1 Navier-Stokes Equation Solution Steps

This guide outlines a step-by-step approach to solving the Navier-Stokes equations, starting with one-dimensional (1D) analysis that includes linear and nonlinear convection, diffusion, and the Burgers' equation. It progresses to two-dimensional (2D) analysis, covering linear and nonlinear convection, diffusion, Burgers' equation, and introduces the Laplace and Poisson equations. The final part focuses on a detailed solution for the Navier-Stokes equation in 2D, particularly on cavity and channel flows.

1.1 Step 1: 1D Linear Convection

The 1D Linear Convection equation simplifies the Navier-Stokes equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{1}$$

Here, u represents the quantity transported by the flow at a constant speed c, similar to wave propagation.

1.1.1 Numerical Discretisation

The spatial domain x is divided into points, indexed from $i = 0$ to X, and time is segmented into intervals, indexed from $n = 0$ to T.
Within this framework:

- Δx indicates the distance between consecutive spatial points.
- Δt represents the time elapsed between successive steps.

1.1.2 Numerical Methods

The approximation of derivatives employs two main methods:

- The **space derivative**, uses a backward difference formula:

$$\frac{\partial u}{\partial x} = \frac{u_i^n - u_{i-1}^n}{\Delta x} \tag{2}$$

- The **time derivative**, uses a forward difference formula:

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t} \tag{3}$$

The discretised convection equation is:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{4}$$

Solving for u_i^{n+1} yields:

$$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) \tag{5}$$

1.1.3 Initial and Boundary Conditions

The initial condition is defined as a piecewise function over the spatial domain

$$u(x,0) = \begin{cases} 2 & \text{if } 0.5 \leq x \leq 1 \\ 1 & \text{otherwise} \end{cases}$$

The boundary conditions are set to ensure that $u = 1$ at the outer edges of the spatial domain, namely at $x = 0$ and $x = 2$. This setup initiates the model with a square wave configuration.

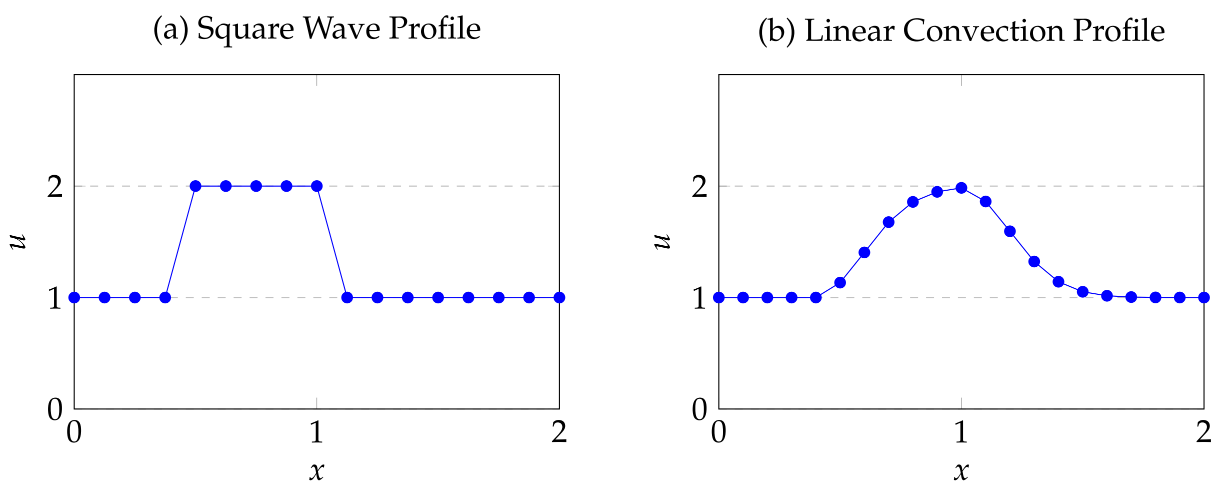

Figure 1: Square Wave and Linear Convection Profiles.

The following algorithm outlines the process for 1D Linear Convection:

Algorithm 1 1D Linear Convection

```
Set: X, T, dt, dx and c                          // Parameters setup
x: 0 → 2 in X steps                              // Spatial grid setup
u ← 1 for all x
u ← 2 where 0.5 ≤ x ≤ 1                          // Wave condition
for n ← 1 to T do
    un ← u                                       // Copy current state
    for i ← 2 to X do
        u(i) ← un(i) − c · (un(i) − un(i − 1)) · dt/dx   // Convection update
    end for
    Plot the current state of u                  // Visualise solution
end for
```

1.2 Step 2: 1D Nonlinear Convection

In Step 2, the previously constant velocity c is replaced with the variable velocity u, leading to a nonlinear expression as follows:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{6}$$

Equation (6), often referred to as the inviscid Burgers' Equation, is notable for its propensity to generate discontinuous solutions from initially smooth conditions, closely resembling shockwaves.

