ORNL/XX-XXXX/XXX

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

Liam M. Pohlmann[1]

November 2024

Draft. Document has not been reviewed and approved for public release.

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/XX-XXXX/XXX

Nuclear Energy and Fuel Cycle Division

SOLUTION OF BURGERS' EQUATION USING THE FINITE VOLUME METHOD AND RUNGE-KUTTA 4 TIME-STEPPING UNDER VARIOUS BOUNDARY CONDITIONS

Liam M. Pohlmann[1]

November 11, 2024

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

CONTENTS

LIST OF FIGURES

Foreword

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.

1 Problem Statement

Burgers' equation is a 1D partial differential equation (PDE) developed as a model to understand fluid flow. This equation specifically enables the exploration of shocks and rarefactions in fluids, which are discontinuous solutions. It is given as (Salih 2016):

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

or, using different notation:

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

Equation (1.1)'s terms are physically understood 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 are 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, information is stored at the center of the volume. The goal has thus been changed from solving directly for u and now 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. Figure 1 provides a graphical representation. The second line in Equation (2.1) defines $V \equiv \int_{\Omega_V} \mathrm{d}\Omega_V$, the cell volume, and simplifies the first line. Notice that Equation (2.1) defines the *volume-averaged* value of u in the cell.

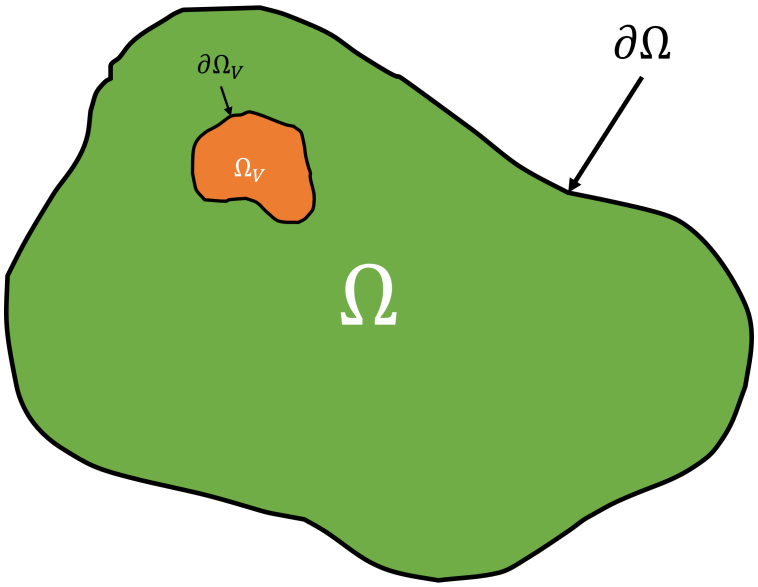


Figure 1. Arbitrary domain and volume subdomain.

The implicit assumption is that u varies only slightly in the volume, and this assumption is valid with a sufficiently small spatial discretization. Substituting Equation (2.1) into Equation (1.1) gives the following:

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

Equation (2.1) has only integrated the function over a discretized volume, not the entire domain, giving the average value of the function within the volume. Integrating over the entire spatial domain gives[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 1D and has been discretized into m finite volumes, the system would instead be written as:

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

The index i represents the center of the volume, so $i \pm 1/2$ is boundary of the volume. Applying the divergence theorem[3], which is defined as (Dawkins 2022):

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

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

2. This is the main step of the FVM, and it carries with it important properties of conservation. Therefore, for applications such as fluid mechanics, properties such as mass are guaranteed to be conserved, leading to a physically-representative solution.

3. This step may seem overcomplicated, but it is ubiquitous in FVM methods, particularly for more complicated equations such as the Navier-Stokes. This step is introduced this way to ensure the reader understands its importance, trivial as it may be for a 1D case.

