# On Accuracy of Operator Splitting Techniques for Unsteady Navier-Stokes Equations

Seongjai Kim* and Hyung-Chun Lee†

November 1, 2000

**Abstract**

The article is concerned with operator splitting techniques for the numerical solution of the unsteady Navier-Stokes equations approximated by the Crank-Nicolson method in time and the second-order finite volume method in space. Most operator splitting methods introduce a splitting error of asymptotically the same order as the discretization error; however, the splitting error and therefore the total error can be much larger than the discretization error unless the timestep size is sufficiently small. We suggest a new operator splitting algorithm that invokes a splitting error of $\mathcal{O}((\Delta t)^3)$, one order higher than the discretization error. **It has been numerically verified that the new algorithm improves the overall accuracy of the solution for relatively large timestep sizes.**

## 1. Introduction

Let $\Omega$ be a bounded domain in the two-dimensional (2D) space and $J = (0, T]$, $T > 0$. Consider the unsteady, incompressible. viscous Navier-Stokes (NS) equations:

$$\begin{aligned} &\text{(a)} \quad \nabla \cdot (\rho \mathbf{v}) = 0, && (\mathbf{x}, t) \in \Omega \times J, \\ &\text{(b)} \quad \frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = \nabla \cdot \mathcal{T} + \rho \mathbf{g}, && (\mathbf{x}, t) \in \Omega \times J, \end{aligned} \tag{1.1}$$

where $\rho$ is the density (mass per unit volume) which is independent of time but may vary in space, $\mathbf{v}$ denotes the velocity (momentum per unit mass), and $\mathbf{g}$ is the gravity

*Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA Email: skim@ms.uky.edu

†Department of Mathematics, Ajou University, San 5, Wonchun-dong, Paldal-gu, Suwon, 442-749 South Korea Email: hclee@minmax.ajou.ac.kr

vector. Here the stress tensor $\mathcal{T}$, which is the molecular transport rate of momentum, reads

$$\mathcal{T} = -pI + \boldsymbol{\tau}, \quad \boldsymbol{\tau} = 2\mu\mathcal{D} + \left[\left(\kappa - \frac{2}{3}\mu\right)\nabla\cdot\mathbf{v}\right]I,$$

where $p$ is the static pressure, $\boldsymbol{\tau}$ is the viscous part of the stress tensor, $\mu$ and $\kappa$ are respectively the shear coefficient of viscosity and the bulk coefficient of viscosity, $I$ is the unit (identity) tensor, and $\mathcal{D}$ is the rate of strain (deformation) tensor defined by

$$\mathcal{D} = \frac{1}{2}\Big(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\Big).$$

In numerical approximation with finite volume methods, it is a common practice to express the NS equations in the integral form: for $\Omega_0 \subset \Omega$,

$$\begin{aligned}
&\text{(a)} \quad \int_{\partial\Omega_0} \rho\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}S = 0,\\
&\text{(b)} \quad \frac{\partial}{\partial t}\int_{\Omega_0} \rho v_i\,\mathrm{d}\Omega_0 + \int_{\partial\Omega_0} \rho v_i\,\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}S\\
&\qquad = -\int_{\partial\Omega_0} pI_i\cdot\mathbf{n}\,\mathrm{d}S + \int_{\partial\Omega_0} \tau_{ij}I_j\cdot\mathbf{n}\,\mathrm{d}S + \int_{\Omega_0} \rho g_i\,\mathrm{d}\Omega_0.
\end{aligned} \tag{1.2}$$

This paper is concerned with operator splitting (OS) techniques for the numerical solution of NS equations (1.1). In an OS method, the velocity components and the pressure in a time step are computed by carrying out a series of fractional steps that are much easier to solve than the original operator. But OS methods introduce a certain degree of extra error, called the *splitting error*.

A popular OS method known in the literature is based on the Crank-Nicolson method in time and a second-order finite difference (FD) or finite volume (FV) method in space, which is similar to that suggested by Choi and Moin [1]. The OS method consists of three fractional steps: (1) an initial velocity solve incorporating the former time level pressure, (2) a velocity update by the removal of half the old pressure gradient, and (3) the pressure update associating the continuity equation and the final velocity update. The splitting error is $\mathcal{O}(\Delta t^2)$, which is the same order as the truncation error already introduced.

The main goal of the paper is to introduce a variant of the Choi-Moin splitting algorithm, which incorporates a smaller splitting error of $\mathcal{O}(\Delta t^3)$.

## 2. Preliminaries

The NS equations (1.1) can be equivalently rewritten in index notation:

$$\begin{aligned}
&\text{(a)} \quad \frac{\partial(\rho v_j)}{\partial x_j} = 0, && (\mathbf{x},t)\in\Omega\times J,\\
&\text{(b)} \quad \frac{\partial(\rho v_i)}{\partial t} + \frac{\partial(\rho v_j v_i)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial\tau_{ij}}{\partial x_j} + \rho g_i, && (\mathbf{x},t)\in\Omega\times J,
\end{aligned} \tag{2.1}$$

where the Einstein convention has been utilized.