1.2.1 Numerical Methods and Discretisation

The numerical methods and discretisation for this step remain consistent with those used in linear convection. However, the equation for updating the velocity is now:

$$u_i^{n+1} = u_i^n - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) \tag{7}$$

1.2.2 Initial and Boundary Conditions

The initial and boundary conditions for this step remain identical to those outlined in Step 1.

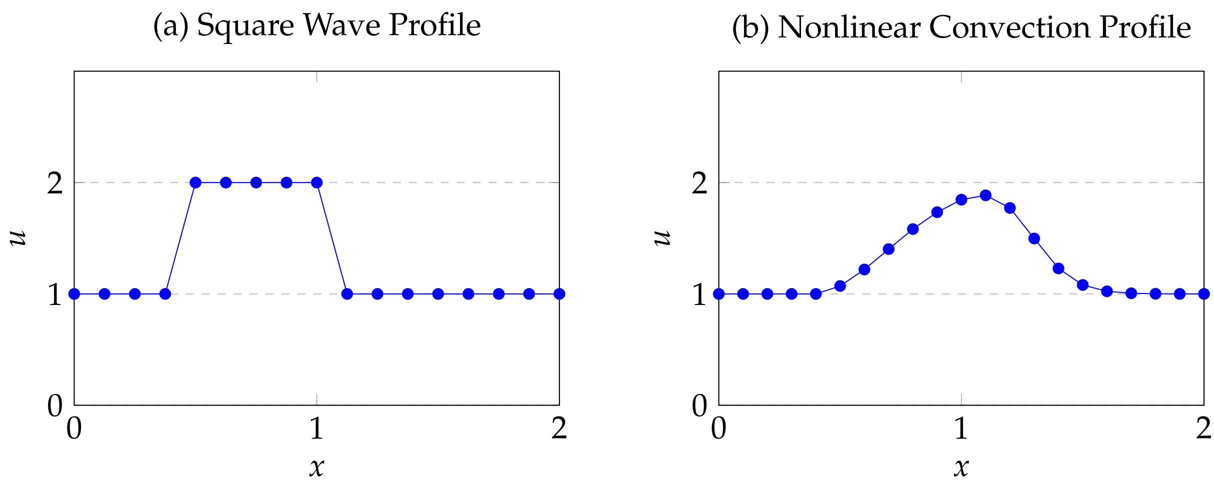

Figure 2: Square Wave and Nonlinear Convection Profiles.

1.3 Step 3: 1D Diffusion

The 1D diffusion equation, also known as the heat equation or pure diffusion, involves a second-order derivative and can be expressed as follows:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \tag{8}$$

where ν signifies the diffusion coefficient. If ν is positive, it indicates physical diffusion, while a negative ν represents an exponentially growing phenomenon, such as expulsion.

1.3.1 Numerical Methods and Discretisation

The physics of diffusion is isotropic, and to accurately represent or simulate this phenomenon, the **central difference formula** is employed for spatial derivatives and the **forward difference formula** for temporal derivatives. The resulting discretised form is given by:

$$\textcolor{red}{\frac{u_i^{n+1} - u_i^n}{\Delta t}} = \nu \textcolor{teal}{\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}} \tag{9}$$

This equation is rearranged to find the unknown variable u_i^{n+1}:

$$u_i^{n+1} = \frac{\nu \Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) + u_i^n \tag{10}$$

1.3.2 Initial and Boundary Conditions

The simulation follows the same initial and boundary conditions as in Step 1.

