Discrete stream function method for the incompressible Navier-Stokes equations with oscillatory flow past a flat plate

Rauan

May 30, 2024

Abstract

The goal of these notes is to present the detailed overview of discrete stream function method for solving incompressible Navier-Stokes equations with oscillatory flow past a flat plate. We will discuss in detail the scheme formulation, transient and spatial discretizations. Special attention will be paid to the boundary conditions and their implementation. After studying these notes one must get a coherent picture of the application of discrete stream function method to incompressible flows and be able to implement the scheme in code.

Contents

1 Introduction

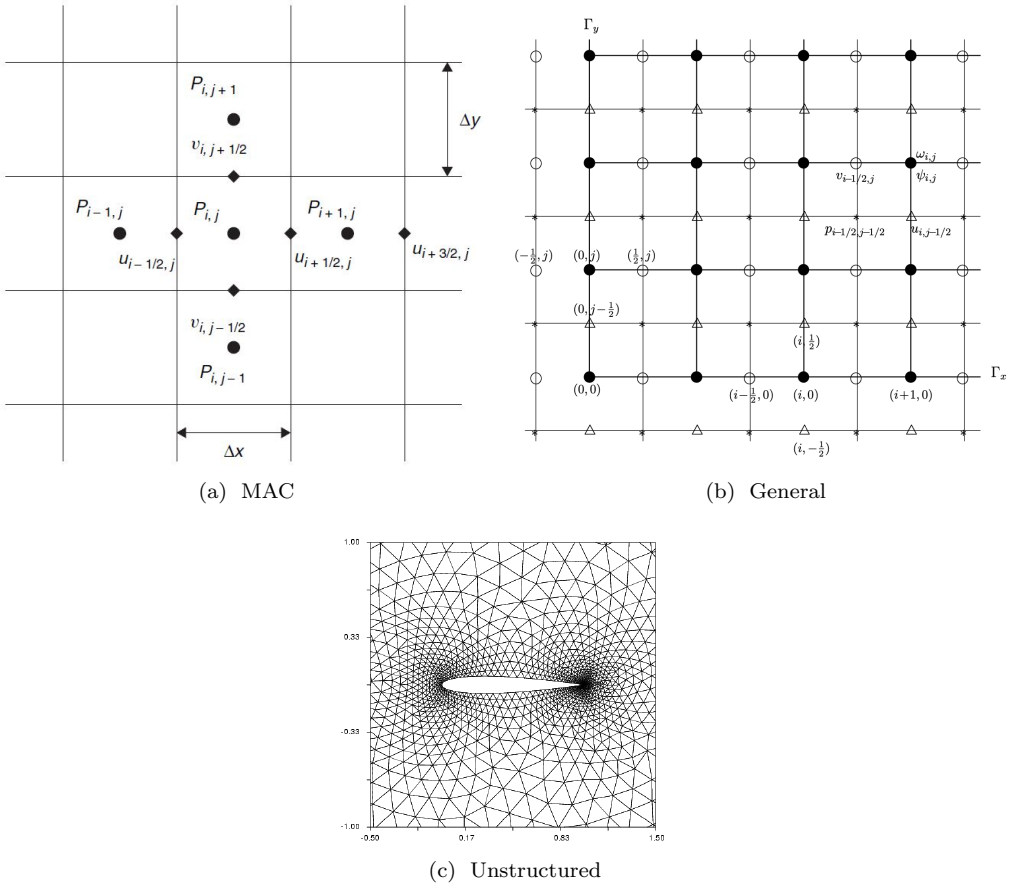
The following notes we will be focused on discretization of the Navier-Stokes equations describing the flow of an incompressible fluid:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\nabla p + \epsilon \nabla \cdot \nabla \boldsymbol{v}, \tag{1.1a}$$

$$\nabla \cdot \boldsymbol{v} = 0, \tag{1.1b}$$

which are written here in the non-dimensional form, i.e. $\epsilon \equiv Re^{-1}$ for brevity. Moreover, $\boldsymbol{v}$ is the velocity and p pressure fields. Since both p and $\boldsymbol{v}$ are functions of time t and space $\boldsymbol{x}$, their discretization will be denoted by f_i^n, where n is the time level t^n and i is *mesh* point $\boldsymbol{x}_i$. The cell centre corresponds to $x_{i+\frac{1}{2}}$. The mesh points x_i and cell centres $x_{i\pm\frac{1}{2}}$ can be regarded as two overlapping interpenetrating meshes, which together are said to constitute a *staggered* mesh as illustrated in Fig. 1(b). When designing a finite difference or finite volume scheme, we have to choose whether to use the same or different sets of grid points for velocity and pressure. The common choice seems to have a single set of points, at which all the variables and all the equations are discretized. Such grid has the name of a *collocated* or *regular* grid. Albeit simple and easy in operation, collocated grids were out of favor for a long time because of their tendency to generate spurious oscillations in the solution (cf. §10.2 in [14]). Hence, staggered grids were used since the work of Harlow and Welch [7] introducing the marker-and-cell (MAC) method, cf. Fig. 1(a). Colonius [3] uses MAC grid Fig. 1(a), where the indexing uses only integer indices, i.e. $u_{i-\frac{1}{2},j}, v_{i,j-\frac{1}{2}}$ are denoted as $u_{i,j}, v_{i,j}$ and lie on left and bottom sides of cell i, j.

Grid indexation starts at $(x_{\frac{1}{2}}, y_{\frac{1}{2}}) = (0, 0)$ and ends at $(x_{M+\frac{1}{2}}, y_{N+\frac{1}{2}}) = (1, 1)$. The width Δx and height Δy of each cell will have the same indexation as the corresponding cells they belong to.



(a) MAC (b) General

(c) Unstructured

Figure 1: Staggered grid.

Staggered arrangement increases complexity of a scheme and programming becomes more difficult since it requires accounting for three (u, v, p) or four (u, v, w, p) indexing systems. Interpolations must be used to compute nonlinear terms of momentum equations. Further complications arise when the grid becomes nonuniform. All these difficulties, however, can be relatively easily handled in computations with structured grids such as those shown in Figs. 1(a) and 1(b). The issues of handling a staggered arrangement increase significantly when unstructured (Fig. 1(c)) grids are used. When such grids started to be broadly applied in general-purpose codes recently, collocated arrangements returned to favor. This area of CFD is still evolving and we only mention that methods have been developed to cure the splitting problem leading to pressure oscillations. However, the cure is not ideal and leads to extra complexities at the implementation level.

In numerical terms, the problem Eqs. (1.1) can be formulated in the following way: given the solution p^n, $\boldsymbol{v}^n$ at the previous time layer t^n, find the next time-layer pressure p^{n+1} and velocity $\boldsymbol{v}^{n+1}$ such that they together satisfy the momentum equation, the velocity is divergence-free $\nabla \cdot \boldsymbol{v}^{n+1} = 0$ and satisfies the boundary conditions.

A conspicuous feature of Eqs. (1.1) is that pressure p is not determined by a time-evolution equation, but rather is implicitly defined by the incompressibility Eq. (1.1b), which plays the role of a constraint that the velocity field $\boldsymbol{v}$ must satisfy. The pressure instantaneously adapts to the evolving velocity field in a way as to satisfy the constraint. This reflects in the fact that p satisfies a Poisson equation, which can be derived by taking the divergence of Eq. (1.1a) and combining the result with Eq. (1.1b). From the mathematical viewpoint, this means that the incompressible flow equations have some features of an elliptic system. We can say that the equations belong to mixed hyperbolic (convective terms), parabolic (viscous terms), and elliptic (pressure and incompressibility) types. The elliptic nature of the pressure solution has a physical meaning: in an incompressible flow, the pressure field in the entire flow domain adjusts instantaneously to any, however localized, perturbation. This is in perfect agreement with the fact that weak perturbations, for example sound waves, propagate at infinite speed

in incompressible fluids.

Given the Poisson equation for p, it may replace the continuity Eq. (1.1b) and its solution can in principle be substituted back into Eq. (1.1a) to obtain an evolution equation for the velocity field alone. Alternatively, the pressure can be immediately eliminated by taking the curl of momentum Eq. (1.1a), which leads to the vorticity-stream function formulation of incompressible flow. Formal manipulations of this type are useful for various theoretical purposes, but the experience has shown that they are rarely advantageous for computational purposes. In most situations, it is preferable to simply approximate and solve Eqs. (1.1) directly for the so-called *primitive* variables $\boldsymbol{v}$ and p.

If, however, one ends up solving the Poisson equation for pressure, an interesting and important question arises: "what boundary conditions should be used for the pressure field?" Such conditions are required at every point of the boundary for the Poisson problem to be well-posed. The conditions, however, do not naturally follow from the flow physics for the boundaries between fluid and solid walls, unless, of course, a full fluid-structure interaction problem is solved. Since the latter option is, in most situations, an unnecessary complication, we have to find a way to derive the pressure boundary conditions from the equations themselves.

The remainder of the present paper is organized as follows. Sections 2 and 3 are dedicated to vorticity-stream function formulation and motivation for numerical scheme. Formulation of the problem in terms of PDE and boundary conditions on a finite domain is performed in Sections 4 to 6. In Sections 7 and 8, we discretize the problem. Boundary conditions are discussed in Section 9. Symmetrization of the resulting system of linear equations is performed in Section 10. We discuss the discrete approach to the algorithm in Section 11, which is based on curl operator from Section 8.4. A summary and discussion are given in Section 12.

- Dong + Chang.x

ToDo

2 Vorticity-stream function formulation

The algorithm we will be using is a combination of classical formulation as in Eqs. (1.1) and vorticity transport equation Eq. (2.4). Consider:

1. Stream function $\psi(x, y, t) : \boldsymbol{v} = \nabla \times \psi$ of an incompressible two-dimensional flow:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \tag{2.1}$$

   where $\boldsymbol{v} = (u, v)$.

   The velocity vector at every point of space and every moment of time is tangential to the line $\psi = const$, such lines represent the streamlines of the flow.

2. Vorticity $\omega = \nabla \times \boldsymbol{v}$ in two-dimensional case (x-y-plane) where the only non-zero component of ω is z, which leads to

$$\omega = -\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{2.2}$$

Importantly, continuity Eq. (1.1b) is satisfied immediately by taking spatial derivatives of stream function Eq. (2.1) components and adding them up. The governing Eqs. (1.1) can now be transformed into the following:

1. Transport equation for vorticity.

   Applying $(\nabla\times)$ to momentum and taking into account continuity Eq. (1.1b) together with the fact that

$$\frac{\partial}{\partial y}\left(\frac{\partial p}{\partial x}\right) - \frac{\partial}{\partial x}\left(\frac{\partial p}{\partial y}\right) = 0 \tag{2.3}$$

   will result in

$$\boxed{\frac{\partial \omega}{\partial t} - \epsilon\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) = -\left(u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y}\right),} \tag{2.4}$$

   where we shifted advection terms to the right-hand side with intent to linearize them later.

2. Vorticity-Stream function equation.

   After substituting stream function Eq. (2.1) into vorticity Eq. (2.2) we obtain

$$\boxed{\nabla^2 \psi = -\omega.} \tag{2.5}$$

The two boxed equations above form a coupled system, and the pressure field does not explicitly appear in either Eq. (2.4) or Eq. (2.5). In principle, this field is not needed in the solution.

The system formed by Eqs. (2.4) and (2.5) requires boundary conditions on ψ and ω. For the stream function imposing physically plausible conditions is not difficult. One has to write the proper boundary conditions for the velocity components and use the definition of Eq. (2.1) to represent them as conditions for ψ. Values for ψ and its derivatives are obtained by integrating Eq. (2.1). In the case of vertical boundary

$$1 = u = \frac{\partial \psi}{\partial y} \implies \frac{\partial \psi}{\partial y} = 1 \implies \psi_1(t, y) = a_1 y + a_2 + \psi_2(t, x), \tag{2.6}$$

$$0 = v = -\frac{\partial \psi}{\partial x} \implies \psi = \psi_1(t, y) + a_3 \text{ (can be made compatible with other BCs)}, \tag{2.7}$$

where a_1, a_2, a_3 are constants. The situation is more difficult in the case of vorticity. There are no natural boundary conditions on ω, but they can be derived from the conditions on ψ by applying Eq. (2.5) to the boundary. The process of finding boundary conditions of ω typically results in expressions containing second derivatives, therefore, a special numerical treatment is required.

3 Prelude to the algorithm using continuous operators

We now modify momentum Eq. (1.1a) in 2D in a similar manner as in Section 2 with the help of differential operators. These operators can be transformed into discrete formats without much difficulty. Hence, we can apply a similar process to system of linear equations generated by Eq. (1.1a) by multiplying it with the corresponding matrices, which is discussed in Sections 8.4 and 11.