For a simple presentation, we define an operator $H(v_i)$ representing discretized convective and diffusive terms:

$$H(v_i) = \frac{\delta(\rho v_j v_i)}{\delta x_j} - \frac{\delta \tau_{ij}}{\delta x_j}.$$

Here and in the remainder of the section, numerical approximations for the spatial derivatives would be formally expressed. The Crank-Nicolson scheme for (2.1.b) can be formulated as follows:

$$\frac{(\rho v_i)^n - (\rho v_i)^{n-1}}{\Delta t^n} + \frac{1}{2}\Big[H(v_i^n) + H(v_i^{n-1})\Big] = -\frac{1}{2}\Big(\frac{\delta p^n}{\delta x_i} + \frac{\delta p^{n-1}}{\delta x_i}\Big) + \rho g_i, \tag{2.2}$$

where $\Delta t^n = t^n - t^{n-1}$. When the spatial derivatives are approximated by second-order schemes, the scheme (2.2) shows the truncation error of

$$\mathcal{O}\Big((\Delta x)^2 + (\Delta y)^2 + (\Delta t^n)^2\Big).$$

In the following we introduce an OS method for (2.2) studied by Choi and Moin [1], which consists of three fractional steps for the advancement of a time level.

In the first step, the velocity is advanced with the pressure explicitly treated:

$$\frac{(\rho v_i)^* - (\rho v_i)^{n-1}}{\Delta t^n} + \frac{1}{2}\Big[H(v_i^*) + H(v_i^{n-1})\Big] = -\frac{\delta p^{n-1}}{\delta x_i} + \rho g_i. \tag{2.3}$$

Note the above is nonlinear; it should be solved iteratively up to the satisfaction of a narrow tolerance. One can employ a Newton method or Picard iteration.

In the second step, half the old pressure gradient is removed:

$$\frac{(\rho v_i)^{**} - (\rho v_i)^*}{\Delta t^n} = \frac{1}{2}\frac{\delta p^{n-1}}{\delta x_i}. \tag{2.4}$$

The third and final step computes the velocity in the new time level, requiring the gradient of the new pressure:

$$\frac{(\rho v_i)^n - (\rho v_i)^{**}}{\Delta t^n} = -\frac{1}{2}\frac{\delta p^n}{\delta x_i}. \tag{2.5}$$

Here the new pressure can be found by solving the Poisson equation

$$\frac{\delta}{\delta x_i}\Big(\frac{\delta p^n}{\delta x_i}\Big) = \frac{2}{\Delta t^n}\frac{\delta(\rho v_i)^{**}}{\delta x_i}, \tag{2.6}$$

which is obtained to fill the requirement that the new velocity should satisfy the continuity equation (2.1.a). Then, the new velocity can be computed through (2.5), using the new pressure just computed.

Thus the above three-step OS method (2.3)-(2.5) satisfies the continuity equation and the momentum equations as follows:

$$\frac{(\rho v_i)^n - (\rho v_i)^{n-1}}{\Delta t^n} + \frac{1}{2}\Big[H(v_i^*) + H(v_i^{n-1})\Big] = -\frac{1}{2}\Big(\frac{\delta p^n}{\delta x_i} + \frac{\delta p^{n-1}}{\delta x_i}\Big) + \rho g_i. \tag{2.7}$$

The splitting error incorporated in the method is

$$H(v_i^n) - H(v_i^*) = H'(\widehat{v}_i)(v_i^n - v_i^*),$$

for some $\widehat{v}_i$ between $v_i^n$ and $v_i^*$. Adding (2.4) and (2.5) leads to

$$v_i^n - v_i^* = -\frac{\Delta t^n}{2\rho}\frac{\delta}{\delta x_i}(p^n - p^{n-1}).$$

Thus the splitting error of (2.3)-(2.5) satisfies

$$H(v_i^n) - H(v_i^*) = \mathcal{O}((\Delta t^n)^2), \tag{2.8}$$

which is the same order as the truncation error of the Crank-Nicolson method already introduced.

Such a splitting technique is rather popular in modern scientific computation for large-scaled problems, in particular, for 3D problems. Even though the splitting error is in the same order as the truncation error, the practical splitting error can be much larger than expected as shown for the ADI method for heat flow problems [2].

## 3. The method

A reasonable remedy for the reduction of the splitting error is to modify the splitting method so as to have a splitting error of one order higher than the truncation error. Consider the following modification:

$$\begin{aligned}
&\text{(a)}\quad \frac{(\rho v_i)^* - (\rho v_i)^{n-1}}{\Delta t^n} + \frac{1}{2}\Big[H(v_i^*) + H(v_i^{n-1})\Big] \\
&\qquad = -\frac{\delta}{\delta x_i}\Big(\Big(1 + \frac{1}{2}\frac{\Delta t^n}{\Delta t^{n-1}}\Big)p^{n-1} - \frac{1}{2}\frac{\Delta t^n}{\Delta t^{n-1}}p^{n-2}\Big) + \rho g_i, \\
&\text{(b)}\quad \frac{(\rho v_i)^{**} - (\rho v_i)^*}{\Delta t^n} = \frac{1}{2}\frac{\delta}{\delta x_i}\Big(\Big(1 + \frac{\Delta t^n}{\Delta t^{n-1}}\Big)p^{n-1} - \frac{\Delta t^n}{\Delta t^{n-1}}p^{n-2}\Big), \\
&\text{(c)}\quad \frac{(\rho v_i)^n - (\rho v_i)^{**}}{\Delta t^n} = -\frac{1}{2}\frac{\delta p^n}{\delta x_i}.
\end{aligned} \tag{3.1}$$

