Time Stepping for Vectorial Operator Splitting

Rossitza S. Marinova

Department of Mathematical and Computing Sciences, Concordia University College of Alberta, 7128 Ada Boulevard, Edmonton, AB, T5B 4E4, CANADA
Phone: 1 (780) 378-8430; Fax: 1 (780) 474 1933

Raymond J. Spiteri

Department of Computer Science, University of Saskatchewan
176 Thorvaldson Bldg, 110 Science Place, Saskatoon, SK S7N 5C9, CANADA

Eddy Essien

Department of Mathematical and Computing Sciences, Concordia University College of Alberta, 7128 Ada Boulevard, Edmonton, AB, T5B 4E4, CANADA

Abstract

We present a fully implicit finite difference method for the unsteady incompressible Navier–Stokes equations. It is based on the one-step θ-method for discretization in time and a special coordinate splitting (called vectorial operator splitting) for efficiently solving the nonlinear stationary problems for the solution at each new time level. The resulting system is solved in a fully coupled approach that does not require a boundary condition for the pressure. A staggered arrangement of velocity and pressure on a structured Cartesian grid combined with the fully implicit treatment of the boundary conditions help to preserve properties of the differential operators and thus lead to excellent stability of the overall algorithm. The convergence properties of the method are confirmed via numerical experiments.

Key words: Unsteady incompressible Navier-Stokes, Implicit method, Stability
1991 MSC: 76D05, 34A09, 65M12

1. Introduction

Fluid fows with high Reynolds numbers or complex geometries are challenging to simulate and of great interest to industry; hence there is significant demand for robust and stable algorithms and software, perhaps even at the expense of a moderately increased computational cost. Fully implicit time-stepping methods are generally more robust and stable than explicit and semi-explicit methods. Therefore, as suggested in [12], fully implicit methods should be further investigated and developed.

* Corresponding author
Email addresses: rmarinova@math.concordia.ab.ca (Rossitza S. Marinova), spiteri@cs.usask.ca (Raymond J. Spiteri), eessien@csa.concordia.ab.ca (Eddy Essien).
URL: www.math.concordia.ab.ca/marinova/ (Rossitza S. Marinova).

The most popular time-stepping methods for the Navier–Stokes equations are the so-called *projection* or *operator-splitting* methods (e.g., fractional step or pressure-correction methods) and are not fully implicit; see [11] and [12]. Decoupling the velocity and pressure reduces the system into simpler sub-problems, but the choice of boundary conditions for the pressure in these procedures is problematic. Moreover, the explicit element introduced by this decoupling requires small time steps to maintain stability. Although operator-splitting methods can work well, they must be used with care in terms of how well the overall solution algorithm behaves. They are usually not suitable for flows with high Reynolds numbers or long simulation times because the requirement of a small time step size.

After discretization in space and time, a fully implicit approach leads to a system of nonlinear equations that may be singular [12]. For this reason, special spatial discretization or stabilization techniques are needed. Strongly coupled solution strategies can improve the stability considerably; however, they also need to be able to handle large nonlinear algebraic systems. Direct solvers can be used for the solution of the linear systems of equations that arise in this process, but they typically require large amounts of memory, and despite increases in computational power, are still not feasible for large-scale computations, particularly for unsteady 3D problems. Hence iterative solvers are the preferred choice for the solution of these systems. Coordinate splitting and multigrid are two powerful methods for solving such systems.

In this paper, we use the linear two-layer (one-step) scheme, which is also known as the θ-method, for the temporal discretization; see e.g., [20]. We employ finite difference approximations in space that utilize computer resources effectively and hence enable efficient computations. For the solution of the nonlinear stationary problems that arise after the temporal discretization, we use coordinate splitting based on the Douglas–Rachford scheme [8]. The splitting procedure is constructed in a way that leaves the system coupled to allow the satisfaction of the boundary conditions but avoids the introduction of artificial boundary conditions for the pressure.

The paper is organized as follows. The problem is formulated in the next section. The time discretization is presented in Section 3, including a discussion on the singularity of direct fully implicit schemes. Issues associated with the solution of the stationary problems that need to be solved after discretization in time are discussed in Section 4. These include requirements to be satisfied by the differential problem and the choice of discretization in space as well as the coordinate splitting method. Finally, numerical results are presented in Section 5 and conclusions in Section 6.

2. Problem Statement

2.1. *Incompressible Navier–Stokes Equations*

We consider the multi-dimensional incompressible Navier–Stokes equations in dimensionless form

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nu \nabla^2 \boldsymbol{u} - \nabla p + \boldsymbol{g} \tag{1}$$

coupled with the continuity equation, also called the incompressibility condition,

$$\operatorname{div} \boldsymbol{u} = \nabla \cdot \boldsymbol{u} = 0 \tag{2}$$

on $\Omega \times (0, T)$, where Ω is a bounded, compact (spatial) domain with a piecewise smooth boundary $\partial\Omega$. Here $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t) = (u, v, w)$ is the fluid velocity at position $\boldsymbol{x} \in \Omega$ and time $t \in (0, T)$ for given T. Also $p = p(\boldsymbol{x}, t)$ is the fluid kinematic pressure, $\nu = 1/\operatorname{Re}$ is the kinematic viscosity, where Re is the Reynolds number, $\boldsymbol{g}$ is an external force, ∇ is the gradient operator, and ∇^2 is the Laplacian operator.

We can write the momentum equation (1) in the following form,

$$\frac{\partial \boldsymbol{u}}{\partial t} + (C + L)\boldsymbol{u} + \nabla p = \boldsymbol{g}, \tag{3}$$

where $C = \boldsymbol{u} \cdot \nabla$ is the nonlinear convection operator and $L = -\nu \nabla^2$ is the linear viscosity operator.