Define

$$\operatorname{curl} = \begin{bmatrix} \frac{\partial}{\partial y} \\ -\frac{\partial}{\partial x} \end{bmatrix}. \tag{3.1}$$

Hence, by stream function Eq. (2.1) we have $\boldsymbol{v} = \operatorname{curl} \psi$. Lastly, define

$$\operatorname{curl}^* = \begin{bmatrix} -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}. \tag{3.2}$$

Let us now compute curl^* applied to velocity vector $\boldsymbol{v}$

$$\operatorname{curl}^* \boldsymbol{v} = \operatorname{curl}^* \operatorname{curl} \psi = \begin{bmatrix} -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial y} \\ -\frac{\partial}{\partial x} \end{bmatrix} \psi = \left(-\frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial x^2} \right) \psi = -\nabla^2 \psi. \tag{3.3}$$

By taking into account the commutativity of partial derivatives with respect to time and space we obtain

$$\operatorname{curl}^* \left(\frac{\partial}{\partial t} \right) \operatorname{curl} \psi = -\frac{\partial}{\partial t} \left(\nabla^2 \psi \right). \tag{3.4}$$

Applying curl^* to Laplacian of velocity leads to

$$\begin{aligned} \operatorname{curl}^* \nabla^2 \boldsymbol{v} &= \operatorname{curl}^* \nabla^2 \operatorname{curl} \psi \\ &= \begin{bmatrix} \frac{\partial}{\partial y} & -\frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} & 0 \\ 0 & \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \end{bmatrix} \begin{bmatrix} -\frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} \end{bmatrix} \psi \\ &= \begin{bmatrix} \frac{\partial}{\partial y} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) & 0 \\ 0 & -\frac{\partial}{\partial x} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \end{bmatrix} \begin{bmatrix} -\frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} \end{bmatrix} \psi \\ &= -\nabla^2 \left(\nabla^2 \psi \right). \end{aligned} \tag{3.5}$$

Our goal is to obtain an equation in terms of ψ by modifying Eq. (1.1a). The procedure lies in transforming this equation into vorticity transport Eq. (2.4) with the help of vorticity stream function substitution Eq. (2.5) and curl, curl^* operators. Firstly, we put transient, gradient and viscous terms to the left-hand side, whereas non-linear terms are linearized and moved to the right. Next, we substitute $\boldsymbol{v} = \operatorname{curl} \boldsymbol{\psi}$ and apply curl^* to rearranged momentum Eq. (1.1a), which eliminates the pressure gradient ($\operatorname{curl}^* \operatorname{grad} = \begin{bmatrix} -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = 0$). For 2-dimensional case from Eqs. (3.4) and (3.5) , we get

$$\operatorname{curl}^* \left(\frac{\partial \boldsymbol{v}}{\partial t} - \epsilon \nabla^2 \boldsymbol{v} \right) = \operatorname{curl}^* \left(\boldsymbol{v} \cdot \nabla \boldsymbol{v} \right) \tag{3.6}$$

$$\operatorname{curl}^* \left(\frac{\partial}{\partial t} \operatorname{curl} \psi - \epsilon \nabla^2 \operatorname{curl} \psi \right) = \operatorname{curl}^* \left(\boldsymbol{v} \cdot \nabla \boldsymbol{v} \right) \tag{3.7}$$

$$\left(\frac{\partial}{\partial t} - \epsilon \nabla^2 \right) \left(-\nabla^2 \psi \right) = \operatorname{curl}^* \left(\boldsymbol{v} \cdot \nabla \boldsymbol{v} \right) \tag{3.8}$$

Solving Eq. (3.8) for $\nabla^2 \psi$ is equivalent to obtaining a solution for Poisson Eq. (2.5). However, we can go through Eq. (3.8) and solve it in one step by obtaining ψ from biharmonic equation

$$\boxed{-\frac{\partial \nabla^2 \psi}{\partial t} + \epsilon \nabla^4 \psi = \operatorname{curl}^* \left(\boldsymbol{v} \cdot \nabla \boldsymbol{v} \right),} \tag{3.9}$$

where the right-hand side contains boundary conditions (in numerical sense) and non-linear advective terms, which were initially stated in terms of $\boldsymbol{v}, p$ and transformed through premultiplication by curl^T.

3.1 Vector fields that vanish at the boundary

This subsection contains side notes for Section 3, here we discuss some properties of differential operators in vector calculus. Consider continuous operators div, curl and grad on bounded domain Ω with boundary $\partial\Omega$. Let all vector $\boldsymbol{X}, \boldsymbol{Y}$ and scalar ϕ, ψ fields vanish at $\partial\Omega$. Denote by SF the set of all scalar fields in Ω, and by VF the set of all vector fields in Ω. In this subsection we need to have inner products in inner product space [1]. Define scalar and vector inner products as

$$\langle \phi, \psi \rangle = \int_\Omega \phi\psi \; dV, \quad \langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \int_\Omega \boldsymbol{X} \cdot \boldsymbol{Y} \; dV. \tag{3.10}$$

Then define the transpose of grad as an operator $\operatorname{grad}^T : VF \to SF$ such that

$$\langle \operatorname{grad} \phi, \boldsymbol{X} \rangle = \left\langle \phi, \operatorname{grad}^T \boldsymbol{X} \right\rangle, \tag{3.11}$$

for all vector fields $\boldsymbol{X}$ and scalar fields ϕ vanishing at the boundary $\partial\Omega$. To identify this operator, integrate the identity

$$\nabla \cdot (\phi \boldsymbol{X}) = \nabla\phi \cdot \boldsymbol{X} + \phi \nabla \cdot \boldsymbol{X}, \tag{3.12}$$

and apply the divergence theorem, to get

$$\int_{\partial\Omega} \phi \boldsymbol{X} \cdot \boldsymbol{n} \; dS = \int_\Omega \nabla \cdot (\phi \boldsymbol{X}) \; dV = \int_\Omega \nabla\phi \cdot \boldsymbol{X} \; dV + \int_\Omega \phi \nabla \cdot \boldsymbol{X} \; dV. \tag{3.13}$$

Since $\phi = 0$ on $\partial\Omega$, left side of Eq. (3.13) vanishes. Hence, we infer

$$0 = \langle \nabla\phi, \boldsymbol{X} \rangle + \langle \phi, \nabla \cdot \boldsymbol{X} \rangle, \tag{3.14}$$

implying that the transpose of the gradient is the negative divergence, and the transpose of the divergence is the negative gradient:

$$\operatorname{grad}^T = -\operatorname{div}, \qquad \operatorname{div}^T = -\operatorname{grad}. \tag{3.15}$$

We now move to the 3-dimensional case, consider the following identity

$$\nabla \cdot (\boldsymbol{X} \times \boldsymbol{Y}) = (\nabla \times \boldsymbol{X}) \cdot \boldsymbol{Y} - \boldsymbol{X} \cdot (\nabla \times \boldsymbol{Y}). \tag{3.16}$$

Navier-Stokes Eqs. (1.1) are defined in the whole domain from $-\infty$ to $+\infty$ in each axis. In our problem operators curl, grad, div are local. Therefore, we are allowed to let velocity vector fields vanish at the boundaries $\pm\infty$. Since we consider only those vector fields that go to zero at $\partial\Omega$, the divergence theorem yields

$$\begin{aligned} 0 = \int_{\partial\Omega} (\boldsymbol{X} \times \boldsymbol{Y}) \cdot \boldsymbol{n} \; dS = \int_\Omega \nabla \cdot (\boldsymbol{X} \times \boldsymbol{Y}) \; dV = \\ = \int_\Omega (\nabla \times \boldsymbol{X}) \cdot \boldsymbol{Y} \; dV - \int_\Omega \boldsymbol{X} \cdot (\nabla \times \boldsymbol{Y}) \; dV = \langle \operatorname{curl} \boldsymbol{X}, \boldsymbol{Y} \rangle - \langle \boldsymbol{X}, \operatorname{curl}^T \boldsymbol{Y} \rangle, \end{aligned} \tag{3.17}$$

which identifies the curl as the transpose of itself in 3D:

$$\operatorname{curl}^T = \operatorname{curl}. \tag{3.18}$$

4 Problem statement

Let the velocity vector $\boldsymbol{v}(t,x,y) = (u(t,x,y), v(t,x,y))$ and pressure $p(t,x,y)$ be the solutions to Eqs. (1.1) on a semi-infinite domain together with boundary conditions:

$$\text{Momentum: } \frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla\cdot\nabla\boldsymbol{v}, \tag{4.1a}$$

ρ - density, μ - dynamic viscosity.

$$\text{Continuity: } \nabla\cdot\boldsymbol{v} = 0, \tag{4.1b}$$

$0 \le x < \infty, 0 \le y < \infty, t \ge 0.$

$$\text{Inlet and Freestream BC: } \boldsymbol{v}(t,0,y) = \boldsymbol{v}(t,x,\infty) = (U_0 + A\cos(kx - \omega t + \phi_0), 0), \tag{4.1c}$$

$\{|A| < U_0, k, \omega, \phi_0\} \subset \mathbb{R}.$

$$\text{No-slip BC: } \boldsymbol{v}(t,x,0) = (0,0). \tag{4.1d}$$

$$\text{Initial condition: } \boldsymbol{v}(0,x,y) = (0,0) \text{ or Blasius profile.} \tag{4.1e}$$

5 Nondimensionalization

In numerical computations, it is advantageous to maintain variables of comparable magnitude. This ensures that operations such as the multiplication of a large dimensional pressure variable with a small velocity are not performed.

We will truncate the domain and normalize all equations by width L and stream velocity U_0. After introducing the following dimensionless variables (marked with prime $'$):

$$x \to Lx', \quad \boldsymbol{v} \to U_0\boldsymbol{v}', \quad \nabla \to \frac{1}{L}\nabla', \quad p \to p'\rho U_0^2, \quad t \to \frac{L}{U_0}t',$$

we obtain

$$\frac{U_0}{\frac{L}{U_0}}\frac{\partial \boldsymbol{v}'}{\partial t'} + U_0\boldsymbol{v}\cdot\left(\frac{1}{L}\nabla'\right)U_0\boldsymbol{v} = -\frac{1}{\rho}\left(\frac{1}{L}\nabla'\right)p'\rho U_0^2 + \frac{\mu}{\rho}\left(\frac{1}{L}\nabla'\right)\cdot\left(\frac{1}{L}\nabla'\right)U_0\boldsymbol{v}'.$$

After multiplying both sides of the above identity by $\frac{L}{U_0^2}$ and introducing well-known Reynolds number

$$\mathrm{Re} = \frac{\rho L U_0}{\mu}$$

we obtain the non-dimensional momentum equation

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\boldsymbol{v},$$

where prime superscript $'$ is suppressed for the dimensionless variables.

The continuity equation is nondimensionalized in a similar manner:

$$\left(\frac{1}{L}\nabla'\right)\cdot U_0\boldsymbol{v}' = 0 \iff \nabla'\cdot\boldsymbol{v}' = 0 \implies \nabla\cdot\boldsymbol{v} = 0,$$

where we drop prime superscripts in the last identity as well.

6 Problem statement on truncated domain

The initial Eqs. (4.1) from Section 4 on semi-infinite domain now become a system of partial differential equations on a unit square domain

$$\text{Momentum: } \frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\nabla p + \epsilon \nabla \cdot \nabla \boldsymbol{v}, \tag{6.1a}$$

$$\epsilon = \frac{1}{\mathrm{Re}}, 0 \le x \le 1, 0 \le y \le 1, t \ge 0.$$

$$\text{Continuity: } \nabla \cdot \boldsymbol{v} = 0, \tag{6.1b}$$

$$0 \le x \le 1, 0 \le y \le 1, t \ge 0.$$

$$\text{Inlet: } \boldsymbol{v}(t, 0, y) = \boldsymbol{v}(t, x, 1) = (1 + A\cos(kx - \omega t + \phi_0), 0), \tag{6.1c}$$

$$\{A \le 1, k, \omega, \phi_0\} \subset \mathbb{R}.$$

$$\text{No-slip wall BC: } \boldsymbol{v}(t, x, 0) = (0, 0). \tag{6.1d}$$

Outlet and freestream BC: Artificial boundary conditions.

Details are discussed in Section 9.

$$\text{Initial condition: } \boldsymbol{v}(0, x, y) = (0, 0) \text{ or Blasius profile.} \tag{6.1e}$$

7 Domain discretization

Following the nondimensionalization of the governing equations, we now turn our attention to the process of domain discretization. This crucial step involves dividing the computational domain into discrete elements, allowing us to compute variables and their derivatives on a finite set of points called the grid. Consider discretizing the area into M intervals horizontally and N intervals vertically forming a mesh of rectangular cells. To enhance the code it is recommended to organize the data using vectors instead of matrices in order to improve memory access time; a single index i works faster than double i, j in most programming languages. In this section, however, for the sake of simplicity, we use two indices i, j (column and row as in Harlow and Welch [7]) to represent coordinates on the grid.

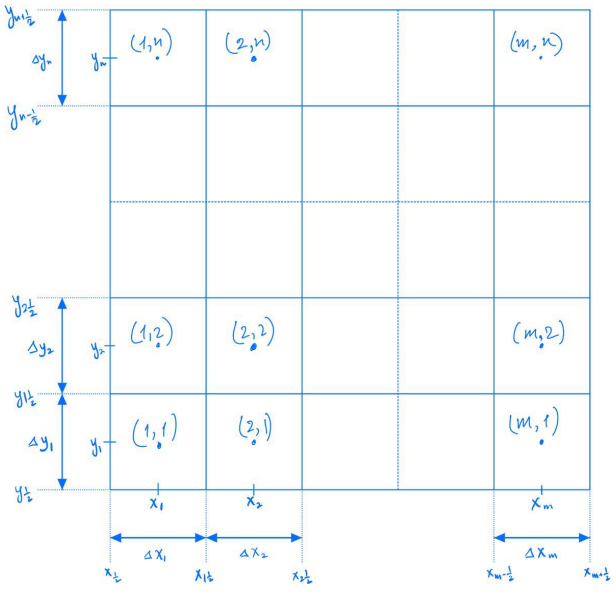


Figure 2: Domain discretization (here $m = M, n = N$)

To eliminate the necessity of solving an additional equation for pressure we will be using a staggered grid arrangement. Details are discussed in Sections 8.2 to 8.4. Moreover, the pressure gradients can

be evaluated directly using central differences on such grids. This calculation on staggered grids does not require any interpolation, furthermore, it is computationally cheaper and simpler to implement.

The u velocity components are stored at the centres of vertically oriented faces, while the values of v are stored at the centres of horizontal ones. The number of momentum equations is $(M+1)N$ for v and $M(N+1)$ for u. These arrays for u and v are then concatenated into a single vector $\boldsymbol{v}$ of size $M(N+1)+(M+1)N$.

