

Finite-difference and -volume methods for the incompressible Navier-Stokes equations

Ruanan

November 11, 2023

Abstract

The goal of these notes is to review the finite-difference schemes for solving incompressible Navier-Stokes equations, with the emphasis on those which are well-established and tested by time. We will discuss in detail the scheme formulation, its advantages and shortcomings including stability and accuracy aspects. Special attention will be paid to the boundary conditions and their implementation. Of particular interest is to highlight the motivation behind the scheme and to trace the historic development of finite-difference methods as applied to incompressible flows through the contributions of the key papers in this field. After studying these notes one must get a coherent picture of Finite-difference and -volume approaches to incompressible flows as well as to see a clear logical connection among methods.

Contents

1	Introduction	2
1.1	Oscillations on collocated grids	3
1.2	Boundary conditions for pressure	5
2	Marker-and-cell scheme	6
3	Chorin's Projection Scheme	6
4	Pressure-correction methods: general view	8
5	Spatial Discretization	8
5.1	Magnetization formulation, not needed?	9
6	Modern schemes for solving Navier-Stokes equations	9
6.1	SIMPLE - Semi-Implicit Method for Pressure Linked Equations	9
6.2	PISO - Pressure Implicit with Splitting of Operators	10
7	Discrete streamfunction method	11
7.1	Problem statement	11
7.2	Domain discretization	11
7.3	Discrete operators	12
7.3.1	Transient terms	12
7.3.2	Laplacian	12
7.3.3	Divergence	15
7.3.4	Gradient	15
7.3.5	Advection	16
7.3.6	Resulting system	18
7.4	Symmetrization	18
7.5	Nullspace method	20
7.6	Resulting algorithm	21
7.7	A particular solution to the discrete continuity equation	21
7.7.1	Particular solution using Lagrange multipliers	21

7.7.2	Particular solution using Graph Theory	22
8	Other interesting methods for solving Naver Stokes (may include algorithms from articles [2, 6, 13, 16])	22
9	Vorticity-Streamfunction formulation	22
A	Appendix	24
A.1	Transient schemes	24
A.2	Laplacian symmetrization in depth	24

1 Introduction

In these notes we will be concerned with the discretization of the Navier-Stokes equations describing the flow of an incompressible fluid:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nu \nabla \cdot \nabla \mathbf{v} \quad (1a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (1b)$$

which are written here in the non-dimensional form, i.e. $\nu \equiv Re^{-1}$ for brevity; also \mathbf{v} is the velocity and p pressure fields. Since both are necessarily functions of time t and space \mathbf{x} , their discretization will be denoted by f_i^n , where n is the time level t^n and i is the *mesh* point \mathbf{x}_i ; for example, in the x -direction the interval $x_i < x < x_{i+1}$ is referred to as the *cell* of the mesh. The cell center corresponds to $x_{i+\frac{1}{2}}$. The mesh points x_i and cell centers $x_{i+\frac{1}{2}}$ can be regarded as two overlapping interpenetrating meshes, which together are said to constitute a *staggered* mesh as illustrated in figure 2(a). 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 obvious choice seems to have a single set of points, at which all the variables and all the equations are discretized. Such a grid has the name of a *collocated* or *regular* grid. Albeit simple and easy in operation, the 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 [21]) and hence staggered grids are used since the work of Harlow and Welch [10] introducing the marker-and-cell (MAC) method, cf. figure 2(a).

The staggered arrangement increases complexity of a scheme. Programming becomes more difficult, since it requires accounting for three (or four in the three-dimensional case) indexing systems. Interpolations must be used to compute nonlinear terms of momentum equations. Further complications arise when the grid is nonuniform. All these difficulties, however, can be relatively easily handled in computations with structured grids such as those shown in figure 2(b). For this reason and because of the benefit of removing the splitting problem, the staggered arrangement was by far the most popular choice during early years of CFD. The difficulties of handling a staggered arrangement increase significantly when unstructured grids are used. When such grids started to be broadly applied in general-purpose codes in recent years, collocated arrangements returned to favor. This area of CFD is still evolving. We only mention that methods have been developed to cure the splitting problem leading to pressure oscillations, but the cure is not ideal and leads to extra complexities at the implementation level.

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

A conspicuous feature of (1) is that pressure p is not determined by a time-evolution equation, but rather is implicitly defined by the incompressibility condition (1b), which plays the role of a constraint that the velocity field \mathbf{v} must satisfy. The pressure instantaneously adapts to the evolving velocity field in such a way as to satisfy that constraint. This is reflected in the fact that p satisfies a Poisson equation, which can be derived by taking the divergence of (35a) and combining the result with (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 are of the mixed hyperbolic (convective terms), parabolic (viscous terms), and elliptic (pressure and incompressibility) type. The elliptic nature of

the pressure solution has a physical meaning. It shows that, 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 equation (1b) and its solution can in principle be substituted back into equation (35a) to obtain an evolution equation for the velocity field alone. Alternatively, the pressure can be immediately eliminated by taking the curl of momentum equation (35a), which leads to the vorticity-stream function formulation of incompressible flow. Formal manipulations of this type are useful for various theoretical purposes, but experience has shown that they are rarely advantageous for computational purposes. In most situations, it is preferable to simply approximate and solve the system (1) directly for the so-called *primitive* variables \mathbf{v} and p .

If, however, one ends up solving the Poisson equation for pressure, an interesting and important question arises as to 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 cases, an unnecessary complication, we have to find a way to derive the pressure boundary conditions from the equations themselves.

- ~~Discuss in detail the origin of spurious oscillations on collocated grids~~
- ~~Discuss the boundary conditions for pressure, cf. §10.3.1 in [21], and how to avoid dealing with the latter.~~
- Consultation appointment to discuss this paper's mistakes and grammar. 15Nov 2PM.

ToDo

1.1 Oscillations on collocated grids

To give a simple and explanatory example to see the reasons for oscillatory behaviour consider a 1-Dimensional ($\mathbf{v} = u$) continuity equation (1b) solved on both Staggered and Collocated Grids (figures 1(a) and 1(b) respectively) with velocities $u_{Left} = 1$ and $u_{Right} = 2$ at the boundaries in Finite Volume formulation.

First, consider the staggered arrangement of the cells. Continuity equation (1b) for cell "O" in figure 1(a) after integrating over the volume and application of Gauss-Ostrogradsky theorem becomes

$$\int_O (\nabla \cdot \mathbf{u}) dV \equiv (u_e^O - u_w^O) \Delta y = 0, \quad (2)$$

where u_e^O, u_w^O represent values of the velocity u at the faces of the "O" cell, solving the equation (2) leads to $u_e^O = u_w^O$. By repeating the process for all control volumes the solution for all cells results in $u = 1$ for all cell faces.

Moving to the collocated grid arrangement in figure 1(b) and solving continuity equation (2) requires interpolation for values of u at cell faces. By applying linear interpolation to discretized continuity equation (2) and assuming uniform grid:

$$u_e^O = \frac{u_E + u_O}{2}, \quad u_w^O = \frac{u_W + u_O}{2}. \quad (3)$$

This leads to the identity $u_E = u_W$, which can be satisfied by infinitely many solutions for velocity u at cell centres, couple of examples could be:

$$u = [1, 0, 1, 0, 1, 0, \dots],$$

$$u = [0, 1, 0, 1, 0, 1, \dots],$$

$$u = [1, 1, 1, 1, 1, 1, \dots] \text{ (the correct solution),}$$

$$u = [1, 0, 3, 1, 0, 3, 1, 0, 3, \dots],$$

and infinitely many more.

To display the infinite solution behaviour from above rigorously, we shall construct a system of linear equations based on a collocated mesh. Consider cell centres at grid coordinates x_{i-1}, x_i, x_{i+1} . Assume that the values of u at the cell centre depend only on neighbouring cells W, E (can be extended to further neighbours such as EE, WW etc. if needed) and cell O . Denoting these x coordinates as W, O, E respectively, the continuity equation (1b) can be represented for i^{th} cell that is not at the boundary

$$A_W^i u_W + A_O^i u_O + A_E^i u_E = 0, \quad (4)$$

where A_W, A_O, A_E represent the coefficients in front of the velocity terms for each cell.

By taking equations ((2),(3)) the coefficients in equation (4) can be obtained, resulting in:

$$A_W^i = -\frac{1}{2}, A_O^i = 0, A_E^i = \frac{1}{2}, \quad \text{where } i = 2, 3, \dots, n-1.$$

The equation (4) changes for the cells next to boundaries:

$$\begin{aligned} A_w^1 u_w + A_O^1 u_O + A_E^1 u_E &= 0, \\ A_W^n u_W + A_O^n u_O + A_e^n u_e &= 0, \end{aligned}$$

with coefficients:

$$\begin{aligned} A_w^1 &= -1, A_O^1 = \frac{1}{2}, A_E^1 = \frac{1}{2}, \\ A_W^n &= -\frac{1}{2}, A_O^n = -\frac{1}{2}, A_e^n = 1. \end{aligned}$$

The values at the boundaries u_w^1 and u_e^n are known, and thus can be moved to the right-hand side of the corresponding equations, which leads to the system of linear equations

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \dots & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \dots & 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} (1)(u_w^1) \\ 0 \\ 0 \\ \vdots \\ 0 \\ (-1)(u_e^n) \end{bmatrix}. \quad (5)$$

The system (5) can be satisfied by many different combinations since it is singular. The correct solution is one of the infinitely many possible ones, therefore the numerical scheme keeps flip-flopping, such phenomena are also known as spurious oscillations or checkerboard oscillations.

The checkerboard oscillations were overcome by the introduction of Staggered Grids [10] in 1965 which were used until Rhie & Chow's (1983) [17] proposal of a special interpolation technique on collocated grids.

The oscillatory behaviour is usually observed for pressure variable in system (1) [21], to display this we take x -component of momentum equation (35a) and by denoting \mathbf{D} , all but pressure terms we integrate over the finite volume of cell O ,

$$\mathbf{D}u = -\frac{\partial p}{\partial x} \implies \sum_{faces} (F^d + F^c) \cdot \mathbf{n} dA = -\frac{\partial p}{\partial x} \Big|_O V_O, \quad (6)$$

where F^d, F^c are diffusive and convective fluxes of u , \mathbf{n} is an outward normal vector to the surface dA enclosing the finite volume of cell O .