Taking into account the incompressibility constraint (2), the nonlinear convective term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ in equation (1) can be written in the equivalent form

$$\begin{aligned} C\boldsymbol{u} &= (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \tfrac{1}{2}\boldsymbol{u}(\nabla \cdot \boldsymbol{u}) \\ &= \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) - \tfrac{1}{2}\boldsymbol{u}(\nabla \cdot \boldsymbol{u}) \\ &= \tfrac{1}{2}[\nabla \cdot (\boldsymbol{u}\boldsymbol{u}) + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}], \end{aligned} \tag{4}$$

which is skew-symmetric. The advantage of using the skew-symmetric form (4) is that it conserves both the square of velocity as well as the kinetic energy, whereas the divergence form $\nabla \cdot (\boldsymbol{u}\boldsymbol{u})$ conserves only the kinetic energy, and the (original) non-divergence form $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ conserves neither the square of the velocity nor the kinetic energy.

2.2. *Initial and Boundary Conditions*

In our investigations, we assume an initial condition

$$\boldsymbol{u}\big|_{t=0} = \boldsymbol{u}_0(\boldsymbol{x}), \tag{5}$$

that is divergence-free, i.e., $\nabla \cdot \boldsymbol{u}_0 = 0$, and the following boundary conditions

$$\boldsymbol{u}\big|_{\partial\Omega} = \boldsymbol{u}_b(t),$$

i.e., the velocity is prescribed at the boundary.

Remark: In order to avoid singularities, the initial and boundary conditions are assumed to agree at $t = 0$ and $\boldsymbol{x} \in \partial\Omega$.

The incompressible Navier–Stokes equations can be classified as partial differential-algebraic equations, e.g., [2]. The challenges in their numerical solution are well known; they are connected with the fact that the Navier–Stokes equations are not an evolutionary system of Cauchy–Kovalevskaya type and that the pressure is an implicit function responsible for the satisfaction of the continuity equation. Furthermore, no boundary conditions on the pressure can be imposed on rigid boundaries. This creates formidable obstacles for the construction of fully implicit schemes.

2.3. *Balanced Pressure Equation*

We now turn to the question of how to construct a robust and stable numerical method, even perhaps at the cost of a moderate increase in computational effort. For the reasons outlined above, we require an implicit time discretization procedure that also preserves the coupling of the velocity and pressure.

A formulation with a pressure equation is preferable compared to one with the continuity equation because we can construct a solver for the resulting nonlinear stationary problem that is not only robust with respect to the physical and numerical parameters but also computationally efficient. For this reason, we use a special pressure equation, which is equivalent to the standard Poisson equation for pressure on a differential level. A similar form for the pressure equation is presented in [10]. In that paper, the Laplacian of the pressure is balanced using the divergence of the momentum equations

$$\nabla \cdot \boldsymbol{u} = \epsilon\,[\nabla^2 p + \nabla \cdot (C\boldsymbol{u} - \boldsymbol{g})]. \tag{6}$$

On the discrete level, the right-hand side of equation (6) does not necessarily vanish. Here, ϵ is a balancing parameter that can be varied. As noted in [10], this parameter is not related to the time step Δt.

The pressure equation used in [15] for the solution of the steady-state problem is similar to (6). The balancing coefficient in the pressure equation is equal to the viscosity ν. As well, a balancing coefficient γ for the term $\nabla \cdot \boldsymbol{u}$ is used as in [16], so that the modified pressure equation becomes

$$\gamma(\nabla \cdot \boldsymbol{u}) = \nu[\nabla^2 p + \nabla \cdot (C\boldsymbol{u} - \boldsymbol{g})]. \tag{7}$$

We find that using a balanced pressure equation such as equation (7) in combination with conservative difference approximations (see Section 4.1) improves the convergence of the difference problem considerably. Conservative discretizations are not considered in [10], nor is a splitting procedure used to improve the efficiency of the solver for the linear systems of equations.

It should be mentioned that the formulation of the problem (1)–(2) is equivalent to the formulation with the pressure equation (1), (7) if and only if the continuity equation is satisfied on the boundary, namely the following boundary conditions are satisfied

$$\boldsymbol{u}\big|_{\partial\Omega} = \boldsymbol{u}_b(t), \quad \nabla\cdot\boldsymbol{u}\big|_{\partial\Omega} = 0. \tag{8}$$

3. Time Discretization

3.1. *Momentum Equation*

The use of a fully implicit approach for time stepping in the momentum equation (3), such as the θ-scheme as applied to stiff systems with $0 < \theta \leq 1$, leads to the solution of the following nonlinear stationary equation at each time step

$$\begin{aligned}\frac{\boldsymbol{u}(\boldsymbol{x},t+\Delta t)-\boldsymbol{u}(\boldsymbol{x},t)}{\Delta t} &+ \theta[(C+L)\,\boldsymbol{u}(\boldsymbol{x},t+\Delta t)+\nabla p(\boldsymbol{x},t+\Delta t)] \\ &+ (1-\theta)[(C+L)\,\boldsymbol{u}(\boldsymbol{x},t)+\nabla p(\boldsymbol{x},t)] \\ &= \theta\,\boldsymbol{g}(\boldsymbol{x},t+\Delta t)+(1-\theta)\boldsymbol{g}(\boldsymbol{x},t),\end{aligned} \tag{9}$$

where $\Delta t = t^{n+1} - t^n$ is the time step and θ is the implicitness parameter. The time discretization (9) for the momentum equation is the second-order Crank–Nicolson method if $\theta = 1/2$, the backward Euler method for $\theta = 1$, and the (explicit) forward Euler method when $\theta = 0$.

In this work, only implicit schemes are considered. Therefore, $\theta \neq 0$, and equation (9) can be written in the following form

$$\frac{1}{\theta\Delta t}\,\boldsymbol{u}(\boldsymbol{x},t+\Delta t)+(C+L)\,\boldsymbol{u}(\boldsymbol{x},t+\Delta t)+\nabla p(\boldsymbol{x},t+\Delta t) = F_{\boldsymbol{u}}(\boldsymbol{x},t), \tag{10}$$

where

$$F_{\boldsymbol{u}}(\boldsymbol{x},t) = \frac{1}{\theta\Delta t}\boldsymbol{u}(\boldsymbol{x},t)+\boldsymbol{g}(\boldsymbol{x},t+\Delta t)+\frac{1-\theta}{\theta}\left[\boldsymbol{g}(\boldsymbol{x},t)-(C+L)\,\boldsymbol{u}(\boldsymbol{x},t)-\nabla p(\boldsymbol{x},t)\right].$$

The stability of the θ-method depends on θ. In the particular case of linear constant-coefficient stiff systems, the constraint for unconditional stability is $1/2 \leq \theta \leq 1$.