Increasing the accuracy closer to the wall and the inlet does look advantageous, hence, we will refine the grid using the standard ratio rule $\Delta x_{i+1} = k_x \Delta x_i, \Delta y_{j+1} = k_y \Delta y_j$ with constants k_x, k_y of values close to 1.

8 Discrete operators

This subsection focuses on the specific operators used in the discretization of Eqs. (6.1), providing the mathematical tools necessary to transform these continuous equations into a form that can be solved numerically.

Denote discrete spatial operators as:

$\hat{L}$: Laplacian.

$\hat{G}$: Gradient.

$\hat{D}$: Divergence.

$\hat{\mathbf{H}}$: Non-linear advective terms.

Then original system of Eqs. (6.1) can be approximated using such operators as

$$\begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \frac{\partial}{\partial t} \begin{pmatrix} \boldsymbol{v} \\ p \end{pmatrix} = \begin{bmatrix} \hat{L} & -\hat{G} \\ -\hat{D} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} + \begin{pmatrix} -\hat{\mathbf{H}}(\boldsymbol{v}) \\ 0 \end{pmatrix} + \mathrm{bc}_{\boldsymbol{v},p}, \tag{8.1}$$

where boundary conditions are in terms of pressure and velocity. In the following subsections, each of the discrete operators is described individually.

Attack Sys. (8.1) with the following schemes as in Colonius [3] (superscript denotes the time step):

Viscous - Implicit trapezoidal - Crank Nicholson scheme (second-order method in time).

$$\hat{L}\boldsymbol{v} = \frac{1}{2}\left(\hat{L}\boldsymbol{v}^{n+1} + \hat{L}\boldsymbol{v}^{n}\right). \tag{8.2}$$

Nonlinear - Explicit Adams-Bashforth (second-order method in time).

$$\hat{\mathbf{H}}(\boldsymbol{v}) = \frac{3}{2}\hat{\mathbf{H}}(\boldsymbol{v}^{n}) - \frac{1}{2}\hat{\mathbf{H}}(\boldsymbol{v})^{n-1}. \tag{8.3}$$

Pressure - Implicit Euler, though the pressure variable will be later eliminated in the algorithm (first-order method in time).

$$\hat{G}p = \hat{G}p^{n+1}. \tag{8.4}$$

The schemes above result in the following time-discretized system:

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbf{I} - \frac{1}{2}\hat{L} & \hat{G} \\ \hat{D} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \left[\frac{1}{\Delta t}\mathbf{I} - \frac{1}{2}\hat{L}\right]\boldsymbol{v}^{n} - \left[\frac{3}{2}\hat{\mathbf{H}}(\boldsymbol{v}^{n}) - \frac{1}{2}\hat{\mathbf{H}}(\boldsymbol{v}^{n-1})\right] \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{bc}_1 \\ \hat{bc}_2 \end{pmatrix}. \tag{8.5}$$

We will use artificial boundary conditions from Dong [4] to generate boundary contributions of outlet and freestream to $\hat{bc}_1$. Their condition was shown to be stable and produced solid results. Details are discussed in Section 9.

After applying the discrete operators (listed in the following Sections 8.1 to 8.3 and 8.5) and making a substitution for explicit terms, we rewrite Sys. (8.5) as

$$\boxed{\begin{bmatrix} \hat{A} & \hat{G} \\ \hat{D} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{r}^{n} \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{bc}_1 \\ \hat{bc}_2 \end{pmatrix},} \tag{8.6}$$

where $\hat{r}^{n} = \left[\frac{1}{\Delta t}\mathbf{I} - \frac{1}{2}\hat{L}\right]\boldsymbol{v}^{n} - \left[\frac{3}{2}\hat{\mathbf{H}}(\boldsymbol{v}^{n}) - \frac{1}{2}\hat{\mathbf{H}}(\boldsymbol{v}^{n-1})\right]$ includes viscous, non-linear and gradient terms from time steps $n, n-1$ and $\hat{bc}_i$ are explicit boundary values in terms of velocity and pressure.

8.1 Advection

Explicit Adams-Bashforth Eq. (8.3) uses information from time steps n and $n-1$, while we solve the system of equations for time step $n+1$. Therefore, we are free to choose between central differences and upwinding techniques. Ferziger & Peric [5] state that upwind difference schemes (UDS) are expected to be more stable than central difference schemes (CDS) for high Peclet numbers Pe > 2. However, we will aim to maintain low time-step and grid sizes to enhance stability and utilize the more accurate second-order CDS (compared to the first-order accuracy of UDS). If the numerical solution exhibits instability, we may switch to UDS as a trade-off between accuracy and stability. An advanced approach for the future involves blending techniques that combine UDS and CDS, following Ferziger & Perics recommendations.

Central approximations are more accurate and less dependent on the flow direction. It is convenient to write such schemes for a conservative form of the advection term. Moreover, CDS approach also make error even lower, since there is no product of velocity with acceleration. Hence, let us rewrite advective derivatives in conservative form.

$$\begin{aligned} u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= u\frac{\partial u}{\partial x} + 0 + v\frac{\partial u}{\partial y} \\ &= u\frac{\partial u}{\partial x} + u\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + v\frac{\partial u}{\partial y} \\ &= \left(u\frac{\partial u}{\partial x} + u\frac{\partial u}{\partial x}\right) + \left(u\frac{\partial v}{\partial y} + v\frac{\partial u}{\partial y}\right) \\ &= \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y}. \end{aligned} \tag{8.7}$$

Our goal now is to approximate advection components for each momentum equation. Below we will describe the discretization schemes for Eq. (8.7).

8.1.1 Inner part

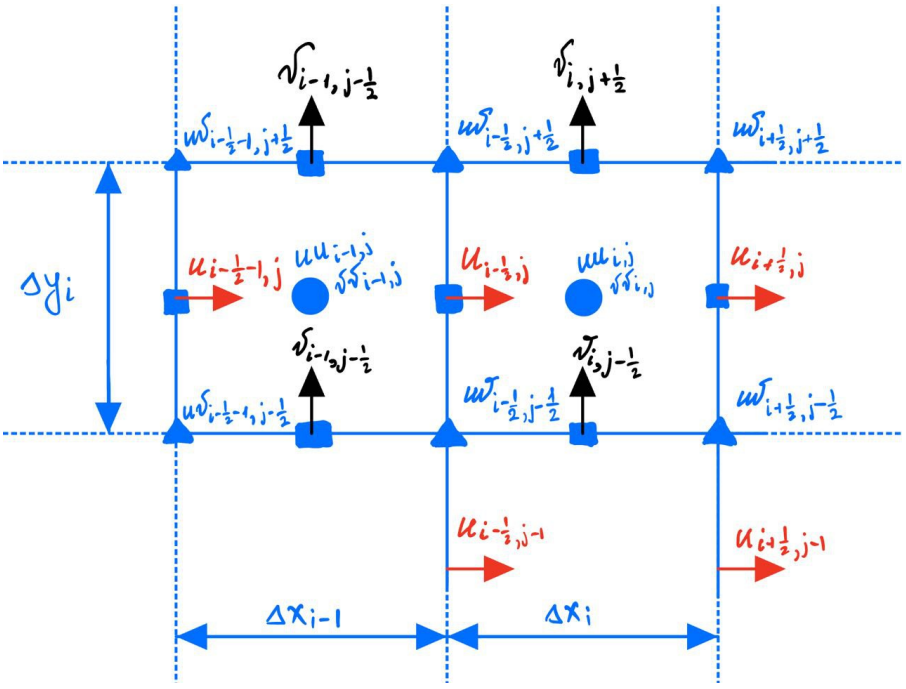


Figure 3: Advection discretization.

Discretizing through central differencing as in Colonius & Taira [3] leads to

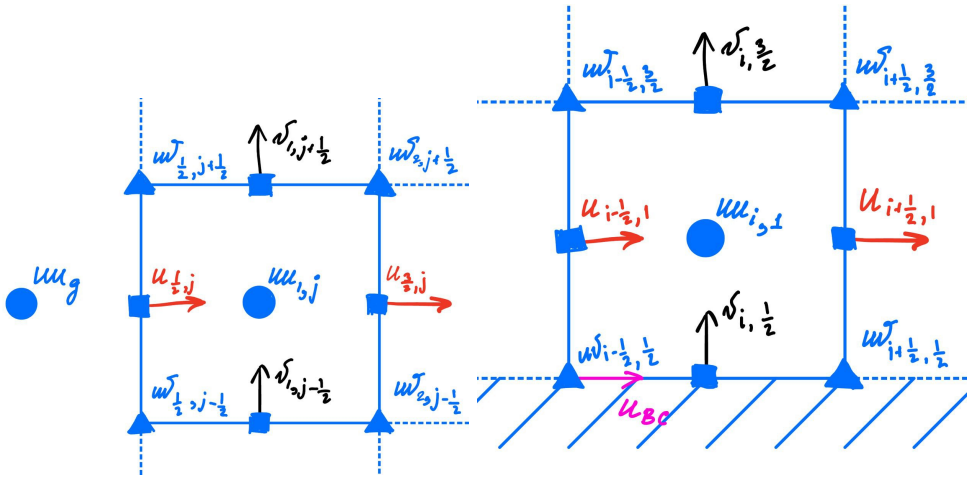
$$\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)_{i-\frac{1}{2},j}=\frac{(uu)_{i,j}-(uu)_{i-1,j}}{\frac{\Delta x_i+\Delta x_{i-1}}{2}}+\frac{(uv)_{i-\frac{1}{2},j+\frac{1}{2}}-(uv)_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y_i}, \tag{8.8}$$

which is evaluated at the same points as $u_{i-\frac{1}{2},j}$ (squares in Fig. 3). It is required to compute uv at the nodes and uu at the cell centres, which are triangles and circles respectively in Fig. 3. Use of linear interpolation for velocity values leads to:

$$\begin{aligned}(uu)_{i,j}&=\left(\frac{u_{i+\frac{1}{2},j}+u_{i-\frac{1}{2},j}}{2}\right)^2,\\ (uv)_{i-\frac{1}{2},j-\frac{1}{2}}&=\left(u_{i-\frac{1}{2},j-1}+\frac{\Delta y_{j-1}}{2}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{1}{2},j-1}}{\frac{\Delta y_{j-1}+\Delta y_j}{2}}\right)\left(v_{i-1,j-\frac{1}{2}}+\frac{\Delta x_{i-1}}{2}\frac{v_{i,j-\frac{1}{2}}-v_{i-1,j-\frac{1}{2}}}{\frac{\Delta x_{i-1}+\Delta x_i}{2}}\right)\\ &=\left(\frac{u_{i-\frac{1}{2},j}\Delta y_{j-1}+u_{i-\frac{1}{2},j-1}\Delta y_j}{\Delta y_{j-1}+\Delta y_j}\right)\left(\frac{v_{i,j-1\frac{1}{2}}\Delta x_{i-1}+v_{i-1,j-\frac{1}{2}}\Delta x_i}{\Delta x_{i-1}+\Delta x_i}\right).\end{aligned}$$

Our final system of linear equations will exclude rows and columns with known velocities. Therefore, Section 8.1.2 is displayed for illustrative purposes only. The actual description of open boundary advection discretization can be found in Section 8.1.3, where unknown velocity components exist at the boundaries.

8.1.2 Left boundary



(a) Left (b) Bottom

Figure 4: Left boundary advection.

The exact value of the uu_g element, which is located outside the boundary (Fig. 4(a)), is known from the inlet boundary conditions, whereas $uv_{\frac{1}{2},j}$ is zero for all j due to $v_{bc}=0$ at the inlet.

Computation of the advective terms just above the bottom boundary is verbatim to the inner part, with the only exception being $uv_{i-\frac{1}{2},\frac{1}{2}}=0$ at the boundary due to the $u_{bc}=0$ condition on the plate (purple on Fig. 4(b)).

The advective terms in the y-momentum equation are computed in a similar manner. After the derivatives are evaluated, they are moved to the right-hand side as in Sys. (8.5) and treated explicitly.

8.1.3 Right and top boundaries

There are unknown velocity variables that are located at the right and top boundaries. Therefore, in order to compute the advection term at these boundary coordinates we need to somehow represent the derivative that is perpendicular to the boundary. As per Ferziger & Peric [5] the inclusion of

additional unknowns should be avoided, as the number of equations corresponds directly to the number of cells. Consequently, only cell-center (cell-faces in case of velocity) values can be considered as unknowns. Since there are no nodes beyond the boundary, these approximations typically rely on one-sided differences or extrapolations. We will stick to the extrapolation technique in computation, which will allow us to use central difference schemes of second order.

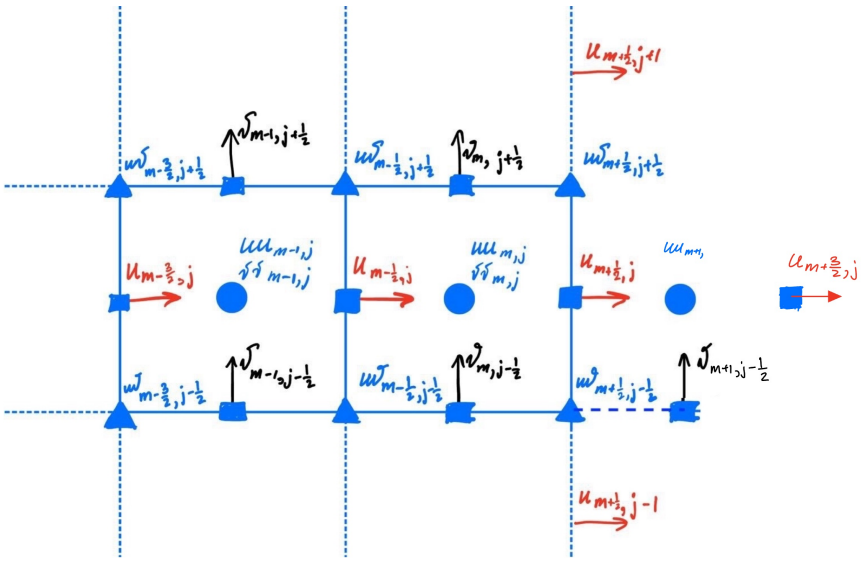


Figure 5: Right boundary advection ($m = M, n = N$).

