Solution of Burgers' Equation using the Finite Volume Method and Runge-Kutta 4 Time-Stepping under Various Boundary Conditions

Liam Pohlmann[1,2]

[1]*Oak Ridge National Laboratory, Thermal Hydraulics Group*
[2]*University of New Mexico, Department of Nuclear Engineering*

August 9, 2024

Contents

Acknowledgement

This research was supported in part by an appointment to the Oak Ridge National Laboratory Research Student Internships Program, sponsored by the U.S. Department of Energy and administered by the Oak Ridge Institute for Science and Education. The author would also like to thank Drs. Vineet Kumar, Vivek Rao, and Arpan Sircar for their support of this project through its lifetime.

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

Overview

The purpose of this document is to outline the steps used to solve Burgers' Equation using the Finite Volume Method (FVM). Many texts exist on the FVM and its applications, though few truly walk the student through clear steps to achieve the desired setup. This paper aims to bridge the gap between textbooks and practice by detailing the steps to solve a difficult partial differential equation (PDE). Code may also be found in the form of a `git` repository titled `FVM-Burgers-Equation`.

Particular instructions will be given to solve for inhomogeneous Dirichlet boundary conditions, inhomogeneous Neumann boundary conditions, and periodic boundary conditions. This article is a set of student-made procedures, and thus should be used as a reference of steps taken for a numerical solution, *not* as a document for rigorous mathematical solutions.

Note to Students

This document was created as part of a personal project of an intern at Oak Ridge National Laboratory in Oak Ridge, Tennessee, United States of America during the summer of 2024. The code and systems outlined have undergone many iterations through trial and error over the course of about two weeks. True understanding of numerical methods and their strengths and limitations requires years of dedication to reading, listening, and experiencing the field. While this paper is not intended to take the place of courses in numerical methods, the author hopes that it may serve as a guide in any project involving numerical methods, particularly those for PDEs.

1 Problem Statement

Burgers' Equations is a 1-D partial differential equation (PDE) developed as a model to understand fluid flow. In particular, it allows for exploration of shocks and rarefactions in fluids, which are discontinuous solutions. It is given as [**?**]:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{1.1}$$

or, using different notation:

$$u_t + uu_x = \nu u_{xx} \tag{1.2}$$

We can physically interpret Equation (1.1)'s terms as:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \text{Time-dependent term}\\ u\frac{\partial u}{\partial x} &= \text{Advection term}\\ \nu\frac{\partial^2 u}{\partial x^2} &= \text{Diffusion term}\end{aligned}$$

For this paper, we choose the initial conditions to be a piecewise-defined step function:

$$u(x,0) = \begin{cases} u_L, & x < L/2 \\ u_R, & x \geq L/2 \end{cases} \tag{1.3}$$

in the domain $x \in [0, L]$, $t \in [0, T]$. To close the system, the following three boundary conditions will be considered:

1. Inhomogeneous Dirichlet
2. Inhomogeneous Neumann
3. Periodic

These will be discussed in further detail in Sections 5.1 to 5.3.

2 Finite Volume Method

In the finite volume method (FVM), the spatial domain is discretized into a set of volumes.[1] In the FVM (unlike the finite element method (FEM)), information is stored at the *center* of the volume. In essence, we are instead solving for $\bar{u}$, which is defined as:

$$\begin{aligned} \bar{u} &= \frac{\int_{\Omega_V} u \mathrm{d}\Omega_V}{\int_{\Omega_V} \mathrm{d}\Omega_V} \\ &= \frac{1}{V} \int_{\Omega_V} u \mathrm{d}\Omega_V \end{aligned} \tag{2.1}$$

where Ω_V defines the spatial domain of a single arbitrary volume in the global domain, Ω (see Figure 1 for a graphic representation). The second line in Equation (2.1) simply defines $V \equiv \int_{\Omega_V} \mathrm{d}\Omega_V$ and simplifies. Notice that Equation (2.1) defines the *volume-averaged* value of u in the cell. In this, we have implicitly assumed that u varies only slightly in the volume, which is valid with a sufficiently small spatial discretization.

[1] In 1-D, the "volumes" are simply line segments.

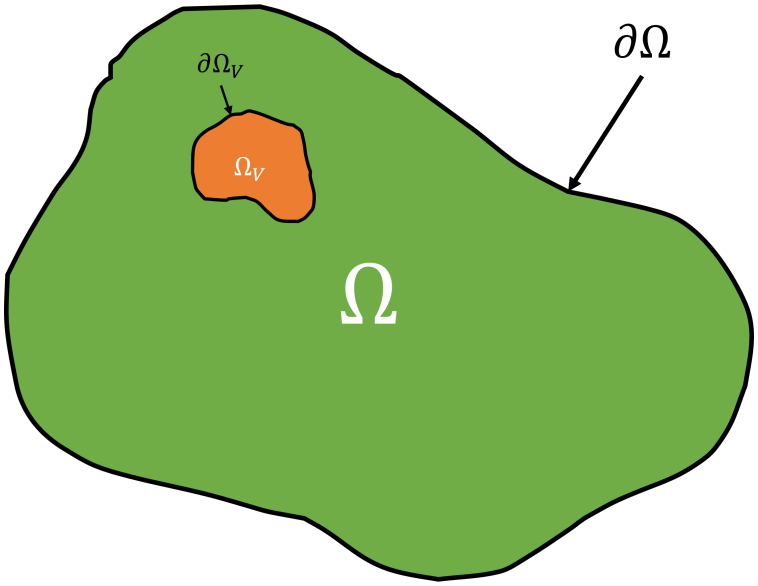


Figure 1: Arbitrary Domain and Volume Subdomain

Substituting Equation (2.1) into Equation (1.1) gives:

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} = \nu \frac{\partial^2 \bar{u}}{\partial x^2} \tag{2.2}$$

Keep in mind that Equation (2.1) has only integrated the function over a *discretized volume*, not the entire domain. This gives the average value of the function within the volume. We now integrate over the *entire* spatial domain[2]:

$$\int_\Omega \bar{u}_t \mathrm{d}\Omega + \int_\Omega \bar{u}\bar{u}_x \mathrm{d}\Omega = \int_\Omega \nu \bar{u}_{xx} \mathrm{d}\Omega \tag{2.3}$$