After application of any interpolation scheme (central, upwinding etc.) to fluxes on the left side of discretized momentum (6), one is able to express the value of u at the cell centre as a linear function of its neighbours $\lambda_N^O u_O = f(u_N^O)$, where λ_N^O is an interpolating coefficient. Expressing u_O in the discretized momentum equation (6) leads to

$$u_O = \frac{1}{\lambda_N^O} \left(f(u_N^O) - \frac{\partial p}{\partial x} \Big|_O V_O \right). \quad (7)$$



Figure 1: Staggered vs Collocated grids.

By substituting the above result (7) into the discretized continuity (2), we obtain

$$\frac{1}{\lambda_E^O} \left(f(u_N^E) - \frac{\partial p}{\partial x} \Big|_E V_E \right) - \frac{1}{\lambda_N^W} \left(f(u_N^W) - \frac{\partial p}{\partial x} \Big|_W V_W \right) = 0. \quad (8)$$

The discretized pressure term $\frac{\partial p}{\partial x} \Big|_O = \frac{p_E - p_W}{\Delta x} = \frac{p_E - p_W}{2\Delta x}$ substituted into the semi-discretized finite volume continuity equation (8), leads to

$$\frac{1}{\lambda_E^O} \left(f(u_N^E) - \frac{p_{EE} - p_O}{2\Delta x} V_E \right) - \frac{1}{\lambda_W^O} \left(f(u_N^W) - \frac{p_O - p_{WW}}{2\Delta x} V_W \right) = 0, \quad (9)$$

where pressure with lower subscripts p_{EE}, p_O, p_{WW} represent the values of pressure at the cell centres with coordinates $i+2, i, i-2$.

From the discretized equation (9) above it is evident that the value of p_O does not depend on the values of its neighbours p_E, p_W , but only on its further neighbours p_{EE}, p_{WW} . Therefore two solutions for pressure (one for even and another for odd cells) may satisfy initial system (1).

1.2 Boundary conditions for pressure

An interesting and important question arises as to 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

Applying $(\nabla \cdot)$ to the momentum equation (35a), first taking into account the commutativity of time derivative and Laplace operator ∇^2 with $(\nabla \cdot)$, then eliminating these terms by applying continuity equation (1b), finally leads to Poisson equation for pressure

$$\nabla \cdot (\nabla p) \equiv \nabla^2 p = \rho \nabla \cdot [-\mathbf{v} \cdot \nabla \mathbf{v}]. \quad (10)$$

The conditions, however, do not naturally follow from the flow physics for the boundaries between fluid and solid walls. We have to find a way to derive the pressure boundary conditions from the equations themselves.

Recalling calculus: the gradient of any function f is defined as the unique vector field whose dot product with any vector v at each point x is the directional derivative of f along v , i.e. $\nabla f \cdot v = \frac{\partial f}{\partial v}$.

Consider \mathbf{n} to be the normal vector to the solid wall boundary. We apply $(\cdot \mathbf{n})$ to the momentum equation (35a) and get

$$\frac{\partial p}{\partial \mathbf{n}} \Big|_{\partial \Omega} = \rho \left[\nu \nabla \cdot \nabla \mathbf{v} - \frac{\partial \mathbf{v}}{\partial t} - \mathbf{v} \cdot \nabla \mathbf{v} \right] \cdot \mathbf{n} \Big|_{\Omega}. \quad (11)$$

To compute the pressure values at the boundaries we need to take into account the impermeability condition. The velocity component perpendicular to the wall is zero, meaning that the flow can not penetrate the wall, mathematically written as

$$\mathbf{v} \cdot \mathbf{n} \Big|_{\partial \Omega} = 0. \quad (12)$$

By taking into account impermeability condition (12) equation (11) leads to boundary condition for the pressure at the wall computed as

$$\frac{\partial p}{\partial \mathbf{n}} \Big|_{\partial \Omega} = \rho \left[\nu \nabla \cdot \nabla \mathbf{v} - \frac{\partial \mathbf{v}}{\partial t} - \mathbf{v} \cdot \nabla \mathbf{v} \right] \cdot \mathbf{n} \Big|_{\Omega} \Rightarrow \frac{\partial p}{\partial \mathbf{n}} \Big|_{\partial \Omega} = \rho [\nu \nabla \cdot \nabla \mathbf{v}] \cdot \mathbf{n} \Big|_{\partial \Omega}. \quad (13)$$

The diffusive term on the right-hand side is nonzero even after the application of no-slip conditions $\mathbf{v} \cdot \mathbf{t} \Big|_{\partial\Omega} = 0$ at the wall, where \mathbf{t} is a tangential vector to the wall, i.e. diffusion at the wall is directed in the normal direction from the boundary.

Boundary conditions for pressure (11) and (13) define a solution of the Poisson equation for pressure (10) up to a constant, i.e. we need to fix value for p at any point of our domain. Another method to obtain boundary conditions for pressure is to apply $(\mathbf{t} \cdot \cdot)$ to the momentum equation (35a) which will lead to Dirichlet boundary condition, where the value of pressure at a single point x_0 is also needed to uniquely determine the boundary values. Gresho [7] describes the equivalence of the derived Dirichlet and Neumann conditions for the Pressure Poisson equation.

The results obtained for pressure boundary conditions require special treatment and recomputation for every change of velocity near the boundaries, which increases the complexity of solvers and generates more errors. Section 9 displays an approach that completely eliminates pressure from the problem statement, thus removing the requirement of its boundary conditions.

2 Marker-and-cell scheme

Since system (1) does not involve $\partial p / \partial t$, the pressure cannot be advanced in time by means of its time derivative. In fact, the only place p appears in the equations at all is in the pressure gradient term in the momentum equation (35a). If this term were differenced in a purely explicit manner, the scheme would not involve p^{n+1} in any way, so there would be no way to compute it. In order to advance the pressure in time, the temporal difference approximation to ∇p in (35a) must be at least partially implicit so that p^{n+1} appears in the difference scheme. This in turn implies that the temporal difference approximation to $\nabla \cdot \mathbf{v}$ in (1b) must likewise be at least partially implicit since the difference approximation to (35a) alone does not provide enough equations to determine both \mathbf{v}^{n+1} and p^{n+1} . But once $\nabla \cdot \mathbf{v}$ is partially implicit, it might as well be fully implicit, which entails no additional labour and has the significant advantage of satisfying (1b) exactly at each time level of the calculation. We, therefore, difference (35a) in a fully implicit manner

$$(\nabla \cdot \mathbf{v})^{n+1} = 0. \quad (14)$$

The momentum equation can be differenced in a fully explicit manner except for the pressure gradient

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} + (\mathbf{v} \cdot \nabla \mathbf{v})^n = -\nabla p^* + \nu (\nabla \cdot \nabla \mathbf{v})^n, \quad (15)$$

where $p^* = \gamma p^{n+1} + (1 - \gamma) p^n$ with the weighting factor $0 < \gamma \leq 1$ controlling the degree of implicitness of the scheme. The system ((14),(15)) is now closed that allows uniquely determine \mathbf{v}^{n+1} and p^* regardless of the value of γ . Varying γ produces small changes of order of Δt in the time history of the pressure field, but has no effect on the velocity field. Since the precise value of γ is immaterial, we may well adopt the simplest choice $\gamma = 1$ so that $p^* = p^{n+1}$, i.e. approximating the pressure gradient in a fully implicit manner. This choice constitutes the MAC scheme [10]. Equations ((14),(15)) are ordinarily solved by iterative methods, of which the traditional choices were successive over-relaxation (SOR).

Include a section on Chorin's scheme [4] before discussing a general view in §4.

ToDo

3 Chorin's Projection Scheme

The proposed method by Chorin [4] was solved on a staggered grid as illustrated in figure 2(b) using Finite Differences. The algorithm can be summarized as follows: the time t is discretized, at every time step two separate equations are solved and combined based on the decomposition theorem of Ladyzhenskaya sometimes referred to as Helmholtz-Hodge Decomposition or simply as Hodge decomposition. The theorem states that the vector field \mathbf{u} defined on a simply connected domain can be uniquely decomposed into a divergence-free (solenoidal) part \mathbf{u}_{sol} and an irrotational part $\mathbf{u}_{\text{irrot}}$, leading to $\mathbf{u} = \mathbf{u}_{\text{sol}} + \mathbf{u}_{\text{irrot}}$. In the case of Navier-Stokes equations, advection-diffusion terms from initial

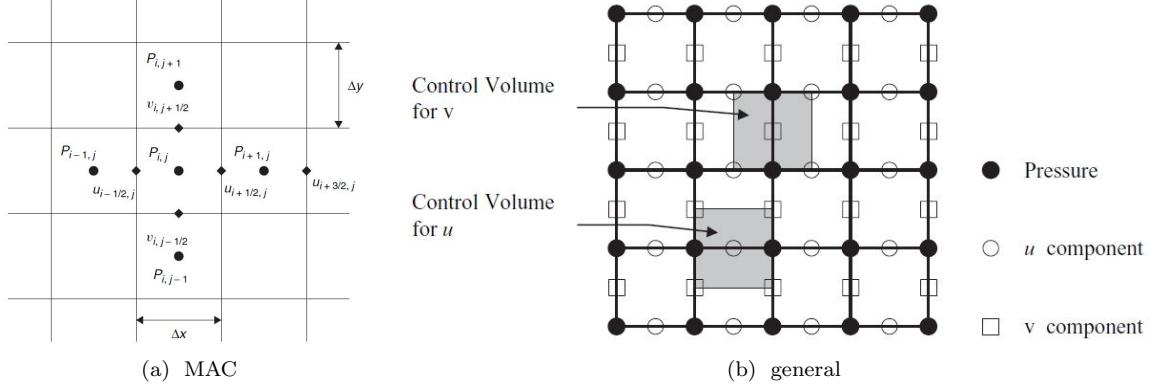


Figure 2: Staggered grid.

to intermediate time are considered as a divergence-free part and the pressure gradient at the time step from intermediate to last as an irrotational part. The method developed by Chorin is focused on solving two equations (Burgers' and Poisson), which are obtained after introducing intermediate velocity $\tilde{\mathbf{v}}$.

Chorin's scheme was computed in single time step as follows:

1. Solve for intermediate velocity field $\tilde{\mathbf{v}}$ equation (16), generally not satisfying incompressible condition, since pressure gradient does not appear in

$$\frac{\tilde{\mathbf{v}} - \mathbf{v}^n}{\Delta t} = -\mathbf{v}^n \cdot \nabla_d \mathbf{v}^n + \nu \Delta_d \mathbf{v}^n, \quad (16)$$

where superscript denotes known values of \mathbf{v} at n 'th time step and subscript d denotes discretized gradient and Laplacian operators. Chorin solved the above Burgers' equation (16) using the alternating direction implicit method.

