



A new class of massively parallel direction splitting for the incompressible Navier–Stokes equations

J.L. Guermond ^{a,b,1}, P.D. Mineev ^{c,*}

^a Department of Mathematics, Texas A&M University, College Station, TX 77843-3368, USA

^b LIMSI (CNRS-UPR 3152), BP 133, 91403, Orsay, France

^c Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada T6G 2G1

ARTICLE INFO

Article history:

Received 5 June 2010

Received in revised form 25 January 2011

Accepted 5 February 2011

Available online 23 February 2011

Keywords:

ADI

Direction splitting

Incompressible flows

Navier–Stokes

Pressure Poisson equation

Time splitting

ABSTRACT

We introduce in this paper a new direction splitting algorithm for solving the incompressible Navier–Stokes equations. The main originality of the method consists of using the operator $(I - \partial_{xx})(I - \partial_{yy})(I - \partial_{zz})$ for approximating the pressure correction instead of the Poisson operator as done in all the contemporary projection methods. The complexity of the proposed algorithm is significantly lower than that of projection methods, and it is shown that they have the same stability properties as the Poisson-based pressure-correction techniques, either in standard or rotational form. The first-order (in time) version of the method is proved to have the same convergence properties as the classical first-order projection techniques. Numerical tests reveal that the second-order version of the method has the same convergence rate as its second-order projection counterpart as well. The method is suitable for parallel implementation and preliminary tests show excellent parallel performance on a distributed memory cluster of up to 1024 processors. The method has been validated on the three-dimensional lid-driven cavity flow using grids composed of up to 2×10^9 points.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Most of the time marching algorithms that are used in Computational Fluid Mechanics to solve large scale incompressible fluids flows are based on the so-called projection methods (see Chorin [5] and Temam [24] for the earliest examples and [12] for a recent review). In all the variants of this strategy, the pressure, or pressure correction, is obtained by solving a Poisson equation equipped with Neumann boundary conditions or a weak version thereof. The so-called pressure Poisson equation results from the decomposition of the velocity field into a divergence-free part and a gradient, and this decomposition has been the main paradigm behind all the improved pressure/velocity decoupling techniques that have been proposed since the pioneering works of Chorin and Temam.

The objective of this paper is to introduce a novel fractional time stepping technique that departs from the projection paradigm. We propose instead to use a pressure equation derived from a perturbed form of the continuity equation in which the incom-

pressibility constraint is penalized in a negative norm induced by direction splitting. Departure from the projection paradigm has already been proved to be very efficient for solving variable density flows in [13]. An early version of the algorithm proposed in this paper has been announced in [9]. In the present work we pursue further the ideas introduced/announced in [9] in the sense that in addition to splitting the pressure-correction, we also apply a direction splitting technique to the momentum equation, thus further reducing the overall computational complexity of the method. We provide stability and convergence results for the first-order variant of the method and a stability result for the (formally) second-order fully split scheme, and we numerically illustrate the convergence properties of the method and its scalability.

The paper is organized as follows. The basic idea is introduced in Section 2. The new paradigm consists of constructing a singular perturbation of the Navier–Stokes equations by using an abstract operator A having generic coercivity properties in $H^1(\Omega)$ (see Section 2.2). The family of Chorin/Temam projection methods is recovered by using $A := -\Delta$ but an entirely new family is obtained by setting $A := (1 - \partial_{xx})(1 - \partial_{yy})$ in two space dimensions and $A := (1 - \partial_{xx})(1 - \partial_{yy})(1 - \partial_{zz})$ in three space dimensions. The merits of the proposed technique are discussed in Section 3. A formally second-order version of the method is introduced in Section 4. The main theoretical result of this paper is Theorem 4.2 which establishes the stability and convergence of this new algorithm. The method is illustrated numerically in Section 5 and

* Corresponding author.

E-mail addresses: guermond@math.tamu.edu (J.L. Guermond), mineev@ualberta.ca (P.D. Mineev).

¹ This publication is based on work partially supported by Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

² The work of this author is also supported by fellowships from the Institute of Applied Mathematics and Computational Science and the Institute of Scientific Computing at Texas A&M University, and a Discovery grant of NSERC.

convergence tests confirm that the rotational form of the incremental version of the method is indeed second-order in the L^2 -norm of the velocity field. Weak scalability tests reported in Section 5 confirm that the method is extremely efficient. Some conclusions are reported in Section 6.

2. The non-incremental scheme

In this section we introduce the most simple first-order version of the method and analyze its stability. The purpose of this section is to introduce the basic concepts of the technique and to avoid technicalities. A formally second-order variant of the method is introduced and analyzed in Section 4.

2.1. Notation and preliminaries

We consider the time-dependent Navier–Stokes equations on a finite time interval $[0, T]$ and in a cubic domain $\Omega = (0, 1)^3$.

Since the nonlinear term in the Navier–Stokes equations does not interfere with the incompressibility constraint, we henceforth mainly focus our attention on the time-dependent Stokes equations written in terms of velocity \mathbf{u} and pressure p :

$$\begin{cases} \partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \times [0, T], \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times [0, T], \\ \mathbf{u}|_{\partial\Omega} = 0 & \text{in } [0, T], \text{ and } \mathbf{u}|_{t=0} = \mathbf{u}_0 & \text{in } \Omega, \end{cases} \quad (2.1)$$

where \mathbf{f} is a smooth source term and \mathbf{u}_0 is a solenoidal initial velocity field with zero normal trace at the boundary of Ω . We consider homogeneous Dirichlet boundary conditions on the velocity for the sake of simplicity.