We will use standard linear extrapolation for u and v to compute necessary velocity products uu, vv and uv that lie outside and at the boundary. Computational process for both of these are exactly the same:

$$u_{M+\frac{3}{2},j} = u_{M-\frac{1}{2},j} + (\Delta x_M + \Delta x_M)\,\frac{u_{M+\frac{1}{2},j} - u_{M-\frac{1}{2},j}}{\Delta x_M}, \tag{8.9}$$

$$v_{M+1,j-\frac{1}{2}} = v_{M-1,j-\frac{1}{2}} + \left(\frac{\Delta x_{M-1}}{2} + \frac{3\Delta x_M}{2}\right)\frac{v_{M,j-\frac{1}{2}} - v_{M-1,j-\frac{1}{2}}}{\frac{\Delta x_M + \Delta x_{M-1}}{2}}, \tag{8.10}$$

where $\Delta x_{M+1} = \Delta x_M$. At the top boundary Δx will change to Δy and extrapolation will be performed for vertical indices rather than horizontal.

8.1.4 Top-right corner

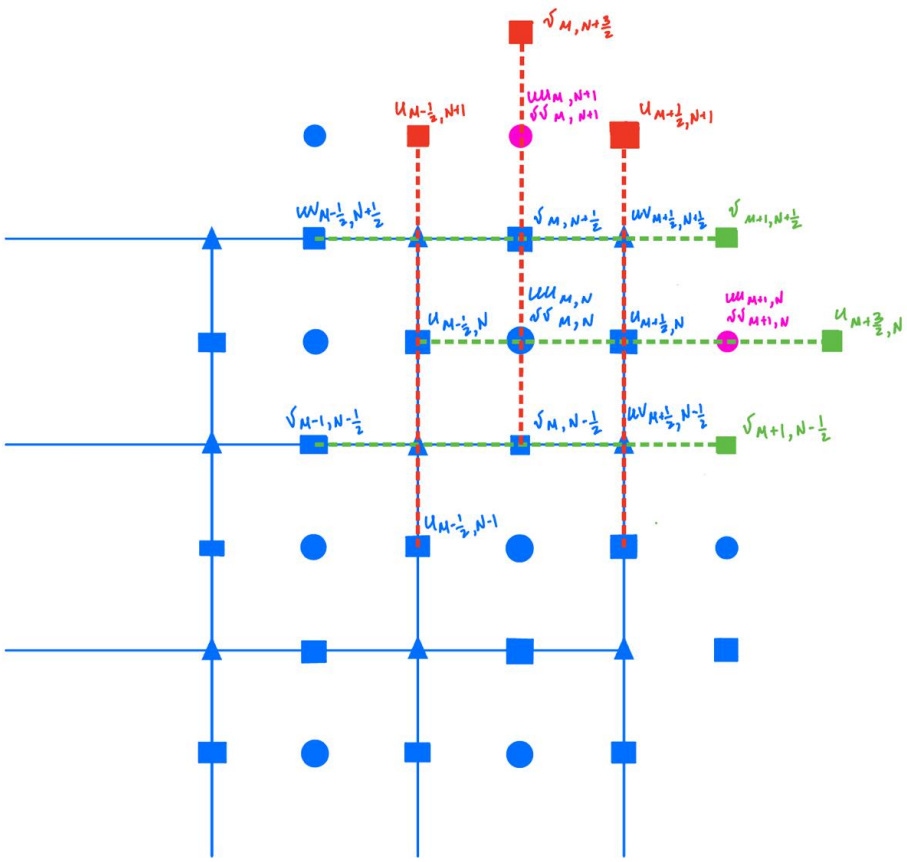


Figure 6: Top-right boundary advection.

At the corner of intersection of two boundaries with unknown velocities we will linearly extrapolate vertically (dashed red lines in Fig. 6) and horizontally (dashed green lines in Fig. 6) to first compute the virtual values of u, v that lie outside the computational domain. After the values of these virtual elements are computed we are able to compute the corresponding uu, vv, uv values and approximate the advection terms as in Section 8.1.1.

8.2 Divergence

One of the advantages of the staggered grid is that the discrete divergence D and gradient G operators are equal to the negative transpose of each other. In this section, we will show one side of why this identity is true (the other part could be found in Section 8.3). The second line of Sys. (8.5) reads:

$$\begin{aligned} \hat{D}\boldsymbol{v} &= \hat{bc}_2, \\ \begin{bmatrix} \hat{D}_x & \hat{D}_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} &= \hat{bc}_2, \\ \frac{1}{\Delta x} D_x u + \frac{1}{\Delta y} D_y v &= \hat{bc}_2, \\ \frac{1}{\Delta_{xy}} \begin{bmatrix} D_x & D_y \end{bmatrix} \begin{bmatrix} u\Delta y \\ v\Delta x \end{bmatrix} &= \frac{1}{\Delta_{xy}} Dq = \hat{bc}_2, \end{aligned} \tag{8.11}$$

where vector $q = \begin{bmatrix} u\Delta y \\ v\Delta x \end{bmatrix}$ is referred to as velocity flux, and the matrix $D = [D_x D_y]$ is the divergence matrix of integer coefficients. Matrix

$$\frac{1}{\Delta_{xy}} = \begin{bmatrix} \frac{1}{\Delta x_1 \Delta y_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\Delta x_2 \Delta y_1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\Delta x_M \Delta y_N} \end{bmatrix} \tag{8.12}$$

is $MN \times MN$ diagonal matrix with entries corresponding to the inverse volumes of cells.

The second-order central difference scheme employed in the construction of D is consistent with other spatial schemes used in this paper. Continuity equation for the cell centre that lies in a finite volume $V_{i,j}$ is expressed as

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)_{i,j} = 0,$$

which can be discretized at cell centre i, j using central difference schemes into

$$\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x_i} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y_j} = 0.$$

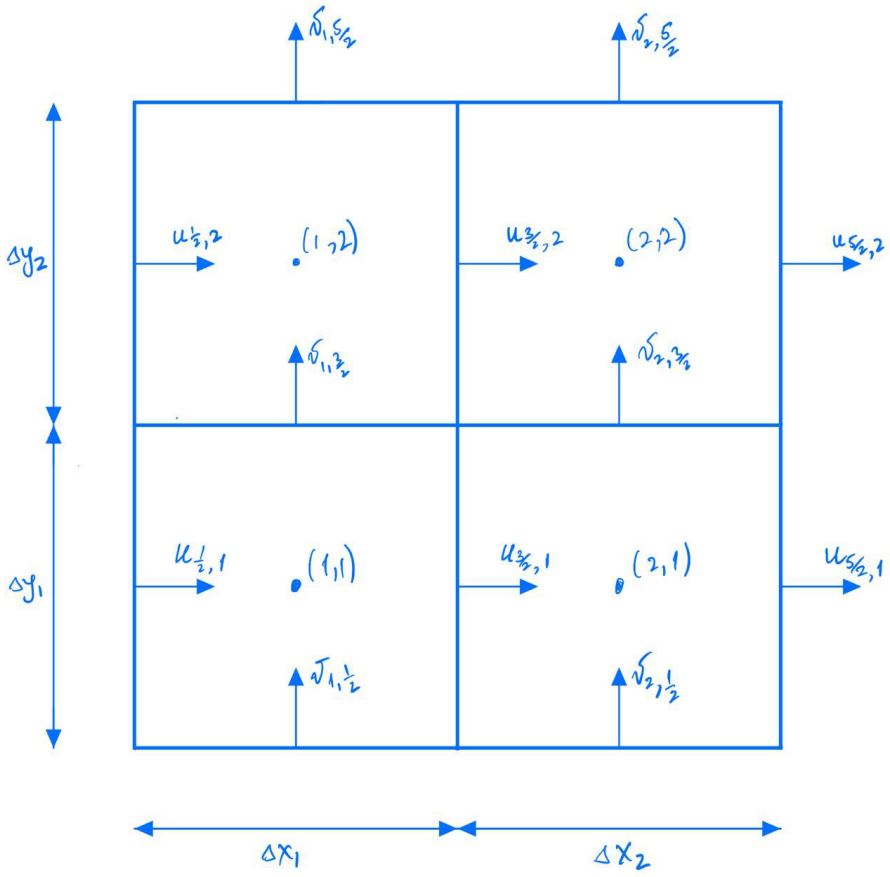


Figure 7: 2×2 grid example for divergence operator.

To provide a good visual example, consider 2×2 grid as in Fig. 7 with the same boundary conditions

as in Eqs. (6.1). The partial derivatives of divergence operator evaluated at cell centres $V_{i,j}$ are:

$$
\begin{aligned}
V_{1,1} &: \frac{u_{\frac{3}{2},1} - u_{\frac{1}{2},1}}{\Delta x_1} + \frac{v_{1,\frac{3}{2}} - v_{1,\frac{1}{2}}}{\Delta y_1} = 0, \\
V_{1,2} &: \frac{u_{\frac{5}{2},1} - u_{\frac{3}{2},1}}{\Delta x_2} + \frac{v_{2,\frac{3}{2}} - v_{2,\frac{1}{2}}}{\Delta y_1} = 0, \\
V_{2,1} &: \frac{u_{\frac{3}{2},2} - u_{\frac{1}{2},2}}{\Delta x_1} + \frac{v_{1,\frac{5}{2}} - v_{1,\frac{3}{2}}}{\Delta y_2} = 0, \\
V_{2,2} &: \frac{u_{\frac{5}{2},2} - u_{\frac{3}{2},2}}{\Delta x_2} + \frac{v_{2,\frac{5}{2}} - v_{2,\frac{3}{2}}}{\Delta y_2} = 0.
\end{aligned}
$$

After applying the same operations as in Eqs. (8.11) to discrete continuity equations from above, we obtain the following system

$$
\operatorname{diag}\begin{pmatrix} \frac{1}{\Delta x_1 \Delta y_1} \\ \frac{1}{\Delta x_2 \Delta y_1} \\ \frac{1}{\Delta x_1 \Delta y_2} \\ \frac{1}{\Delta x_2 \Delta y_2} \end{pmatrix}
\begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
u_{\frac{1}{2},1}\Delta y_1 \\ u_{\frac{3}{2},1}\Delta y_1 \\ u_{\frac{5}{2},1}\Delta y_1 \\
u_{\frac{1}{2},2}\Delta y_2 \\ u_{\frac{3}{2},2}\Delta y_2 \\ u_{\frac{5}{2},2}\Delta y_2 \\
v_{1,\frac{1}{2}}\Delta x_1 \\ v_{2,\frac{1}{2}}\Delta x_2 \\
v_{1,\frac{3}{2}}\Delta x_1 \\ v_{2,\frac{3}{2}}\Delta x_2 \\
v_{1,\frac{5}{2}}\Delta x_1 \\ v_{2,\frac{5}{2}}\Delta x_2
\end{bmatrix}
= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8.13}
$$

Sys. (8.13) in general case is written as $\frac{1}{\Delta_{xy}} Dq = \hat{bc}_2$. We have used boundary velocities in vector $\boldsymbol{v}$ (as recommended by Chang [2]) to eliminate any explicit boundary condition treatment of the incompressibility constraint. The divergence matrix D contains zeros and ± 1 as shown in Sys. (8.13).

8.3 Gradient

The pressure gradients are computed at the same coordinates as the unknown velocities in Sys. (8.1). In this section, we will demonstrate that the discrete gradient and divergence operators comply with

$$
G = -D^T \tag{8.14}
$$

on staggered/MAC grids by construction.