2. This step is based on the application of Helmholtz-Decomposition to final velocity field \mathbf{v}^{n+1} ,

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}} - \nabla_d (\Delta t p^{n+1}). \quad (17)$$

By imposing the divergence operator to the above equation (17) and making use of the continuity (1b) at time step $n + 1$ a discretized version of the pressure-Poisson equation (10) is written as

$$\Delta_d p^{n+1} = \frac{1}{\Delta t} \nabla_d \tilde{\mathbf{v}}. \quad (18)$$

Pressure condition at the boundary as in section 1.2 is

$$\mathbf{n} \cdot \nabla_d p^{n+1} = \mathbf{n} \cdot \left(-\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} - (\mathbf{v} \cdot \nabla_d \mathbf{v})^n + \nu \Delta_d \mathbf{v}^n \right). \quad (19)$$

Equation (18) is solved with the corresponding boundary condition (19). Chorin used the Successive Point Over-Relaxation method, however, any appropriate iterative/direct/multigrid approaches might be used.

3. Use the solution for pressure from Step 2 and compute velocity at the next time step

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}} - \Delta t \nabla_d p^{n+1}. \quad (20)$$

4 Pressure-correction methods: general view

As discussed by Dukowicz & Dvinsky [6], both MAC [10] and Chorin's [4, 18, 19] methods can be put in a single form for time-discretization:

$$\text{Predictor : } \frac{\tilde{\mathbf{v}} - \mathbf{v}^n}{\Delta t} + \mathbf{v}^n \cdot \nabla \mathbf{v}^* = \nu \nabla \cdot \nabla \mathbf{v}^*, \quad (21a)$$

$$\text{Corrector : } \frac{\mathbf{v}^{n+1} - \tilde{\mathbf{v}}}{\Delta t} = -\nabla p^*, \quad (21b)$$

$$\nabla \cdot \mathbf{v}^{n+1} = 0,$$

where Δt is the time step and \mathbf{v}^* stands for \mathbf{v}^n in MAC and the intermediate velocity $\tilde{\mathbf{v}}$ in Chorin's method; the pressure time level is unspecified in MAC and is taken to be the new time level by Chorin. The general strategy in (21) is to decompose each time step into two substeps. On the first substep, the momentum equation is solved for the velocity components. The pressure gradient is either removed from the equation or approximated by an estimate. The obtained velocity field cannot be considered a solution at the new time level since it does not satisfy the incompressibility condition. The second substep is, therefore, needed, at which the correct pressure distribution is found and the correction of velocity is made.

If one adopts the *projection* point of view, as inspired by the existence theory of the NSEs, then the scheme (21) can be interpreted as a projection of the intermediate velocity field $\tilde{\mathbf{v}}$ onto a subspace of vector fields with zero divergence. Namely, the term projection reflects the fact that we find a preliminary solution, which is not divergence-free, and then project it onto the space of divergence-free vector functions. This projection point of view, pioneered by Chorin [4] is based on the observation that equation (35a) can be put in the form $\frac{\partial \mathbf{v}}{\partial t} + \nabla p = -\mathbf{v} \cdot \nabla \mathbf{v} + \nu \nabla \cdot \nabla \mathbf{v}$, the left-hand side of which is a Hodge decomposition, so that by acting with the operator \mathbf{P} , which projects a vector field onto the space of divergence-free vector fields with appropriate boundary conditions, we get $\frac{\partial \mathbf{v}}{\partial t} = \mathbf{P}[-\mathbf{v} \cdot \nabla \mathbf{v} + \nu \nabla \cdot \nabla \mathbf{v}]$, which is the continuous version of (21a). On the other hand, the scheme (21) can be interpreted from the *operator-splitting* or *fractional time-step* (aka *time-splitting*) point of view on, which originates as a purely numerical method which is general and not specific to a particular set of equations [20], since by adding the two sets of equations to eliminate the intermediate velocity $\tilde{\mathbf{v}}$. Regardless of which point of view to take in order to interpret (21), the conceptual and practical advantage of decoupling the "pressure correction" from the momentum equation calculations.

- Add a section on spatial discretization using §4.5 in [1] as well as research papers [6, 13, 16, 2], etc. Use the notations for discrete advection, Laplacian, divergence, and gradient operators as per equation (3) in Perot [16]. The articles listed describe time discretization techniques, and spatial operators are kept arbitrary.
- Also discuss other numerical schemes in detail such as the ~~semi-implicit method for pressure linked equations (SIMPLE)~~ [15, 14], ~~the pressure implicit with splitting of operators (PISO)~~ [11, 12], Colonius' (discrete streamfunction), etc.
- Add a section on velocity-vorticity methods (§10.5.1 [21], §4.6 [1]).
- Since the scheme (21) is first-order accurate in time, discuss its improvements in terms of accuracy [13, 6, 16, 2].
- When a reference is made to specific schemes such as the Crank-Nicholson, Adams-Bashforth, and so on, for self-contained discussion their description should be included in an appendix.

ToDo

5 Spatial Discretization

The initial system of equations (1) can be rewritten using discrete spatial operators as

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{H}(\mathbf{v}) = -Gp^{n+1} + \nu L(\mathbf{v}),$$

$$D\mathbf{v}^{n+1} = 0, \quad (22)$$

where \mathbf{H}, G, L, D are discrete convective, gradient, Laplacian and divergence operators respectively.

Bringen and Chow [1] performed discretization on uniform staggered MAC grid illustrated on figure 2(a) and used the discrete operators at time step n :

$$\begin{aligned}
\mathbf{H}(\phi_{i+1/2,j}^n, \psi_{i,j+1/2}^n) &= \left(\frac{(\phi_{i,j}^n)^2 - (\phi_{i+1,j}^n)^2}{\Delta x} + \frac{(\phi\psi)_{i+1/2,j-1/2}^n - (\phi\psi)_{i+1/2,j+1/2}^n}{\Delta y}, \right. \\
&\quad \left. \frac{(\phi\psi)_{i-1/2,j+1/2}^n - (\phi\psi)_{i+1/2,j+1/2}^n}{\Delta x} + \frac{(\psi_{i,j}^n)^2 - (\psi_{i,j+1}^n)^2}{\Delta y} \right), \\
(G_x(\phi_{i+1/2,j}^n), G_y(\phi_{i,j+1/2}^n)) &= \left(\frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{\Delta x}, \frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{\Delta y} \right), \\
L(\phi_{i+1/2,j}^n, \psi_{i,j+1/2}^n) &= \left(\frac{\phi_{i+3/2,j}^n - 2\phi_{i+1/2,j}^n + \phi_{i-1/2,j}^n}{\Delta x^2} + \frac{\phi_{i+1/2,j+1}^n - 2\phi_{i+1/2,j}^n + \phi_{i+1/2,j-1}^n}{\Delta y^2}, \right. \\
&\quad \left. \frac{\psi_{i+1,j+1/2}^n - 2\psi_{i,j+1/2}^n + \psi_{i-1,j+1/2}^n}{\Delta x^2} + \frac{\psi_{i,j+3/2}^n - 2\psi_{i,j+1/2}^n + \psi_{i,j-1/2}^n}{\Delta y^2} \right), \\
D(\phi_{i,j}^n, \psi_{i,j}^n) &= \left(\frac{\phi_{i+1/2,j}^n - \phi_{i-1/2,j}^n}{\Delta x} + \frac{\psi_{i,j+1/2}^n - \psi_{i,j-1/2}^n}{\Delta y} \right), \\
\end{aligned} \tag{23}$$

where ϕ, ψ are physical quantities. In the case of velocities they are evaluated at the cell surface leading to $1/2$ indices appearing, whereas for pressure the quantities are computed at the cell centre (marker). Convective operator \mathbf{H} was evaluated in a conservative form

$$\mathbf{H}(\mathbf{v}) = \mathbf{H}(u, v) = \left(\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y}, \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} \right). \tag{24}$$

5.1 Magnetization formulation, not needed?

An alternative formulation of the incompressible NavierStokes equations (1) based on a new variable called "magnetization", impulse, or gauge can be made [2]. Two new variables, \mathbf{m} and χ , are introduced that are related to the fluid velocity by

$$\mathbf{m} = \mathbf{v} + \nabla \chi.$$

The vector field \mathbf{m} and the potential χ can be chosen to satisfy evolution equations in such a way that the fluid velocity and pressure derived from them satisfy the Navier-Stokes equations. Given \mathbf{m} , one possibility is to let \mathbf{m} satisfy in Ω the evolution equation

$$\begin{aligned}
\mathbf{m}_t + (\mathbf{v} \cdot \nabla) \mathbf{v} &= \nu \nabla^2 \mathbf{m} \\
\mathbf{v}|_{\partial\Omega} &= \mathbf{v}_b,
\end{aligned}$$

where $\mathbf{v} = \mathbf{P}(\mathbf{m})$ is the operator which projects a vector field onto the space of divergence-free vector fields with appropriate boundary conditions.

6 Modern schemes for solving Navier-Stokes equations

This section is focused on several popular methods described in two-dimensional cases.

6.1 SIMPLE - Semi-Implicit Method for Pressure Linked Equations

As discussed in section 1.1 there is a possibility that a solution on collocated grids will generate spurious oscillations due to interpolation, the use of staggered grids can address this issue. One of the methods on staggered grids was developed by Patankar and Spalding [15]. They named it SIMPLE, algorithm was based on a predictor-corrector procedure with successive pressure correction

$$p = \bar{p} + p', \tag{25}$$

where p is the actual pressure, \bar{p} is the estimated pressure, and p' is the pressure correction. Similarly to the pressure correction (25), the velocity vector is decomposed into two terms

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}' = (\bar{u} + u', \bar{v} + v'). \quad (26)$$

Corrections of pressure are related to the velocity corrections by approximate momentum equation,

$$\frac{\partial \mathbf{v}'}{\partial t} = -Gp \implies \mathbf{v}' = -G(p\Delta t). \quad (27)$$

Combining velocity correction (26) with right hand side of expression (27) and substituting the result into the continuity equation (1b), we obtain the so-called pressure-correction Poisson equation

$$DGp' \equiv \nabla \cdot \nabla p' = -\frac{1}{\Delta t} (D\mathbf{v} - D\bar{\mathbf{v}}) = \frac{1}{\Delta t} D\bar{\mathbf{v}}, \quad (28)$$

where we set $\nabla \cdot \mathbf{v} = 0$ to enforce the mass conservation at the current time step. An iterative procedure used to obtain the solution is described below.