It should be noted that (3.1.a) and (3.1.b) have been obtained by subtracting or adding the term

$$\frac{1}{2}\frac{\Delta t^n}{\Delta t^{n-1}}(p^{n-1} - p^{n-2})$$

to the right sides of (2.3) and (2.4), respectively. The OS algorithm (3.1) thus solves the equation of the form (2.7), in addition to satisfying the continuity equation. But, here, the splitting error is one-order higher than (2.8). Indeed, adding (3.1.b) and (3.1.c) reads

$$v_i^n - v_i^* = -\frac{\Delta t^n}{2\rho}\frac{\delta}{\delta x_i}\Big(p^n - \Big(1+\frac{\Delta t^n}{\Delta t^{n-1}}\Big)p^{n-1} + \frac{\Delta t^n}{\Delta t^{n-1}}p^{n-2}\Big),$$

which is $\mathcal{O}((\Delta t^n)^3)$. Thus

$$H(v_i^n) - H(v_i^*) = \mathcal{O}((\Delta t^n)^3). \tag{3.2}$$

*Remark.* Algorithm (3.1) can be applied for $n \geq 2$. A special care would better be taken when $n = 1$. However, for simplicity, we utilize (2.3)-(2.5) for $n = 1$, which is the same as assuming $p^{-1} = p^0$.

## 4. Implementation details

For a purpose of numerical analysis, we implement algorithms (2.3)-(2.5) and (3.1) on a rectangular domain: $\Omega = (a_x, b_x) \times (a_y, b_y)$. Let $\{K_{\ell,m}\} \subset \Omega$ be the control volumes which differ for the pressure and different components of the velocity. (See §4.1.) We rewrite the algorithm (3.1) in its integral form:

$$\begin{aligned}
&\text{(a)}\quad \frac{1}{\Delta t^n}\int_{K_{\ell,m}} \rho v_i^*\,d\Omega + \frac{1}{2}\int_{\partial K_{\ell,m}} \rho v_i^* \mathbf{v}^* \cdot \mathbf{n}\,dS - \frac{1}{2}\int_{\partial K_{\ell,m}} \tau_{ij}^* I_j \cdot \mathbf{n}\,dS = Q_{\ell,m}^{n-1},\\
&\text{(b)}\quad \frac{1}{\Delta t^n}\int_{K_{\ell,m}} \rho v_i^{**}\,d\Omega = \frac{1}{\Delta t^n}\int_{K_{\ell,m}} \rho v_i^*\,d\Omega\\
&\qquad\qquad + \frac{1}{2}\int_{\partial K_{\ell,m}} \Big(\Big(1+\frac{\Delta t^n}{\Delta t^{n-1}}\Big)p^{n-1} - \frac{\Delta t^n}{\Delta t^{n-1}}p^{n-2}\Big) I_i \cdot \mathbf{n}\,dS,\\
&\text{(c)}\quad \frac{1}{\Delta t^n}\int_{K_{\ell,m}} \rho v_i^n\,d\Omega = \frac{1}{\Delta t^n}\int_{K_{\ell,m}} \rho v_i^{**}\,d\Omega - \frac{1}{2}\int_{\partial K_{\ell,m}} p^n I_i \cdot \mathbf{n}\,dS,
\end{aligned} \tag{4.1}$$

where

$$\begin{aligned}
Q_{\ell,m}^{n-1} = {} & \frac{1}{\Delta t^n}\int_{K_{\ell,m}} \rho v_i^{n-1}\,d\Omega - \frac{1}{2}\int_{\partial K_{\ell,m}} \rho v_i^{n-1}\mathbf{v}^{n-1}\cdot \mathbf{n}\,dS + \frac{1}{2}\int_{\partial K_{\ell,m}} \tau_{ij}^{n-1} I_j \cdot \mathbf{n}\,dS\\
& - \int_{\partial K_{\ell,m}} \Big(\Big(1+\frac{1}{2}\frac{\Delta t^n}{\Delta t^{n-1}}\Big)p^{n-1} - \frac{1}{2}\frac{\Delta t^n}{\Delta t^{n-1}}p^{n-2}\Big) I_i \cdot \mathbf{n}\,dS + \int_{K_{\ell,m}} \rho g_i\,\mathrm{d}\Omega.
\end{aligned}$$

In the following, we consider a staggered mesh and corresponding control volumes, an implicit method of velocity updates for (4.1), Stone's method, and a pressure solver.