As an illustrative example, we may consider gradient operator on 2×2 grid as in Fig. 8 with boundary conditions from Eqs. (6.1). The normal distances between the computational domain and virtual pressure values located outside of it are considered to be infinitesimally small [2]. It is required to determine the pressure gradients across each unknown velocity (square nodes on Fig. 8):

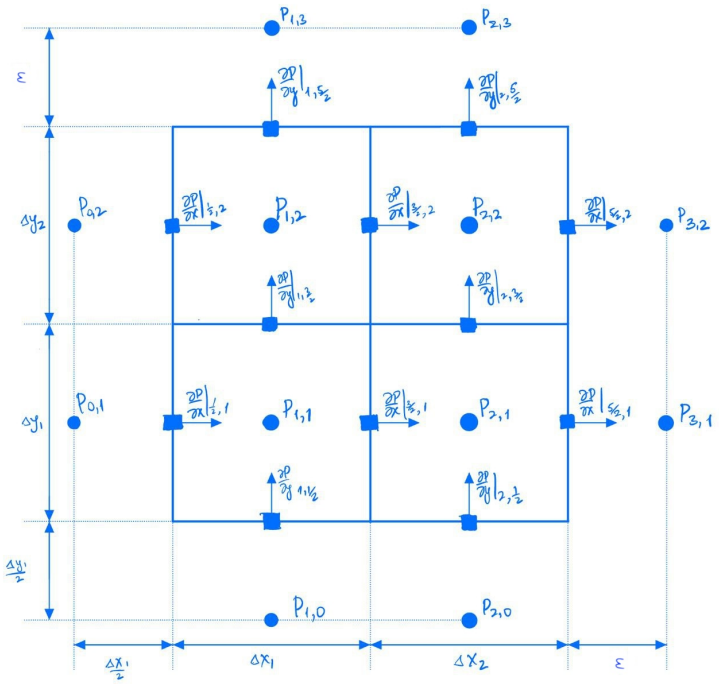


Figure 8: 2×2 grid example for gradient operator ($\varepsilon \to 0$).

$$
\begin{aligned}
u_{\frac{1}{2},1} &: \frac{2}{\Delta x_1}\left(p_{1,1} - p_{0,1}\right), \\
u_{\frac{3}{2},1} &: \frac{2}{\Delta x_1 + \Delta x_2}\left(p_{2,1} - p_{1,1}\right), \\
u_{\frac{5}{2},1} &: \frac{2}{\Delta x_2}\left(p_{3,1} - p_{2,1}\right), \\
u_{\frac{1}{2},2} &: \frac{2}{\Delta x_1}\left(p_{1,2} - p_{0,2}\right), \\
u_{\frac{3}{2},2} &: \frac{2}{\Delta x_1 + \Delta x_2}\left(p_{2,2} - p_{1,2}\right), \\
u_{\frac{5}{2},2} &: \frac{2}{\Delta x_2}\left(p_{3,2} - p_{2,2}\right), \\
v_{1,\frac{1}{2}} &: \frac{2}{\Delta y_1}\left(p_{1,1} - p_{1,0}\right), \\
v_{2,\frac{1}{2}} &: \frac{2}{\Delta y_1}\left(p_{2,1} - p_{2,0}\right), \\
v_{1,\frac{3}{2}} &: \frac{2}{\Delta y_1 + \Delta y_2}\left(p_{1,2} - p_{1,1}\right), \\
v_{2,\frac{3}{2}} &: \frac{2}{\Delta y_1 + \Delta y_2}\left(p_{2,2} - p_{2,1}\right), \\
v_{1,\frac{5}{2}} &: \frac{2}{\Delta y_2}\left(p_{1,3} - p_{1,2}\right), \\
v_{2,\frac{5}{2}} &: \frac{2}{\Delta y_2}\left(p_{2,3} - p_{2,2}\right).
\end{aligned}
$$

The next step is to rewrite the above expressions in matrix form using the pressure boundary

conditions. The resultant system of linear equations then becomes

$$\text{diag}\begin{pmatrix} \frac{2}{\Delta x_1} \\ \frac{2}{\Delta x_1+\Delta x_2} \\ \frac{2}{\Delta x_2} \\ \frac{2}{\Delta x_1} \\ \frac{2}{\Delta x_1+\Delta x_2} \\ \frac{2}{\Delta x_2} \\ \frac{2}{\Delta y_1} \\ \frac{2}{\Delta y_1} \\ \frac{2}{\Delta y_1+\Delta y_2} \\ \frac{2}{\Delta y_1+\Delta y_2} \\ \frac{2}{\Delta y_2} \\ \frac{2}{\Delta y_2} \end{pmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} p_{1,1} \\ p_{2,1} \\ p_{1,2} \\ p_{2,2} \end{bmatrix} + \begin{bmatrix} p_{0,1} \\ 0 \\ p_{3,1} \\ p_{0,2} \\ 0 \\ p_{3,2} \\ p_{1,0} \\ p_{2,0} \\ 0 \\ 0 \\ p_{1,3} \\ p_{2,3} \end{bmatrix} \right), \tag{8.15}$$

which can be rewritten in compact format as

$$\nabla p + bc_p \Longleftrightarrow \hat{M}^{-1} \left(\hat{G} \begin{bmatrix} p_{1,1} \\ p_{2,1} \\ \vdots \\ p_{M,N} \end{bmatrix} + bc_p \right),$$

where $\hat{M}^{-1}$ is the diagonal matrix containing the inverse distances between neighbouring pressure coordinates; G is integer gradient matrix and $[p_{1,1}, p_{2,1}, ..., p_{M,N}]^T$ is the vector of pressure values at the cell centres. There are some pressure BC values in bc_p since there is a pressure gradient that has to be computed across the boundary. It can be observed from Sys. (8.13) and (8.15) that Eq. (8.14) holds and the general case is true by construction.

Lastly, it is worth mentioning that pressure values at the boundaries with known velocities (inlet and wall) are not needed. The reason is that after constructing all momentum equations the corresponding rows are replaced by zeroes.

8.4 Curl operator and pressure elimination

The goal of this subsection is to show how unknown pressure variables can be eliminated from Sys. (10.1) similarly to Section 2. Publications of Chang [2] and Hall [6] use the idea that in Sys. (10.1) matrix D is wider than tall, hence it defines a nullspace which we denote as C.

The number of rows in the nullspace C is equal to the number of momentum equations for all the velocities. In two dimensions C has N_n columns, which is equal to the number of the nodes in the grid, whereas in three dimensions the nullspace has N_e columns being the number of edges.

In the two-dimensional case, the matrix C has two non-zero elements in each row ($+1$ and -1). The $+1$ value corresponds to the node $90°$ from the normal velocity vector, whereas -1 corresponds to the node $-90°$ from the normal velocity vector. For three dimensional case see Chang [2].

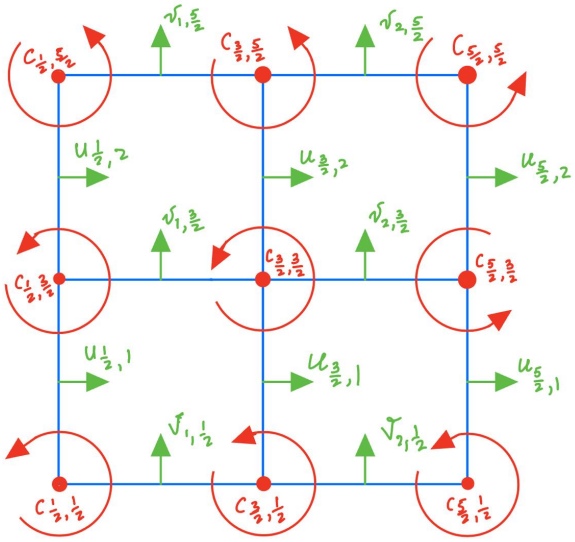


Figure 9: 2×2 example for C matrix.

A more intuitive way of constructing the matrix C relies on the utilization of counterclockwise vorticity around the nodes within the domain (Fig. 9). If the direction of the velocity vector on the adjacent face aligns with the vorticity's direction, +1 is assigned to the corresponding row; conversely, -1 is assigned in the case of opposite directions of velocity and vorticity. After applying the above procedure to all nodes we obtain

$$C = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \tag{8.16}$$

The desired product then becomes $DC = 0$. $D = -G^T$ (derived in Sections 8.2 and 8.3) leads to important property $(DC)^T = C^T D^T = -C^T G = 0$. Premultiplying momentum equation from Sys. (10.1) by C^T creates $C^T Gp = 0$ term, which eliminates the pressure from our system, however, leaves pressure boundary conditions which we will discuss in Section 9.

In order to make use of efficient solvers, the matrix $C^T A$ can be made symmetric if multiplied by C from the right. The matrix $C^T C$ is equivalent to Laplacian (Chang [2] uses RC notation), whereas $C^T LC$ is equivalent to symmetric biharmonic operator of streamfunction ψ as in Eq. (3.9). Laplacian operator discretization will be discussed in the following Section 8.5.

8.5 Laplacian

As per Sys. (8.5), it is required to approximate implicit viscous terms spatially. In order to obtain the discrete operator acting on a velocity vector we will rewrite Laplacian as a block matrix:

$$\hat{L} = \begin{bmatrix} \hat{L}^u_{xx} + \hat{L}^u_{yy} & 0 \\ 0 & \hat{L}^v_{xx} + \hat{L}^v_{yy} \end{bmatrix}.$$

Each pair of the matrices $\hat{L}^u_{xx}, \hat{L}^v_{xx}$ and $\hat{L}^u_{yy}, \hat{L}^v_{yy}$ will be equal on a uniform, but different on non-uniform grids. These matrices are computed in a similar manner. Below we will discuss how the Laplacian matrix is constructed on different parts of the grid. We will only display inner, bottom and left part discretizations in this section, the rest will be derived in artificial boundary conditions Section 9.

8.5.1 Inner part.

The uniform grid refinement $\Delta x_{i+1} = k_x \Delta x_i, \Delta y_{j+1} = k_y \Delta y_j$ makes the order of the scheme consistent throughout the whole domain. Without loss of generality, let us compute $\hat{L}^u_{xx}$. The other three matrices can be constructed in the same way. The power series of of $u_{i-\frac{1}{2}\pm1,j}$ at nodes $x_{i-\frac{1}{2}\pm1}$ with respect to $u_{i-\frac{1}{2},j}$ at node $x_{i-\frac{1}{2}}$ are:

$$u_{i-\frac{1}{2}+1,j} = u_{i-\frac{1}{2},j} + \left.\frac{\partial u}{\partial x}\right|_{i-\frac{1}{2},j} \left(x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}}\right) + \frac{1}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{i-\frac{1}{2},j} \left(x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}}\right)^2 + O\left(\Delta x^3\right), \quad (8.17)$$

$$u_{i-\frac{1}{2}-1,j} = u_{i-\frac{1}{2},j} + \left.\frac{\partial u}{\partial x}\right|_{i-\frac{1}{2},j} \left(x_{i-\frac{1}{2}-1} - x_{i-\frac{1}{2}}\right) + \frac{1}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{i-\frac{1}{2},j} \left(x_{i-\frac{1}{2}-1} - x_{i-\frac{1}{2}}\right)^2 + O\left(\Delta x^3\right). \quad (8.18)$$

We can combine Eqs. (8.17) and (8.18) by cancelling the first derivative, which will result in

$$\begin{aligned}\left.\frac{\partial^2 u}{\partial x^2}\right|_{i-\frac{1}{2},j} &\approx \frac{u_{i-\frac{1}{2}+1,j}\left(x_{i-\frac{1}{2}} - x_{i-\frac{1}{2}-1}\right) - u_{i-\frac{1}{2},j}\left(x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}-1}\right) + u_{i-\frac{1}{2}-1,j}\left(x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}}\right)}{\left(\frac{x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}-1}}{2}\right)\left(x_{i-\frac{1}{2}} - x_{i-\frac{1}{2}-1}\right)\left(x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}}\right)} + O(\Delta x) \\ &\approx u_{i-\frac{1}{2}+1}\frac{1}{h_c h_e} - u_{i-\frac{1}{2}}\frac{2}{h_w h_e} + u_{i-\frac{1}{2}-1}\frac{1}{h_c h_w} + O(\Delta x), \end{aligned} \quad (8.19)$$

where $h_w = x_{i-\frac{1}{2}} - x_{i-\frac{1}{2}-1}, h_c = \frac{x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}-1}}{2}, h_e = x_{i-\frac{1}{2}+1} - x_{i-\frac{1}{2}}$ (cf. Fig. 10). The coefficients after the velocity components are then placed into the $\hat{L}^u_{xx}$ matrix. The order of this approximation becomes 2^{nd} on uniform grids.

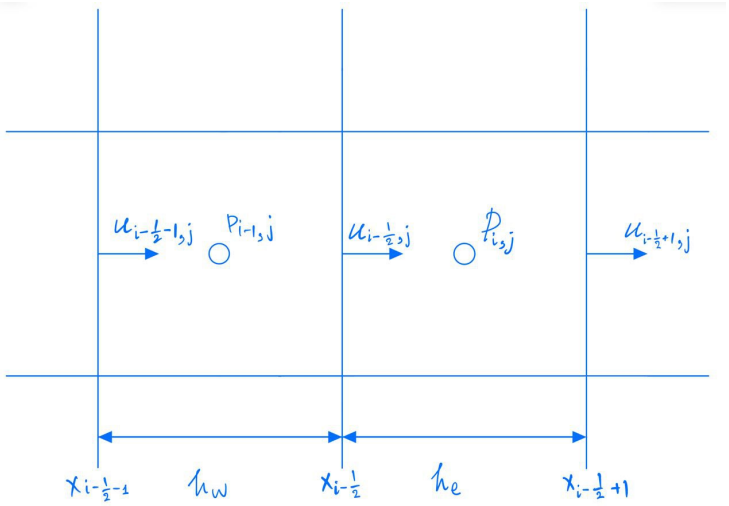


Figure 10: $\hat{L}^u_{xx}$ inner part.

8.5.2 Left boundary.

The exact value at the left boundary (normal velocity $u_{\frac{1}{2},j}$ on Fig. 11) is given as a Dirichlet boundary condition. The value of $u_{\frac{1}{2},j}$ is known, therefore it is possible to move it to the vector $\hat{bc}_1$. Discretization

of Laplacian operator of $u_{\frac{1}{2}+1,j}$ term is then

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{\frac{1}{2}+1,j} = \frac{u_{\frac{1}{2},j}\left(\Delta x_2\right) + u_{\frac{1}{2}+1,j}\left(-[\Delta x_1 + \Delta x_2]\right) + u_{\frac{1}{2}+2,j}\left(\Delta x_1\right)}{\Delta x_1 \frac{\Delta x_1+\Delta x_2}{2} \Delta x_2}, \tag{8.20}$$

which in terms of h_w, h_c, h_e and corresponding row of $\hat{L}^u_{xx}$ becomes

$$\left.\hat{L}^u_{xx}\right|_{\frac{1}{2}+1,j} + \left.\hat{L}^u_{\hat{bc}_1}\right|_{\frac{1}{2}+1,j} \iff \left[\frac{u_{\frac{1}{2}+1,j}\left(-2h_c\right) + u_{\frac{1}{2}+2,j}\left(h_w\right)}{h_e h_c h_w}\right] + \left[\frac{u_{\frac{1}{2},j}\left(h_e\right)}{h_e h_c h_w}\right], \tag{8.21}$$

where $\hat{L}^u_{\hat{bc}_1}$ is contribution of boundary condition from $\hat{L}^u_{xx}$ to $\hat{bc}_1$. We treat normal boundary conditions explicitly in momentum equations (Chang [2] suggests that only incompressibility constraint to be not treated explicitly). Therefore, there is no need to discretize Laplacian at $u_{\frac{1}{2},j}$.