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

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

It is noted that a shortcut has been taken in the first integral:

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

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

Herein lies a conundrum. To solve Equation (2.6), both the functional values and the derivatives at the boundaries must be known. As discussed with Equation (2.1), only the average value of the cell is known. Necessarily, the solution is assumed and often expected to be discontinuous on the volume boundaries.[4]

Approximation techniques are now used to describe these values. The derivatives at the boundaries can be approximated in a straightforward manner with finite differencing. To derive the stencil, one expands $\frac{\partial u}{\partial x}\big|_{i\pm1/2} = u_x\big|_{i\pm1/2}$ using two Taylor series (Yew 2011):

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

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

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 $O()$ is known as "big-O" notation, and it describes the dominant term in an infinite series. Because $(\Delta x)^3 > (\Delta x)^4 > \ldots$, it is said that the error terms increase on an "order of $(\Delta x)^3$." Subtracting Equation (2.7b) from Equation (2.7a) gives the following:

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

An approximation for $u_{xx}\big|_i$ must now be found. Fortunately, this process is simple because resources such as (Yew 2011) have already derived high-order methods, such as two-point central differencing, for finding the value of the derivative at a volume center.

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

Equation (2.9), finally arrives at the following:

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

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

The nonlinear terms in Equation (2.6), however, prove to be significantly more difficult. Defining $f \equiv \frac{u^2}{2}$ and rewriting Equation (2.6) gives the following:

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

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

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

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

Equation (2.12) has only one approximation, and that approximation has an error that scales quadratically with the spatial step size. In order to not waste efforts in achieving second-order accuracy, time and flux discretizations that are at least second-order accurate must be similarly chosen.

3 Numerical Flux Selection

The numerical fluxes in Equation (2.12) are not able to be numerically solved in their current form. These fluxes must be written in terms of volume centers (i.e., in terms of integer values of i, not $i \pm 1/2$).

A flux scheme that not only captures the data accurately but also prevents the nonlinear terms from blowing up completely must be selected. The following flux schemes 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 other resources such as (LeVeque 1992) for more information.

Because of its second-order accuracy and reduced dissipative properties, the Lax–Wendroff flux scheme was selected. The Lax–Wendroff method is not unique for a nonlinear PDE like it is for a linear PDE (Toro 2023). One particular method was given by Richtmyer. In this method, the flux is evaluated in two steps, given as the following:

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

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

The fluxes are 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), although 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; unfortunately, this method 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, as opposed to standard linear solvers such as LU factorization, which adds a significant layer of computational complexity.

To handle time-stepping, the tried-and-true Runge-Kutta methods are used instead. These methods handle time stepping in a Forward Euler-like method, in which the derivatives are approximated with a weighted sum. These methods take the following form:

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

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 (Cheever 2022):

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

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

In this case, $m^* = \frac{k_1+2k_2+2k_3+k_4}{6}$. The RK4 method is $O\left((\Delta t)^4\right)$ globally (Cheever 2022), meaning the accuracy is significantly more likely to be restricted by spacial discretization rather than temporal discretization. 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), 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 take a timestep.

5 Boundary Conditions and Implementation

At this point in the process, the actual setup of the numerical methods can be discussed. Because 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 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 computational 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 (Mazumder 2016). For this 1D case, a simple, constant-interval method was selected, where each volume will be the same size.

For now, Δt and Δx are specified. It is clear, then, that $m = L/\Delta x + 1$ (total number of volume centers) and $n = T/\Delta t + 1$ (total number of time points). If a system for $\mathbf{k}_1$ was built using the approximation in Equation (2.12), the equations would be represented as follows:

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

Although Equation (5.1) is a good start, two values, u_{-1} and u_m, raise an issue because they are not within the domain of the problem (including their appearances in the flux values). Ultimately, it will be the boundary conditions of the system which will determine these values.