Equation (2.3) was written with a more general notation where Ω is some arbitrary domain. When the domain is in 1-D and has been discretized into m finite volumes, the system would instead be:

$$\sum_{i=0}^{m-1} \left[\int_{i-1/2}^{i+1/2} \bar{u}_t \mathrm{d}\left(\Omega_V\right)_i + \int_{i-1/2}^{i+1/2} \bar{u}\bar{u}_x \mathrm{d}\left(\Omega_V\right)_i \right] = \sum_{i=0}^{m-1} \left[\int_{i-1/2}^{i+1/2} \nu \bar{u}_{xx} \mathrm{d}\left(\Omega_V\right)_i \right] \tag{2.4}$$

The index i represents the *center* of the volume, so $i \pm 1/2$ is boundary of the volume. Applying the divergence theorem, which is defined as [**?**][3]:

$$\iiint_\Omega \nabla \cdot \vec{f} \mathrm{d}V = \oiint_{\partial\Omega} \vec{f} \cdot \hat{n} \mathrm{d}S \tag{2.5}$$

[2]This is the main step of the FVM, and it carries with it important properties of conservation. This means that for applications such as fluid mechanics, we guarantee properties such as mass will be conserved, leading to a physically-representative solution.

[3]This may seem like an overcomplicated step, but it is a ubiquitous one in FVM methods, particularly for more complicated equations such as the Navier-Stokes. It is introduced this way to make sure the reader understands the importance of this step, trivial as it may be for a 1-D case.

where $\vec{f}$ is some vector-valued function, Ω is an arbitrary domain and $\partial\Omega$ is the domain's boundary, gives:

$$\bar{u}_t \Delta x + \left(\left(\frac{\bar{u}^2}{2} \right)_{i+1/2} - \left(\frac{\bar{u}^2}{2} \right)_{i-1/2} \right) = \nu \left(\left. \frac{\partial \bar{u}}{\partial x} \right|_{i+1/2} - \left. \frac{\partial \bar{u}}{\partial x} \right|_{i-1/2} \right) \tag{2.6}$$

We note that a shortcut has been taken in the first integral, as:

$$\begin{aligned} \int_{i-1/2}^{i+1/2} \bar{u}_t \mathrm{d}\left(\Omega_V\right)_i &= \bar{u}_t \int_{i-1/2}^{i+1/2} \mathrm{d}\left(\Omega_V\right)_i \\ &= \bar{u}_t V_i \end{aligned}$$

where V_i is the volume of i^{th} volume. V_i in 1-D is simply the length along the domain. More on this will be discussed in Section 5, but for now, assume the domain has been discretized into equally-spaced 1-D volumes of length Δx.

Herein lies a conundrum. To solve Equation (2.6), we must know both the functional values and the derivatives *at the boundaries*. As we discussed with Equation (2.1), we only know the average value of the cell. Necessarily, we assume (and oftentimes *expect*) the solution to be discontinuous on the volume boundaries.[4]

We now look to approximation techniques to describe these values. The derivatives at the boundaries can be approximated in a straightforward manner with finite differencing. To derive the stencil, we first expand $\left. \frac{\partial u}{\partial x} \right|_{i\pm1/2} = \left. u_x \right|_{i\pm1/2}$ using two Taylor series [**?**]:

$$\left. u_x \right|_{i+1/2} = \left. u_x \right|_i + \left(\frac{\Delta x}{2} \right) \left. u_{xx} \right|_i + \frac{1}{2!} \left(\frac{\Delta x}{2} \right)^2 \left. u_{xxx} \right|_i + \mathcal{O}\Big((\Delta x)^3 \Big) \tag{2.7a}$$

$$\left. u_x \right|_{i-1/2} = \left. u_x \right|_i - \left(\frac{\Delta x}{2} \right) \left. u_{xx} \right|_i + \frac{1}{2!} \left(\frac{\Delta x}{2} \right)^2 \left. u_{xxx} \right|_i + \mathcal{O}\Big((\Delta x)^3 \Big) \tag{2.7b}$$

Note that the subscript x in u_x indicates a derivative with respect to x, so $u_{xx} = \frac{\partial^2 u}{\partial x^2}$ and $u_{xxx} = \frac{\partial^3 u}{\partial x^3}$. The symbol $\mathcal{O}()$ is known as "big-O" notation, and it describes the dominant term in an infinite series. Because $(\Delta x)^3 > (\Delta x)^4 > \ldots$, we say that the error terms increase on an "order of $(\Delta x)^3$." We then subtract Equation (2.7b) from Equation (2.7a) to get:

$$\left. u_x \right|_{i+1/2} - \left. u_x \right|_{i-1/2} = \Delta x \left. u_{xx} \right|_i + \mathcal{O}\big((\Delta x)^3 \big) \tag{2.8}$$

We must now find an approximation for $\left. u_{xx} \right|_i$. Fortunately, because we simply must find the value of the derivative *at a volume center*, we can look to resources such as [**?**] which have already derived high-order methods, such as 2-point central differencing:

$$\left. u_{xx} \right|_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} + \mathcal{O}\big((\Delta x^2) \big) \tag{2.9}$$

[4] A perfectly continuous case would imply a steady solution with no spatial variation.

Using Equation (2.9), we finally arrive at:

$$u_x\Big|_{i+1/2} - u_x\Big|_{i-1/2} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x} + \mathcal{O}\big((\Delta x)^3\big) \tag{2.10}$$

The nonlinear terms in Equation (2.6), however, prove to be much more difficult. Let us define $f \equiv \frac{u^2}{2}$ and rewrite Equation (2.6):

$$u_t + \frac{1}{\Delta x}\left(f_{i+1/2} - f_{i-1/2}\right) = \frac{1}{\Delta x}\nu\left(\frac{\partial \bar{u}}{\partial x}\Big|_{i+1/2} - \frac{\partial \bar{u}}{\partial x}\Big|_{i-1/2}\right) \tag{2.11}$$

Substituting Equation (2.10) to the right hand side gives:

$$u_t + \frac{1}{\Delta x}\left(f_{i+1/2} - f_{i-1/2}\right) = \frac{\nu}{(\Delta x)^2}\left(u_{i+1} - 2u_1 + u_{i-1}\right) + \mathcal{O}\Big((\Delta x)^2\Big) \tag{2.12}$$

The terms $f_{i\pm1/2}$ are known as *numerical flux* terms, and are defining characteristics of advection equations. They will be discussed in the next section. Additionally, the bar seen above the function terms u will be omitted for brevity, though the reader is advised to remember that all solutions discussed will be volume-averaged values defined by Equation (2.1).