3.2. *Pressure Equation*

In addition to using equation (7) in place of the continuity equation (2), a pressure equation can also be derived from the momentum equation discretized in time. We consider the following two approaches for discretizing pressure.

3.2.1. *Pressure equation obtained from time-discretized momentum equation (10)*

To derive a pressure equation, we apply the divergence operator to equation (10) and note that the continuity equation $\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t+\Delta t) = 0$ must be satisfied as well as $\nabla\cdot(L\,\boldsymbol{u}) = L\,(\nabla\cdot\boldsymbol{u}) = L\,(0) = 0$, a property that stems from the linearity of the operator $L = -\nu\nabla^2$. It follows that

$$\frac{1}{\theta\Delta t}[\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t+\Delta t)]+\nabla\cdot[C\,\boldsymbol{u}(\boldsymbol{x},t+\Delta t)]+\nabla^2 p(\boldsymbol{x},t+\Delta t) = \nabla\cdot[F_{\boldsymbol{u}}(\boldsymbol{x},t)]. \tag{11}$$

After multiplying equation (11) by ν, we obtain the following pressure equation

$$L\,p(\boldsymbol{x},t+\Delta t)-\frac{\nu}{\theta\Delta t}[\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t+\Delta t)] = \nu\nabla\cdot[C\,\boldsymbol{u}(\boldsymbol{x},t+\Delta t)]-\nu\nabla\cdot[F_{\boldsymbol{u}}(\boldsymbol{x},t)]. \tag{12}$$

We introduce a coefficient $\gamma = -\frac{\nu}{\theta\Delta t}$ that controls the stability of the system; so the pressure equation becomes

$$L\,p(\boldsymbol{x},t+\Delta t)+\gamma[\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t+\Delta t)] = F_p(\boldsymbol{x},t), \qquad F_p(\boldsymbol{x},t) = \nu\nabla\cdot[C\,\boldsymbol{u}(\boldsymbol{x},t+\Delta t)-F_{\boldsymbol{u}}(\boldsymbol{x},t)]. \tag{13}$$

3.2.2. *Pressure equation obtained from continuous momentum (7)*

As an alternative, we also consider equation (7), which after discretization in time is

$$L\ p(\boldsymbol{x}, t+\Delta t) + \gamma[\nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t+\Delta t)] = F_p(\boldsymbol{x}, t), \qquad F_p(\boldsymbol{x}, t) = \nu \nabla \cdot [C\ \boldsymbol{u}(\boldsymbol{x}, t+\Delta t) - \boldsymbol{g}(\boldsymbol{x}, t)], \tag{14}$$

where γ is chosen to enhance convergence and stability. The choice $\gamma = 1$ works well in practice.

3.3. *Boundary Conditions*

The boundary conditions (8) must be also discretized in time; they then take the form

$$\boldsymbol{u}(\boldsymbol{x}, t+\Delta t)\big|_{\partial\Omega} = \boldsymbol{u}_b(t+\Delta t), \quad \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t+\Delta t)\big|_{\partial\Omega} = 0. \tag{15}$$

3.4. *Nonlinear Stationary Problems*

Finally, after the discretization in time, in order to obtain the solution at the next time level, it is necessary to solve the nonlinear stationary problem (10), (15), (5) evolved to the current time level, and (13) (or (14)). Although equations (13) and (14) are different, we use the generic variables γ and F_p in the remainder of the paper; which definition is being used should be clear from the context.

The system of equations (10), (13) can be written in a matrix form as

$$\begin{bmatrix} L + C + \dfrac{1}{\theta \Delta t} & \text{grad} \\ \gamma\ \text{div} & L \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} F_{\boldsymbol{u}} \\ F_p \end{bmatrix}, \tag{16}$$

where $\boldsymbol{u}$ and p are evaluated at $\boldsymbol{x}$ and $t + \Delta t$.

Discretization in space of (16), with appropriate boundary conditions added, leads to the solution of a nonlinear algebraic system that must be solved iteratively. In the process of solving the nonlinear system, a system of linear equations must be solved at each iteration. In the case of the fully implicit approach for time discretization, the matrix obtained after spatial discretization of the system is not symmetric positive definite, in contrast to systems arising from an explicit treatment of the convective term. Therefore, the choice of discretization in space is crucial for the stability of the scheme. Because the equations to be solved are conservation laws, it is highly desirable that the numerical scheme should also preserve these laws [17].

Stabilization techniques are usually based on perturbed versions of the continuity equation. There exist many variations of pressure stabilization techniques [3]; see [12] and [11] for reviews. Although not originally derived as stabilization methods, the artificial incompressibility method [4] and the penalty method [18] can be also placed into this category. Because they are usually used with finite element discretizations, these methods aim at stabilizing pressure oscillations and allowing standard grids and elements.

Most popular time-stepping methods, including fully implicit methods such as the backward Euler method, typically do not solve the resulting system in a fully coupled manner. Velocity and pressure are usually decoupled, and this requires imposition of pressure boundary conditions. Solving the system (16) in a fully coupled approach is preferred because it preserves the implicitness of the scheme, but such solvers require further development. Large linear systems must be solved as part of this process. Direct linear solvers, such as Gaussian elimination, are generally not efficient for 3D problems. Iterative strategies, such as BiCGStab and GMRES, combined with suitable preconditioners, can be effectively used for solving the linear systems of equations that arise. Coordinate splitting is also effective because it can reduce the number of operations for solving these linear systems by an order of magnitude [8,14].