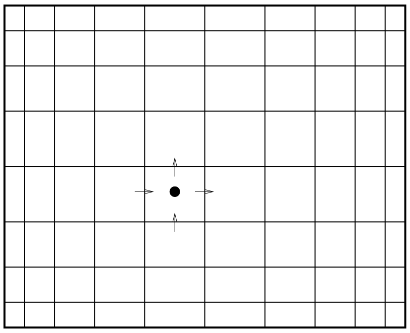

Figure 1: The mesh and staggered grid. The nodal points for scalar quantities are cell-centered (marked by $\bullet$) and those of velocities are located at centroids of cell boundaries.

| | Nodal points | Control volumes |
|---|---|---|
| $p$ | $(x_{c,\ell}, y_{c,m})$ | $(x_{\ell-1}, x_\ell) \times (y_{m-1}, y_m)$ |
| $v_1$ | $(x_\ell, y_{c,m})$ | $(x_{c,\ell}, x_{c,\ell+1}) \times (y_{m-1}, y_m)$ |
| $v_2$ | $(x_{c,\ell}, y_m)$ | $(x_{\ell-1}, x_\ell) \times (y_{c,m}, y_{c,m+1})$ |

Table 1: Nodal points and the corresponding control volumes $K_{\ell,m}$ for the pressure and the velocity components on the staggered grid mesh.

**4.1. Mesh generation and staggered control volumes**

The mesh is generated as follows. For the partitioning in the $x$-direction, let $c_x$ be the center of the interval $(a_x, b_x)$, i.e., $c_x = (a_x + b_x)/2$. The interval $(a_x, c_x)$ is divided into $k$ subintervals by choosing $k+1$ points: $a_x = x_0 < x_1 < \cdots < x_k = c_x$ satisfying

$$\Delta x_{\ell+1} = r\Delta x_\ell, \quad \ell = 1, \cdots, k-1, \tag{4.2}$$

where $\Delta x_\ell = x_\ell - x_{\ell-1}$ and $r \geq 1$. The other half, $(c_x, b_x)$, is partitioned by the points that are the reflection image of $\{x_0, x_1, \cdots, x_k\}$ with respect to $c_x = x_k$. So the grid points in the $x$-direction are $a_x = x_0 < x_1 < \cdots < x_{n_x} = b_x$, where $n_x = 2k$. See Figure 1. The same can be applied to the generation of $y$-directional grid points: $a_y = y_0 < y_1 < \cdots < y_{n_y} = b_y$. The cells composed by the grid lines are

$$(x_{\ell-1}, x_\ell) \times (y_{m-1}, y_m), \quad \ell = 1, \cdots, n_x, \;\; m = 1, \cdots, n_y.$$

Now we bulid the control volumes $K_{\ell,m}$ which differ for the pressure and the velocity components. Let

$$x_{c,\ell} = (x_{\ell-1} + x_\ell)/2, \;\; \ell = 1, \cdots, n_x, \quad y_{c,m} = (y_{m-1} + y_m)/2, \;\; m = 1, \cdots, n_y.$$

The nodal points and the corresponding control volumes for the pressure and the velocity components are given in Table 1 and depicted in Figure 2. The control

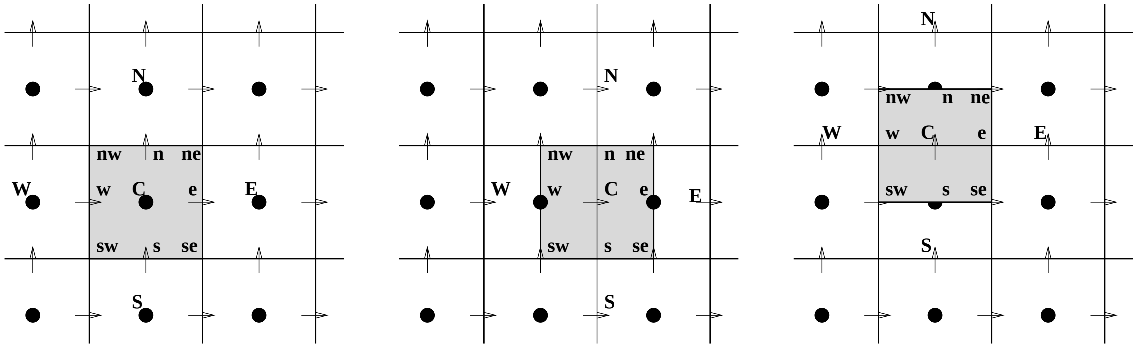

Figure 2: Control volumes $K_{\ell,m}$ for the pressure (left), the $x$-component of the velocity (center), and $y$-component of the velocity (right). In each figure, $C$ denotes the nodal point of the control volume. The capital letters $E$, $W$, $S$, and $N$ indicate the adjacent nodal points with which a row of the algebraic system is to be constructed for $C$.

volumes for the pressure coincide with the cells of the mesh. Note that nodal points for the velocity components may not be located at the center of the corresponding control volumes.