Equation (2.12) thus far has only made one approximation, and that approximation has an error that scales quadratically with the spatial step size. In order to not waste our efforts in achieving 2^{nd}-order accuracy, we must similarly choose time and flux discretizations that are *at least* 2^{nd}-order accurate.

3 Numerical Flux Selection

The numerical fluxes given in Equation (2.12) are not able to be numerically solved in their current form. We must write them in terms of volume centers (i.e. in terms of integer values of i, not $i \pm 1/2$). To do this, we look to the wisdom passed along to us by mathematicians.

It is of great importance to select a flux scheme that not only captures the data accurately, but also prevents the nonlinear terms from blowing up completely. Some flux schemes that may be of interest:

1. Upwind
2. Lax-Friedrichs
3. Lax-Wendroff
4. Godunov

Analyses and explanations of the above schemes would require a significant divergence from the goal of this paper, so the reader is referred to resources such as [**?**] for more information.

Because of its 2nd-order accuracy and reduced dissipative properties, the Lax-Wendroff flux scheme was selected. As mentioned by [**?**], the Lax-Wendroff method is not unique for a nonlinear PDE like it is for a linear PDE. One particular method was given by Richtmyer, and in it, the flux is evaluated in two steps. This is given as:

$$u_{i+1/2}^{k+1/2} = \frac{1}{2}\left(u_i^k + u_{i+1}^k\right) + \frac{1}{2}\frac{\Delta t}{\Delta x}\left(f_i^k - f_{i+1}^k\right) \tag{3.1a}$$

$$f_{i+1/2} = f\left(u_{i+1/2}^{k+1/2}\right) \tag{3.1b}$$

With the fluxes now able to be evaluated, all that is left is to discretize the time derivative and assemble the volumes into a system of equations.

4 Time Discretization

There are many methods derived by mathematicians that could be implemented to approximate our time derivative. An obvious case might be to approximate with forward differencing (also known as the Forward Euler method), though this method suffers greatly from instability and only first-order accuracy. Another option is backward differencing (also known as the Backward Euler method) which solves the issue of stability because of its implicit nature, though it also suffers from first-order accuracy. This stability, however, is at the cost of inverting a system. Due to the nonlinear nature of Equation (1.1), solving each time step would require root-finding algorithms such as Newton-Raphson, which adds a significant layer of computational complexity.

To handle time-stepping, we instead look to the tried-and-true Runge-Kutta methods. These methods look to handle time stepping in a forward Euler-like method, where one instead approximates the derivative with a weighted sum. These take the form of:

$$u^{k+1} = u^k + m^* \times (\Delta t). \tag{4.1}$$

Where Δt is the size of a single time step. In particular, the Runge-Kutta 4 (RK4) method is implemented. For some differential equation $\frac{\mathrm{d}u}{\mathrm{d}t} = g(u(t), t)$, given $u(t_k) = u^k$ and $t_k = t_0 + k \times (\Delta t)$

for $k \in [0, n]$ time steps [**?**]:

$$\begin{aligned} k_1 &= g\left(u^k, t_k\right) \\ k_2 &= g\left(u^k + k_1\frac{\Delta t}{2}, t_k + \frac{\Delta t}{2}\right) \\ k_3 &= g\left(u^k + k_2\frac{\Delta t}{2}, t_k + \frac{\Delta t}{2}\right) \\ k_4 &= g\left(u^k + k_3\Delta t, t_k + \Delta t\right) \end{aligned} \tag{4.2}$$

$$u^{k+1} = u^k + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \times (\Delta t)$$

In this case, $m^* = \frac{k_1+2k_2+2k_3+k_4}{6}$. The RK4 method is $\mathcal{O}\big((\Delta t)^4\big)$ globally [**?**], meaning we are much more likely to be restricted in accuracy by our spatial discretization rather than temporal. Note that in applying Equation (4.2) to Equation (1.1), the values u^{k+1},u^k,and $k_{1,2,3,4}$ are all *vectors*. These vectors contain the function values at each point in space at a particular time and will be denoted with bold font going forward.

To implement Equation (4.2), we simply let $g(u^k, t_k)$ equal Equation (2.12), solve for the flux terms, $\mathbf{k_1}, \mathbf{k_2}, \mathbf{k_3}, \mathbf{k_4}$, calculate $\mathbf{m}^*$, and finally take a timestep.

5 Boundary Conditions and Implementation

It is at this point in the solving process that we are finally able to discuss the actual setup of the numerical methods. Since the entire point of numerical methods is to convert a differential equation into a set of much-simpler algebraic equations, a brief discussion on the domain discretization is imperative.

One of the major benefits of the FVM (and the reason why it is implemented in commercial Computational Fluid Dynamics (CFD) codes) is its enforcement of conservation laws regardless of the geometry. This allows for arbitrary, irregular shapes to be used to discretize the domain, an advantage particularly useful when the computation domain is very complicated.[5] The discretized domain is known as a *mesh*. Creating complicated meshes is well-outside the scope of this text, and professionals have worked over the years to develop algorithms to do this [**?**]. For our 1-D case, we will use a simple, constant-interval method, where each volume will be the same size.