1. Guess the pressure \bar{p} distribution at each cell centre.
2. Solve the momentum equation in control volumes formulation at the staggered grid points $(i+1/2, i-1/2, j+1/2, j-1/2)$ to find $\bar{\mathbf{v}}$.
3. Solve the pressure correction equation (28) to find p' at integer points $(i, j), (i, j-1), (i, j+1), (i-1, j), (i+1, j)$. Since the corner grid points are avoided, the scheme is called "semi-implicit".
4. Correct the pressure and velocity using the equations (25), (26) and (27):

$$\begin{aligned} p &= \bar{p} + p', \\ \mathbf{v} &= \bar{\mathbf{v}} - \Delta t [Gp' - \mathbf{H}(\mathbf{v}') + \nu L(\mathbf{v}')]. \end{aligned} \quad (29)$$

5. Replace the previous intermediate values of pressure and velocity $(\bar{p}, \bar{\mathbf{v}})$ with the corrected values (p, \mathbf{v}) and return to Step 2.
6. Repeat Steps 2 to 5 until desired convergence is achieved.

There are several modifications of the current method that lead to better convergence. These algorithms were named SIMPLER (SIMPLE revised) and SIMPLEC (SIMPLE consistent), with SIMPLEC being the most effective scheme.

6.2 PISO - Pressure Implicit with Splitting of Operators

Issa [11, 12] proposed an additional correction step to the SIMPLE algorithm. The sequential procedure is described below.

1. Set p^n, \mathbf{v}^n from previous timestep.
2. Solve the momentum equation in control volumes formulation at the staggered grid points $(i+1/2, i-1/2, j+1/2, j-1/2)$ to find \mathbf{v}' .

$$\frac{1}{\Delta t} (\mathbf{v}' - \mathbf{v}^n) = -\mathbf{H}(\mathbf{v}) + \nu L(\mathbf{v}) - G\bar{p}. \quad (30)$$

3. Solve first pressure correction equation (28) to find p' at integer points $(i, j), (i, j-1), (i, j+1), (i-1, j), (i+1, j)$, the equation below has conservation of mass enforced $\nabla \cdot \mathbf{v}' = 0$.

$$DGp'' \equiv \nabla \cdot \nabla p' = -\frac{1}{\Delta t} (\nabla \cdot \mathbf{v}' - \nabla \cdot \mathbf{v}^n) = \frac{1}{\Delta t} (\nabla \cdot \mathbf{v}^n) = \frac{1}{\Delta t} D\mathbf{v}^n. \quad (31)$$

4. Apply first correction and obtain \mathbf{v}'' from

$$\frac{1}{\Delta t} (\mathbf{v}'' - \mathbf{v}^n) = D(-\mathbf{H}(\mathbf{v}') + \nu L(\mathbf{v}')) - G(p'). \quad (32)$$

5. Solve the second pressure correction

$$DGp'' \equiv \nabla \cdot \nabla p'' = \frac{1}{\Delta t} D\mathbf{v} + D(-\mathbf{H}(\mathbf{v}'') + \nu L(\mathbf{v}'')). \quad (33)$$

6. Apply second correction and obtain $\mathbf{v}''' = \mathbf{v}^{n+1}$ from

$$\frac{1}{\Delta t} (\mathbf{v}''' - \mathbf{v}^n) = -\mathbf{H}(\mathbf{v}'') + \nu L(\mathbf{v}'') \quad (34)$$

7 Discrete streamfunction method

7.1 Problem statement

Let the velocity vector $\mathbf{v}(t, x, y) = (u(t, x, y), v(t, x, y))$ and pressure $p(t, x, y)$ be the solutions to the system (1) on a unit square domain together with boundary conditions:

$$\text{Momentum: } \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nu \nabla \cdot \nabla \mathbf{v}, \quad 0 \leq x \leq 1, 0 \leq y \leq 1, t \geq 0, \quad (35a)$$

$$\text{Continuity: } \nabla \cdot \mathbf{v} = 0, \quad 0 \leq x \leq 1, 0 \leq y \leq 1, t \geq 0, \quad (35b)$$

$$\text{Freestream BC: } \mathbf{v}(t, 0, y) = \mathbf{v}(t, x, 1) = (u_0 + A \cos(kx - \omega t + \phi_0), 0), \quad \{A, k, \omega, \phi_0\} \subset \mathbb{R}, \quad (35c)$$

$$\text{No-slip BC: } \mathbf{v}(t, x, 0) = (0, 0), \quad (35d)$$

$$\text{Outlet BC: } \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0, v = \mathbf{h}(t, y), \quad (x = 1). \quad (35e)$$

7.2 Domain discretization

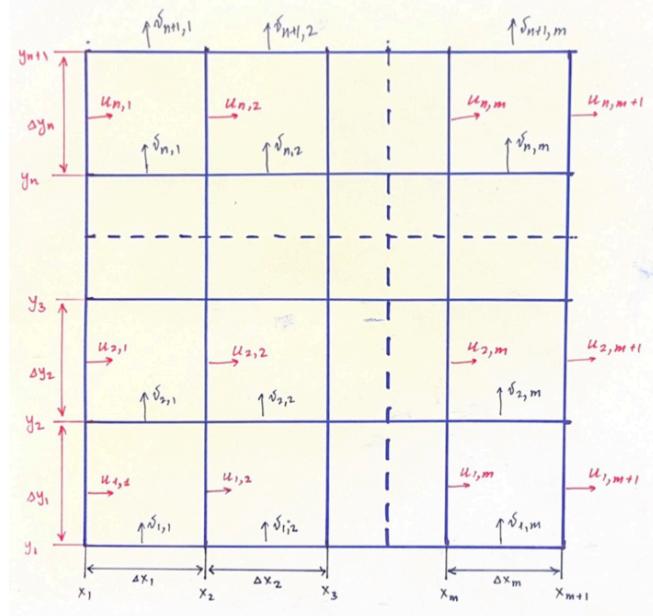


Figure 3: Domain discretization

The method described below is based on discrete volume formulation. Consider discretizing the area into m intervals horizontally and n intervals vertically, forming a grid of rectangular cells. To enhance memory access, it is recommended to organize the data using arrays instead of matrices. To

make things easier, label the cells by counting from the bottom left corner of the domain, going from left to right, and moving upwards after we reach the end of a row until we hit the top right corner of the domain.

In the context of two-dimensional analysis, the component u is stored at the center of the left face, while the value for v is stored at the center of the bottom face. To appropriately address boundary conditions, additional ghost cells with zero volume are introduced along the top and right boundaries of the entire domain.

The number of unknown components inside the domain (excluding boundaries) is $m(n - 1)$ for v and $(m - 1)n$ for u . In the given problem statement (35e), the values of u for the outlet are not explicitly defined. To address this, we introduce an additional n unknowns to the u array making the size of it $(m - 1)n + n = mn$. The counting process follows the same sequence as for cells, first for u and then for v . These resultant arrays are then concatenated into a single vector \mathbf{v} of size $mn + (n - 1)m$.

Grid indexation starts at $x_1 = y_1 = (0, 0)$ and ends at $x_{m+1} = (1, 0)$, $y_{n+1} = (0, 1)$. The intervals $\Delta x, \Delta y$ have the same indexation as the corresponding cells they belong to. For the grid refinement the standard constant ratio $k_x, k_y, \Delta x_{i+1} = k_x \Delta x_i, \Delta y_{j+1} = k_y \Delta y_j$ is used in both x and y directions.

7.3 Discrete operators

The initial system of equations (35) can be rewritten using discrete spatial operators as

$$\begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \frac{\partial}{\partial t} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{bmatrix} \hat{L} & -\hat{G} \\ -\hat{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ p \end{bmatrix} + \begin{pmatrix} -\hat{\mathbf{H}}(\mathbf{v}) \\ 0 \end{pmatrix} + \text{bc}_{\mathbf{v}, p}, \quad (36)$$

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

7.3.1 Transient terms

Attack the system (36) with the following schemes as in Colonius [5]:

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

$$\hat{L}\mathbf{v} = \frac{1}{2} (\hat{L}\mathbf{v}^{n+1} + \hat{L}\mathbf{v}^n) \quad (37)$$

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

$$\hat{\mathbf{H}}(\mathbf{v}) = \frac{3}{2} \hat{\mathbf{H}}(\mathbf{v}^n) - \frac{1}{2} \hat{\mathbf{H}}(\mathbf{v})^{n-1}. \quad (38)$$

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

$$\hat{G}p = \hat{G}p^{n+1}. \quad (39)$$

The above schemes result in the following discretized system:

$$\begin{bmatrix} \frac{1}{\Delta t} \mathbf{I} - \frac{1}{2} \hat{L} & \hat{G} \\ \hat{D} & 0 \end{bmatrix} \begin{pmatrix} \mathbf{v}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \left[\frac{1}{\Delta t} \mathbf{I} - \frac{1}{2} \hat{L} \right] \mathbf{v}^n - \left[\frac{3}{2} \hat{\mathbf{H}}(\mathbf{v}^n) - \frac{1}{2} \hat{\mathbf{H}}(\mathbf{v}^{n-1}) \right] \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{b}c_1 \\ \hat{b}c_2 \end{pmatrix}. \quad (40)$$

7.3.2 Laplacian

Laplacian can be rewritten as block matrix:

$$\hat{L} = \begin{bmatrix} \hat{L}_{xx}^u + \hat{L}_{yy}^u & 0 \\ 0 & \hat{L}_{xx}^v + \hat{L}_{yy}^v \end{bmatrix}. \quad (41)$$

Each pair of the matrices $\hat{L}_{xx}^u, \hat{L}_{xx}^v$ and $\hat{L}_{yy}^u, \hat{L}_{yy}^v$ will be equal on a uniform, but different on non-uniform grids. These matrices are computed in a similar way. Below we will discuss how the Laplacian matrix is constructed at different parts of the grid.