Another issue with Equation (5.1) is that it does not represent a system as we known linear algebra. This issue 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:

$$\mathbf{u}^k = \begin{bmatrix} u^k_0 \\ u^k_1 \\ u^k_2 \\ \\ \vdots \\ \\ u^k_{m-1} \end{bmatrix}, \quad \mathbf{f}^k = \begin{bmatrix} f^k_0 \\ f^k_1 \\ f^k_2 \\ \\ \vdots \\ \\ f^k_{m-1} \end{bmatrix}. \tag{5.3}$$

These vectors both have a length of m, where m is the number of volumes. To evaluate the fluxes, Equa-

5. An example would be instances which the domain is generated with help of Computer Aided Disign (CAD).

tions (3.1a) and (3.1b) are used. Because $f \equiv \frac{1}{2}u^2$, all that is needed is to write out systems for $\mathbf{u}_{\pm 1/2}^{k+1/2}$.

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

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

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 a system of equations describing a domain outside of the problem statement cannot be created with the current information. Sections 5.1 to 5.3 discusses what to put in these rows.

5.1 Dirichlet Boundary Conditions

Dirichlet boundary conditions specify the values of the function on the boundary and complete the problem statement in Section 1:

Find u which solves Equation (1.1) *and satisfies the following:*

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

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

Although simple in concept, determining the numerical scheme is more challenging than it may seem. Necessarily, the known quantities in $\mathbf{u}^k$ and $\mathbf{f}^k$ must be separated from the unknown. Appendix A provides details on the necessary linear algebra. Once separated, Equation (5.2) now reads as follows:

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

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):

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

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

Note 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, an approach is presented for satisfying inhomogeneous Neumann boundary conditions; that is, the derivatives at the boundaries are set 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, the above condition does not directly help solve Equation (2.2) because the goal is to solve for the *function*, not its derivative. However, finite differencing techniques, can be used to put the derivative in terms of the function. In particular, the central differencing technique can be used to achieve second-order accuracy (Yew 2011).

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

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

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

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

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

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, the half-step values for the Lax–Wendroff fluxes are derived using

Equations (5.4a) and (5.4b):

$$\mathbf{u}_{+1/2}^{k+1/2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & & \\ & 1 & 1 & & & & \\ & & 1 & 1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & 1 \\ & & & & & 1 & 1 \end{bmatrix}\mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x}\begin{bmatrix} 1 & -1 & & & & & \\ & 1 & -1 & & & & \\ & & 1 & -1 & & & \\ & & & \ddots & \ddots & & \\ & & & & & 1 & -1 \\ & & & & & -1 & 1 \end{bmatrix}\mathbf{f}^k + \\ + \beta(\Delta x)\left(1 - \Delta t\beta\right)\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} - \beta(\Delta t)u_{m-2}^k\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{5.14a}$$

$$\mathbf{u}_{-1/2}^{k+1/2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & \\ 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & 1 \end{bmatrix}\mathbf{u}^k + \frac{1}{2}\frac{\Delta t}{\Delta x}\begin{bmatrix} -1 & 1 & & & & \\ 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & -1 \end{bmatrix}\mathbf{f}^k + \\ + \alpha(\Delta x)\left(\Delta t\alpha - 1\right)\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \alpha(\Delta t)u_1^k\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.14b}$$

5.3 Periodic Boundary Conditions

Taking advantage of repeating patterns is often helpful when dealing with complicated geometries. This is accomplished by demanding *periodic* boundary conditions, which say that what comes through the domain will be repeated. The 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 the system, which is now represented as:

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

This case is easier to implement than the prior cases, and the half-step flux inputs are calculated in a 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:

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

The quantity $\Delta t/\Delta x$ is known as the Courant–Friedrich–Lewy (CFL) 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.[6] (Caminha 2017) For many schemes, a CFL number of less 1 is safe from instability while time-stepping.