For now, we say that Δt and Δx are specified. It is clear, then, that $m = L/\Delta x + 1$ and $n = T/\Delta t + 1$, which are the total number of volume centers and time points, respectively. If we were to start to build the system for $\mathbf{k}_1$ using the approximation in Equation (2.12), the equations

[5] such as when the domain is generated with help of Computer Aided Disign (CAD)

would be:

$$\mathbf{k}_1 = \begin{bmatrix} \frac{1}{\Delta x}\left(f_{-1/2} - f_{1/2}\right) + \frac{\nu}{(\Delta x)^2}\left(u_1 - 2u_0 + u_{-1}\right) \\ \frac{1}{\Delta x}\left(f_{1/2} - f_{3/2}\right) + \frac{\nu}{(\Delta x)^2}\left(u_2 - 2u_1 + u_0\right) \\ \frac{1}{\Delta x}\left(f_{3/2} - f_{5/2}\right) + \frac{\nu}{(\Delta x)^2}\left(u_3 - 2u_2 + u_1\right) \\ \vdots \\ \frac{1}{\Delta x}\left(f_{m-5/2} - f_{m-3/2}\right) + \frac{\nu}{(\Delta x)^2}\left(u_{m-1} - 2u_{m-2} + u_{m-3}\right) \\ \frac{1}{\Delta x}\left(f_{m-3/2} - f_{m-1/2}\right) + \frac{\nu}{(\Delta x)^2}\left(u_m - 2u_{m-1} + u_{m-2}\right) \end{bmatrix} \tag{5.1}$$

Though a good start, we realize there is an issue: there are two values in this system, u_{-1} and u_m, that are not within the domain of the problem (including their appearances in the flux values). The way to address these values comes down to the kind of boundary conditions specified.

Another issue with Equation (5.1) is that it does not represent a "system" as we know in linear algebra. This can be fixed by specifying a couple of vectors of values and separating the coefficients.

$$\mathbf{k}_1 = \frac{\nu}{(\Delta x)^2}\begin{bmatrix} \cdots & & & & & & & \\ 1 & -2 & 1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & & 1 & -2 & 1 & \\ & & & & & & \cdots \end{bmatrix}\mathbf{u}^k + \frac{1}{(\Delta x)}\mathbf{f}^k_{-1/2} - \frac{1}{(\Delta x)}\mathbf{f}^k_{+1/2} \tag{5.2}$$

where:

$$\mathbf{u}^k = \begin{bmatrix} u_0^k \\ u_1^k \\ u_2^k \\ \\ \vdots \\ \\ \\ u_{m-1}^k \end{bmatrix}, \quad \mathbf{f}^k = \begin{bmatrix} f_0^k \\ f_1^k \\ f_2^k \\ \\ \vdots \\ \\ \\ f_{m-1}^k \end{bmatrix}. \tag{5.3}$$

These vectors both have a length of m, where m is the number of volumes. To evaluate the fluxes, we look back to Equations (3.1a) and (3.1b). Because $f \equiv \frac{1}{2}u^2$, all we need to do is write out systems for $\mathbf{u}^{k+1/2}_{\pm 1/2}$.

$$\mathbf{u}^{k+1/2}_{+1/2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & & & \\ & 1 & 1 & & & & & \\ & & 1 & 1 & & & & \\ & & & \ddots & \ddots & & & \\ & & & & & 1 & 1 & \\ & & & & & & \cdots \end{bmatrix}\mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x}\begin{bmatrix} 1 & -1 & & & & & & \\ & 1 & -1 & & & & & \\ & & 1 & -1 & & & & \\ & & & \ddots & \ddots & & & \\ & & & & & 1 & -1 & \\ & & & & & & \cdots \end{bmatrix}\mathbf{f}^k \tag{5.4a}$$

$$\mathbf{u}_{-1/2}^{k+1/2} = \frac{1}{2}\begin{bmatrix} \cdots & & & & & & \\ 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & 1 & 1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & 1 \end{bmatrix} \mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x}\begin{bmatrix} \cdots & & & & & & \\ 1 & -1 & & & & & \\ & 1 & -1 & & & & \\ & & 1 & -1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & -1 \end{bmatrix} \mathbf{f}^k \tag{5.4b}$$

The reader may notice that the top and bottom rows of the matrices in Equations (5.4a), (5.4b) and (5.2) are not filled in. This is to emphasize that we cannot create a system of equations describing a domain outside of the problem statement. What to put in these rows will be discussed in Sections 5.1 to 5.3.

5.1 Dirichlet Boundary Conditions

A Dirichlet boundary condition is where the values of the function are specified on the boundary. This is completes the problem statement in Section 1:

Find u which solves Equation (1.1) and satisfies:

$$\begin{aligned} u(0,t) &= u_L \\ u(L,t) &= u_R \\ u(x,0) &= \begin{cases} u_L, & x < L/2 \\ u_R, & x \geq L/2 \end{cases} \end{aligned} \tag{5.5}$$

for $x \in [0, L]$, $t \in [0, T]$.

Though simple in concept, determining the numerical scheme is more challenging than it may seem. Necessarily, we must separate the known quantities in $\mathbf{u}^k$ and $\mathbf{f}^k$. To accomplish this, the reader is referred to Appendix A and is encouraged to work through the algebra themselves. Once separated, Equation (5.2) is now:

$$\mathbf{k}_1 = \frac{1}{\Delta x}\left((\mathbf{f}^*)_{-1/2}^k - (\mathbf{f}^*)_{1/2}^k\right) + \frac{\nu}{(\Delta x)^2}\begin{bmatrix} u_L \\ 0 \\ \vdots \\ 0 \\ u_R \end{bmatrix} + \frac{\nu}{(\Delta x)^2}\begin{bmatrix} -2 & 1 & & & & & & \\ 1 & -2 & 1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & & 1 & -2 & 1 & \\ & & & & & & 1 & -2 \end{bmatrix} (\mathbf{u}^*)^k \tag{5.6}$$

where:

$$(\mathbf{u}^*)^k = \begin{bmatrix} u_1^k \\ u_2^k \\ u_3^k \\ \\ \vdots \\ \\ u_{m-2}^k \end{bmatrix}, \quad (\mathbf{f}^*)^k = \begin{bmatrix} f_1^k \\ f_2^k \\ f_3^k \\ \\ \vdots \\ \\ f_{m-2}^k \end{bmatrix} \tag{5.7}$$

The fluxes are calculated using Equations (5.4a) and (5.4b):

$$\begin{aligned} \mathbf{u}_{+1/2}^{k+1/2} = \frac{1}{2} &\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix} (\mathbf{u}^*)^k + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ u_R \end{bmatrix} + \\ &+ \frac{1}{2}\frac{\Delta t}{\Delta x} \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -1 \\ & & & & & \cdots \end{bmatrix} (\mathbf{f}^*)^k + \frac{1}{2}\frac{\Delta t}{\Delta x} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -f_R \end{bmatrix} \end{aligned} \tag{5.8a}$$

$$\begin{aligned} \mathbf{u}_{-1/2}^{k+1/2} = \frac{1}{2} &\begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix} (\mathbf{u}^*)^k + \frac{1}{2} \begin{bmatrix} u_L \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} + \\ &+ \frac{1}{2}\frac{\Delta t}{\Delta x} \begin{bmatrix} -1 & & & & & \\ 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & -1 \\ & & & & & 1 \end{bmatrix} (\mathbf{f}^*)^k + \frac{1}{2}\frac{\Delta t}{\Delta x} \begin{bmatrix} f_L \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{5.8b}$$

It is noted that the lengths of $(\mathbf{u}^*)^k$ and $(\mathbf{f}^*)^k$ are both $m-2$, where m is the number of volumes.