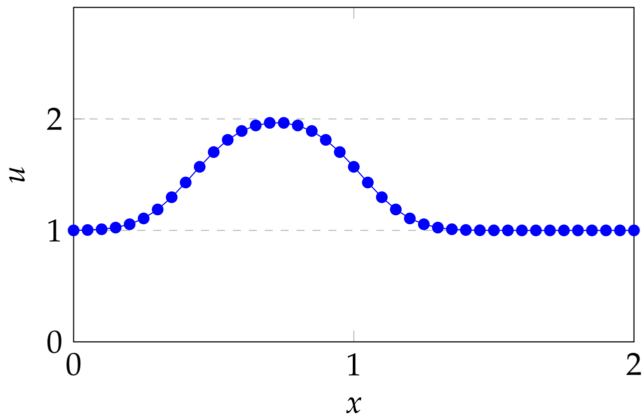

Figure 3: Evolution of Diffusion

The following algorithm outlines the process for 1D Diffusion:

Algorithm 2 1D Diffusion

Set: X, T, dt, dx and v `// Parameters setup`
$x : 0 \to 2$ in X steps `// Spatial grid setup`
$u \leftarrow 1$ for all x
$u \leftarrow 2$ where $0.5 \leq x \leq 1$ `// Wave condition`
for $n \leftarrow 1$ **to** T **do**
  $un \leftarrow u$ `// Copy current state`
  for $i \leftarrow 2$ **to** X **do**
    $u(i) \leftarrow un(i) + (un(i+1) - 2un(i) + un(i-1)) \cdot \frac{v \cdot dt}{dx}$ `// Diffusion update`
  end for
  Plot the current state of u `// Visualise solution`
end for

1.4 Step 4: 1D Burgers' Equation

Step 4 combines principles from Steps 2 (pure diffusion) and 3 (pure convection) to construct the 1D Burgers' Equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{11}$$

1.4.1 Numerical Methods and Discretisation

Employing forward difference for time, backward difference for space, and the central difference formula for the second-order derivatives yields:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{12}$$

Which can be rearranged as:

$$u_i^{n+1} = \nu \frac{\Delta t}{\Delta x^2}(u_{i+1}^n - 2u_i^n + u_{i-1}^n) - u_i^n \frac{\Delta t}{\Delta x}(u_i^n - u_{i-1}^n) + u_i^n \tag{13}$$

1.4.2 Analytical solution

The analytical solution to the problem is represented by the following equations:

$$\phi = \exp\left(\frac{-(x-4t)^2}{4\nu(1+t)}\right) + \exp\left(\frac{-(x-4t-2\pi)^2}{4\nu(1+t)}\right) \tag{14}$$

$$u = 4 - \frac{2\nu}{\phi}\frac{\partial \phi}{\partial x} \tag{15}$$

1.4.3 Initial and Boundary Conditions

The initial condition is determined by inserting $t = 0$ into Equation (15). Moreover, to ensure the system behaves periodically across the range $x = 0$ to $x = 2\pi$, the boundary condition is defined as $u(0) = 2\pi$.

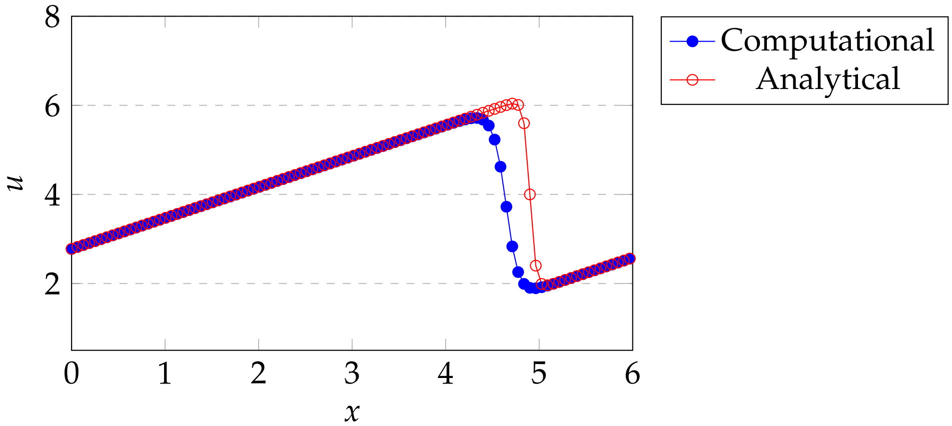

Figure 4: Evolution of Burgers' Equation

1.5 Step 5: 2D Linear Convection

The 2D Linear Convection equation is expressed as:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} + c\frac{\partial u}{\partial y} = 0 \tag{16}$$

This is identical to the 1D Linear Convection, with the addition of two spatial dimensions as time progresses. The discretisation of the timestep is achieved using a forward difference, while both spatial steps are discretised using backward differences.

