.17416	0.17404
0.5	0.0.22059	0.22044	0.19968	0.19953
0.6	0.23285	0.23265	0.20899	0.20877
0.7	0.22003	0.21985	0.19551	0.19535
0.8	0.17543	0.17528	0.15437	0.15424
0.9	0.09841	0.09833	0.08597	0.08589
1	0	0	0	0

The given table showcases the difference between the analytical solution and the determined solution. For smaller time steps, it becomes more accurate.

Table 6: Comparison of the numerical solution with the exact solution at different h for $t = 1$, $\nu = 0.1$, and $k = 5 * 10^{-3}$.

x	$h = 0.1$ FEM	$h = 0.05$ FEM	$h = 0.02$ FEM	Exact
0	0	0	0	0
0.1	0.06765	0.06756	0.06753	0.06749
0.2	0.13394	0.13374	0.13368	0.13359
0.3	0.19701	0.19666	0.19656	0.19643
0.4	0.25383	0.25334	0.25320	0.25308
0.5	0.29916	0.29864	0.29850	0.29830
0.6	0.32420	0.32401	0.32394	0.32373
0.7	0.31589	0.31652	0.31667	0.31650
0.8	0.25958	0.26116	0.26159	0.26148
0.9	0.14899	0.15054	0.15097	0.15093
1	0	0	0	0
L2	0.06467	0.01490	0.00806	

Table 7: Comparison of the numerical solution with the exact solution at different times for $\nu = 1.0$, $\nu = 0.01$, $h = 0.0125$ and $k = 5 * 10^{-4}$.

x	t	$\nu = 1.0$		$\nu = 0.01$	
		FEM	Exact	FEM	Exact
0.25	0.10	0.262100	0.261480	0.607429	0.607363
	0.15	0.162029	0.161478	0.549523	0.549421
	0.20	0.099922	0.099470	0.499956	0.499828
	0.25	0.061437	0.061088	0.457555	0.457413
0.50	0.10	0.384313	0.383422	0.955870	0.956007
	0.15	0.234870	0.234055	0.914298	0.914426
	0.20	0.143550	0.142888	0.867052	0.867136
	0.25	0.087738	0.087233	0.818309	0.818337
0.75	0.10	0.282216	0.281573	0.886613	0.886767
	0.15	0.170340	0.169738	0.938182	0.938437
	0.20	0.103140	0.102655	0.969361	0.969741
	0.25	0.062655	0.062290	0.979027	0.979469

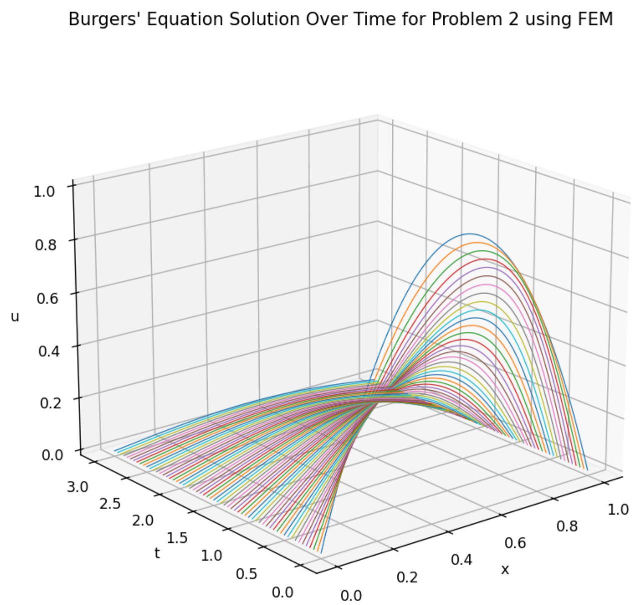

Figure 5: FEM Solution for Problem 2

Table 8: Convergence Results for Problem 2 at $t = 0.2$ and $x = 0.75$

Time Step (Δt)	**Numerical Solution (u)**	**Error ($\|u - u_{\mathbf{exact}}\|$)**
1×10^{-2}	0.112702	0.010047
5×10^{-3}	0.107708	0.005053
1×10^{-3}	0.103651	0.001996
5×10^{-4}	0.103140	0.001485
Exact Solution	0.102655	0.000000