5.2 Neumann Boundary Conditions

Another case is where we specify not the values of the function, but the function's derivative at the boundary. For this paper, we look to satisfy inhomogeneous Neumann boundary conditions; that is, we set the derivatives at the boundaries to a value other than 0. The problem becomes:

Find u which solves Equation (1.1) and satisfies:

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_0 &= \alpha \\ \left.\frac{\partial u}{\partial x}\right|_L &= \beta \\ u(x,0) &= \begin{cases} u_L, & x < L/2 \\ u_R, & x \geq L/2 \end{cases} \end{aligned} \tag{5.9}$$

for $x \in [0, L]$, $t \in [0, T]$.

At this point, we realize that the above condition does not directly help us in solving Equation (2.2), as we are interested in solving for the *function*, not its derivative. However, using finite differencing techniques, we can put the derivative in terms of the function. In particular, the *central differencing* technique can be used to achieve 2nd-order accuracy [**?**].

$$\frac{\mathrm{d}u}{\mathrm{d}x} = \frac{u_{i+1} - u_{i-1}}{2(\Delta x)} + \mathcal{O}\big((\Delta x)^2\big) \tag{5.10}$$

To get the derivative at $i = [0, m-1]$, we perform basic algebra to find:

$$\begin{aligned} \left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_0 = \alpha \approx \frac{u_1 - u_{-1}}{2(\Delta x)} &\Rightarrow u_{-1} \approx u_1 - 2\alpha(\Delta x) \\ \left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_L = \alpha \approx \frac{u_m - u_{m-2}}{2(\Delta x)} &\Rightarrow u_m \approx u_{m-2} + 2\beta(\Delta x) \end{aligned} \tag{5.11}$$

For the boundary equations at $i = [0, m-1]$, we plug in Equation (5.11) to the first and last lines of Equation (5.1) and get:

$$\begin{aligned} k_1^{i=0} &= \frac{\nu}{(\Delta x)^2}\left(2u_1 - 2u_0\right) - \frac{2\nu\alpha}{\Delta x} \\ k_1^{i=m-1} &= \frac{\nu}{(\Delta x)^2}\left(2u_{m-2} - 2u_{m-1}\right) + \frac{2\nu\beta}{\Delta x} \end{aligned} \tag{5.12}$$

Applying this to Equation (5.2) gives:

$$\mathbf{k}_1 = \frac{\nu}{(\Delta x)^2}\begin{bmatrix} -2 & 2 & & & & & \\ 1 & -2 & 1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 \\ & & & & & 2 & -2 \end{bmatrix}\mathbf{u}^k + \frac{1}{\Delta x}\mathbf{f}^k_{-1/2} - \frac{1}{\Delta x}\mathbf{f}^k_{+1/2} + \frac{2\nu}{\Delta x}\begin{bmatrix} -\alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix} \tag{5.13}$$

In a similar process as Section 5.1, we derive the half-step values for the Lax-Wendroff fluxes using Equations (5.4a) and (5.4b):

$$\mathbf{u}^{k+1/2}_{+1/2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & 1 & 1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & 1 \\ & & & & & 1 & 1 \end{bmatrix}\mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x}\begin{bmatrix} 1 & -1 & & & & & \\ & 1 & -1 & & & & \\ & & 1 & -1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & -1 \\ & & & & & -1 & 1 \end{bmatrix}\mathbf{f}^k + \tag{5.14a}$$

$$+ \beta(\Delta x)\left(1 - \Delta t\beta\right)\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} - \beta(\Delta t)u^k_{m-2}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{u}^{k+1/2}_{-1/2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & 1 \end{bmatrix}\mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x}\begin{bmatrix} -1 & 1 & & & & \\ 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & -1 \end{bmatrix}\mathbf{f}^k + \tag{5.14b}$$

$$+ \alpha(\Delta x)\left(\Delta t\alpha - 1\right)\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \alpha(\Delta t)u^k_1\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

5.3 Periodic Boundary Conditions

When dealing with complicated geometries, it is often beneficial to take advantage of repeating patterns. This is accomplished by demanding *periodic* boundary conditions, which say that what comes through the domain will be repeated. Our problem statement thus becomes:

Find u which solves Equation (1.1) and satisfies:

$$\begin{aligned} u(0_-, t) &= u(L, t) \\ u(L_+, t) &= u(0, t) \\ u(x, 0) &= \begin{cases} u_L, & x < L/2 \\ u_R, & x \geq L/2 \end{cases} \end{aligned} \tag{5.15}$$

for $x \in [0, L]$, $t \in [0, T]$.