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + c\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x} + c\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta y} = 0 \tag{17}$$

Solving for the only unknown, $u_{i,j}^{n+1}$, yields:

$$u_{i,j}^{n+1} = u_{i,j}^{n} - c\frac{\Delta t}{\Delta x}(u_{i,j}^{n} - u_{i-1,j}^{n}) - c\frac{\Delta t}{\Delta y}(u_{i,j}^{n} - u_{i,j-1}^{n}) \tag{18}$$

The initial conditions are given as:

$$u(x,y) = \begin{cases} 2 & \text{for} \quad 0.5 \leq x,y \leq 1 \\ 1 & \text{for} \quad \text{everywhere else} \end{cases} \tag{19}$$

The boundary conditions are specified as $u = 1$ for $x = 0, 2$ and $y = 0, 2$. The following algorithm details the procedure for 2D Linear Convection

Algorithm 3 2D Linear Convection

Set: X, Y, T, dt, dx and c `// Parameters setup`
$x: 0 \rightarrow 2$ in X steps `// Spatial grid setup`
$y: 0 \rightarrow 2$ in Y steps `// Spatial grid setup`
$u = 1$ for all (x, y)
$u = 2$ for $0.5 \leq x \leq 1$ and $0.5 \leq y \leq 1$ `// Initial condition`
Create a grid (nX, nY) from x and y coordinates.
for $n = 1$ **to** T **do**
  $un \leftarrow u$ `// Copy current state`
  for $i = 2$ **to** $X - 1$ **do**
    for $j = 2$ **to** $Y - 1$ **do**
      $u(i,j) \leftarrow un(i,j) - c \cdot dt \cdot \left(\frac{un(i,j) - un(i-1,j)}{dx} + \frac{un(i,j) - un(i,j-1)}{dx} \right)$
    end for
  end for
  Set u at boundaries to 1 `// Boundary conditions`
  Plot u as a surface over (nX, nY).
end for

(a) Spatial Domain (b) Wave Propagation

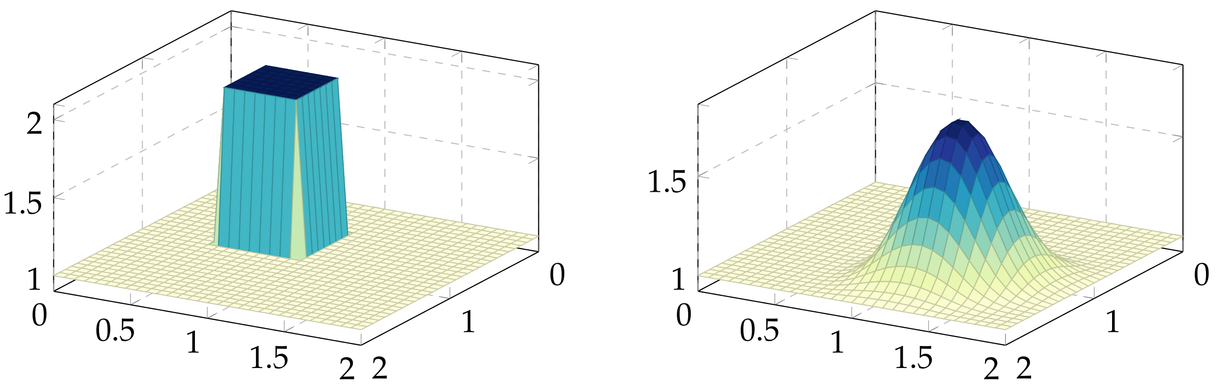

Figure 5: Analysis of 2D Linear Convection

1.6 Step 6: 2D Nonlinear Convection

For 2D nonlinear convection, the dynamic behavior is governed by a set of coupled partial differential equations that incorporate variable velocities in the x and y directions:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = 0 \tag{20}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = 0 \tag{21}$$

These formulations capture the temporal and spatial evolution of velocities u and v. The equations are discretised using the forward difference method for the time derivative and the backward difference method for spatial derivatives, as outlined below:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + u_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x} + v_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta y} = 0 \tag{22}$$

$$\frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta t} + u_{i,j}^{n}\frac{v_{i,j}^{n} - v_{i-1,j}^{n}}{\Delta x} + v_{i,j}^{n}\frac{v_{i,j}^{n} - v_{i,j-1}^{n}}{\Delta y} = 0 \tag{23}$$

Solving for the only unknowns $u_{i,j}^{n+1}$ and $v_{i,j}^{n+1}$, yields:

$$u_{i,j}^{n+1} = u_{i,j}^{n} - \frac{u_{i,j}^{n}\Delta t}{\Delta x}(u_{i,j}^{n} - u_{i-1,j}^{n}) - \frac{v_{i,j}^{n}\Delta t}{\Delta y}(u_{i,j}^{n} - u_{i,j-1}^{n}) \tag{24}$$

$$v_{i,j}^{n+1} = v_{i,j}^{n} - \frac{u_{i,j}^{n}\Delta t}{\Delta x}(v_{i,j}^{n} - v_{i-1,j}^{n}) - \frac{v_{i,j}^{n}\Delta t}{\Delta y}(v_{i,j}^{n} - v_{i,j-1}^{n}) \tag{25}$$

The initial and boundary conditions are the same as those established in Step 5.