### 4.2. Implicit velocity update

Equation (4.1.a) includes nonlinear terms and its linearized problem results in an algebraic system of equations which requires an efficient solver, while equations (4.1.b) and (4.1.c) are easy to solve. As a part of iterative procedure for (4.1.a), we utilize the Picard iteration and a deferred correction (with blending) for the convection term as follows: Set $v_i^{*,0} = v_i^{n-1}$ and find $v_i^{*,k}$, $k \geq 1$, satisfying

$$\begin{aligned} \frac{1}{\Delta t^n} \int_{K_{\ell,m}} \rho v_i^{*,k}\, d\Omega + \frac{1}{2} \int_{\partial K_{\ell,m}} \rho v_i^{*,k,UDS} \mathbf{v}^{*,k-1} \cdot \mathbf{n}\, dS - \frac{1}{2} \int_{\partial K_{\ell,m}} \tau_{ij}^{*,k} I_j \cdot \mathbf{n}\, dS \\ = \beta \frac{1}{2} \int_{\partial K_{\ell,m}} \rho (v_i^{*,k-1,UDS} - v_i^{*,k-1,CDS}) \mathbf{v}^{*,k-1} \cdot \mathbf{n}\, dS + Q_{\ell,m}^{n-1}, \end{aligned} \tag{4.3}$$

where $0 \leq \beta \leq 1$ and $v_i^{*,k,UDS}$ and $v_i^{*,k,CDS}$ denote respectively the upwind and central approximations when evaluated on the interfaces of the control volumes. Other terms are approximated using central schemes. Note that the limit of the iterates, if any, solves (4.1.a) with the term $v_i^*$ in the convection part approximated by the upwind scheme ($\beta = 0$), the central scheme ($\beta = 1$), or their blending ($0 < \beta < 1$).

We consider FV methods for the convection and diffusion terms in (4.3); the other terms can be treated in a similar way. Let the four sides of the control volume $K_{\ell,m}$ be denoted by $S_e$, $S_w$, $S_s$, and $S_n$; see Figure 2. Then, the convection term in the left side of (4.3) reads

$$\begin{aligned}\int_{\partial K_{\ell,m}} \rho v_i^k \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS &= \int_{S_e} \rho v_i^k \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS + \int_{S_w} \rho v_i^k \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS \\ &\quad + \int_{S_n} \rho v_i^k \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS + \int_{S_s} \rho v_i^k \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS,\end{aligned} \tag{4.4}$$

where the superscripts $*$ and $UDS$ are dropped for convenience. For the first term in the right side of the above equation, we have

$$\int_{S_e} \rho v_i^k \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS \approx v_{i,e}^k \cdot m_e^{k-1},$$

where the subscript $e$ denotes the corresponding value at the point $e \in S_e$ and

$$m_e^{k-1} := \int_{S_e} \rho \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS \approx \rho_e v_{1,e}^{k-1}\cdot |S_e|,$$

where $|S_e|$, the length of the line segment $S_e$, is $(y_m - y_{m-1})$ for $i = 1$ and $(y_{c,m+1} - y_{c,m})$ for $i = 2$. When the point $e$ is not a nodal point, $\rho_e$ and $v_{1,e}^{k-1}$ are approximated by arithmetic averages. For example,

$$v_{1,e}^{k-1} \approx \begin{cases} (v_{1,C}^{k-1} + v_{1,E}^{k-1})/2, & \text{when } i = 1, \\ r v_{1,ne}^{k-1} + (1-r) v_{1,se}^{k-1}, & \text{when } i = 2, \end{cases}$$

where $r = (y_e - y_{se})/|S_e|$, where $y_e$ and $y_{se}$ are the $y$-components of the points $e$ and $se$, respectively. (Indeed, $(y_e - y_{se})$ is $(y_m - y_{m-1})/2$ for $i = 1$ and $(y_m - y_{c,m})$ for $i = 2$.) The same average technique can be applied for the central approximation $v_{i,e}^{*,k-1,CDS}$ when evaluating the right side of (4.3). The upwind approximation for $v_{i,e}^k$, $v_{i,e}^{k,UDS}$, is computed as in

$$v_{i,e}^{k,UDS}\cdot m_e^{k-1} \approx v_{i,C}^k \cdot \max(m_e^{k-1},0) + v_{i,E}^k \cdot \min(m_e^{k-1},0).$$

After approximating the other terms in the right side of (4.4) similarly, we have

$$\begin{aligned}\int_{\partial K_{\ell,m}} \rho v_i^{k,UDS} \mathbf{v}^{k-1}\cdot\mathbf{n}\,dS &\approx v_{i,E}^k \cdot \min(m_e^{k-1},0) + v_{i,W}^k \cdot \min(m_w^{k-1},0) \\ &\quad + v_{i,N}^k \cdot \min(m_n^{k-1},0) + v_{i,S}^k \cdot \min(m_s^{k-1},0) - v_{i,C}^k \cdot M_C,\end{aligned} \tag{4.5}$$

where

$$M_C = \min(m_e^{k-1},0) + \min(m_w^{k-1},0) + \min(m_n^{k-1},0) + \min(m_s^{k-1},0).$$

Note that equation (4.5) results in a five-point stencil for the algebraic system of (4.3).