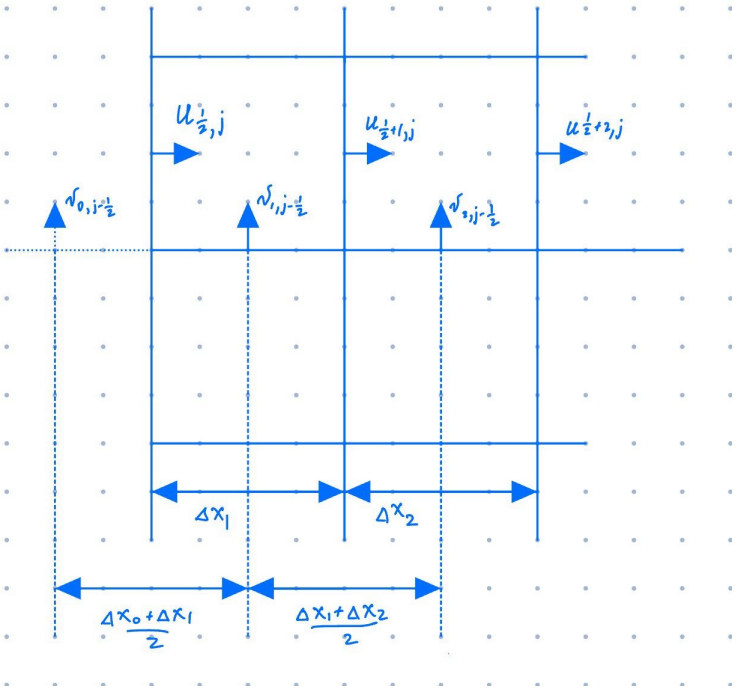


Figure 11: $\hat{L}^u_{xx}$ at the left boundary.

In fact, it is possible to directly use one of the velocities directly from the boundary when computing Laplacian at $v_{1,j-\frac{1}{2}}$. However, the order of such a scheme will then be reduced due to the large ratio of distances between the neighbouring velocities. If, on the other hand, we introduce a ghost velocity $v_{0,j-\frac{1}{2}}$, we have the freedom to chose its coordinate. The corresponding viscous term at $v_{1,j-\frac{1}{2}}$ component becomes

$$\left.\frac{\partial^2 v}{\partial x^2}\right|_{1,j-\frac{1}{2}} = \frac{v_{0,j-\frac{1}{2}}\left(\frac{\Delta x_1+\Delta x_2}{2}\right) + v_{1,j-\frac{1}{2}}\left(-\frac{\Delta x_1+\Delta x_2}{2} - \frac{\Delta x_0+\Delta x_1}{2}\right) + v_{2,j-\frac{1}{2}}\left(\frac{\Delta x_0+\Delta x_1}{2}\right)}{\frac{\Delta x_1+\Delta x_2}{2}\frac{\frac{\Delta x_1+\Delta x_2}{2}+\frac{\Delta x_0+\Delta x_1}{2}}{2}\frac{\Delta x_0+\Delta x_1}{2}}, \tag{8.22}$$

applying $v_{0,j-\frac{1}{2}} = 0$, using h_e, h_c, h_w and writing $\hat{L}^v_{xx}$ with $\hat{L}^v_{\hat{bc}_1}$ leads to

$$\left.\hat{L}^v_{xx}\right|_{1,j-\frac{1}{2}} + \left.\hat{L}^v_{\hat{bc}_1}\right|_{1,j-\frac{1}{2}} \iff \left[\frac{v_{1,j-\frac{1}{2}}\left(-2h_c\right) + v_{2,j-\frac{1}{2}}\left(h_e\right)}{h_e h_c h_w}\right] + \left[0\right]. \tag{8.23}$$

8.5.3 Bottom boundary.

Let us use Eq. (8.19) and change the distances between velocities as in Fig. 14. This leads to

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_{i-\frac{1}{2},1} = u_{i-\frac{1}{2},2}\left(\frac{1}{h_n h_c}\right) + u_{i-\frac{1}{2},1}\left(\frac{-2}{h_n h_s}\right) + u_{i-\frac{1}{2},0}\left(\frac{1}{h_s h_c}\right). \tag{8.24}$$

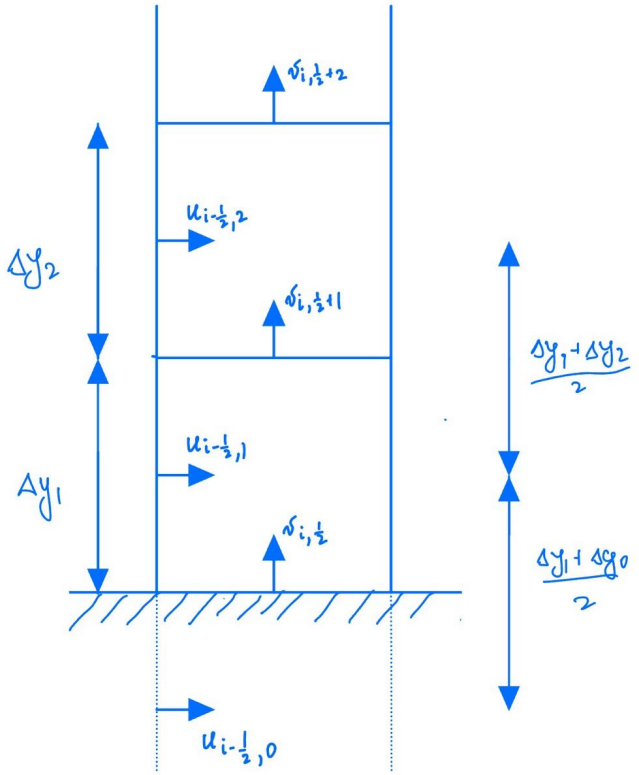


Figure 12: $\hat{L}^u_{yy}$ at the bottom boundary.

The position of $u_{i-\frac{1}{2},0}$ is not fixed, we can make the order of the scheme $O(\Delta y^2)$ if $h_s = \frac{\Delta y_1 + \Delta y_0}{2} \equiv h_n = \frac{\Delta y_1 + \Delta y_2}{2}$ when computing Laplacian for $u_{i-\frac{1}{2},1}$. Interpolation of velocity at the wall $u_w = \frac{u_{i-\frac{1}{2},1} + u_{i-\frac{1}{2},0}}{2} = 0 \Rightarrow u_{i-\frac{1}{2},0} = -u_{i-\frac{1}{2},1}$, then

$$\begin{aligned}\left.\frac{\partial^2 u}{\partial y^2}\right|_{i-\frac{1}{2},1} &= u_{i-\frac{1}{2},2}\left(\frac{1}{h_n h_c}\right) + u_{i-\frac{1}{2},1}\left(\frac{-2}{h_n h_s} + \frac{-1}{h_s h_c}\right) \\ &= u_{i-\frac{1}{2},2}\left(\frac{1}{h_n h_c}\right) + u_{i-\frac{1}{2},1}\left(\frac{-2h_c - h_n}{h_s h_c h_n}\right).\end{aligned} \tag{8.25}$$

The coefficients from Eq. (8.25) are distributed into the $\hat{L}^u_{yy}$ matrix as

$$\hat{L}^u_{yy}\Big|_{i-\frac{1}{2},1} + \hat{L}^u_{\hat{bc}_1}\Big|_{i-\frac{1}{2},1} \iff \left[u_{i-\frac{1}{2},2}\left(\frac{1}{h_n h_c}\right) + u_{i-\frac{1}{2},1}\left(\frac{-2h_c - h_n}{h_s h_c h_n}\right)\right] + [0]\,. \tag{8.26}$$

Computation of Laplacian at $v_{i,\frac{1}{2}+1}$ can be performed similarly to the $\hat{L}^u_{xx}$ at the left boundary

$$\hat{L}^v_{yy}\Big|_{i,\frac{1}{2}+1} + \hat{L}^v_{\hat{bc}_1}\Big|_{i,\frac{1}{2}+1} \iff \left[v_{i,\frac{1}{2}+2}\left(\frac{1}{h_n h_c}\right) + v_{i,\frac{1}{2}+1}\left(\frac{-2}{h_s h_n}\right)\right] + [0]\,, \tag{8.27}$$

where $h_s = \Delta y_1, h_n = \Delta y_2, h_c = \frac{h_s + h_n}{2}$ and wall velocity $v_{i,\frac{1}{2}+1} = 0$ leads to $\hat{L}^v_{\hat{bc}_1}\Big|_{i,\frac{1}{2}+1} = 0$.

9 Artificial boundary conditions

In numerical simulations of the fluid flow, solving problems on unbounded domains with finite computational resources necessitates the implementation of artificial boundary conditions (ABCs). These conditions effectively mimic the behaviour of the flow at the boundaries, ensuring that numerical simulations accurately capture the dynamics within the computational domain. Numerous studies have contributed to the development of ABCs, each aiming to minimize spurious reflections and maintain fidelity to physical phenomena.

Key contributions in this area include seminal works such as from Braza and collaborators [8,9,11] nonreflecting outlet boundary condition for incompressible unsteady Navier-Stokes calculations. Liu & Liu [10] exploration of high-order finite difference methods for instability in planar channels has significantly advanced our understanding of ABCs. Studies by Sani & Gresho [12] on open boundary conditions and Tsynkov's [13] comprehensive review on numerical solutions for unbounded domains provided valuable insights into the area of artificial boundary conditions.

Among these foundational studies, the method proposed by Dong, Karniadakis & Chryssostomidis in their 2014 paper stands out for its robustness and accuracy in handling outflow boundaries on severely truncated unbounded domains [4]. This method provided a reliable framework for simulating flows where conventional ABCs may fail, offering a comprehensive solution to the challenges posed by unbounded domains in CFD simulations. Notably, this method also stated Dirichlet boundary conditions for pressure at the outlet, which are essential for our discrete streamfunction method, as highlighted by Chang [2]. In the subsequent sections, we delve into the principles underlying these ABCs, with a particular focus on the method proposed by Dong et al.

The general equation at the open boundary is given by

$$-p\hat{\boldsymbol{n}}+\epsilon\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{v}-\left[\frac{1}{2}|\boldsymbol{v}|^2S_0(\hat{\boldsymbol{n}}\cdot\boldsymbol{v})\right]\hat{\boldsymbol{n}}=\mathbf{f}_b(\mathbf{x},t),\quad\text{on }\partial\Omega_0:=\{x=1,0\le y\le 1\}\cup\{0\le x\le 1,y=1\},\tag{9.1}$$

were $\hat{\boldsymbol{n}}$ represents the outward-pointing unit vector normal to $\partial\Omega_0$, and $|\boldsymbol{v}|$ denotes the velocity magnitude. Element $\mathbf{f}_b$ is a vector function on $\partial\Omega_0$, mainly used for numerical testing and set to zero during actual simulations. In the scope of this paper, we will set $\mathbf{f}_b$ to zero.

The function $S_0(\hat{\boldsymbol{n}}\cdot\boldsymbol{v})$ is defined as:

$$S_0(\hat{\boldsymbol{n}}\cdot\boldsymbol{v})=\frac{1}{2}\left(1-\tanh\frac{\hat{\boldsymbol{n}}\cdot\boldsymbol{v}}{U_0\delta}\right),\tag{9.2}$$

where U_0 represents the characteristic velocity scale, and $\delta>0$ is a chosen non-dimensional positive constant. δ controls the sharpness of the smoothed step function, being sharper with smaller δ values. Essentially, $S_0(\hat{\boldsymbol{n}}\cdot\boldsymbol{v})$ behaves like a smoothed step function, predominantly being equal to one when $\hat{\boldsymbol{n}}\cdot\boldsymbol{v}<0$ and zero elsewhere. It is worth noting that the numerical tests of Dong et al. demonstrate that the simulation results are not significantly affected by the specific value of δ when it is chosen to be relatively small.