1.7 Step 7: 2D Diffusion

The 2D diffusion equation is expressed as:

$$\frac{\partial u}{\partial t} = \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{26}$$

This equation models the spread or diffusion of a quantity, u, across a two-dimensional space, where ν is the diffusion coefficient indicating the rate of diffusion. For the discretisation of this equation, a central difference scheme is employed for both x and y spatial derivatives, offering a precise approximation of the spatial changes in u. The discretised version is presented as:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \nu \left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2} \right) \tag{27}$$

To solve for the forthcoming time step $u_{i,j}^{n+1}$, the equation is rearranged as:

$$u_{i,j}^{n+1} = \nu \Delta t \left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2} \right) + u_{i,j}^{n} \tag{28}$$

The progression of diffusion in 2D is depicted in the figure below.

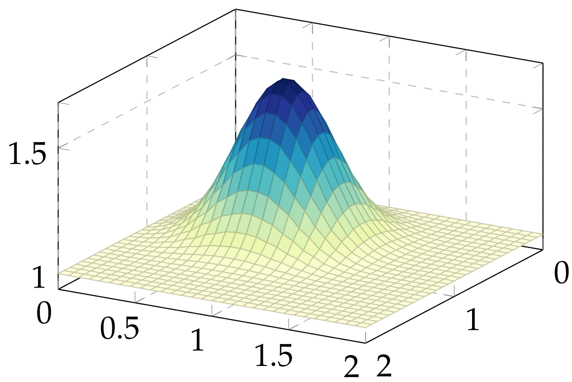

Figure 6: Evolution of Diffusion in 2D

1.8 Step 8: 2D Burgers' Equation

The Burgers' Equation in 2D, which combines the effects of convection and diffusion, is formulated as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{29}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \tag{30}$$

The discretisation of these equations incorporates both forward difference for time derivatives and central differences for spatial derivatives. The resulting discretised forms are:

$$
\begin{aligned}
&\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y} \\
&=\nu\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2}\right)
\end{aligned}
\tag{31}
$$

And

$$
\begin{aligned}
&\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta t}+u_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\Delta y} \\
&=\nu\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x^2}+\frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{\Delta y^2}\right)
\end{aligned}
\tag{32}
$$

Calculating $u_{i,j}^{n+1}$ and $v_{i,j}^{n+1}$ involves rearranging the discretised equations to determine the fluid velocities at the upcoming time step:

$$
\begin{aligned}
u_{i,j}^{n+1}=u_{i,j}^{n}&-\Delta t\left(u_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{u_{i,j}^{n}-u_{i,j-1}^{n}}{\Delta y}\right) \\
&+\nu\Delta t\left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2}\right).
\end{aligned}
\tag{33}
$$

$$
\begin{aligned}
v_{i,j}^{n+1}=v_{i,j}^{n}&-\Delta t\left(u_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i-1,j}^{n}}{\Delta x}+v_{i,j}^{n}\frac{v_{i,j}^{n}-v_{i,j-1}^{n}}{\Delta y}\right) \\
&+\nu\Delta t\left(\frac{v_{i+1,j}^{n}-2v_{i,j}^{n}+v_{i-1,j}^{n}}{\Delta x^2}+\frac{v_{i,j+1}^{n}-2v_{i,j}^{n}+v_{i,j-1}^{n}}{\Delta y^2}\right).
\end{aligned}
\tag{34}
$$

With the initial and boundary conditions consistent with those delineated in Step 5, the visualisation of the Burgers' Equation evolution in 2D is captured in the figure below.