4. Difference problem

4.1. *Analytical properties*

No matter what iterative strategy is used, in order to create a difference scheme that solves the problem accurately and efficiently, it is highly desirable for the scheme to satisfy the following analytical properties:

(i) *Conservation properties*

Following [17], we call an operator $T[\varphi]$ *conservative* if it can be written in divergence form $T[\cdot] = \nabla \cdot (S[\cdot])$, where $\varphi(\boldsymbol{x}, t + \Delta t)$ is a function, such as a velocity component, kinetic energy, etc., and S is an operator that can be used to express the system of equations in an equivalent form on the continuous level provided the continuity equation is satisfied. In general, however, these forms are not equivalent on the discrete level.

Assuming the continuity equation (2) is satisfied, it is known that [17]

(a) The *mass* is conserved *a priori* for the exact solution because the continuity equation (2) appears in divergence form.

(b) The *momentum* is conserved *a priori* for the exact solution; the pressure and viscous terms are conservative *a priori*; the convective term is also conservative *a priori*.

(c) The *square of a velocity component* φ^2 is of importance in case of coordinate splitting. If the convective term is written in a skew-symmetric form (4), then it conserves φ^2. For instance, in direction x

$$\varphi\ C_x[\varphi] = \frac{\varphi}{2}\left[\frac{\partial(\varphi u)}{\partial x} + u\frac{\partial \varphi}{\partial x}\right] = \frac{1}{2}\frac{\partial(\varphi^2 u)}{\partial x}\cdot$$

The convective term in a skew-symmetric form is conservative *a priori*, whereas the pressure and viscous term are not conservative.

(d) The *kinetic energy* $K \overset{\text{def}}{=} \frac{1}{2}(u^2+v^2+w^2)$: The skew-symmetric convective term is energy conservative, the pressure term is energy conservative, whereas the viscous term is not energy conservative.

In addition to conservation we also ensure that the scheme satisfies the following properties:

(ii) *Compatibility for Poisson's equation for pressure.*

(iii) *Commutativity* of the Laplacian and divergence operators.

(iv) *Consistency* between gradient and divergence operators.

(v) A velocity field that is *solenoidal* at each time step; i.e., $\nabla \cdot \boldsymbol{u} = 0$.

Satisfaction of properties (i)–(v) leads to excellent l^2-stability of the scheme [16].

4.2. *Coordinate Operator Splitting*

We consider a flow in a region with rectilinear boundaries in Cartesian coordinates. The boundary conditions derived from (15) in 3D take the form

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{(x=c,y,z,t+\Delta t)} &= \psi_1(y, z, t+\Delta t), \\ \left.\frac{\partial v}{\partial y}\right|_{(x,y=c,z,t+\Delta t)} &= \psi_2(x, z, t+\Delta t), \\ \left.\frac{\partial w}{\partial z}\right|_{(x,y,z=c,t+\Delta t)} &= \psi_3(x, y, t+\Delta t), \end{aligned}$$

where $(x = c, y, z, t{+}\Delta t)$, $(x, y = c, z, t{+}\Delta t)$, and $(x, y, z = c, t{+}\Delta t)$ represent boundary points, c is a generic constant meant to denote the constant co-ordinate values on the boundary, and $\psi_i,\ i = 1, 2, 3$ are known functions. We keep the coupling between the pressure and the respective velocity component through the boundary conditions at each fractional step. This allows us to construct efficient implicit splitting schemes.

The stationary system of equations (16) can be written in the following general form

$$A\ \boldsymbol{v} = F, \tag{17}$$

where vector $\boldsymbol{v} = (\boldsymbol{u}, p)^{\mathrm{T}}$, A is the coefficient matrix in (16), and $F = (F_{\boldsymbol{u}}^{\mathrm{T}}, F_p^{\mathrm{T}})^{\mathrm{T}}$.

We construct an iterative scheme based on coordinate splitting by introducing the operators A_i consisting of derivatives with respect to a particular direction x, y, or z. Then, the operator A can be written as $A = A_1 + \cdots + A_d$, where d is equal to the number of spatial dimensions.

The splitting procedure used here is a generalization of the scheme of Douglas and Rachford [8]. After regularization with a derivative with respect to artificial time (or false transient) s [13], the solution of (17) can be obtained as

$$\begin{aligned} \frac{\boldsymbol{v}^{n+1/d}-\boldsymbol{v}^{n}}{\Delta s} &+ A_1\boldsymbol{v}^{n+1/d}+\sum_{i=2}^{l}A_i\boldsymbol{v}^{n}=F^{n} \\ \frac{\boldsymbol{v}^{n+i/d}-\boldsymbol{v}^{n+(i-1)/d}}{\Delta s} &+ A_i(\boldsymbol{v}^{n+i/d}-\boldsymbol{v}^{n})=0, \quad i=2,\dots,d. \end{aligned} \tag{18}$$

In equations (18), Δs is a regularization parameter that can be chosen (usually between 0.05 and 0.5 to ensure and/or accelerate the convergence of the iterative scheme; F^n is the right hand side F at iteration n. The solution $\boldsymbol{v}^{n+1}$ of equations (18) approximates the solution $\boldsymbol{v}$ of equation (17).

In 3D, the splitting equations take the form

$$\begin{aligned} (I+\Delta s\ A_1)\boldsymbol{v}^{n+1/3} &= \boldsymbol{v}^{n}-\Delta s\ (A_2+A_3)\boldsymbol{v}^{n}+\Delta s\ F^{n}, \\ (I+\Delta s\ A_2)\boldsymbol{v}^{n+2/3} &= \boldsymbol{v}^{n+1/3}+\Delta s\ A_2\ \boldsymbol{v}^{n}, \\ (I+\Delta s\ A_3)\boldsymbol{v}^{n+1} &= \boldsymbol{v}^{n+2/3}+\Delta s\ A_3\ \boldsymbol{v}^{n}. \end{aligned} \tag{19}$$

In (19), I is the identity matrix of size $(d+1)$ times the number of unknowns.