In numerical terms, this means $u_{-1} = u_{m-1}$ and $u_m = u_0$, closing our system. This will yield the following system:

$$\mathbf{k}_1 = \frac{\nu}{(\Delta x)^2} \begin{bmatrix} -2 & 1 & & & & & 1 \\ 1 & -2 & 1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & & 1 & -2 & 1 \\ 1 & & & & & 1 & -2 \end{bmatrix} \mathbf{u}^k + \frac{1}{\Delta x}\mathbf{f}^k_{-1/2} - \frac{1}{\Delta x}\mathbf{f}^k_{+1/2} \tag{5.16}$$

This case is much easier to implement, and the half-step flux inputs are calculated in a much more straightforward manner:

$$\mathbf{u}^{k+1/2}_{+1/2} = \frac{1}{2} \begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & 1 & 1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & 1 \\ 1 & & & & & & 1 \end{bmatrix} \mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x} \begin{bmatrix} 1 & -1 & & & & & \\ & 1 & -1 & & & & \\ & & 1 & -1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & -1 \\ -1 & & & & & & 1 \end{bmatrix} \mathbf{f}^k \tag{5.17a}$$

$$\mathbf{u}^{k+1/2}_{-1/2} = \frac{1}{2} \begin{bmatrix} 1 & & & & & & 1 \\ 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & & & 1 & 1 \end{bmatrix} \mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x} \begin{bmatrix} -1 & & & & & & 1 \\ 1 & -1 & & & & & \\ & 1 & -1 & & & & \\ & & \ddots & \ddots & & & \\ & & & & & 1 & -1 \end{bmatrix} \mathbf{f}^k \tag{5.17b}$$

6 Results

To showcase the implementation of the above schemes, numerical values were prescribed in Python code. The numerical conditions are as follows:

$$\begin{aligned} &\Delta t = 0.001, \quad \frac{\Delta t}{\Delta x} = 0.2 \\ &\nu = 10^{-2}, \quad L = 20, \quad T = 0.5 \\ &u_L = 0.5, \quad u_R = 1 \\ &\alpha = -1, \quad \beta = -1 \end{aligned} \tag{6.1}$$

The quantity $\Delta t/\Delta x$ is known as the CFL[6] number and is commonly seen in stability analyses of numerical methods. It is generally introduced using a simple linear problem, such as:

$$\frac{\partial u}{\partial x} + a\frac{\partial u}{\partial x} = 0 \tag{6.2}$$

The CFL number for Equation (6.2) would be $a\frac{\Delta t}{\Delta x}$ for a first-order explicit upwind scheme.[7] [**?**] For many schemes, a CFL number of less 1 is safe from instability while time-stepping. If this number is too large, meaning $\Delta t \geq \Delta x$, we would likely find

With the above specifications, each computation began with the same initial condition (see Figure 2). For convenience, intermediate steps of $t = 0.10$ (Figure 3) and $t = 0.25$ (Figure 4) have been included along with the final time, $t = 0.50$ (Figure 5). Additional solution sets can be created by running the code in the repository; however, only $\nu = 10^{-2}$ is included in this report. Additionally, the analytical solution adapted from [**?**] for Dirichlet boundary conditions has been plotted as a proof of methods.[8] It is given as:

$$u(x,t) = \frac{u_R + u_L}{2} - \frac{u_L - u_R}{2}\tanh\left(\frac{(x_0 - x - st)\,(u_L - u_R)}{4\nu}\right) \tag{6.3}$$

where the shock speed, s, is defined as:

$$\begin{aligned} s &= \frac{f(u_L) - f(u_R)}{u_L - u_R} \\ &= \left(\frac{u_L^2}{2} - \frac{u_R^2}{2}\right) / \left(u_L - u_R\right) \\ &= \frac{u_L + u_R}{2} \end{aligned} \tag{6.4}$$

u_L and u_R are defined as in the initial conditions (Equation (1.3)) and $x_0 = 10$. Analytical solutions for Neumann and periodic boundary conditions are not included in this report, and thus

[6] "Courant-Friedrich-Lewy" number, also known as the "Courant" number.

[7] Do not let the math jargon frighten you – this just means a simple finite difference scheme.

[8] The solution given in [**?**] is specified for an infinite domain. Since this is computationally impossible, the reader is urged to take the analytical solution plotted as an approximation.

Equations (6.3) and (6.4) are only found in the Dirichlet code.

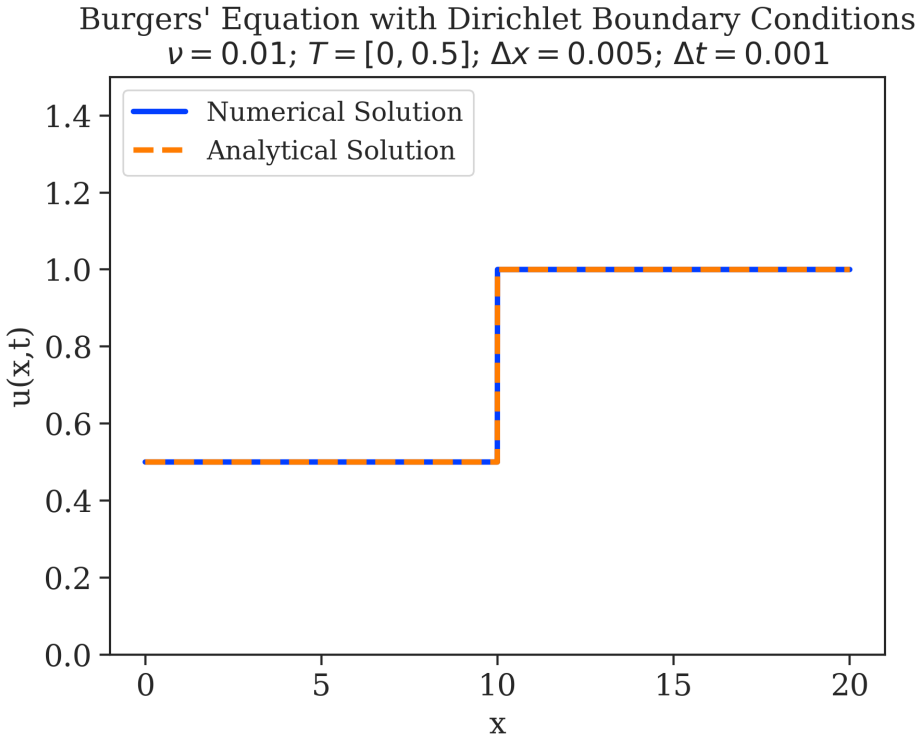
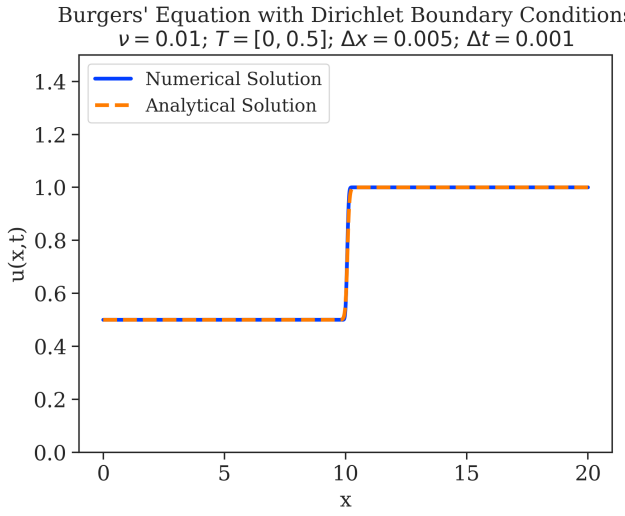
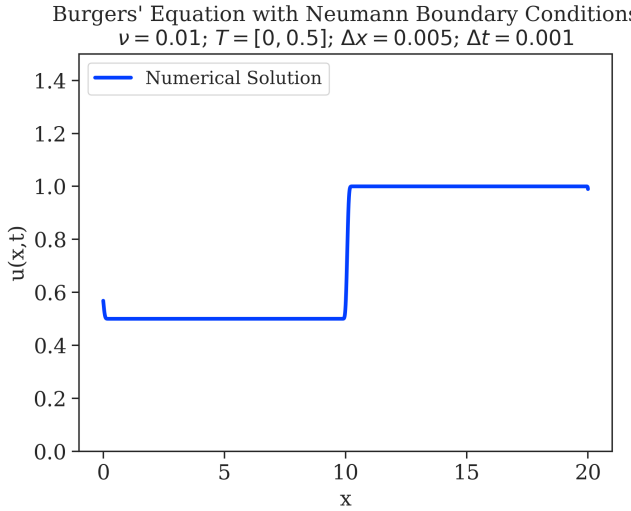