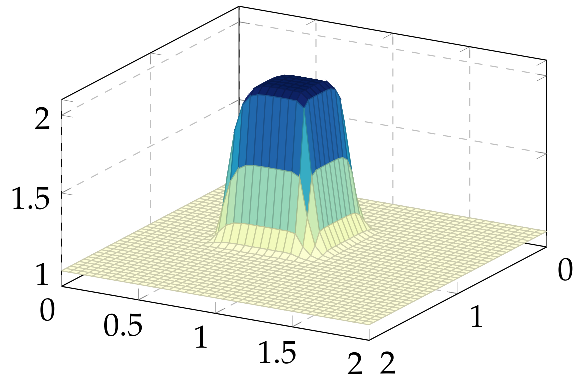

Figure 7: Evolution of Burgers' Equation in 2D

1.9 Step 9: Laplace Equation

The Laplace equation is crucial for understanding how pressure evenly distributes or diffuses within a fluid at a steady state. Mathematically, it is expressed as:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = 0 \tag{35}$$

The discretisation of the Laplace equation utilises the second-order central difference approximation in both spatial dimensions. This method, often referred to as the "5-point difference operator," offers a balanced treatment of the spatial derivatives by involving the central point and its immediate neighbours in both directions. The discrete form of the Laplace equation becomes:

$$\frac{p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}}{\Delta x^2} + \frac{p^n_{i,j+1} - 2p^n_{i,j} + p^n_{i,j-1}}{\Delta y^2} = 0 \tag{36}$$

Rearranging the equation to solve for the pressure at the central grid point $p^n_{i,j}$, yields:

$$\frac{(p^n_{i+1,j} + p^n_{i-1,j})\Delta y^2 + (p^n_{i,j+1} + p^n_{i,j-1})\Delta x^2}{2(\Delta x^2 + \Delta y^2)} = p^n_{i,j} \tag{37}$$

The initial conditions set $p = 0$ everywhere in a domain of $(0, 2)$ in the x-direction and $(0, 1)$ in the y-direction. The boundary conditions are $p = 0$ at $x = 0$, $p = y$ at $x = 2$, and $\frac{\partial p}{\partial y} = 0$ at $y = 0$ and 1.

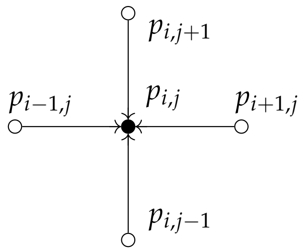

Figure 8: 5 points difference operator.

The analytical solution for $p(x, y)$, derived under specific conditions, is given by:

$$p(x,y) = \frac{x}{4} - 4\sum_{n=1}^{\infty} \frac{1}{(n\pi)^2 \sinh(2\pi n)} \sinh(2\pi x)\cos(n\pi y) \tag{38}$$

The numerical solution to the Laplace equation is obtained through an iterative method suited for steady-state problems. The absence of time derivatives implies that time does not directly influence the solution. Instead, iterations simulate a pseudo-time advancement, progressively refining the pressure distribution until convergence is achieved. The following algorithm details this procedure

Algorithm 4 Laplace Equation

Set: X, Y, T, dx, dy `// Parameters setup`
x: $0 \to 2$ in X steps `// Spatial grid setup`
y: $0 \to 2$ in Y steps `// Spatial grid setup`
$p = 0$ for all (x, y)
$p(\text{right edge}) \leftarrow y$ `// Boundary condition`
Create a grid (nX, nY) from x and y coordinates.
for $n = 1$ **to** T **do**
  for $i = 2$ **to** $X - 1$ **do**
    for $j = 2$ **to** $Y - 1$ **do**
      $p(i,j) \leftarrow \dfrac{dy^2(p_{i+1,j} + p_{i-1,j}) + dx^2(p_{i,j+1} + p_{i,j-1})}{2(dy^2 + dx^2)}$
    end for
  end for
  Apply Neumann boundary conditions on top and bottom boundaries.
  Plot the current state of p over (nX, nY).
end for

1.10 Step 10: Poisson Equation

The Poisson equation extends the Laplace equation by introducing a non-homogeneous term. It is written as:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = b \tag{39}$$

This equation resembles the Laplace equation but includes the right-hand side b, which can represent various physical phenomena such as charge distribution in electromagnetism or force fields in fluid dynamics. The numerical approach for solving the Poisson equation utilises the second-order central difference scheme in space. The discretised form is given by:

$$\frac{p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}}{\Delta x^2} + \frac{p^n_{i,j+1} - 2p^n_{i,j} + p^n_{i,j-1}}{\Delta y^2} = b^n_{i,j} \tag{40}$$

In this context, $b^n_{i,j}$ denotes the source term at the grid position (i, j) during the n-th iteration. Typically, this term remains zero except at specific points that denote sources or sinks.