It should be noted that the vectorial splitting procedure does not eliminate iterations for finding the solution of the nonlinear system (17). The splitting is used for the purpose of reducing the number of operations necessary to obtain a sufficiently accurate approximation to $\boldsymbol{v}$.

4.3. *Spatial Discretization*

We discretize the differential equations and boundary conditions such that the numerical scheme preserves the integral properties of the underlying continuous problem. Standard central three-point differences are used for the second derivatives that inherit the negative definiteness of the respective differential operators. The first derivatives for pressure are discretized with central second-order differences.

The grid is staggered in each direction; i.e., it is staggered for u in the x-direction, etc. For boundary conditions involving derivatives, this allows the use of second-order central differences with two-point stencils. In three dimensions, we denote the number of main grid lines (which are the grid lines for p) in the x-, y- and z-directions respectively by N_x, N_y, and N_z. The coordinates of the grid points are denoted $(x_i,\ y_j,\ z_k)$ for $i=1,2,\dots,N_x$, $j=1,2,\dots,N_y$, $k=1,2,\dots,N_z$. The grid spacings are given by $h^p_{x,i}=x_{i+1}-x_i$, $i=1,2,\dots,N_x-1$, $h^p_{y,j}=y_{j+1}-y_j$, $j=1,2,\dots,N_y-1$, and $h^p_{z,k}=z_{k+1}-z_k$, $k=1,2,\dots,N_z-1$. The grid spacings for the function u in direction x are defined as

$$h^u_{x,1}=h^p_{x,1},\ h^u_{x,i}=\tfrac{1}{2}(h^p_{x,i}+h^p_{x,i-1}) \text{ for } i=2,\dots,N_x-1, \text{ and } h^u_{x,N_x}=h^p_{x,N_x-1},$$

with the spacings for v in direction y and for w in direction z defined similarly. The pressure is sampled at the points labelled by $\bullet$; function u at $\circ$; function v at $*$, and function w at $\diamond$. We denote

$$\begin{gathered} p_{i,j,k}=p(x_i,y_j,z_k), \quad u_{i,j,k}=u(x_i-\tfrac{1}{2}h^p_{x,i-1},y_j,z_k), \\ v_{i,j,k}=v(x_i,y_j-\tfrac{1}{2}h^p_{y,j-1},z_k), \quad w_{i,j,k}=w(x_i,y_j,z_k-\tfrac{1}{2}h^p_{z,k-1}). \end{gathered}$$

Also, we keep the coupling between the pressure and the respective velocity component through the boundary conditions at each fractional step. This allows us to construct a robust implicit splitting scheme with excellent l^2-stability.

The first derivatives for pressure at the mesh-point labelled by $\circ$, $*$, and $\diamond$ as

$$\left.\frac{\partial p}{\partial x}\right|_{\circ}\approx\frac{p_{i,j,k}-p_{i-1,j,k}}{h^p_{x,i-1}}, \quad \left.\frac{\partial p}{\partial y}\right|_{*}\approx\frac{p_{i,j,k}-p_{i,j-1,k}}{h^p_{y,j-1}}, \quad \left.\frac{\partial p}{\partial z}\right|_{\diamond}\approx\frac{p_{i,j,k}-p_{i,j,k-1}}{h^p_{z,k-1}}.$$

On the other hand, the derivatives $\partial u/\partial x$, $\partial v/\partial y$, and $\partial w/\partial z$ in $\nabla\cdot\boldsymbol{u}$ at each interior mesh-point labelled by "$\bullet$" are approximated as

$$\left.\frac{\partial u}{\partial x}\right|_{\bullet} \approx \frac{u_{i+1,j,k} - u_{i,j,k}}{h^u_{x,i}}, \quad \left.\frac{\partial v}{\partial y}\right|_{\bullet} \approx \frac{v_{i,j+1,k} - v_{i,j,k}}{h^v_{y,j}}, \quad \left.\frac{\partial w}{\partial z}\right|_{\bullet} \approx \frac{w_{i,j,k+1} - w_{i,j,k}}{h^w_{z,k}} \, .$$

The variables u, v, and w in F in (17) are evaluated at the previous iteration.

We consider second-order conservative approximations of the nonlinear operators on a uniform staggered grid that are akin to the ones proposed by Arakawa [1] for the streamline-vorticity formulation for ideal flows. A similar idea in terms of primitive variables was described in [14], with special reference to operator-splitting schemes, and implemented in [6] on a uniform grid and in [15] on a non-uniform grid. On a non-uniform staggered grid, we employ the following conservative differences for the nonlinear terms in the equation for velocity component u:

$$C^h_x[u] = \left.\left(\frac{\partial(u^2)}{\partial x} - \frac{u}{2}\frac{\partial u}{\partial x}\right)\right|_{\circ} = \frac{u^n_{i+1/2,j,k}u_{i+1,j,k} - u^n_{i-1/2,j,k}u_{i-1,j,k}}{h^u_{x,i} + h^u_{x,i-1}} \, ,$$

$$C^h_y[u] = \left.\left(\frac{\partial(uv)}{\partial y} - \frac{u}{2}\frac{\partial v}{\partial y}\right)\right|_{\circ} = \frac{v^n_{i-1/2,j+1,k}u_{i,j+1,k} - v^n_{i-1/2,j,k}u_{i,j-1,k}}{h^p_{y,j} + h^p_{y,j-1}} \, ,$$

$$C^h_z[u] = \left.\left(\frac{\partial(uw)}{\partial z} - \frac{u}{2}\frac{\partial w}{\partial z}\right)\right|_{\circ} = \frac{w^n_{i-1/2,j,k+1}u_{i,j,k+1} - w^n_{i-1/2,j,k}u_{i,j,k-1}}{h^p_{z,k} + h^p_{z,k-1}} \, ,$$

where $u^n_{i+1/2,j,k} = (u^n_{i+1,j,k} + u^n_{i,j,k})/2$, $u^n_{i-1/2,j,k} = (u^n_{i,j,k} + u^n_{i+1,j,k})/2$, etc. Conservative differences for the nonlinear terms in the equations for v and w are similar.