Inner part. Without loss of generality, let us show how to compute \hat{L}_{xx}^u , for the other three matrices the distances between velocity components are different on non-uniform grids, while the rest is verbatim. The power series of $u_{i\pm 1}$ at nodes $x_{i\pm 1}$ with respect to u_i at node x_i are

$$u_{i+1} = u_i + \frac{\partial u}{\partial x} \Big|_i (x_{i+1} - x_i) + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Big|_i (x_{i+1} - x_i)^2 + O(\Delta x^3), \quad (42)$$

$$u_{i-1} = u_i + \frac{\partial u}{\partial x} \Big|_i (x_{i-1} - x_i) + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Big|_i (x_{i-1} - x_i)^2 + O(\Delta x^3). \quad (43)$$

We can combine (43) and (42) by cancelling the first derivative, which will result in

$$\frac{\partial^2 u}{\partial x^2} \Big|_i \approx \frac{u_{i+1}(x_i - x_{i-1}) - u_i(x_{i+1} - x_{i-1}) + u_{i-1}(x_{i+1} - x_i)}{\left(\frac{x_{i+1} - x_{i-1}}{2}\right)(x_i - x_{i-1})(x_{i+1} - x_i)} + O(\Delta x) \quad (44)$$

$$\approx \frac{1}{h_c h_e} u_{i+1} + \frac{2}{h_w h_e} u_i + \frac{1}{h_c h_w} u_{i-1} + O(\Delta x). \quad (45)$$

where $h_w = x_i - x_{i-1}$, $h_c = \frac{x_{i+1} - x_{i-1}}{2}$, $h_e = x_{i+1} - x_i$ as in figure 4. Moreover, the order of this approximation becomes 2nd for uniform grids. The coefficients in front of the velocity components are placed into the \hat{L}_{xx}^u matrix.

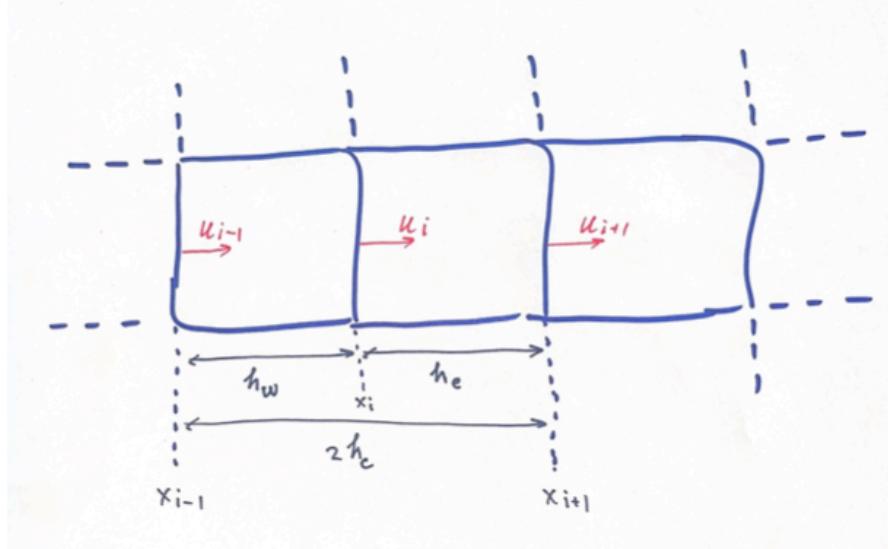


Figure 4: \hat{L}_{xx}^u inner part.

Right boundary. To be determined. To be determined.

Left boundary. Since the exact value on the left boundary (u_1 in a box on figure 5) is given as a Dirichlet boundary condition, it is possible to move the corresponding terms together with the coefficients to the right-hand side of the linear system and treat the values explicitly. The values are moved to the vector \hat{b}_1 on the right-hand side of the equation (40).

Top boundary. The Dirichlet condition over the top boundary affects only the \hat{L}_{yy}^u and \hat{L}_{yy}^v due to the 5-point stencil discretization. We will focus on the \hat{L}_{yy}^u in this part; the \hat{L}_{yy}^v is done similarly, except that the distances between the v velocities differ due to the non-uniform grid. Consider the top-most value of u at the J 'th node, with u_g being the ghost velocity outside the grid. The velocities u_g and u_J are equidistant from the top domain boundary

$$\frac{\partial^2 u}{\partial y^2} \Big|_J = \frac{u_g(h_s) + u_J(-2h_c) + u_{J-(m-1)}(h_n)}{h_n h_s h_c}, \quad (46)$$

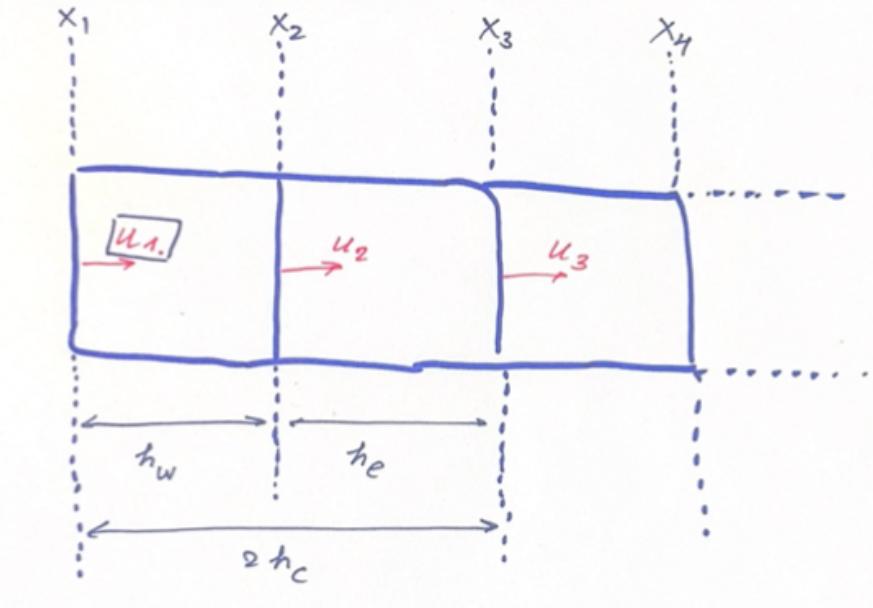


Figure 5: \hat{L}_{xx}^u at the left boundary.

where $h_s = y_J - y_{J-1}$, $h_c = \frac{y_{J+1} - y_J}{2}$, $h_n = y_{J+1} - y_J$.

Since the value of u_g at the centre of the topmost face lies outside the domain on a staggered grid, interpolation is necessary. The value of u along the top edge is known to be U , interpolation of $U = \frac{u_g + u_J}{2} \implies u_g = 2U - u_J$, which we can plug into the equation (46) above

$$\begin{aligned} \frac{\partial^2 u}{\partial y^2} \Big|_J &= \frac{2Uh_s}{h_n h_s h_c} + u_J \left(\frac{-h_s}{h_n h_s h_c} + \frac{-2h_c}{h_n h_s h_c} \right) + u_{J-(m-1)} \frac{h_n}{h_c h_s h}, \\ \frac{\partial^2 u}{\partial y^2} \Big|_J &= \frac{2U}{h_c h_c} + u_J \left(\frac{-(2h_c + h_s)}{h_c h_s h_c} \right) + u_{J-(m-1)} \frac{1}{h_s h_c}, \\ \frac{\partial^2 u}{\partial y^2} \Big|_J &= \frac{2U}{h_c^2} + u_J \left(\frac{-(2h_c + h_s)}{h_c^2 h_s} \right) + u_{J-(m-1)} \frac{1}{h_s h_c}. \end{aligned} \quad (47)$$

The first summand from the above equation is treated explicitly, i.e. moved to the vector \hat{bc}_1 on the right-hand side in equations (40), while the other two coefficients are used as elements for the \hat{L}_{yy}^u matrix.

Bottom boundary. We use the equation (44) and change the distances between velocities as in figure 6, which lead to

$$\frac{\partial^2 u}{\partial y^2} \Big|_j = u_{j+1} \left(\frac{1}{h_n h_c} \right) + u_j \left(\frac{-2}{h_n h_s} \right) + u_g \left(\frac{1}{h_s h_c} \right). \quad (48)$$

Interpolation of $u_{BC} = 0 = \frac{u_j + u_g}{2} \implies u_g = -2u_j$, then

$$\begin{aligned} \frac{\partial^2 u}{\partial y^2} \Big|_j &= u_{j+1} \left(\frac{1}{h_n h_c} \right) + u_j \left(\frac{-2}{h_n h_s} + \frac{-2}{h_s h_c} \right) \\ &= u_{j+1} \left(\frac{1}{h_n h_c} \right) + u_j \left(\frac{-2h_c - 2h_n}{h_s h_c h_n} \right). \end{aligned} \quad (49)$$

The coefficients are distributed into the \hat{L}_{yy}^u matrix similarly to left and top domain cases. It is, in fact, possible to use the third velocity from the boundary instead of the ghost u_g outside. However, the order of such a scheme will be reduced due to the large ratio of the distances between the neighbouring velocities.

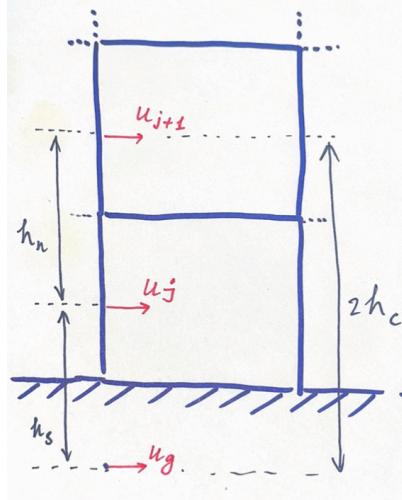


Figure 6: \hat{L}_{yy}^u at the bottom boundary.