- **Initial Conditions:**
  * Pressure p is initialised to zero throughout the domain.
  * The domain is defined from 0 to 2 in the x-direction and from 0 to 1 in the y-direction.
- **Boundary Conditions:**
  * At the left boundary ($x = 0$), p is set to zero.
  * At the right boundary ($x = 2$), p is equal to the y-coordinate.
  * The source term b is specified as follows:
    · $b_{i,j} = 100$ at $i = \frac{X}{4}$ and $j = \frac{Y}{4}$.
    · $b_{i,j} = -100$ at $i = \frac{3X}{4}$ and $j = \frac{3Y}{4}$.
    · $b_{i,j}$ is zero at all other points.

1.11 Step 11: Cavity Flow

Cavity flow, a fundamental study in fluid dynamics, focuses on incompressible flow in a confined rectangular space, driven by a moving top lid against stationary boundaries. This setup is crucial for understanding fluid behaviour under restrictions. The mathematical foundation of cavity flow consists of:

– The continuity equation for fluid incompressibility:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{41}$$

– The momentum equations for velocity changes due to forces:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{42}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{43}$$

The pressure Poisson equation, derived from the momentum and continuity equations, is formulated as follows:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = -\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right) + \rho\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{44}$$

The discretisation of the momentum equations utilises the Forward Difference Method for time derivatives and the Central Difference Method for spatial derivatives.

– Discretised u-momentum equation:

$$\begin{aligned}
&\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + u_{i,j}^{n}\left(\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x}\right) + v_{i,j}^{n}\left(\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta y}\right) \\
&= -\frac{1}{\rho}\left(\frac{p_{i+1,j}^{n} - p_{i-1,j}^{n}}{2\Delta x}\right) + \nu\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2}\right)
\end{aligned} \tag{45}$$

– Discretised v-momentum equation:

$$\frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta t} + u_{i,j}^{n}\left(\frac{v_{i,j}^{n} - v_{i-1,j}^{n}}{\Delta x}\right) + v_{i,j}^{n}\left(\frac{v_{i,j}^{n} - v_{i,j-1}^{n}}{\Delta y}\right)$$
$$= -\frac{1}{\rho}\left(\frac{p_{i,j+1}^{n} - p_{i,j-1}^{n}}{2\Delta y}\right) + \nu\left(\frac{v_{i+1,j}^{n} - 2v_{i,j}^{n} + v_{i-1,j}^{n}}{\Delta x^2} + \frac{v_{i,j+1}^{n} - 2v_{i,j}^{n} + v_{i,j-1}^{n}}{\Delta y^2}\right) \tag{46}$$

For discretising the pressure equation, the Central Difference Method is utilised for spatial derivatives.

$$\frac{p_{i+1,j}^{n} - 2p_{i,j}^{n} + p_{i-1,j}^{n}}{\Delta x^2} + \frac{p_{i,j+1}^{n} - 2p_{i,j}^{n} + p_{i,j-1}^{n}}{\Delta y^2} =$$
$$-\rho\left[\left(\frac{u_{i+1,j}^{n} - u_{i-1,j}^{n}}{2\Delta x}\right)^2 + 2\left(\frac{u_{i,j+1}^{n} - u_{i,j-1}^{n}}{2\Delta y}\right)\left(\frac{v_{i+1,j}^{n} - v_{i-1,j}^{n}}{2\Delta x}\right) + \left(\frac{v_{i,j+1}^{n} - v_{i,j-1}^{n}}{2\Delta y}\right)^2\right] \tag{47}$$

Next, expressions for $u_{i,j}^{n+1}$, $v_{i,j}^{n+1}$, and $p_{i,j}^{n}$ are derived to progress the simulation. The initial and boundary conditions are set as follows:

– **Initial Conditions:** At the beginning of the simulation, the velocity components $u_{i,j}$ and $v_{i,j}$, along with the pressure $p_{i,j}$, are initialised to zero throughout the domain. This setup represents a stationary fluid, providing a starting point for the simulation of fluid dynamics.

– **Boundary Conditions:**

  * The velocity u is set to 1 at the top boundary of the domain to simulate the lid-driven flow, representing the movement of the top lid in a cavity flow scenario.

  * For all other boundaries (the bottom, left, and right walls), the no-slip condition is applied by setting $u = 0$ and $v = 0$, reflecting the behavior of stationary walls.

  * The pressure p is normalised, typically to a reference value of zero, at one of the boundaries to establish a reference level for the pressure field.

* Neumann boundary conditions for pressure are implemented at the domain's periphery to maintain a divergence-free velocity field, ensuring the fluid's incompressibility.