After the discretization for each fractional step, we linearize and solve a linear algebraic system in a similar manner to [15]. The multi-diagonal systems are solved by means of a specialized Gaussian-elimination solver [5] with pivoting that is a generalization of the Thomas algorithm [19]. The algorithm for solving the difference equations is also easily vectorized; the sequence of one-dimensional problems (penta- and tri-diagonal systems) at each time step can be solved in parallel.

4.4. *Algorithm*

The numerical algorithm for solving the problem is

```
(i) Initialization
    (a) Set values of the problem parameters:
        dimension; steady/unsteady problem; $\nu$; $\gamma$; geometry information; initial time $t_0$; final time $T$
    (b) Set values of the time-stepping method parameters:
        implicitness parameter $0 < \theta \leq 1$; time step $\Delta t$; number of time steps $n_t$
    (c) Define grid:
        Specify $N_x$, $N_y$, and $N_z$ (if applicable) for uniform grid or list of points for non-uniform grid
    (d) Set values of the iterative solver parameters: tolerance $\varepsilon$ for the uniform norms of residuals of the
        equations for velocity components and regularization parameter $\Delta s$
    (e) Set initial conditions at $t = t_0$. Set time $t := t + \Delta t$
(ii) Do While $t \leq T$
    (a) Update the boundary conditions at $t$
    (b) Update the right-hand side $F$ from (17) at $t$
    (c) Solve the stationary problem (10), (13), and (15) to find the values of $\boldsymbol{u}$ and $p$ at the time level $t$
        with vectorial operator splitting (18). The following criterion is used for terminating the iterations

        $$\max\{R^u(s), R^v(s), R^w(s)\} \leq \varepsilon, \quad \text{where} \quad R^f(s) \overset{\text{def}}{=} \frac{\max\limits_{i,j,k} |f^{n+1}_{i,j,k} - f^n_{i,j,k}|}{\Delta s \max\limits_{i,j,k} |f^{n+1}_{i,j,k}|}.$$

    (d) Set time $t := t + \Delta t$
    (e) End Do
(iii) End
```

5. Numerical Results

We now verify the convergence properties of the method in space and time. All computations are performed using double-precision arithmetic. No boundary conditions are imposed on the pressure p, and, unless otherwise specified, we use $\Delta s = 0.05$, $\varepsilon = 10^{-10}$, $\gamma = 1$, and $t_0 = 0$. We have tried calculations with different values of γ and have found that $\gamma = 1$ is a good choice in most cases. If γ is chosen to be close to zero or negative, the method may become unstable. In general, γ is varied to enhance stability.

5.1. *Convergence of Time Discretization*

In order to validate convergence of the 3D unsteady algorithm, we perform a convergence test with the following analytical solution of the incompressible Navier–Stokes equations

$$u = v = w = e^{-t}, \quad p = (x + y + z)\, e^{-t}, \tag{20}$$

in the unit square. Because all functions in (20) are linear in x, y, and z, this test allows us to directly verify the convergence rate of the time discretization.

We use fixed values for the following problem parameters: $\nu = 1/15$; $h_x = h_y = h_z = 1/16$; and $T = 1$. Figure 1 presents the l^2-norms of the residuals $R(s) = \sqrt{(R^u(s))^2 + (R^v(s))^2 + (R^w(s))^2}$ versus s ($s = n\Delta s$, where n is the iteration number in the stationary problem solver). We clearly see the convergence of the vectorial operator splitting iterations for finding the solution at time $T = 1$ starting from time $t = 0$ for two different time steps, $\Delta t = 1$ (left) and $\Delta t = 0.5$ (right).

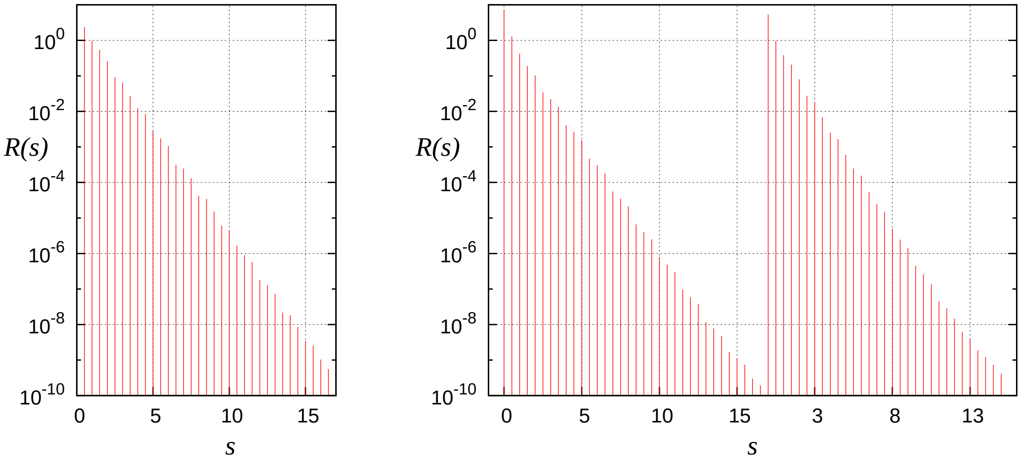

Fig. 1. Residual norm $R(s)$ for $\Delta t = 1$ with one step in time (left) and $\Delta t = 0.5$ with two steps in time (right).