7.3.3 Divergence

The second line of the system (36) reads

$$\begin{aligned} \hat{D}\mathbf{v} &= bc_2, \\ \frac{1}{\Delta x} D_x u + \frac{1}{\Delta y} D_y v &= bc_2, \\ \frac{1}{\Delta x \Delta y} \begin{bmatrix} D_x & D_y \end{bmatrix} \begin{bmatrix} u \Delta y \\ v \Delta x \end{bmatrix} &= \frac{1}{\Delta x \Delta y} Dq = bc_2, \end{aligned} \quad (50)$$

where the vector q is referred to as velocity flux, and the matrix D is the divergence matrix. The first-order central difference scheme was employed in the construction of this matrix. Now, consider a finite volume $V_{i,j}$ with velocities $u_{i,j}$ and $v_{i,j}$ on the left and bottom faces respectively. Two indices (row, column) are used for illustrative purposes, it is always recommended to use a single array of data, which will enhance the memory access. The continuity equation for the cell center is expressed as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (51)$$

which can be discretized using a central difference scheme into

$$\frac{u_{i+1,j} - u_{i,j}}{\Delta x_i} + \frac{v_{i,j+1} - v_{i,j}}{\Delta y_i} = 0. \quad (52)$$

Since the vector of velocities is $\mathbf{v} = [u_{i+1,j}, u_{i,j}, v_{i,j+1}, v_{i,j}]^T$, we can express this equation in matrix form as follows:

$$\frac{1}{\Delta x_i \Delta y_i} [1 \quad -1 \quad 1 \quad -1] [\Delta y_i u_{i+1,j}, \Delta y_i u_{i,j}, \Delta x_i v_{i,j+1}, \Delta x_i v_{i,j}]^T = 0. \quad (53)$$

The velocity values known at the boundaries can be shifted to the right-hand side and explicitly treated. The divergence matrix contains ± 1 , as illustrated in the figure 7.

7.3.4 Gradient

Since the unknowns are the velocities, the pressure gradients are computed at the corresponding points. Interestingly, the gradient operator is the transpose of divergence on staggered/MAC grids

$$\hat{G} = -\hat{D}^T. \quad (54)$$

- D ($m=5, n=4$)

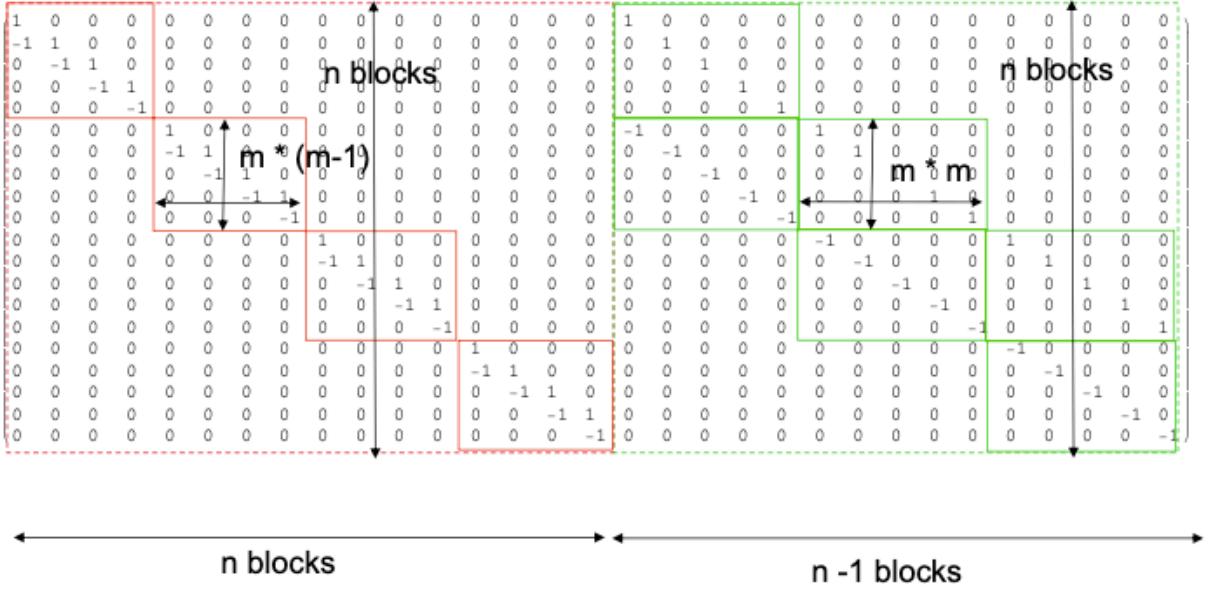


Figure 7: Divergence matrix.

To show why consider velocity $u_{i,j}$, compute pressure gradient across the corresponding point

$$\left. \frac{\partial p}{\partial x} \right|_{u_{i,j}} = \frac{p_{i,j} - p_{i-1,j}}{\frac{\Delta x_i - \Delta x_{i-1}}{2}}. \quad (55)$$

The above equation will give

$$\frac{2}{\Delta x_i - \Delta x_{i-1}} [-1 \ 1] [p_{i-1,j} \ p_{i,j}]^T, \quad (56)$$

which in matrix notation for all velocity points will become

$$\nabla p = \Delta_{xy} \hat{G} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{mn} \end{bmatrix}, \quad (57)$$

where Δ_{xy} is the diagonal matrix containing the distances between neighbouring pressure coordinates, \hat{G} is the gradient matrix and $[p_1, p_2, \dots, p_{mn}]^T$ is the vector of pressure values at the cell centres. Note that on staggered/MAC grids pressure gradient is computed directly at each velocity coordinate without the need of any interpolation.

7.3.5 Advection

Photo in the figure 8 displays the discretization of nonlinear terms on a 4×4 staggered grid. Values of uu and vv are defined in cell centres using the (left/right) interpolation, and $uv = vu$ values are defined on grid corners using the cross (left/right times up/down) interpolation. The edges and outer corners are dropped at the end since the system of linear equations is determined by the inner velocity values.

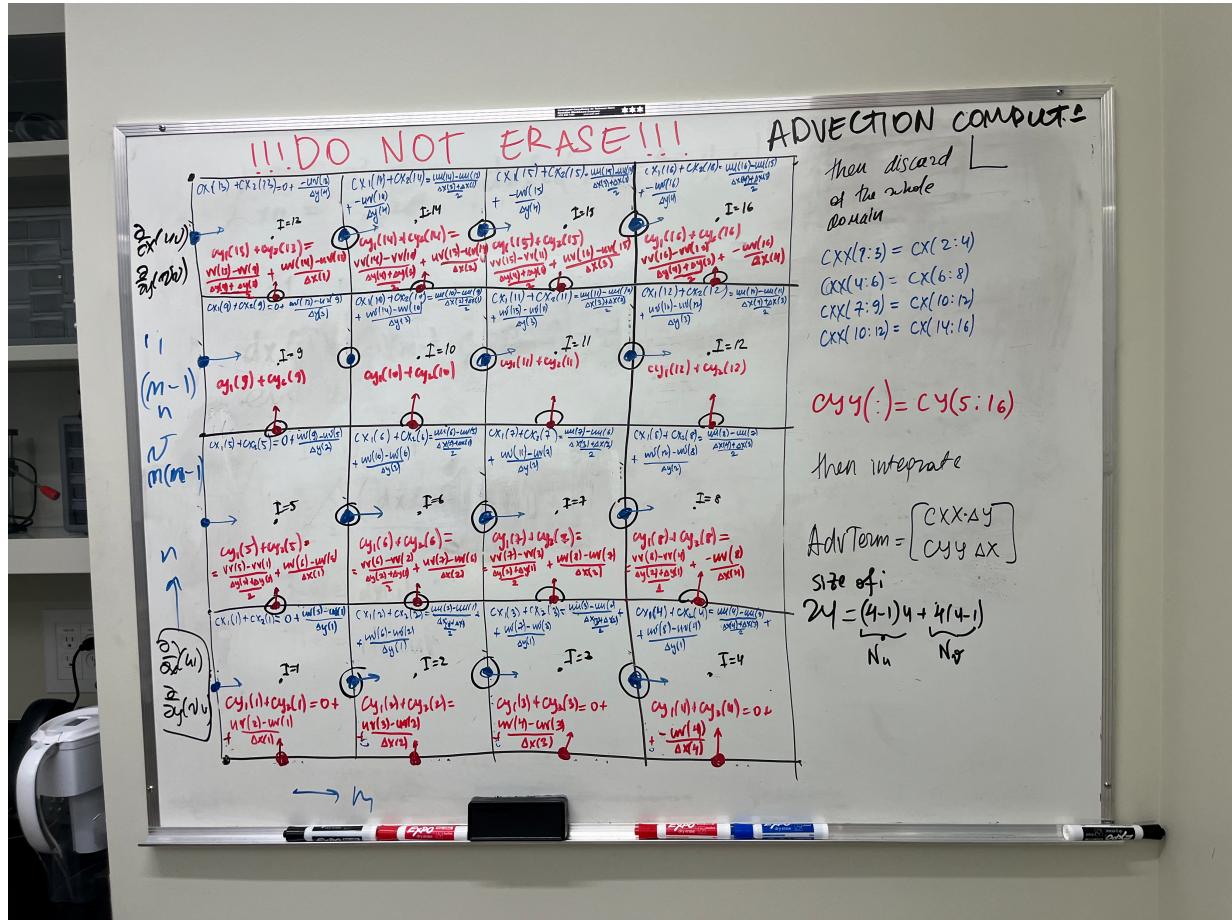


Figure 8: Advection on a staggered grid.

Inside the domain x -momentum advective derivative can be written in a conservative form as

$$\begin{aligned}
u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= u \frac{\partial u}{\partial x} + 0 + v \frac{\partial u}{\partial y} \\
&= u \frac{\partial u}{\partial x} + u \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + v \frac{\partial u}{\partial y} \\
&= \left(u \frac{\partial u}{\partial x} + u \frac{\partial u}{\partial x} \right) + \left(\frac{\partial v}{\partial y} + v \frac{\partial u}{\partial y} \right) \\
&= \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y}, \\
\left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right)_{u_{i,j}} &= \frac{(uu)_{i+\frac{1}{2},j} - (uu)_{i-\frac{1}{2},j}}{\frac{\Delta x_{i+1} + \Delta x_{i-1}}{2}} + \frac{(uv)_{i,j+\frac{1}{2}} - (uv)_{i,j-\frac{1}{2}}}{\Delta y_i},
\end{aligned}$$

where the half indices correspond to the coordinates at the cell centres for uu and cell edges for uv . The computation of uu, uv values is performed using the interpolation, i.e.

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

The advective term in the y -momentum equation is computed in a similar manner. After the derivative values are computed they are moved to the right-hand side as in equation (40) and treated explicitly.

7.3.6 Resulting system

After applying the above discrete operators we can combine the summands into \hat{A}, \hat{r}^n ; we then rewrite the system (40) as

$$\begin{bmatrix} \hat{A} & \hat{G} \\ \hat{D} & 0 \end{bmatrix} \begin{pmatrix} \mathbf{v}^{n+1} \\ p \end{pmatrix} = \begin{pmatrix} \hat{r}^n \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{bc}_1 \\ \hat{bc}_2 \end{pmatrix}. \quad (58)$$

7.4 Symmetrization

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 symmetricity, we introduce two matrices: the diagonal scaling matrix \hat{M} and the diagonal flux matrix R . These matrices are defined as

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

Using the above two we can express $q^{n+1} = R\mathbf{v}^{n+1} \implies \mathbf{v}^{n+1} = R^{-1}q^{n+1}$.