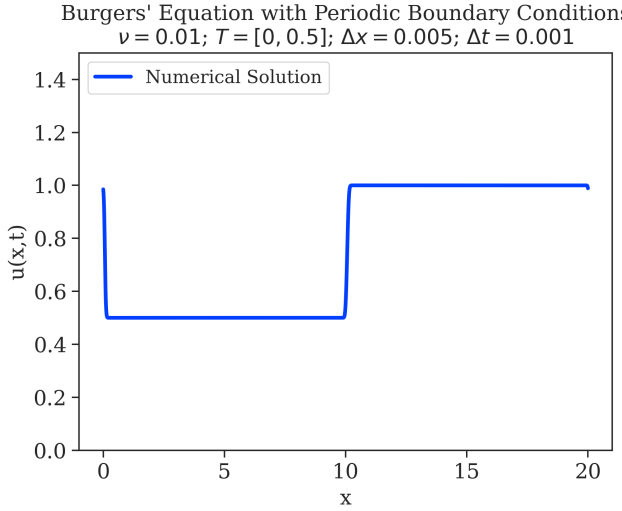
Figure 2: Implemented Initial Condition



(a) Inhomogeneous Dirichlet

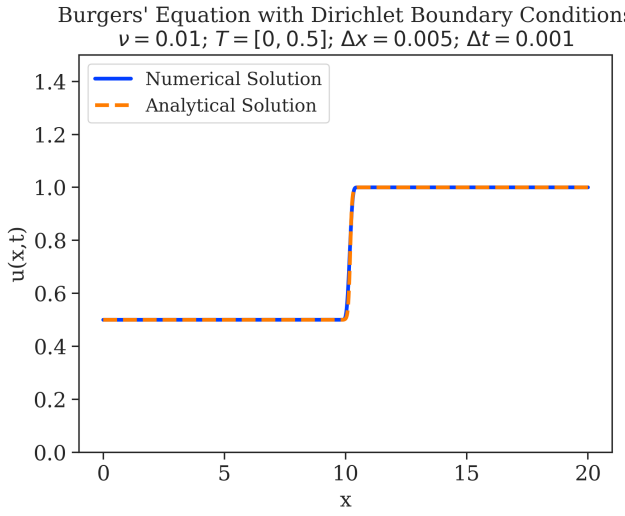


(b) Inhomogeneous Neumann

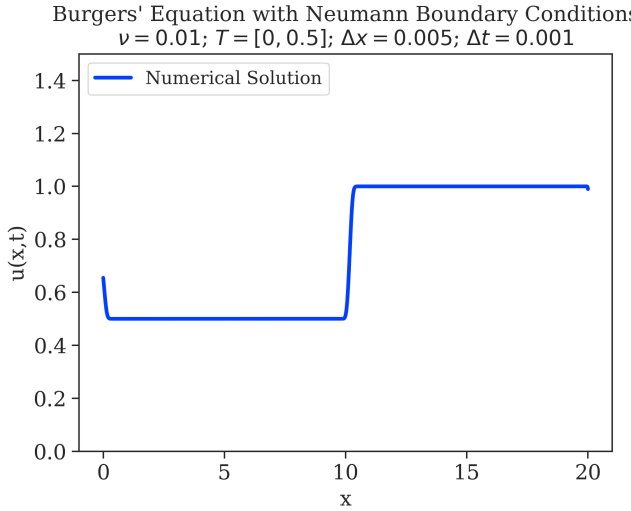


(c) Periodic

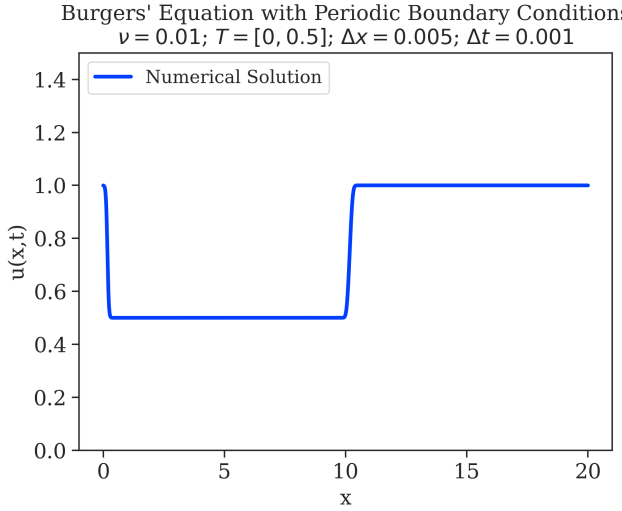
Figure 3: Implemented Conditions $t = 0.10$