Now, we consider the diffusion term

$$\int_{\partial K_{\ell,m}} \tau_{ij} I_j \cdot \mathbf{n}\,dS,$$

where we have again omitted the superscripts on $\tau_{ij}$ for convenience. For a simpler presentation, we approximate $\tau_{ij} \approx \mu(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})$. Then,

$$\begin{aligned}
\int_{S_e} \tau_{ij} I_j \cdot \mathbf{n}\, dS &= \int_{S_e} \tau_{i1}\, dS \approx (\tau_{i1})_e \cdot |S_e|, \\
\int_{S_w} \tau_{ij} I_j \cdot \mathbf{n}\, dS &= \int_{S_w} -\tau_{i1}\, dS \approx -(\tau_{i1})_w \cdot |S_w|, \\
\int_{S_n} \tau_{ij} I_j \cdot \mathbf{n}\, dS &= \int_{S_n} \tau_{i2}\, dS \approx (\tau_{i2})_n \cdot |S_n|, \\
\int_{S_s} \tau_{ij} I_j \cdot \mathbf{n}\, dS &= \int_{S_s} -\tau_{i2}\, dS \approx -(\tau_{i2})_s \cdot |S_s|,
\end{aligned}$$

Here we apply central approximations for the evaluation of $\tau_{ij}$. For example,

$$\begin{aligned}
(\tau_{11})_e &= 2\mu_e \Big(\frac{\partial v_1}{\partial x_1}\Big)_e \approx 2\mu_e \frac{v_{1,E} - v_{1,C}}{x_E - x_C}, \\
(\tau_{21})_e &= \mu_e \Big(\frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2}\Big)_e \approx \mu_e \Big(\frac{v_{2,E} - v_{2,C}}{x_E - x_C} + \frac{v_{1,ne} - v_{1,se}}{y_{ne} - y_{se}}\Big),
\end{aligned}$$

where $\mu_e$ in the first equation is a nodal value, while that in the second equation is an average of $\mu$'s obtained from surrounding four cells. Again, we can see that the diffusion term contributes to the five-point stencil.

### 4.3. Solver for the linear system

When every term in (4.3) is approximated by the FV method, each of the equations $(i = 1, 2)$ results in an algebraic system of the form

$$A\mathbf{u} = \mathbf{b}. \tag{4.6}$$

To solve it, we adopt Stone's *strongly implicit procedure* (SIP) [4]. For completeness of the paper, we briefly describe the basic idea of SIP.

As for other iterative methods, SIP is based on a regular splitting, $A = M - N$, with $M$ being an incomplete $LU$ (ILU) factorization;

$$M = L_I U_I = A + N, \tag{4.7}$$

where $L_I$ and $U_I$ are respectively the lower and upper triangular components of the ILU factorization of $A$, where the entries of the main diagonal of $U_I$ are all one. Note that the iteration corresponding to the splitting (4.7) is formulated as

$$L_I U_I \mathbf{u}^k = N\mathbf{u}^{k-1} + \mathbf{b}, \tag{4.8}$$

or, equivalently,

$$\begin{aligned}
&\text{(a)} \quad \mathbf{r}^{k-1} = \mathbf{b} - A\mathbf{u}^{k-1}, \\
&\text{(b)} \quad L_I U_I \boldsymbol{\delta}^{k-1} = \mathbf{r}^{k-1}, \\
&\text{(c)} \quad \mathbf{u}^k = \mathbf{u}^{k-1} + \boldsymbol{\delta}^{k-1}.
\end{aligned} \tag{4.9}$$

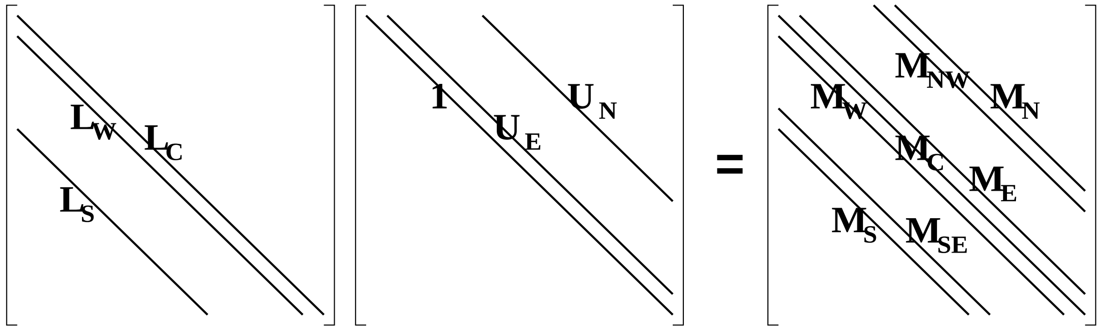


Figure 3: Systematic presentation of $L_I U_I = M$. The subscripts $S$, $W$, $E$, $N$, and $C$ denote respectively south, west, east, north, and center. Note that diagonals of $M$ marked by subscripts $SE$ and $NW$ are not found in $A$.