The only matrix which is asymmetric in (40) is Laplacian. In order to get a symmetric diffusion matrix we need to move to the new variables. As discussed in 7.2 the domain has m intervals in x direction and n intervals in y direction. The number of unknown velocities is $N_u = (m-1)n + n = mn$ and $N_v = m(n-1)$. The number of unknowns is equal to the number of face velocities in the interior of the domain plus the outlet boundary (extra n in N_u). We can rewrite \mathbf{v} as a vector of unknowns

$$\mathbf{v} = \underbrace{[u_1; u_2; \dots; u_m; u_1; u_2; \dots; u_m; \dots; u_1; u_2; \dots; u_m]}_{n \text{ blocks of } 1, \dots, m}, \underbrace{[v_1; v_2; \dots; v_m; v_1; v_2; \dots; v_m; \dots; v_1; v_2; \dots; v_m]}_{n-1 \text{ of each } 1, \dots, m \text{ block}}$$

In order to maintain consistency - indexation goes left to right from bottom to top as discussed earlier. Define matrices

$$\Delta y_j = \text{diag}([\underbrace{\Delta y_1; \Delta y_1; \dots; \Delta y_1}_{m \text{ times}}; \underbrace{\Delta y_2; \Delta y_2; \dots; \Delta y_2}_{m \text{ times}}; \underbrace{\Delta y_n; \Delta y_n; \dots; \Delta y_n}_{m \text{ times}}]),$$

$$\Delta x_i = \text{diag}([\underbrace{\Delta x_1; \Delta x_2; \dots; \Delta x_m; \Delta x_1; \Delta x_2; \dots; \Delta x_m; \dots; \Delta x_1; \Delta x_2; \dots; \Delta x_m}_{n-1 \text{ blocks of } 1, \dots, m}]),$$

combined into

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

Let the new vector variable of unknowns (mass flux) be defined as

$$q = R\mathbf{v},$$

where R is a diagonal matrix of size $mn \times m(n-1)$. We need to define the difference matrices to cancel out the $\frac{x_{i+1} - x_{i-1}}{2}$ central term in diffusion discretization. Let the diagonal matrices

$$\Delta x_i + \Delta x_{i-1} = \text{diag}[\underbrace{\Delta x_2 + \Delta x_1; \Delta x_3 + \Delta x_2; \dots; \Delta x_{m+1} + \Delta x_m; \dots; \Delta x_2 + \Delta x_1; \Delta x_3 + \Delta x_2; \dots; \Delta x_{m+1} + \Delta x_m}_{n \text{ times for each block of } \Delta x_2 + \Delta x_1 \text{ to } \Delta x_{m+1} + \Delta x_m}],$$

where the $x_{m+1} = x_m$ is a ghost interval lying outside the domain.

$$\Delta y_j + \Delta y_{j-1} = \text{diag}[\underbrace{\Delta y_2 + \Delta y_1; \dots; \Delta y_2 + \Delta y_1}_{m \text{ times}}; \underbrace{\Delta y_3 + \Delta y_2; \dots; \Delta y_3 + \Delta y_2}_{m \text{ times}}; \underbrace{\Delta y_n + \Delta y_{n-1}; \dots; \Delta y_n + \Delta y_{n-1}}_{m \text{ times}}],$$

be combined into

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

Now, since $q = R\mathbf{v}$ then $\mathbf{v} = R^{-1}q$, The Laplacian matrix multiplied by the velocity vector is

$$L\mathbf{v} = LR^{-1}q,$$

premultiply by \hat{M} , then the expression becomes

$$\hat{M}LR^{-1}q,$$

where $\hat{M}LR^{-1}$ is symmetric by construction.

Using the above transformation we can modify the initial system of linear equations

$$\begin{bmatrix} \hat{A} & \hat{G} \\ \hat{D} & 0 \end{bmatrix} \begin{pmatrix} \mathbf{v}^{n+1} \\ p \end{pmatrix} = \begin{pmatrix} \hat{r}^n \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{b}c_1 \\ \hat{b}c_2 \end{pmatrix} \quad (59)$$

into

$$\begin{bmatrix} \hat{M}\hat{A}R^{-1} & \hat{M}\hat{G} \\ \hat{D}R^{-1} & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ p \end{pmatrix} = \begin{pmatrix} \hat{M}\hat{r}^n \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{M}\hat{b}c_1 \\ \hat{b}c_2 \end{pmatrix}, \quad (60)$$

which can be rewritten as

$$\begin{bmatrix} A & G \\ D & 0 \end{bmatrix} \begin{pmatrix} q^{n+1} \\ p \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \end{pmatrix}, \quad (61)$$

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.

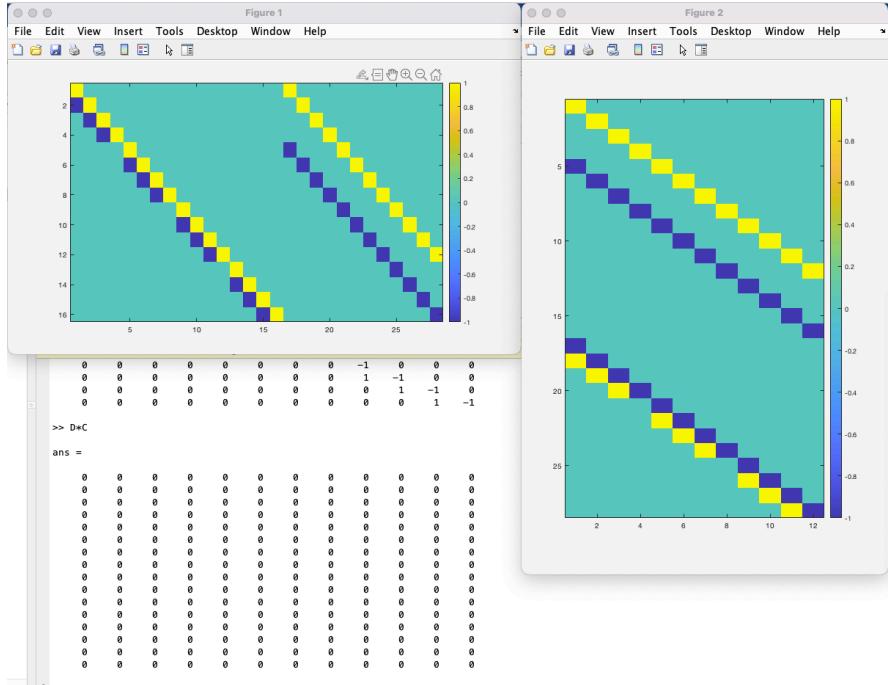


Figure 9: Divergence and Curl matrices.

7.5 Nullspace method

Publications of Chang [3] and Hall [8] use the idea that in the system (61) matrix D is wider than tall, hence it defines a nullspace. The nullspace is the set of all solutions to the homogeneous linear system $Dx = 0$, where x is a vector in the null space of D . Let C be the nullspace matrix containing such vectors x .

The number of rows in the nullspace C is equal to the number of faces with unknown velocities (N_f). In two dimensions C has N_n columns, which is equal to the number of 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, which are $+1$ 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 [3].

A more intuitive way of constructing the matrix C relies on the utilization of counterclockwise vorticity around the nodes within the domain and along the boundary if such nodes have adjacent velocities from \mathbf{v} . In the case of the open boundary vortices around the nodes belonging to the open segment are taken into consideration. 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.

The matrix C has dimensions corresponding to unknown velocities multiplied by the number of nodes around which these velocities revolve. Figure 9 illustrates matrices D and C for a 4×4 grid with an open boundary on the right side of the domain. Yellow squares represent $+1$ entries, while blue squares correspond to -1 entries.

The desired product, then, becomes $DC = 0$. Since $D = -G^T$, as discussed earlier in subsections 7.3.3 and 7.3.4. The identity leads to important property $(DC)^T = C^T D^T = -C^T G = 0$, which is worth keeping in mind for future use.

The final chord of this method is the introduction of another new variable called streamfunction $\psi : q_h = C\psi$, where q_h is a homogeneous solution to the problem (61) from subsection 7.4 and is discussed in the following subsection 7.6.

7.6 Resulting algorithm

Let us consider the solution to the discretized system (61)

$$q^{n+1} = q_p^{n+1} + q_h^{n+1}, \quad (62)$$

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

$$Dq_h^{n+1} = 0, \quad (63)$$

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

$$Dq_p^{n+1} = bc_2. \quad (64)$$

The algorithm to solve the discrete Navier-Stokes system of equations (61) can be described as follows:

1. Construct matrix C , such that $DC = 0$ and $q_h = C\psi$.
2. Rewrite $q^{n+1} = q_p^{n+1} + q_h^{n+1}$.
3. Find q_p^{n+1} from the non-homogeneous continuity equation using the methods described in the next subsection 7.7.
4. Split the solution q^{n+1} into homogeneous and particular, then eliminate the pressure terms in the momentum equation.

$$\begin{aligned} Aq^{n+1} &= -Gp^{n+1} + S_1 + bc_1, & \text{where } S_1 \text{ is momentum source term,} \\ A(q_h^{n+1} + q_p^{n+1}) &= D^T p^{n+1} + S_1 + bc_1, & \text{premultiply by } C^T, \\ C^T Aq_h^{n+1} &= C^T (S_1 + bc_1 - Aq_p^{n+1}), & \text{use } q_h^{n+1} = C\psi^{n+1}, \\ C^T A C \psi^{n+1} &= C^T (S_1 + bc_1 - Aq_p^{n+1}). \end{aligned}$$

5. Solve the resulting system from above for ψ^{n+1} .
6. Obtain $q_h^{n+1} = C\psi^{n+1}$.
7. Compute $q^{n+1} = q_p^{n+1} + q_h^{n+1}$.
8. Repeat the steps (3-7) since the boundary values in q^{n+1} were changed.