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

6. This just means a simple finite difference scheme.

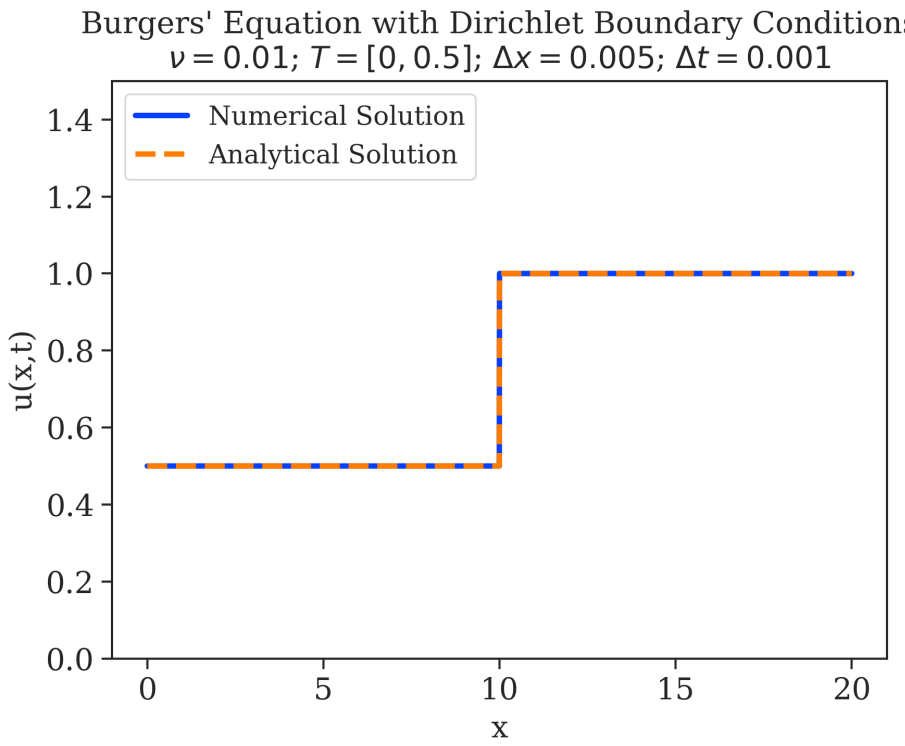


Figure 2. Implemented initial condition.

repository; however, only $\nu = 10^{-2}$ is included in this report. Additionally, the analytical solution adapted from (Cameron, n.d.) for Dirichlet boundary conditions has been plotted as a proof of methods.[7] The solution is given as:

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

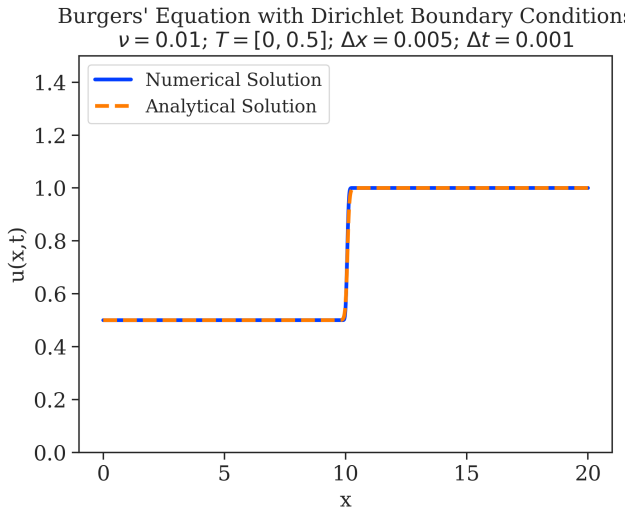
where the shock speed, s, is defined as:

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

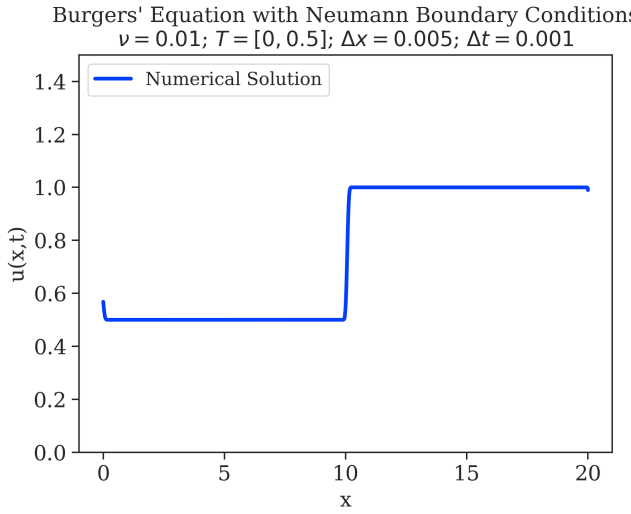
Variables u_L and u_R are defined as in the initial conditions (Equation (1.3)) and $x_0 = 10$. Analytical solutions for Neumann and periodic boundary conditions are not included in this report, so Equations (6.3) and (6.4) are only found in the Dirichlet code.

As mentioned in Section 1, Burgers' equation was developed to study shocks and rarefactions in fluids, which that occur due to the nonlinear advective term present in the Navier-Stokes equations. Because 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 (Salih 2016; Cameron, n.d.) among other resources.

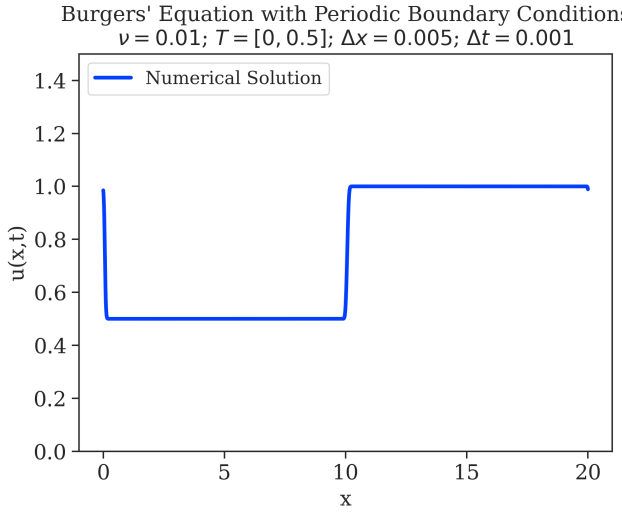
7. The solution given in (Cameron, n.d.) is specified for an infinite domain. Because this is computationally impossible, the reader is urged to take the analytical solution plotted as an approximation.