Algorithm 5 Algorithm for Cavity Flow Simulation

Initialise domain dimensions X, Y and time steps T.
Set fluid properties: kinematic viscosity ν, density ρ.
Define time step dt and calculate grid spacing dx, dy.
Initialise velocity vectors u, v and pressure p to zero.
Set lid-driven cavity condition by assigning u at the top boundary to 1.
for $n = 1$ to T **do**
    Copy current velocity vectors to temporary variables un, vn.
    Calculate source term b based on the velocity field.
    Solve the Pressure Poisson equation iteratively for pressure p.
    Update velocity vectors u, v using the temporary variables and new pressure p.
    Apply boundary conditions for velocities and pressure.
end for
Plot the final pressure and velocity field.

The graphical representation of the pressure and velocity field is illustrated in the figure below:

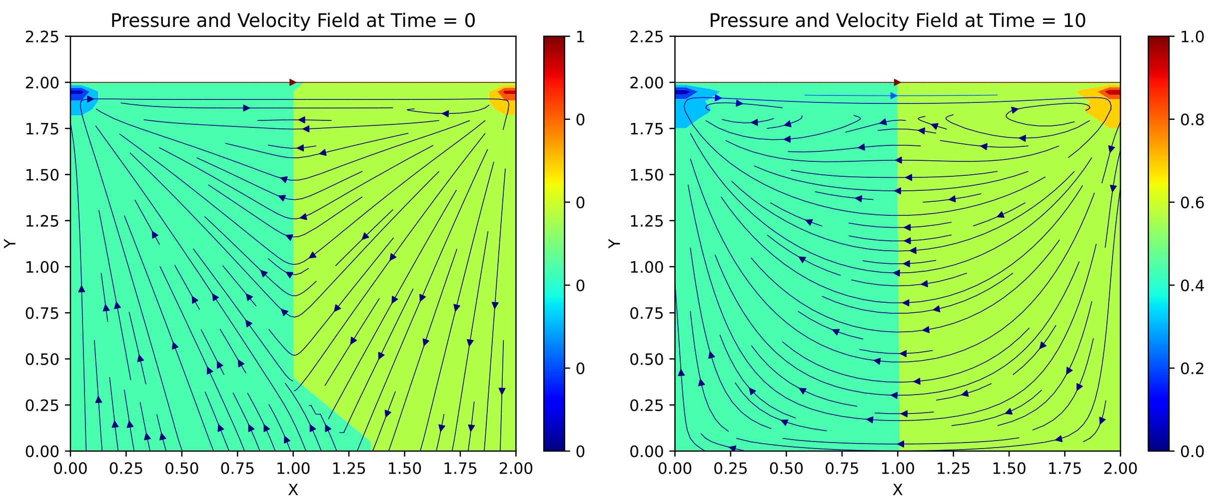

Figure 9: Cavity Flow Simulation

1.12 Step 12: Channel Flow

Channel Flow is an extension of the previous cavity flow problem, utilising identical equations with the inclusion of an additional source term F, which is set to 1 to represent a driving force. The governing equations for Channel Flow, derived from the Navier-Stokes equations, are as follows:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + F \tag{48}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{49}$$

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = -\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right) \tag{50}$$

Initial conditions for the simulation involve setting u, v, and p to zero throughout the domain. Boundary conditions are adapted to be periodic in x for u, v, and p at $x = 0, 2$, allowing for continuous flow through the channel's horizontal edges. At $y = 0, 2$, u and v are set to zero, maintaining the no-slip condition at the channel's top and bottom, while the pressure gradient $\frac{\partial p}{\partial y}$ is set to zero, ensuring a consistent pressure across these boundaries.

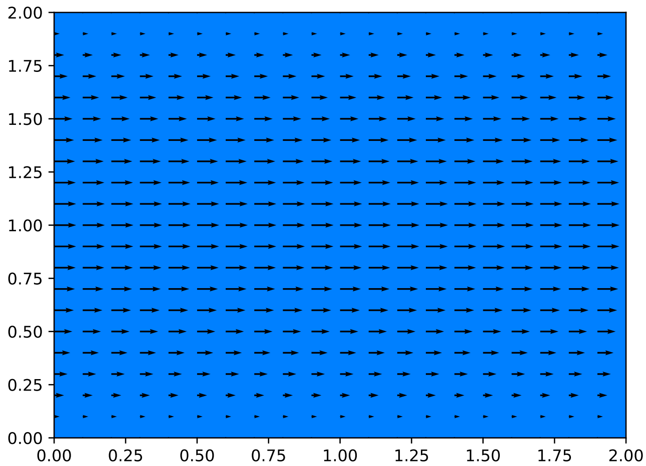

Figure 10: Channel Flow Simulation