7.7 A particular solution to the discrete continuity equation

7.7.1 Particular solution using Lagrange multipliers

It is possible to find one particular solution to the discrete non-homogeneous continuity equation (64) using the method of Lagrange multipliers as follows

$$Dq_p^{n+1} = bc_2, \implies \mathcal{L}(q_p, \lambda) = \|q_p\|^2 + \lambda^T (bc_2 - Dq_p). \quad (65)$$

Differentiating w.r.t q_p and finding the minimum (derivative equal to zero) yields to

$$2q_p - D^T \lambda = 0. \quad (66)$$

We may premultiply by D to obtain

$$2Dq_p - DD^T \lambda = 0, \quad (67)$$

the substitution of $bc_2 = Dq_p$ will lead to

$$\begin{aligned} 2bc_2 &= DD^T \lambda, \\ \lambda &= 2(DD^T)^{-1} bc_2, \end{aligned} \quad (68)$$

plugging the above into (66) results in

$$q_p = D^T (DD^T)^{-1} bc_2 = D^\dagger bc_2, \quad (69)$$

where $D^\dagger = D^T (DD^T)^{-1}$ is called a pseudo-inverse. In the method above we minimize the square of mass flux, which is equivalent to the minimization of kinetic energy.

7.7.2 Particular solution using Graph Theory

TO KEEP OR NOT TO KEEP Another option to find a particular solution to (64) was proposed by Hall in [8] and [9]. The method uses the spanning tree of graph \mathcal{N} which has nodes and edges corresponding to the cell centres and velocity vectors from our grid. This method is discussed below.

Let us write useful notations:

- \mathcal{N} - directed planar network (graph).
- K - a set of nodes (control volume centres).
 N - number of nodes.
- Λ - a set of interior Λ^0 and boundary $\partial\Lambda$ links (face velocities).
 M - total number of links.
 M^0 - number of interior links.
- Pendant node - a node which is an extremity of only one interior link.
- \mathcal{T} - a spanning tree of directed network \mathcal{N} . \mathcal{T} is a connected partial network of \mathcal{N} which has no cycles.

The procedure for finding a particular solution is as follows:

1. Construct a spanning tree \mathcal{T} of \mathcal{N} with the pressure specified nodes appended.
2. Velocities associated with links not in the tree are set to zero.
3. Choose a node in \mathcal{T} which is an extremity of a boundary link and call it root.
4. Set the boundary link velocities equal to zero, except the link incident on the root.
5. At each pendant node except the root, the continuity equation was reduced to a simple equation relating the velocity only at the internal connection of the tree to that node.
6. Move from the pendant node along the tree towards the root. At each node the continuity equation involves a single unknown velocity.
7. Except the juncture nodes, where the chains from one or more pendant nodes connect. Before solving the continuity equation associated with such a juncture node, the velocities on the chains from pendant nodes to the juncture node have to be determined.
8. The process is completed after using the continuity equation at the root to determine the final velocity of the boundary link.

8 Other interesting methods for solving Naver Stokes (may include algorithms from articles [2, 6, 13, 16])

Citing Colonius [5] The second type of error associated with vorticity advecting or diffusing through the boundary is typically handled by posing outflow boundary conditions. For incompressible flow, these are usually called convective boundary conditions, whereas for compressible flow the term non-reflecting boundary condition is often used. Multidomain boundary conditions used on 3 domains: fine, medium, and coarse.

9 Vorticity-Streamfunction formulation

Consider:

1. Streamfunction $\psi(x, y, t)$ of an incompressible two-dimensional flow:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \quad (70)$$

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

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

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

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \quad (71)$$

Importantly, continuity equation (1b) is satisfied immediately by taking spatial derivatives of streamfunction (70) components and adding them up. The governing equations (1) can now be transformed. The new set of equations becomes:

1. Transport equation for vorticity.

Application of $(\nabla \times)$ to momentum and taking into account continuity (1b) together with the fact

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

results in

$$\boxed{\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right)}. \quad (72)$$

2. Vorticity-Streamfunction equation.

After substituting streamfunction definition (70) into vorticity (71) we obtain

$$\boxed{\nabla^2 \psi = -\omega.} \quad (73)$$

These two equations above form a coupled system, and the pressure field (as promised at the end of Section 4) does not explicitly appear in neither equations (72) nor (73) and, in principle, is not needed in the solution.

The system of equations ((72),(73)) requires boundary conditions on ψ and ω . For the streamfunction, imposing physically plausible conditions is not difficult. One has to write the proper boundary conditions for the velocity components and use vorticity (71) to represent them as conditions for ψ and its derivatives. 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 application of the equation (73) at the boundary. Since the process of finding boundary conditions of ω typically results in expressions containing second derivatives, special numerical treatment is required.

References

- [1] S. Biringen and C.-Y. Chow. *An Introduction to Computational Fluid Mechanics by Example*. John Wiley & Sons, Inc., 2011.
- [2] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 168:464–499, 2001.
- [3] Wang Chang, Francis Giraldo, and Blair Perot. Analysis of an exact fractional step method. *Journal of Computational Physics*, 180(1):183–199, 2002.
- [4] A. J. Chorin. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

- [5] 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.
- [6] J. Dukowicz and A. Dvinsky. Approximate factorisation as a high order splitting for the incompressible flow equations. *Journal of Computational Physics*, 102:336–347, 1992.
- [7] Philip M. Gresho and Robert L. Sani. On pressure boundary conditions for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 7(10):1111–1145, 1987.
- [8] 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.
- [9] Charles A. Hall. Numerical solution of Navier-Stokes problems by the dual variable method. *SIAM Journal on Algebraic Discrete Methods*, 6:220–236, 1985.
- [10] 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.
- [11] R. I. Issa. Solution of the implicitly discretized fluid flow equations by operator splitting. 62:40–65, 1985.
- [12] R. I. Issa, A. D. Gosman, and A. P. Watkins. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. 62:66–82, 1986.
- [13] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, 59:308–323, 1985.
- [14] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere, Washington, D.C., 1980.
- [15] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15:1787–1806, 1972.
- [16] J. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108:249, 1993.
- [17] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, 1983.
- [18] R. Témam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I). *Archive for Rational Mechanics and Analysis*, 32:135–153, 1969.
- [19] R. Témam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II). *Archive for Rational Mechanics and Analysis*, 33:377–385, 1969.
- [20] N. N. Yanenko. *The Method of Fractional Steps*. Springer-Verlag, New York, 1971.
- [21] O. Zikanov. *Essential Computational Fluid Dynamics*. Wiley, Hoboken, N.J., 2010.

A Appendix

A.1 Transient schemes

A.2 Laplacian symmetrization in depth

In order to get a symmetric diffusion matrix we need to move to the new variables.

Let us have $N_x = m$ intervals in x direction and $N_y = n$ intervals in y direction. The number of unknowns then will be $N_u = (N_x - 1)N_y + N_y = mn$ and $N_v = N_x(N_y - 1) = m(n - 1)$. The number

of unknowns is equal to the number of face velocities in the interior of the domain plus the outlet boundary (extra N_y in N_u). Rewrite \mathbf{v} as a vector of unknowns

$$\mathbf{v} = \underbrace{[u_1; u_2; \dots; u_m; u_1; u_2; \dots; u_m; \dots; u_1; u_2; \dots; u_m]}_{n \text{ times for each } 1 \dots m \text{ block}}, \underbrace{v_1; v_2; \dots; v_m; v_1; v_2; \dots; v_m; \dots; v_1; v_2; \dots; v_m]}_{n-1 \text{ of each } 1, \dots, m \text{ block}}.$$

To maintain consistency - indexation goes left to right from bottom to top.

Define matrices

$$\begin{aligned} \Delta y_j &= \text{diag}([\underbrace{\Delta y_1; \Delta y_1; \dots; \Delta y_1}_{m \text{ times}}; \underbrace{\Delta y_2; \Delta y_2; \dots; \Delta y_2}_{m \text{ times}}; \dots; \underbrace{\Delta y_n; \Delta y_n; \dots; \Delta y_n}_{m \text{ times}}]), \\ \Delta x_i &= \text{diag}([\underbrace{\Delta x_1; \Delta x_2; \dots; \Delta x_m}_{n-1 \text{ times}}; \underbrace{\Delta x_1; \Delta x_2; \dots; \Delta x_m}_{m \text{ times}}; \dots; \underbrace{\Delta x_1; \Delta x_2; \dots; \Delta x_m}_{m \text{ times}}]) \\ R &= \begin{bmatrix} \Delta y_j & 0 \\ 0 & \Delta x_i \end{bmatrix}. \end{aligned}$$

Let the new vector variable of unknowns (something like mass flux) be defined as

$$\mathbf{q} = R\mathbf{v},$$

where R is a diagonal matrix of size $mn \times m(n-1)$ defined above.

Next, we need to define the difference matrices to cancel out the $\frac{x_{i+1}x_{i-1}}{2}$ central term in diffusion discretization. Define the diagonal matrices

$$\Delta x_i + \Delta x_{i-1} = \text{diag}[\underbrace{\Delta x_2 + \Delta x_1; \Delta x_3 + \Delta x_2; \dots; \Delta x_{m+1} + \Delta x_m}_{n \text{ times for each block of } \Delta x_2 + \Delta x_1 \text{ to } \Delta x_{m+1} + \Delta x_m}; \dots; \Delta x_2 + \Delta x_1; \Delta x_3 + \Delta x_2; \dots; \Delta x_{m+1} + \Delta x_m],$$

where the $x_{m+1} = x_m$ is a ghost interval lying outside the domain.

$$\begin{aligned} \Delta y_j + \Delta y_{j-1} &= \text{diag}[\underbrace{\Delta y_2 + \Delta y_1; \dots; \Delta y_2 + \Delta y_1}_{m \text{ times}}; \underbrace{\Delta y_3 + \Delta y_2; \dots; \Delta y_3 + \Delta y_2}_{m \text{ times}}; \dots; \underbrace{\Delta y_n + \Delta y_{n-1}; \dots; \Delta y_n + \Delta y_{n-1}}_{m \text{ times}}], \\ \hat{M} &= \begin{bmatrix} \frac{1}{2}(\Delta x_i + \Delta x_{i-1}) & 0 \\ 0 & \frac{1}{2}(\Delta y_j + \Delta y_{j-1}) \end{bmatrix}. \end{aligned}$$

Now, since $\mathbf{q} = R\mathbf{v}$ then $\mathbf{v} = R^{-1}\mathbf{q}$, Going back to Laplacian matrix

$$L\mathbf{v} = LR^{-1}\mathbf{q},$$

premultiply by \hat{M} , the expression will become

$$\hat{M}LR^{-1}\mathbf{q},$$

where $\hat{M}LR^{-1}$ is symmetric by construction.