4.3 Problem 3

4.3.1 Boundary Conditions

We solve the Burgers Equation for the initial condition

$$u(x, 1) = x/(1 + e^{(x^2-1/4)/4v}), \quad a < x < b \tag{6}$$

and the boundary conditions:

$$u(a, t) = u(b, t) = 0, \quad t > 0 \tag{7}$$

4.3.2 Calculation of the Analytical solutions

The exact solution is given by:

$$u(x, t) = x/t/(1 + e^{x^2/4vt}(t/e^{1/8v})^{0.5}), \quad a < x < b \tag{8}$$

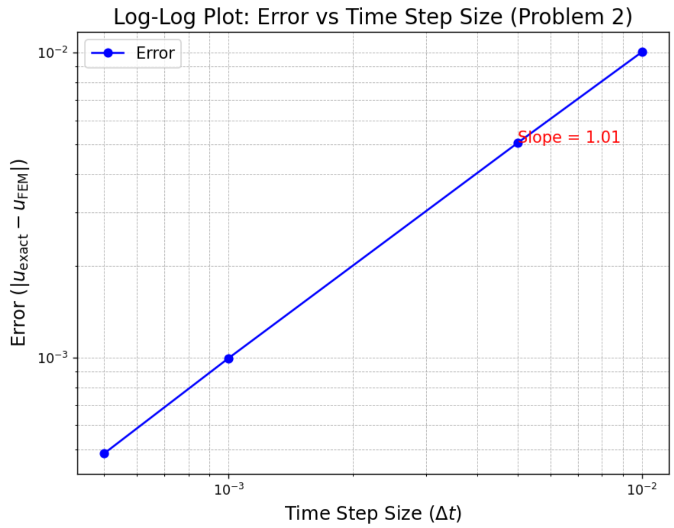

Figure 6: Log error vs Log Δt plot for Problem 2

Table 9: Comparison of the numerical solution with the exact solution at different times for $\nu = 0.005$, $h = 0.01$ and $k = 5 * 10^{-3}$.

x	$t = 2$		$t = 2.5$	
	FEM	Exact	FEM	Exact
0	0	0	0	0
0.12	0.06	0.05994	0.04800	0.04795
0.24	0.12001	0.11989	0.09601	0.0959
0.36	0.18001	0.17982	0.14401	0.14384
0.48	0.23966	0.23940	0.19192	0.19169
0.60	0.28789	0.28755	0.23821	0.23789
0.72	0.11100	0.11181	0.24251	0.24265
0.84	0.00164	0.00175	0.03748	0.03806
0.96	7.483e-6	9.1423e-6	0.00060	0.00065
1.08	1.522e-8	2.272e-8	4.6e-6	5.5392e-6
1.2	0	2.717e-11	0	2.6e-8

The given table showcases the difference between the analytical solution and the determined solution. For smaller time steps, it becomes more accurate.

Table 10: Comparison of the numerical solution with the exact solution at different h for $t = 2.5$, $\nu = 0.005$, and $k = 5 * 10^{-3}$.

x	$h = 0.1$ FEM	$h = 0.05$ FEM	$h = 0.02$ FEM	$h = 0.01$ FEM	Exact
0	0	0	0	0	0
0.12	0.06765	0.06756	0.04800	0.04800	0.04795
0.24	0.13394	0.13374	0.096015	0.09601	0.0959
0.36	0.19701	0.19666	0.14401	0.14401	0.14384
0.48	0.25383	0.25334	0.19194	0.19192	0.19169
0.6	0.29916	0.29864	0.23838	0.23821	0.23789
0.72	0.32420	0.32401	0.24284	0.24251	0.24265
0.84	0.31589	0.31652	0.037	0.03748	0.03806
0.96	0.25958	0.26116	0.00048	0.00060	0.00065
1.08	0.14899	0.15054	2.25e-6	4.6e-6	5.5392e-6
1.2	0	0	0	0	2.6e-8
L2	0.06467	0.165	0.0237	0.007778	