(a) Inhomogeneous Dirichlet

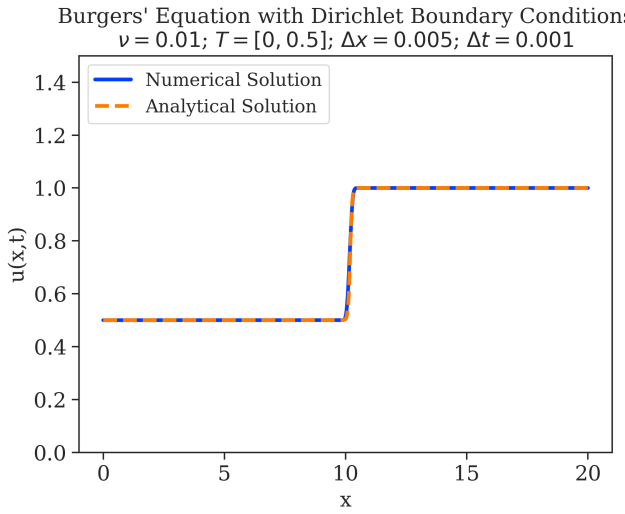


(b) Inhomogeneous Neumann

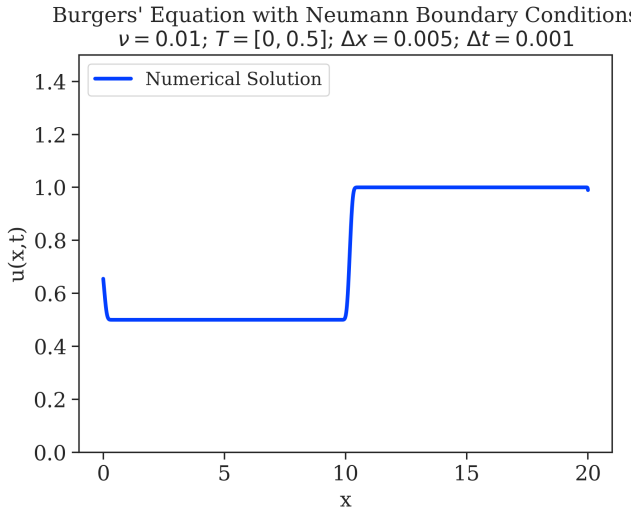


(c) Periodic

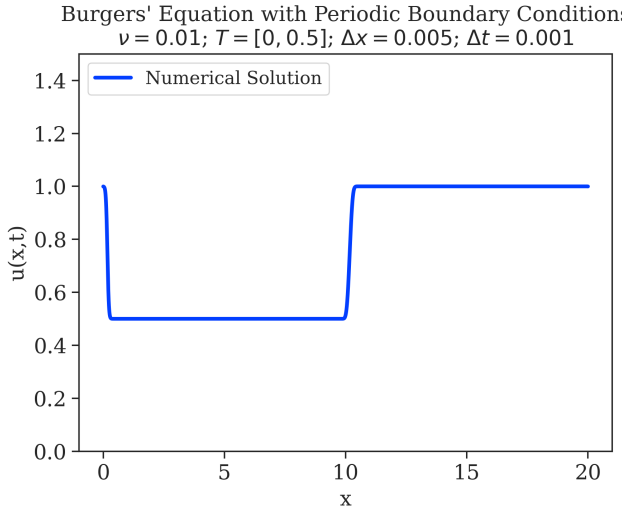
Figure 3. Implemented conditions $t = 0.10$**.**