To make the iteration (4.8) converge fast, we may have to choose elements of $L_I$ and $U_I$ in a way that $N$ is as small as possible. For a 2D problem in a rectangular mesh where the grid points are ordered in the row-wise manner, the ILU factorization is in the form as in Figure 3 and the row of $M$ corresponding to the $(\ell, m)$-th grid point is given as

$$\begin{aligned} M_S^{\ell,m} &= L_S^{\ell,m}, \\ M_{SE}^{\ell,m} &= L_S^{\ell,m} U_E^{\ell,m-1}, \\ M_W^{\ell,m} &= L_W^{\ell,m}, \\ M_C^{\ell,m} &= L_S^{\ell,m} U_N^{\ell,m-1} + L_W^{\ell,m} U_E^{\ell-1,m} + L_C^{\ell,m}, \\ M_E^{\ell,m} &= L_C^{\ell,m} U_E^{\ell,m}, \\ M_{NW}^{\ell,m} &= L_W^{\ell,m} U_N^{\ell-1,m}, \\ M_N^{\ell,m} &= L_C^{\ell,m} U_N^{\ell,m}. \end{aligned} \tag{4.10}$$

The $(\ell, m)$-th component of $N\mathbf{u}$ is

$$\begin{aligned} (N\mathbf{u})_{\ell,m} = {} & N_C^{\ell,m} u_{\ell,m} + N_S^{\ell,m} u_{\ell,m-1} + N_W^{\ell,m} u_{\ell-1,m} + N_E^{\ell,m} u_{\ell+1,m} + N_N^{\ell,m} u_{\ell,m+1} \\ & + M_{SE}^{\ell,m} u_{\ell+1,m-1} + M_{NW}^{\ell,m} u_{\ell-1,m+1}. \end{aligned} \tag{4.11}$$

By utilizing the approximations

$$\begin{aligned} u_{\ell+1,m-1} &\approx \alpha(u_{\ell,m-1} + u_{\ell+1,m} - u_{\ell,m}), \\ u_{\ell-1,m+1} &\approx \alpha(u_{\ell,m+1} + u_{\ell-1,m} - u_{\ell,m}), \end{aligned} \qquad 0 < \alpha < 1, \tag{4.12}$$

we can rewrite (4.11) as

$$\begin{aligned} (N\mathbf{u})_{\ell,m} \approx {} & (N_C^{\ell,m} - \alpha M_{SE}^{\ell,m} - \alpha M_{NW}^{\ell,m}) u_{\ell,m} \\ & + (N_S^{\ell,m} + \alpha M_{SE}^{\ell,m}) u_{\ell,m-1} + (N_W^{\ell,m} + \alpha M_{NW}^{\ell,m}) u_{\ell-1,m} \\ & + (N_E^{\ell,m} + \alpha M_{SE}^{\ell,m}) u_{\ell+1,m} + (N_N^{\ell,m} + \alpha M_{NW}^{\ell,m}) u_{\ell,m+1}. \end{aligned} \tag{4.13}$$

Set each of coefficients in the right-side of (4.13) to be zero. Then, it follows from (4.10) that entries of $N$ are presented by those of $L_I$ and $U_I$:

$$\begin{aligned} N_S^{\ell,m} &= -\alpha M_{SE}^{\ell,m} = -\alpha L_S^{\ell,m} U_E^{\ell,m-1}, \\ N_W^{\ell,m} &= -\alpha M_{NW}^{\ell,m} = -\alpha L_W^{\ell,m} U_N^{\ell-1,m}, \\ N_C^{\ell,m} &= \alpha(M_{SE}^{\ell,m} + M_{NW}^{\ell,m}) = \alpha(L_S^{\ell,m} U_E^{\ell,m-1} + L_W^{\ell,m} U_N^{\ell-1,m}), \\ N_E^{\ell,m} &= -\alpha M_{SE}^{\ell,m} = -\alpha L_S^{\ell,m} U_E^{\ell,m-1}, \\ N_N^{\ell,m} &= -\alpha M_{NW}^{\ell,m} = -\alpha L_W^{\ell,m} U_N^{\ell-1,m}. \end{aligned} \tag{4.14}$$

Now, utilizing $M = A + N$, (4.10), and (4.14), one can obtain Stone's SIP [4]:

$$\begin{aligned} L_S^{\ell,m} &= A_S^{\ell,m}/(1 + \alpha U_E^{\ell,m-1}), \\ L_W^{\ell,m} &= A_W^{\ell,m}/(1 + \alpha U_N^{\ell-1,m}), \\ L_C^{\ell,m} &= A_C^{\ell,m} + \alpha(L_S^{\ell,m} U_E^{\ell,m-1} + L_W^{\ell,m} U_N^{\ell-1,m}) \\ &\quad - L_S^{\ell,m} U_N^{\ell,m-1} - L_W^{\ell,m} U_E^{\ell-1,m}, \\ U_E^{\ell,m} &= (A_E^{\ell,m} - \alpha L_S^{\ell,m} U_E^{\ell,m-1})/L_C^{\ell,m}, \\ U_N^{\ell,m} &= (A_N^{\ell,m} - \alpha L_W^{\ell,m} U_N^{\ell-1,m})/L_C^{\ell,m}. \end{aligned} \tag{4.15}$$

*Remark.* The approximations in (4.12) are second-order accurate when $\alpha = 1$. But the algorithm (4.15) can be unstable for the case; the parameter $\alpha$ is often chosen between 0.92 and 0.96 [3]. Entries of $L_I$ and $U_I$ used in (4.15) whose indices are outside the index boundaries should be set zero.