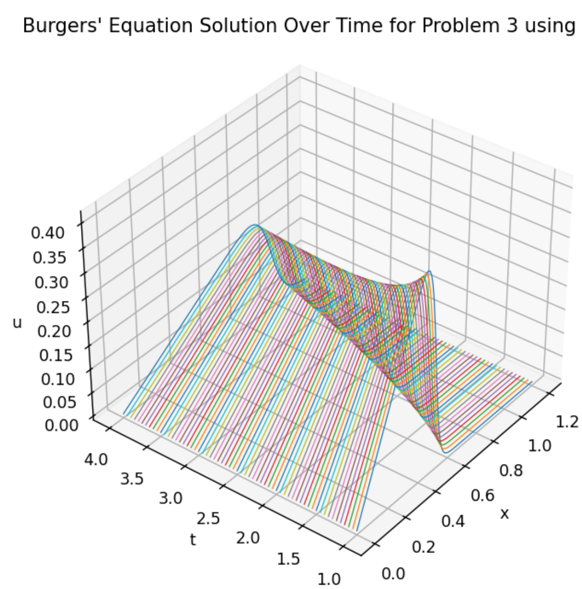

Figure 8: FEM Solution for Problem 3

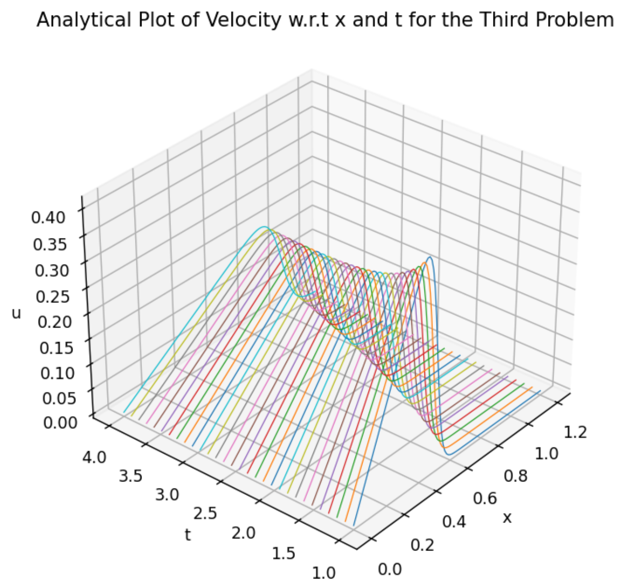

Figure 7: Analytical Solution for Problem 3

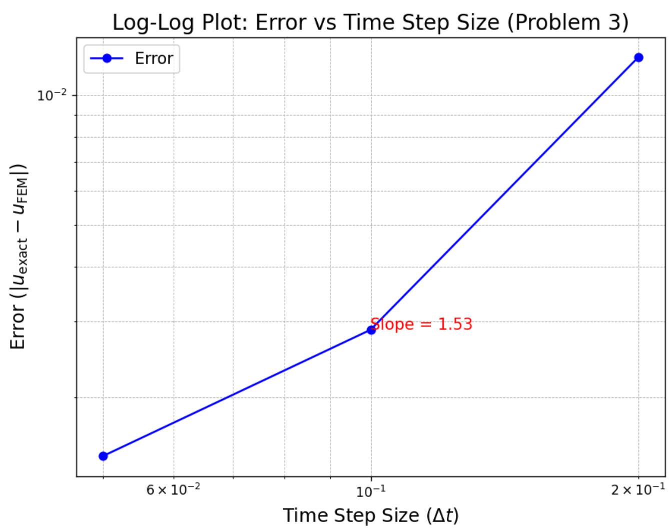

Figure 9: Log error vs Log Δt plot for Problem 3

Table 11: Convergence Results for Problem 3 at $t = 1.5$ and $x = 1.5$

Time Step (Δt)	**Numerical Solution** (u)	**Error** ($|u - u_{\mathbf{exact}}|$)
2×10^{-1}	0.316344	0.010359
1×10^{-1}	0.306989	0.000634
5×10^{-2}	0.305587	0.001461
Exact Solution	0.304125	0.000000

5 Conclusions and Future Scope

In this paper, we defined the Finite Element Method for solving Non-Linear Partial Differential Equations using Picard iteration. Numerical solutions for three test problems are found, and comprehensively studied. The results showed that the implicit finite difference method offers great accuracy in the viscosity ranges explored in the problems (going as low as 0.005), provided the time step and element size are small enough to enable accurate approximation. However, it does have some limitations. It is observed that the L2 error becomes constant after a while as the element size decreases. This might be due to the floating point limitation, causing round-off errors and not enabling it to decrease. Future scope includes possibly using Wavelet Basis for shape functions, possibly eliminating the non-linear term due to the orthogonality nature of the wavelet basis, w.r.t the Haar Wavelet.

Acknowledgments and Contributions

We would like to thank Prof. Sundararajan Natarajan for giving us the opportunity to undertake this project, as well as his guidance throughout its course.

Koneru Nitish : Led the conceptualization of the problem, implementation of the analytical solutions for the problem, assisting in the formulation of Picard and handled spacial mesh convergence, contributing to documentation of the results.

Ashmit Sinha : Focused on coding the part of implementing Picard iteration and the convergence studies of time and viscosity, proving that it is viable. Contributed overall to plotting and documentation.

References

[1] Nand Kishor Kumar. A review on burgers' equations and it's applications. *Journal of Applied Mathematics and Computational Mechanics.*

[2] Bilge Inan and Ahmet Refik Bahadir. Numerical solution of the one-dimensional burgers' equation: Implicit and fully implicit exponential finite difference methods. *Applied Mathematics and Computation*, 175(2):1243–1254, 2006.