In order to finish the construction of the Laplace matrix and pressure boundary conditions vector we will need velocity and pressure equations from BC (9.1). Dong et al. provide the following identities:

$$p^{n+1}=\epsilon\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{v}^{*,n+1}\cdot\hat{\boldsymbol{n}}-\frac{1}{2}\left|\boldsymbol{v}^{*,n+1}\right|^2S_0\left(\hat{\boldsymbol{n}}\cdot\boldsymbol{v}^{*,n+1}\right)-\mathbf{f}_b^{n+1}\cdot\hat{\boldsymbol{n}},\quad\text{on }\partial\Omega_0,\tag{9.3}$$

$$\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{v}^{n+1}=\frac{1}{\epsilon}\left[p^{n+1}\hat{\boldsymbol{n}}+\frac{1}{2}\left|\boldsymbol{v}^{*,n+1}\right|^2S_0\left(\hat{\boldsymbol{n}}\cdot\boldsymbol{v}^{*,n+1}\right)\hat{\boldsymbol{n}}-\epsilon\left(\nabla\cdot\boldsymbol{v}^{*,n+1}\right)\hat{\boldsymbol{n}}+\mathbf{f}_b^{n+1}\right],\quad\text{on }\partial\Omega_0,\tag{9.4}$$

where $\boldsymbol{v}^{*,n+1}$ variable represents a k-th order approximation of $\boldsymbol{v}^{n+1}$ as

$$\boldsymbol{v}^{*,n+1}=\begin{cases}\boldsymbol{v}^n, & \text{if } k=1,\\ 2\boldsymbol{v}^n-\boldsymbol{v}^{n-1}, & \text{if } k=2,\end{cases}\tag{9.5}$$

and $\left|\boldsymbol{v}^{*,n+1}\right|=\left|\left(u^{*,n+1}\right)^2+\left(v^{*,n+1}\right)^2\right|$. Rewriting BCs (9.3) and (9.4) when $k=1$ for each component and using $\mathbf{f}^{n+1}=0$ leads to:

$$p^{n+1}=\epsilon\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{v}^n\cdot\hat{\boldsymbol{n}}-\frac{1}{2}\left|\boldsymbol{v}^n\right|^2S_0\left(\hat{\boldsymbol{n}}\cdot\boldsymbol{v}^n\right),\quad\text{on }\partial\Omega_0,\tag{9.6}$$

$$\hat{\boldsymbol{n}} \cdot \nabla \boldsymbol{v}^{n+1} = \frac{1}{\epsilon}\left[p^{n+1}\hat{\boldsymbol{n}} + \frac{1}{2}\left|\boldsymbol{v}^n\right|^2 S_0\left(\hat{\boldsymbol{n}} \cdot \boldsymbol{v}^n\right)\hat{\boldsymbol{n}} - \epsilon\left(\nabla \cdot \boldsymbol{v}^n\right)\hat{\boldsymbol{n}}\right], \quad \text{on } \partial\Omega_0. \tag{9.7}$$

9.1 Pressure vector for top and right boundaries

In order to use BC (9.6) it is necessary to express velocity gradient term $\epsilon\hat{\boldsymbol{n}} \cdot \nabla \boldsymbol{v}^n \cdot \hat{\boldsymbol{n}}$ at time step n. Let us calculate this pressure value at the right boundary $\hat{\boldsymbol{n}} = (1, 0)$:

$$p^{n+1} = \epsilon\frac{\partial u^n}{\partial x} - \frac{1}{4}\left|(u^n)^2 + (v^n)^2\right|\left(1 - \tanh\frac{u^n}{U_0\delta}\right). \tag{9.8}$$

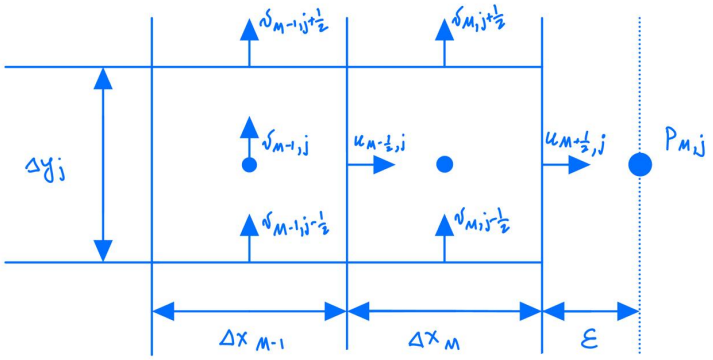


Figure 13: Pressure boundary condition at the right boundary ($\epsilon \to 0$).

Due to $\varepsilon \to 0$, position of $P_{M,j} \to u_{M+\frac{1}{2},j}$. Therefore, it is necessary to evaluate velocity magnitude at coordinate $M + \frac{1}{2}, j$. Component u is inherently located at this coordinate, however, v component needs to be extrapolated. This is possible since we already computed all necessary velocity values at previous time step n. First derivative $\frac{\partial u^n}{\partial x}$ will be approximated using backward difference. Finally, we obtain contribution to pressure boundary condition vector as

$$p^{n+1}_{M,j} = \epsilon\frac{u^n_{M+\frac{1}{2},j} - u^n_{M-\frac{1}{2},j}}{\Delta x_M} - \frac{1}{4}\left|\left(u^n_{M+\frac{1}{2},j}\right)^2 + \left(v^n_{M+\frac{1}{2},j}\right)^2\right|\left(1 - \tanh\frac{u^n_{M+\frac{1}{2},j}}{U_0\delta}\right). \tag{9.9}$$

Similarly, at the top boundary we get

$$p^{n+1}_{i,N} = \epsilon\frac{v^n_{i,N+\frac{1}{2}} - v^n_{i,N-\frac{1}{2}}}{\Delta x_M} - \frac{1}{4}\left|\left(u^n_{i,N+\frac{1}{2}}\right)^2 + \left(v^n_{i,N+\frac{1}{2}}\right)^2\right|\left(1 - \tanh\frac{v^n_{i,N+\frac{1}{2}}}{U_0\delta}\right). \tag{9.10}$$

9.2 Laplacian for top and right boundaries

In Section 8.5 we computed discrete Laplacian operator for inner part, wall and inlet. Here we will approximate this operator using artificial boundary conditions. We will first simplify BC (9.7) algebraically. This removes unnecessary numerical calculations involving p^{n+1} term. Taking BC (9.6) and plugging it into BC (9.7) leads to

$$\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{v}^{n+1} = \frac{1}{\epsilon}\left[\left(\epsilon\hat{\boldsymbol{n}} \cdot \nabla\boldsymbol{v}^n \cdot \hat{\boldsymbol{n}} - \frac{1}{2}\left|\boldsymbol{v}^n\right|^2 S_0\left(\hat{\boldsymbol{n}} \cdot \boldsymbol{v}^n\right)\right)\hat{\boldsymbol{n}} + \frac{1}{2}\left|\boldsymbol{v}^n\right|^2 S_0\left(\hat{\boldsymbol{n}} \cdot \boldsymbol{v}^n\right)\hat{\boldsymbol{n}} - \epsilon\left(\nabla \cdot \boldsymbol{v}^n\right)\hat{\boldsymbol{n}}\right], \quad \text{on } \partial\Omega_0, \tag{9.11}$$

which after simplification reads as

$$\hat{\boldsymbol{n}} \cdot \nabla\boldsymbol{v}^{n+1} = \left(\hat{\boldsymbol{n}} \cdot \nabla\boldsymbol{v}^n \cdot \hat{\boldsymbol{n}}\right)\hat{\boldsymbol{n}} - \left(\nabla \cdot \boldsymbol{v}^n\right)\hat{\boldsymbol{n}}, \quad \text{on } \partial\Omega_0. \tag{9.12}$$

Now we can directly use BC (9.12) to compute the remaining Laplacian components.

Right boundary. At $x = 1$ we have $\hat{\boldsymbol{n}} = (1, 0)$, this reduces BC (9.12) to

$$\begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial x} \end{pmatrix}^{n+1} = \begin{pmatrix} -\frac{\partial v}{\partial y} \\ 0 \end{pmatrix}^{n}, \tag{9.13}$$

which in discrete form can be written as

$$\frac{u^{n+1}_{M+\frac{3}{2},j} - u^{n+1}_{M+\frac{1}{2},j}}{\Delta x_{M+1}} = -\frac{v^{n}_{M+1,j+\frac{1}{2}} - v^{n}_{M+1,j-\frac{1}{2}}}{\Delta y_j}. \tag{9.14}$$

Expressing the outer term leads to

$$u^{n+1}_{M+\frac{3}{2},j} = -\frac{v^{n}_{M+1,j+\frac{1}{2}} - v^{n}_{M+1,j-\frac{1}{2}}}{\Delta y_j}\Delta x_{M+1} + u^{n+1}_{M+\frac{1}{2},j}. \tag{9.15}$$

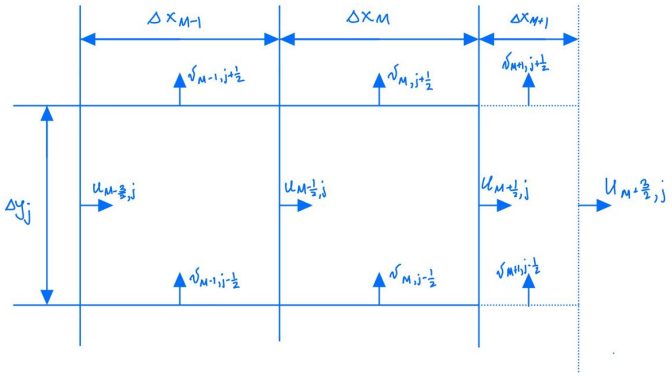


Figure 14: Laplacian at the right boundary.

Let us write the second derivative of u w.r.t. x using inner part Eq. (8.19) at $M + \frac{1}{2}, j$ coordinate

$$\left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1}_{M+\frac{1}{2},j} = u^{n+1}_{M-\frac{1}{2},j}\frac{1}{\frac{\Delta x_M + \Delta x_{M+1}}{2}\Delta x_M} + u^{n+1}_{M+\frac{1}{2},j}\frac{-2}{\Delta x_M \Delta x_{M+1}} + u^{n+1}_{M+\frac{3}{2},j}\frac{1}{\Delta x_{M+1}\frac{\Delta x_M + \Delta x_{M+1}}{2}}, \tag{9.16}$$

denoting $h_w = x_M, h_e = x_{M+1}, h_c = \frac{\Delta x_M + \Delta x_{M+1}}{2}$ gives

$$\left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1}_{M+\frac{1}{2},j} = u^{n+1}_{M-\frac{1}{2},j}\frac{1}{h_w h_c} + u^{n+1}_{M+\frac{1}{2},j}\frac{-2}{h_e h_w} + u^{n+1}_{M+\frac{3}{2},j}\frac{1}{h_e h_c}. \tag{9.17}$$

We can substitute value of $u^{n+1}_{M+\frac{3}{2},j}$ from Eq. (9.15) into the above expression

$$\left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1}_{M+\frac{1}{2},j} = u^{n+1}_{M-\frac{1}{2},j}\frac{1}{h_w h_c} + u^{n+1}_{M+\frac{1}{2},j}\frac{-2}{h_e h_w} + \left(-\frac{v^{n}_{M+1,j+\frac{1}{2}} - v^{n}_{M+1,j-\frac{1}{2}}}{\Delta y_j}\Delta h_e + u^{n+1}_{M+\frac{1}{2},j}\right)\frac{1}{h_e h_c} \tag{9.18}$$

$$= u^{n+1}_{M-\frac{1}{2},j}\frac{1}{h_c h_w} - \frac{\frac{1}{\Delta y}\left(v^{n}_{M+1,j+\frac{1}{2}} - v^{n}_{M+1,j-\frac{1}{2}}\right)}{h_c} + u^{n+1}_{M+\frac{1}{2},j}\frac{h_w - 2h_c}{h_c h_e h_w}. \tag{9.19}$$

Simplification using $h_e = h_w - 2h_c$ leads to

$$\left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1}_{M+\frac{1}{2},j} = u^{n+1}_{M-\frac{1}{2},j}\frac{1}{h_c h_w} - \frac{\frac{1}{\Delta y}\left(v^{n}_{M+1,j+\frac{1}{2}} - v^{n}_{M+1,j-\frac{1}{2}}\right)}{h_c} + u^{n+1}_{M+\frac{1}{2},j}\frac{1}{h_c h_w}. \tag{9.20}$$

Eq. (9.20) is independent on h_e, therefore, we can set $h_e \to 0$, such process will also shift v components locations from $v_{M+1,j+\frac{1}{2}}, v_{M+1,j-\frac{1}{2}}$ to $v_{M+\frac{1}{2},j+\frac{1}{2}}, v_{M+\frac{1}{2},j-\frac{1}{2}}$. Finally, we can express $\hat{L}^u_{xx}$ together with the corresponding boundary contribution $\hat{L}^u_{\hat{bc}_1}$.

$$\hat{L}^u_{xx}\Big|_{M+\frac{1}{2},j} + \hat{L}^u_{\hat{bc}_1}\Big|_{M+\frac{1}{2},j} \iff \left[u^{n+1}_{M-\frac{1}{2},j}\frac{1}{h_c h_w} + u^{n+1}_{M+\frac{1}{2},j}\frac{1}{h_c h_w}\right] - \left[\frac{\frac{1}{\Delta y}\left(v^n_{M+1,j+\frac{1}{2}} - v^n_{M+1,j-\frac{1}{2}}\right)}{h_c}\right]. \tag{9.21}$$

Let us rewrite Eq. (8.19) for v component at $M, j-\frac{1}{2}$ as

$$\frac{\partial^2 v}{\partial x^2}\Big|^{n+1}_{M,j-\frac{1}{2}} = \frac{v^{n+1}_{M-1,j-\frac{1}{2}}\left(\frac{\Delta x_M+\Delta x_{M+1}}{2}\right) + v^{n+1}_{M,j-\frac{1}{2}}\left(-\frac{\Delta x_M+\Delta x_{M+1}}{2} - \frac{\Delta x_{M-1}+\Delta x_M}{2}\right) + v^{n+1}_{M+1,j-\frac{1}{2}}\left(\frac{\Delta x_{M-1}+\Delta x_M}{2}\right)}{\frac{\Delta x_M+\Delta x_{M+1}}{2}\,\frac{\frac{\Delta x_M+\Delta x_{M+1}}{2}+\frac{\Delta x_{M-1}+\Delta x_M}{2}}{2}\,\frac{\Delta x_{M-1}+\Delta x_M}{2}}. \tag{9.22}$$

Since the second line in BC (9.13) reads as $\frac{\partial v}{\partial x}^{n+1} = 0$ we use second order central difference at the boundary index $M+\frac{1}{2}, j-\frac{1}{2}$

$$\frac{\partial v}{\partial x}\Big|^{n+1}_{M+\frac{1}{2},j-\frac{1}{2}} = \frac{v^{n+1}_{M+1,j-\frac{1}{2}} - v^{n+1}_{M,j-\frac{1}{2}}}{\frac{\Delta x_M+\Delta x_{M+1}}{2}} \tag{9.23}$$

to express

$$v^{n+1}_{M+1,j-\frac{1}{2}} = v^{n+1}_{M,j-\frac{1}{2}}. \tag{9.24}$$

Plugging Eq. (9.24) into Eq. (9.22) we obtain expression

$$\hat{L}^v_{xx}\Big|_{M+\frac{1}{2},j-\frac{1}{2}} + \hat{L}^v_{\hat{bc}_1}\Big|_{M+\frac{1}{2},j-\frac{1}{2}} \iff \frac{v^{n+1}_{M-1,j-\frac{1}{2}}\left(\frac{\Delta x_M+\Delta x_{M+1}}{2}\right) + v^{n+1}_{M,j-\frac{1}{2}}\left(-\frac{\Delta x_M+\Delta x_{M+1}}{2}\right)}{\frac{\Delta x_M+\Delta x_{M+1}}{2}\,\frac{\frac{\Delta x_M+\Delta x_{M+1}}{2}+\frac{\Delta x_{M-1}+\Delta x_M}{2}}{2}\,\frac{\Delta x_{M-1}+\Delta x_M}{2}}. \tag{9.25}$$

The above Eq. (9.25) in terms of $h_e = \frac{\Delta x_M+\Delta x_{M+1}}{2}, h_w = \frac{\Delta x_{M-1}+\Delta x_M}{2}, h_c = \frac{\frac{\Delta x_M+\Delta x_{M+1}}{2}+\frac{\Delta x_{M-1}+\Delta x_M}{2}}{2}$ is written as

$$\hat{L}^v_{xx}\Big|_{M+\frac{1}{2},j-\frac{1}{2}} + \hat{L}^v_{\hat{bc}_1}\Big|_{M+\frac{1}{2},j-\frac{1}{2}} \iff \left[\frac{v^{n+1}_{M-1,j-\frac{1}{2}}(h_e) + v^{n+1}_{M,j-\frac{1}{2}}(-h_e)}{h_w h_c h_e}\right] + [0]. \tag{9.26}$$

Top boundary.
Similarly to BC (9.13), we get

$$\begin{pmatrix}\frac{\partial u}{\partial x}\\ \frac{\partial v}{\partial x}\end{pmatrix}^{n+1} = \begin{pmatrix}0\\ -\frac{\partial u}{\partial x}\end{pmatrix}^n \tag{9.27}$$

at top, where x changes to y, u changes to v and the rest is verbatim to discretization of corresponding operator at the right boundary that we discussed earlier.