The maximum, average, and l^2-norms of the difference between the numerical solution and the exact solution (20) at the final time are on the order of the round-off error in double-precision arithmetic. Figure 2 shows the l^2-error taken over all grid points for the numerical solution $\boldsymbol{u}$ and the exact solution $\boldsymbol{u}(\boldsymbol{x}, T)$

$$l^2\text{–error} \overset{\text{def}}{=} \|\boldsymbol{u} - \boldsymbol{u}(\cdot, T)\|_2 = \sqrt{(u - u(\cdot, T))^2 + (v - v(\cdot, T))^2 + (w - w(\cdot, T))^2}. \tag{21}$$

The errors are small due to fact that the solution is only linearly dependent in space. The l^2-error for $\boldsymbol{u}$ decreases for $2^{-8} = 0.125 \leq \Delta t \leq 1$ and increases for time steps $\Delta t \leq 2^{-16}$. This is not abnormal for numerical errors and is due to the increased number of arithmetic operations. The errors are all on the order of round-off errors.

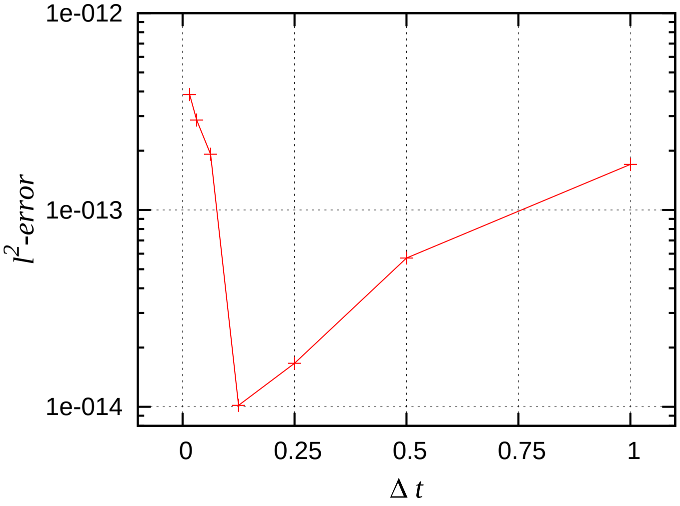

Fig. 2. The l^2-error for $\boldsymbol{u}$ versus time step Δt.

5.2. *Convergence of Spatial Discretization*

To confirm the convergence of the spatial discretization of the unsteady algorithm, we perform calculations on a uniform grid $h = h_x = h_y = h_z$ for a problem with the following exact solution

$$u = \sqrt{2}\exp(-\sqrt{2}x)\cos(y+z), \quad v = w = \exp(-\sqrt{2}x)\sin(y+z), \quad p = -\exp(-2\sqrt{2}x), \tag{22}$$

in the unit square. We choose $\nu = 1/15$, $\Delta t = 1$, $T = 2$, and vary the spacing $h = h_x = h_y = h_z$. As can be expected for a steady-state solution, the changes (if any) in the numerically obtained values of the sought functions at $t = 1$ and $t = 2$ are small, i.e., on the order of round-off error.

We also present results for the calculated maximum absolute value of the divergence $\nabla \cdot \boldsymbol{u}$ and the l^2-error for $\boldsymbol{u}$ defined in (21) at time T as well as their convergence rates, calculated as $\alpha = \log_2 \left| \frac{l^2\text{-error}(h)}{l^2\text{-error}(2h)} \right|$; see Table 1. It can be seen that the convergence rate for the divergence is second order, and in fact the convergence rate appears to be higher than second order for the l^2-errors for $\boldsymbol{u}$.

Table 1
Discretization errors as a function of $h = h_x = h_y = h_z$.

$\frac{1}{h}$	h^2	$\max_{i,j,k} \lvert\nabla \cdot \boldsymbol{u}\rvert$		l^2-error	
		value	α	value	α
8	1.56250×10^{-2}	3.90701×10^{-3}	-	3.86078×10^{-6}	-
16	3.90625×10^{-3}	9.76610×10^{-4}	2.00021138	4.91198×10^{-7}	2.974512989
32	9.76563×10^{-4}	2.44144×10^{-4}	2.00005284	5.02422×10^{-8}	3.289334615
64	2.44141×10^{-4}	6.10353×10^{-5}	2.00001321	4.71955×10^{-9}	3.412176909

All the results described in Section 5.1 and Section 5.2 use the fully implicit backward Euler scheme. They confirm the expected convergence properties of the method. The convergence results for such large time steps demonstrates the excellent stability of the algorithm.

5.3. *Convergence of Overall Method*

Finally, in order to verify the convergence of the overall method in both time and space, we use a third test with a 3D analytical solution presented in [9]. The particular solution used here is

$$\begin{aligned}
u =& -\left[e^{x}\sin(y+z)+e^{z}\cos(x+y)\right]e^{-t}] \\
v =& -\left[e^{y}\sin(z+x)+e^{x}\cos(y+z)\right]e^{-t}] \\
w =& -\left[e^{z}\sin(x+y)+e^{y}\cos(z+x)\right]e^{-t}] \\
p =& -\frac{1}{2}[e^{2x}+e^{2y}+e^{2z}+2\sin(x+y)\cos(z+x)e^{y+z} \\
& +2\sin(y+z)\cos(x+y)e^{z+x}+2\sin(z+x)\cos(y+z)e^{x+y}]e^{-2t}
\end{aligned} \tag{23}$$

in $\Omega=[-0.5,0.5]^3$. In this test the values of the problem parameters are specified as $\nu=1/15$, $\Delta t=0.25$, $h=2^{-4}$, and $T=1$.

Unlike the numerical solution, the analytical solution (23) does not depend on the viscosity ν nor do the analytical solutions (20) and (22). However, it is a good problem for benchmarking because it does depend on both time and space. There are terms that are growing exponentially with the increase of x, y, and z. With regard to dependence on time, all of the analytic solutions used here are decreasing in time due to terms such as e^{-kt}, where k is a positive constant equal to 1 or 2.