(a) Inhomogeneous Dirichlet

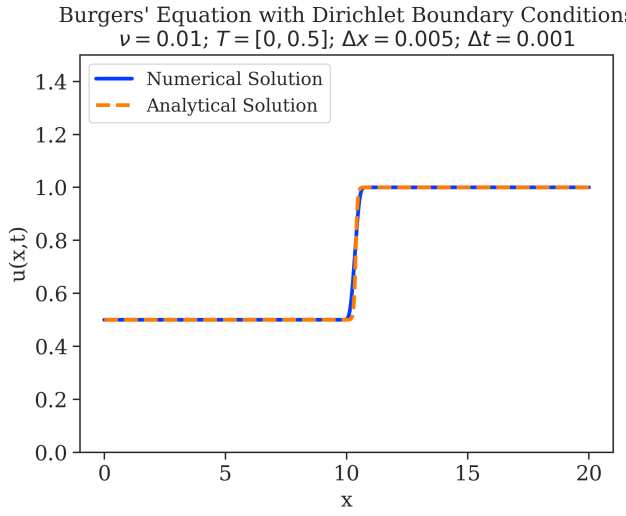


(b) Inhomogeneous Neumann

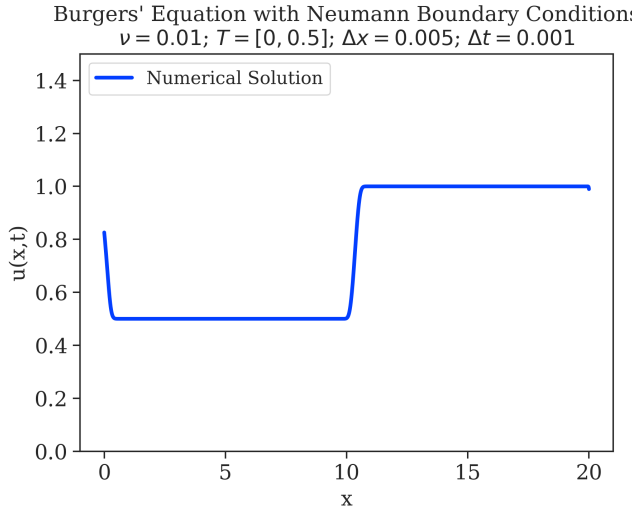


(c) Periodic

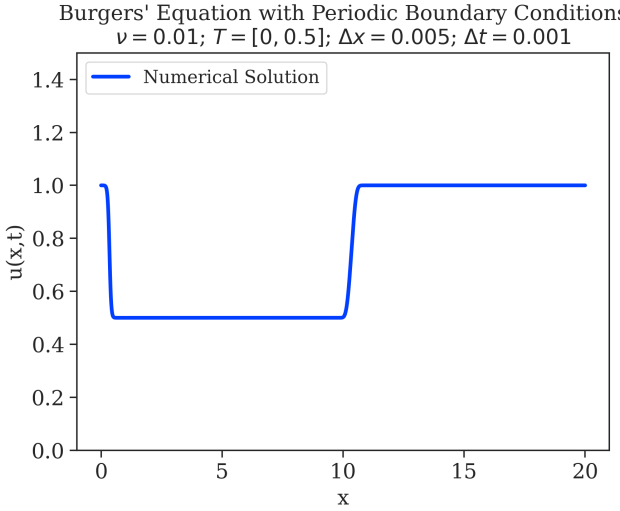
Figure 4: Implemented Conditions $t = 0.25$



(a) Inhomogeneous Dirichlet



(b) Inhomogeneous Neumann



(c) Periodic

Figure 5: Implemented Conditions $t = 0.50$

As mentioned in Section 1, Burgers' Equation was developed to study shocks and rarefactions that occur in fluids that occur due to the nonlinear advective term present in the Navier-Stokes equations. Since this paper's focus was on numerical methods and implementation, a mathematical analysis of these phenomena are not included. For more studies on shocks and Burgers' Equation, the author recommends exploring [**?**, **?**] among many other resources.

7 Conclusion and Next Steps

For any who have been able to run the included code above $\nu = 0.01$, you will notice that the solution rapidly diverges at the interface of the two velocities in the initial condition. Though it is not immediately clear what the cause of this instability is, the next steps will likely include explorations of higher-resolution spatial and flux schemes. A finer mesh may also greatly improve the quality of the solution, though this will be difficult to achieve without the aid of parallel computing to handle the large matrix-vector products. Finally, a conversion from an explicit time-stepping method (RK4) to an implicit method may also improve stability properties. As can be shown in the literature [**?**, **?**], the discontinuity present in the solution is a known computational difficulty, and its study will help greatly in understanding more complex equations such as the Navier-Stokes, which also have nonlinearities.

A Splitting a Matrix-Vector Multiply

Say we have a system $A\mathbf{x} = \mathbf{b}$, and we simply want to evaluate the left-hand-side to verify the result. If A is 3×3 where:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{A.1}$$

and:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{A.2}$$

then their product is:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} ax_1 + bx_2 + cx_3 \\ dx_1 + ex_2 + fx_3 \\ gx_1 + hx_2 + ix_3 \end{bmatrix} \tag{A.3}$$

which can also be expressed as:

$$\begin{aligned}\begin{bmatrix} ax_1 + bx_2 + cx_3 \\ dx_1 + ex_2 + fx_3 \\ gx_1 + hx_2 + ix_3 \end{bmatrix} &= \begin{bmatrix} ax_1 \\ bx_1 \\ cx_1 \end{bmatrix} + \begin{bmatrix} bx_2 + cx_3 \\ ex_2 + fx_3 \\ hx_2 + ix_3 \end{bmatrix} \\ &= \begin{bmatrix} a \\ b \\ c \end{bmatrix} x_1 + \begin{bmatrix} b & c \\ e & f \\ h & i \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}\end{aligned}. \tag{A.4}$$

The above is particularly useful when a value is known, such as in the case of Dirichlet boundary conditions for a PDE.