10 Normalization

The most effective methods for solving systems of linear equations often work optimally with symmetric matrices, making symmetry a desirable property. In order to achieve a symmetric system, we introduce two matrices - the diagonal scaling matrix $\hat{M}$ and the diagonal flux matrix R. These matrices are defined as

$$R \equiv \begin{bmatrix} \Delta y_j & 0 \\ 0 & \Delta x_i \end{bmatrix},$$
$$\hat{M} \equiv \begin{bmatrix} \frac{1}{2}(\Delta x_i + \Delta x_{i-1}) & 0 \\ 0 & \frac{1}{2}(\Delta y_j + \Delta y_{j-1}) \end{bmatrix}.$$

Define matrices

$$\Delta y_j = \mathrm{diag}([\underbrace{\Delta y_1; \Delta y_1; \dots; \Delta y_1}_{M-1 \text{ times}}; \underbrace{\Delta y_2; \Delta y_2; \dots; \Delta y_2}_{M-1 \text{ times}}; \dots; \underbrace{\Delta y_N; \Delta y_N; \dots; \Delta y_N}_{M-1 \text{ times}}]),$$
$$\Delta x_i = \mathrm{diag}([\underbrace{\Delta x_1; \Delta x_2; \dots \Delta x_M; \Delta x_1; \Delta x_2; \dots \Delta x_M; \dots; \Delta x_1; \Delta x_2; \dots; \Delta x_M}_{N-1 \text{ blocks of } 1,\dots,M}]),$$

which are then combined into

$$R = \begin{bmatrix} \Delta y_j & 0 \\ 0 & \Delta x_i \end{bmatrix}.$$

By moving to new variable $q = R\boldsymbol{v}$, we use matrix R, which inverse R^{-1} performs half of the symmetrization process for $\hat{L}$. Here, R is a diagonal matrix of size $(M-1)N \times M(N-1)$. Using the R matrix we can express mass flux $q^{n+1} = R\boldsymbol{v}^{n+1} \implies \boldsymbol{v}^{n+1} = R^{-1}q^{n+1}$. It is required that the difference matrices cancel out $\frac{x_{i+1}-x_{i-1}}{2}$ central term in diffusion discretization to make Laplacian partially symmetric. Define the elements of diagonal mass matrix as

$$\Delta x_i + \Delta x_{i-1} = \mathrm{diag}[\underbrace{\Delta x_2 + \Delta x_1; \Delta x_3 + \Delta x_2; \dots; \Delta x_M + \Delta x_{M-1}; \dots; \Delta x_2 + \Delta x_1; \Delta x_3 + \Delta x_2; \dots \Delta x_M + \Delta x_M}_{N \text{ times for each block of } \Delta x_2+\Delta x_1 \text{ to } \Delta x_M+\Delta x_{M-1}}]$$

and

$$\Delta y_j + \Delta y_{j-1} = \mathrm{diag}[\underbrace{\Delta y_2 + \Delta y_1; \dots \Delta y_2 + \Delta y_1}_{M \text{ times}}; \underbrace{\Delta y_3 + \Delta y_2; \dots \Delta y_3 + \Delta y_2}_{M \text{ times}}; \underbrace{\Delta y_N + \Delta y_{N-1} \dots \Delta y_N + \Delta y_{N-1}}_{M \text{ times}}],$$

which are then combined into

$$\hat{M} = \begin{bmatrix} \frac{1}{2}(\Delta x_i + \Delta x_{i-1}) & 0 \\ 0 & \frac{1}{2}(\Delta y_j + \Delta y_{j-1}) \end{bmatrix}.$$

This matrix both removes the M^{-1} in $\hat{G} = \hat{M}^{-1}G$ if left multiplied and completes the symmetrization process of $\hat{L}$.

Identity $q = R\boldsymbol{v}$ implies $\boldsymbol{v} = R^{-1}q$, therefore, Laplacian matrix multiplied by the velocity vector becomes

$$\hat{L}\boldsymbol{v} = \hat{L}R^{-1}q.$$

Premultiplying by $\hat{M}$ we obtain

$$\hat{M}\hat{L}R^{-1}q,$$

where $\hat{M}LR^{-1}$ is symmetric by construction, hence, $\hat{M}\hat{A}R^{-1} = \hat{M}(\frac{1}{\Delta t}\mathbf{I} - \frac{1}{2}L)R^{-1}$ is also symmetric.

Using the above transformations we can modify Sys. (8.6) into

$$\begin{bmatrix} \hat{M}\hat{A}R^{-1} & \hat{M}\hat{G} \\ \hat{D}R^{-1} & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{M}\hat{r}^n \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{M}\widehat{bc}_1 \\ \widehat{bc}_2 \end{pmatrix},$$

which can be rewritten as

$$\boxed{\begin{bmatrix} A & G \\ D & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \end{pmatrix},} \tag{10.1}$$

where $A = \hat{M}\hat{A}R^{-1} = \frac{1}{\Delta t}\hat{M}R^{-1} - \frac{1}{2}\hat{M}\hat{L}R^{-1}$ is symmetric and $\hat{G} = \hat{M}^{-1}G$ according to Sys. (8.15). It is also possible to transform the divergence operator $\hat{D}$ with non-integer coefficients into D with integer coefficients in continuity part of Sys. (10.1) using Sys. (8.13). Multiplying both sides of continuity equation by Δ_{xy} matrix from Eq. (8.12) leads to

$$
\begin{aligned}
\hat{D}\boldsymbol{v}^{n+1} &= \hat{bc}_2, \\
(\Delta_{xy})\,\hat{D}\boldsymbol{v}^{n+1} &= (\Delta_{xy})\,\hat{bc}_2, \\
(\Delta_{xy})\,\frac{1}{\Delta_{xy}}DR\boldsymbol{v}^{n+1} &= (\Delta_{xy})\,\hat{bc}_2, \\
(\Delta_{xy})\,\frac{1}{\Delta_{xy}}Dq^{n+1} &= (\Delta_{xy})\,\hat{bc}_2, \\
Dq^{n+1} &= bc_2.
\end{aligned}
$$

11 Resulting algorithm

Let us consider the solution to discretized Sys. (10.1)

$$q^{n+1} = q_p^{n+1} + q_h^{n+1},$$

where q_h^{n+1} is a solution of homogeneous continuity equation

$$Dq_h^{n+1} = 0, \tag{11.1}$$

and q_p^{n+1} is a particular solution to its non-homogeneous counterpart

$$Dq_p^{n+1} = bc_2. \tag{11.2}$$

Since we eliminated any explicit boundary condition treatment of the incompressibility constraint and there is no change in density, we obtain $bc_2 = 0$. Hence, it is not required to solve inhomogeneous Eq. (11.2).

We assume that momentum equation from Sys. (10.1) is modified to streamfunction Eq. (3.9), where biharmonic operator satisfies homogeneous Eq. (11.1) as in Colonius-Taira [3].

Taking into account continuous operator identities in Section 3, the resulting algorithm for discrete Navier-Stokes Sys. (10.1) can be described as follows:

1. Construct curl matrix C, such that $DC = 0$ and $q_h = C\psi$.
2. Rewrite $q^{n+1} = q_h^{n+1}$.
3. Precompute bc_1^n and r^n.
4. Eliminate pressure terms in the momentum equation.

$$\begin{aligned}
Aq^{n+1} &= -Gp^{n+1} + r^n + bc_1^n, && \\
A(q_h^{n+1}) &= D^T p^{n+1} + r^n + bc_1^n, && \text{premultiply by } C^T, \\
C^T A q_h^{n+1} &= C^T(r^n + bc_1^n), && \text{use } q_h^{n+1} = C\psi^{n+1}, \\
C^T AC\psi^{n+1} &= C^T(r^n + bc_1^n). &&
\end{aligned} \tag{11.3}$$

5. Solve the resulting Sys. (11.3) for ψ^{n+1}. This Sys. (11.3) is equivalent to biharmonic Eq. (3.9).
6. Obtain $q_h^{n+1} = C\psi^{n+1}$.
7. Compute $q^{n+1} = q_h^{n+1}$ and convert it to velocity vector field $\boldsymbol{v}^{n+1} = R^{-1}q^{n+1}$.
8. Transition to next time instance by repeating Steps (3) to (7).

12 Summary

References

[1] MATH 248: Honours Vector Calculus Fall 2019. https://www.math.mcgill.ca/gantumur/math248f19/.

[2] Wang Chang, Francis Giraldo, and Blair Perot. Analysis of an exact fractional step method. *Journal of Computational Physics*, 180(1):183–199, 2002.

[3] Tim Colonius and Kunihiko Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197:2131–2146, April 2008.

[4] S. Dong, G. E. Karniadakis, and C. Chryssostomidis. A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *Journal of Computational Physics*, 261:83–105, March 2014.

[5] Joel H. Ferziger and Milovan Perić. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[6] C. Hall, T. Porsching, R. Dougall, R. Amit, A. Cha, C. Cullen, George Mesina, and Samir Moujaes. Numerical methods for thermally expandable two-phase flow-computational techniques for steam generator modeling. *NASA STI/Recon Technical Report N*, 1980.

[7] F. Harlow and E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.

[8] G. Jin and M. Braza. A Nonreflecting Outlet Boundary Condition for Incompressible Unsteady Navier-Stokes Calculations. *Journal of Computational Physics*, 107(2):239–253, August 1993.

[9] A. Kourta, M. Braza, P. Chassaing, and H. Haminh. Numerical analysis of a natural and excited two-dimensional mixing layer. *AIAA Journal*, 25(2):279–286, February 1987.

[10] C. Liu and Z. Liu. High Order Finite Difference and Multigrid Methods for Spatially Evolving Instability in a Planar Channel. *Journal of Computational Physics*, 106(1):92–100, May 1993.

[11] Hélène Persillon and Marianna Braza. Physical analysis of the transition to turbulence in the wake of a circular cylinder by three-dimensional Navier–Stokes simulation. *Journal of Fluid Mechanics*, 365:23–88, June 1998.

[12] R. L. Sani and P. M. Gresho. Résumé and remarks on the open boundary condition minisymposium. *International Journal for Numerical Methods in Fluids*, 18(10):983–1008, 1994.

[13] Semyon V. Tsynkov. Numerical solution of problems on unbounded domains. A review. *Applied Numerical Mathematics*, 27(4):465–532, August 1998.

[14] O. Zikanov. *Essential Computational Fluid Dynamics*. Wiley, Hoboken, N.J., 2010.

A Treating boundary conditions implicitly and maintaining the symmetric property of system

Consider the following discrete Poisson equation

$$\left[\begin{array}{ccc|c|cccc} -4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -4 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -4 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -4 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & -4 \end{array}\right] \left[\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{array}\right] = \left[\begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \\ r_7 \\ r_8 \end{array}\right]. \tag{A.1}$$

Row 4 is treating boundary condition of the corresponding unknown u_4 implicitly. Let us multiply the whole row by a very large number 10^{10}.

$$\left[\begin{array}{ccc|c|cccc} -4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -4 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 10^{10} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -4 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -4 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & -4 \end{array}\right] \cdot \left[\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{array}\right] = \left[\begin{array}{c} r_1 \\ r_2 \\ r_3 \\ 10^{10} r_4 \\ r_5 \\ r_6 \\ r_7 \\ r_8 \end{array}\right] \tag{A.2}$$

Let us now add or subtract any relatively small constants from different columns of this row. In order to make the matrix symmetric we can modify it to

$$\left[\begin{array}{ccc|c|cccc} -4 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -4 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 10^{10} & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & -4 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -4 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -4 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & -4 \end{array}\right] \left[\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{array}\right] = \left[\begin{array}{c} r_1 \\ r_2 \\ r_3 \\ 10^{10} r_4 \\ r_5 \\ r_6 \\ r_7 \\ r_8 \end{array}\right]. \tag{A.3}$$

Row 4 now becomes $u_2 + u_3 + 10^{10} u_4 - u_8 \approx 10^{10} u_4$, since relatively small values of u_2, u_3, u_8 can be treated as round-off errors.