Regardless of the fact that the analytical functions do not depend on the parameter ν, the numerical scheme may depend on ν due to the discretization. We mentioned that some of the properties of the differential operators are preserved by their discrete counterparts at each iteration. However, many of the conservation properties are subject to fulfillment of the continuity equation, or they are satisfied upon convergence of the iterations. The difference scheme described here is implicit for the momentum equation (except for the convective term, which requires linearization) and the boundary conditions. The scheme is also implicit for all operators in the pressure equation except for $\nabla\cdot(C\boldsymbol{u}(\boldsymbol{x},t+\Delta t))$. This term is taken from the previous iteration in the stationary problem solver. After convergence, it satisfies the equations at the new time level; however, this may cause some problem with convergence of the iterative process. In fact, we have observed instability in the calculations for $\nu=1/15$ with (23). The iterative process diverges for some values of Δt and Δs if h is small. Again, this can be controlled to some extent by choosing appropriate (typically smaller) values of Δs.

The test problems in Section 5.1 and Section 5.2 are not suitable for testing the Crank–Nicolson method as well as the two versions of pressure equation because the terms that would make any difference are not present in these test problems. Even the analytical solution (23) does not allow us to verify the second-order convergence in time of the Crank–Nicolson method because the spatial discretization errors dominate when Δt is small.

We have also performed calculations for $\theta=0.5$ with the two pressure equations (13) and (14) derived in Section 3.2.1 and Section 3.2.2, respectively. The algorithm performs comparably (there are small differences in the divergence for larger time steps) if equation (13) is used for pressure.

Table 2
l^2-errors with equation (14), $h=2^{-4}$, $\nu=1/15$.

Δt	$\theta=1.0$	$\theta=0.5$
1	5.78908×10^{-4}	9.97307×10^{-4}
0.5	4.98314×10^{-4}	2.95910×10^{-4}
0.25	4.51934×10^{-4}	4.03451×10^{-4}
0.125	4.25608×10^{-4}	3.98005×10^{-4}

The results for the l^2-error from calculations using pressure equation (14) and for $\theta=1$ and $\theta=0.5$ are presented in Table 2. Although the convergence in time cannot be observed from these results because the errors are dominated by the spatial component, it can be seen that the l^2-errors are small, even for relatively large h, attesting to the accuracy and stability of the overall method.

6. Conclusion

We have demonstrated the convergence for a numerical method for the unsteady incompressible Navier–Stokes equations that is based on a fully implicit time integration and a conservative spatial discretization.

The resulting discrete system is solved efficiently using vectorial operator splitting. The most important properties of the method are the overall stability due to the implicit treatment of the time-stepping and boundary conditions and the conservative spatial discretization. The results from numerical experiments we give indicate that the discretization errors are dominated by the spatial component because the spatial step size h is relatively large. Nonetheless, the l^2-errors are small and suggest the method has excellent stability properties. Future work will focus on the development and investigation of higher-order discretizations and comparison with other existing methods.

Acknowledgment

This work was partially supported by MITACS and NSERC.

References

[1] A. Arakawa, Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *J. Comp. Phys.* 1, 119–143 (1966).

[2] U. M. Ascher, L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, 1998.

[3] F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods*, Berlin:Springer, 1991.

[4] A. J. Chorin, Numerical Solution of the Navier-Stokes Equations, *Mathematics of Computation* 22(104), 745–762 (1968).

[5] C. I. Christov, Gaussian Elimination with Pivoting for Multidiagonal Systems. University of Reading, Internal Report 4, 1994.

[6] C. I. Christov, R. S. Marinova. Implicit scheme for Navier–Stokes equations in primitive variables via vectorial operator splitting. In M. Griebel, O. P. Iliev, S. D. Margenov, and P. S. Vassilevski, editors, *Notes on Numer. Fluid Mech.* 62, 251–259, Vieweg (1998).

[7] C. R. Doering, The 3D Navier-Stokes Problem, *Annu. Rev. Fluid Mech.*, 41, 109128 (2009).

[8] J. Douglas, H. H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, *Trans. Amer. Math. Soc.* 82, 421–439 (1956).

[9] C. R. Ethier and D. A. Steinman, Exact fully 3D Navier-Stokes solutions for benchmarking, *International Journal for Numerical Methods in Fluids* 19(5), 369–375 (1994).

[10] M. Hafez, M. Soliman, Numerical Solutions of the Incompressible Navier-Stokes Equations in Primitive Variables, *Incompressible computational fluid dynamics*, Cambridge University Press, Editors: Max D. Gunzburger and Roy A. Nicolaides, 183–201, 1993.

[11] D. Kwak, C. Kiris, C. S. Kim, Computational challenges of viscous incompressible flows, *Computers & Fluids* 34, 283–299 (2005).

[12] H. P. Langtangen, K.-A. Mardal, R. Winther, Numerical Methods for Incompressible Viscous Flow, *Advances in Water Resources* 25, 1125–1146 (2002).

[13] G. D. Mallison and G. de Vahl Davis. The method of false transients for the solution of coupled elliptic equations, *Journal of Computational Physics* 12, 435–461 (1973).

[14] G. I. Marchuk, *Methods of Numerical Mathematics*, Springer, Berlin, 1982.

[15] R. S. Marinova, C. I. Christov, T. T. Marinov, A Fully Coupled Solver for Incompressible Navier-Stokes Equations, *Int. Journal of Computational Fluid Dynamics*, 17(5), 371–385 (2003).

[16] R.S. Marinova, T. Takahashi, H. Aiso, C.I. Christov, T.T. Marinov, Conservation Properties of Vectorial Operator Splitting, *Journal of Computational and Applied Mathematics*, 152(1–2), 289–303 (2003).

[17] Y. Morinishi, T.S. Lund, O.V. Vasiliev, P. Moin, Fully conservative higher order finite difference schemes for incompressible flow, *Journal of Computational Physics* 143, 90–124, (1998).

[18] J. N. Reddy, On Penalty function methods in the finite element analysis of flow problems, *International Journal for Numerical Methods in Fluids* 18, 853–870 (1982).

[19] L. H. Thomas, Elliptic Problems in Linear Difference Equations over a Network, Watson Scientific Computing Laboratory, Columbia University, New York, 1949.

[20] N. N. Yanenko. *Method of Fractional Steps.* Springer, Berlin, 1971.