### 4.4. Pressure solver

The Poisson equation (2.6) reads, in its integral form,

$$\int_{\partial K_{\ell,m}} \nabla p \cdot \mathbf{n}\, dS = \frac{2}{\Delta t^n} \int_{\partial K_{\ell,m}} \rho \mathbf{v}^{**} \cdot \mathbf{n}\, dS. \tag{4.16}$$

No flux condition is often imposed for the boundary condition. When using the second-order FV method, we can obtain the five-point scheme as follows:

$$\begin{aligned} \int_{\partial K_{\ell,m}} \nabla p \cdot \mathbf{n}\, dS \;\approx\;& (y_m - y_{m-1})\Big(\frac{p_E - p_C}{x_{c,m+1} - x_{c,m}} - \frac{p_C - p_W}{x_{c,m} - x_{c,m-1}}\Big) \\ +\;& (x_\ell - x_{\ell-1})\Big(\frac{p_N - p_C}{y_{c,\ell+1} - y_{c,\ell}} - \frac{p_C - p_S}{y_{c,\ell} - y_{c,\ell-1}}\Big). \end{aligned}$$

The right side of (4.16) can be easily integrated without averaging the velocity components.

The resulting algebraic system can be written as

$$A_p \mathbf{p} = \mathbf{k}. \tag{4.17}$$

The above algebraic system can be solved by employing SIP as in §4.3. Since $A_p$ is symmetric and positive definite (when the pressure is fixed at a nodal point on the boundary), we may explore more efficient algorithms which possibly take advantage of such nice properties of algebraic systems. As an alternative to SIP, we adopt a preconditioned conjugate gradient (PCG) algorithm of which the preconditioner $M$ is either the Stone's decomposition (4.15) or another ILU factorization:

select $\mathbf{p}_0$, $\varepsilon$;
$\mathbf{r}_0 = \mathbf{k} - A_p\mathbf{p}_0$; $M\mathbf{z}_0 = \mathbf{r}_0$;
$\mathbf{q}_0 = \mathbf{z}_0$; $\rho_0 = \mathbf{z}_0 \cdot \mathbf{r}_0$;
do $k = 0, 1, \cdots$
    $\alpha_k = \rho_k / \mathbf{q}_k \cdot A_p\mathbf{q}_k$;
    $\mathbf{p}_{k+1} = \mathbf{p}_k + \alpha_k\mathbf{q}_k$;
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A_p\mathbf{q}_k$;
    if $\| \mathbf{r}_{k+1} \|_2 \leq \varepsilon$, stop;
    $M\mathbf{z}_{k+1} = \mathbf{r}_{k+1}$;
    $\rho_{k+1} = \mathbf{z}_{k+1} \cdot \mathbf{r}_{k+1}$;
    $\beta_k = \rho_{k+1}/\rho_k$;
    $\mathbf{q}_{k+1} = \mathbf{z}_{k+1} + \beta_k\mathbf{q}_k$;
enddo

The above PCG is the same as applying the CG method to the following preconditioned system:

$$M^{-1}A_p\mathbf{p} = M^{-1}\mathbf{k}, \tag{4.18}$$

where $M$ is the preconditioner. The smaller the condition number of $M^{-1}A_p$ is, the faster the PCG converges.

## 5. Numerical experiments

## 6. Conclusions

## References

[1] H. Choi and P. Moin, *Effects of the computational time step on numerical solutions of turbulent flow*, J. Comput. Phys., 113 (1994), pp. 1–4.

[2] J. Douglas, Jr. and S. Kim, *On accuracy of alternating direction implicit methods for parabolic equations.* Preprint, 1999.

[3] J. Ferziger and M. Peric, *Computational methods for fluid dynamics, 2nd Edition*, Springer-Verlag, Berlin, Heidelberg, New York, 1999.

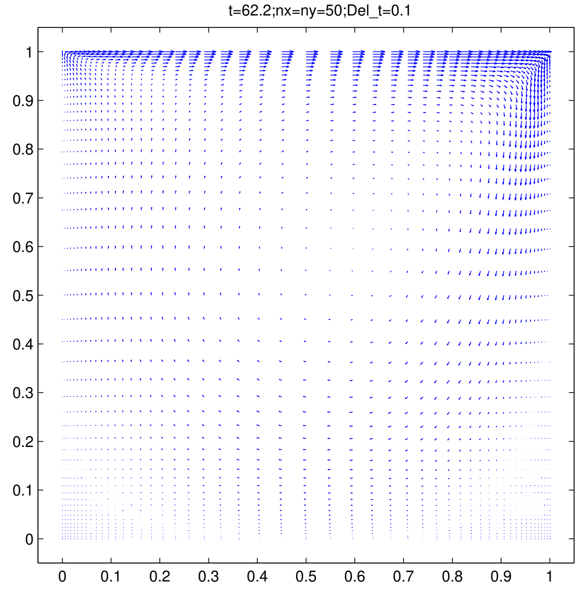


Figure 4: The velocity field for a lid-driven cavity when $t = 62.2$. The lid velocity $\mathbf{v}_L = (1,0)$, the density $\rho = 1$, and the viscosity $\mu = 0.001$. Set $\Omega = (0,1)^2$, $n_x = n_y = 50$, $\Delta t = 0.1$ and the grid-ratio in (4.2) $r = 1.1$.

[4] H. Stone, *Iterative solution of implicit approximations of multidimensional partial differential equations*, SIAM J. Numer. Anal., 5 (1968), pp. 530–558.