(a) Inhomogeneous Dirichlet

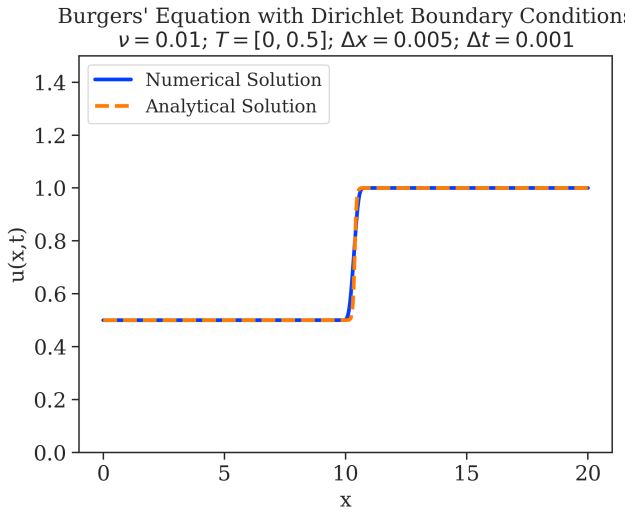


(b) Inhomogeneous Neumann

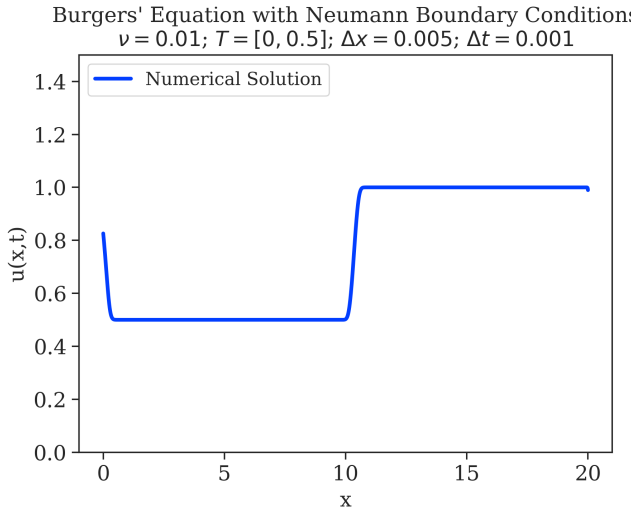


(c) Periodic

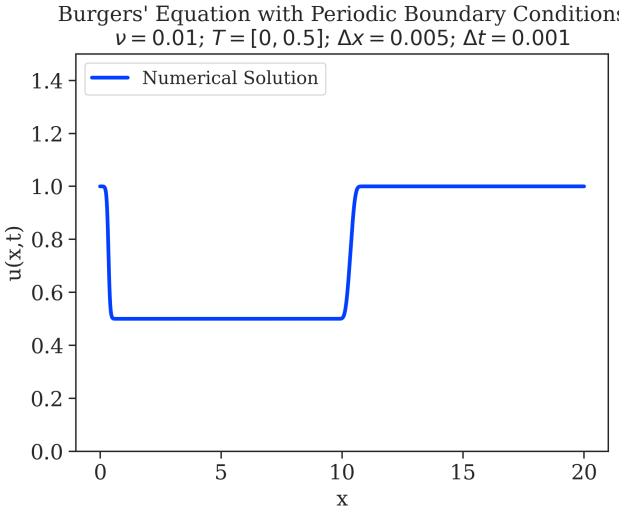
Figure 4. Implemented conditions $t = 0.25$**.**



(a) Inhomogeneous Dirichlet



(b) Inhomogeneous Neumann



(c) Periodic

Figure 5. Implemented conditions $t = 0.50$**.**

7 Conclusion and Next Steps

If the included code has been run with a viscosity value above $\nu = 0.01$, then the solution will rapidly diverge at the interface of the two velocities given in the initial condition. Although the cause of this stability is not immediately clear, 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, but this will be difficult to achieve without the aid of parallel computing to handle the large matrix-vector products. Finally, conversion from an explicit time-stepping method (RK4) to an implicit method may also improve stability properties. As shown in the literature (Cameron, n.d.; Salih 2016), the discontinuity present in the solution is a known computational difficulty. Studying this difficulty will greatly improve the understanding of more complex equations, such as the Navier-Stokes, that also have nonlinearities.

8 REFERENCES

Cameron, Maria. n.d. "NOTES ON BURGERS'S EQUATION."

Caminha, Guilherme. 2017. "CFL Condition: How to Choose Your Timestep Size." *SimScale* (August). Accessed August 8, 2024.

Cheever, Erik. 2022. "Fourth Order Runge-Kutta," accessed July 30, 2024.

Dawkins, Paul. 2022. "Calculus III - Divergence Theorem" (November 16, 2022). Accessed July 30, 2024.

LeVeque, Randall J. 1992. "Numerical Methods for Conservation Laws." In *Numerical Methods for Conservation Laws,* 2nd ed. 1992. Lectures in Mathematics. ETH Zürich. Basel: Birkhäuser Basel. ISBN: 3-0348-8629-2.

Mazumder, Sandip. 2016. *Numerical Methods for Partial Differential Equations.* Elsevier. ISBN: 978-0-12-849894-1.

Salih, A. 2016. "Burgers' Equation" (February). Accessed July 30, 2024.

Toro, E.F. 2023. "Lax–Wendroff Method - Encyclopedia of Mathematics" (March 26, 2023). Accessed August 2, 2024.

Yew, A. 2011. "Numerical Differentiation: Finite Differences."

A Splitting a Matrix-Vector Multiply

For a system $A\mathbf{x} = \mathbf{b}$, if $A \in \mathcal{R}^{3\times3}$ 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 partial differential equation.