2.2. Singular perturbation analysis

Let us start by recalling some fundamental properties of projection methods. Let Δt be the time step in the Chorin–Temam algorithm, then it can be shown (see e.g., [22]) that the Chorin–Temam algorithm is a singular perturbation of (2.1) which can be written as follows:

$$\begin{cases} \partial_t \mathbf{u}_\epsilon - \nu \Delta \mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f} & \text{in } \Omega \times [0, T], \\ -\Delta t \Delta p_\epsilon + \nabla \cdot \mathbf{u}_\epsilon = 0 & \text{in } \Omega \times [0, T], \\ \mathbf{u}_\epsilon|_{\partial\Omega} = 0, \partial_n p_\epsilon|_{\partial\Omega} = 0 & \text{in } [0, T], \\ \text{and } \mathbf{u}_\epsilon|_{t=0} = \mathbf{u}_0, p_\epsilon|_{t=0} = p_0 & \text{in } \Omega, \end{cases} \quad (2.2)$$

where $p_0 := p(t=0)$ and Δt is the perturbation parameter (i.e., $\epsilon := \Delta t$). A key property of the Chorin–Temam technique is that the quantity $\mathbf{v}_\epsilon := -\Delta t \nabla p_\epsilon + \mathbf{u}_\epsilon$ is solenoidal and has a zero normal trace at the boundary. Denoting $\mathbf{H} := \{\mathbf{z} \in \mathbf{L}(\Omega)^2; \nabla \cdot \mathbf{z} \in L^2(\Omega); \mathbf{z} \cdot \mathbf{n}|_{\partial\Omega} = 0\}$ and setting $P_{\mathbf{H}} : \mathbf{L}^2(\Omega) \rightarrow \mathbf{H}$ to be the L^2 -projection onto \mathbf{H} , we have $\mathbf{v}_\epsilon = P_{\mathbf{H}} \mathbf{u}_\epsilon$, which is the earmark of Chorin–Temam type algorithms. Actually $\mathbf{u}_\epsilon = \mathbf{v}_\epsilon + \Delta t \nabla p_\epsilon$ is the so-called Helmholtz decomposition of \mathbf{u}_ϵ and this decomposition is L^2 -orthogonal, i.e., $L^2(\Omega) = \mathbf{H} \oplus \nabla H^1_{j=0}(\Omega)$ where we have defined $L^2_{j=0}(\Omega)$ and $H^1_{j=0}(\Omega)$ to be the spaces that are composed of those functions in $L^2(\Omega)$ and $H^1(\Omega)$ that are of zero mean, respectively. The quantity \mathbf{v}_ϵ is often used as an approximation for \mathbf{u} .

We now propose to depart from (2.2) by considering the following alternative $\mathcal{O}(\Delta t)$ -perturbation of (2.1):

$$\begin{cases} \partial_t \mathbf{u}_\epsilon - \nu \Delta \mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f} & \text{in } \Omega \times [0, T], \\ \Delta t \Delta p_\epsilon + \nabla \cdot \mathbf{u}_\epsilon = 0 & \text{in } \Omega \times [0, T], \\ \mathbf{u}_\epsilon|_{\partial\Omega} = 0, p_\epsilon \in D(A) & \text{in } [0, T], \\ \text{and } \mathbf{u}_\epsilon|_{t=0} = \mathbf{u}_0, p_\epsilon|_{t=0} = p_0 & \text{in } \Omega, \end{cases} \quad (2.3)$$

where the operator $A : D(A) \subset L^2_{j=0}(\Omega) \rightarrow L^2_{j=0}(\Omega)$ is assumed to be unbounded and closed and to be such that the bilinear form $a(p, q) := \int_{\Omega} q A p \, d\mathbf{x}$ satisfies the following properties:

$$a \text{ is symmetric, and } \|\nabla q\|_{L^2}^2 \leq a(q, q), \quad \forall q \in D(A). \quad (2.4)$$

As a consequence of these hypotheses, the following scalar product and norm can be defined:

$$\begin{aligned} \langle p, q \rangle_A &:= \langle A p, q \rangle, \quad \forall p, q \in D(A), \quad \|q\|_A \\ &:= \langle A q, q \rangle^{\frac{1}{2}}, \quad \forall q \in D(A). \end{aligned} \quad (2.5)$$

Many admissible choices are possible for the operator A . For instance one recovers the Chorin–Temam technique by using $A = -\Delta_N$, where $-\Delta_N$ is the Laplace operator supplemented with homogeneous Neumann boundary conditions. One could also use $A = I - \Delta_N$ where I is the identity operator. The key to the method presented in this paper is that $A := (1 - \partial_{xx})(1 - \partial_{yy})(1 - \partial_{zz})$ with appropriate boundary conditions satisfies the requirement (2.4) in three space dimensions. In two space dimensions the operator $A := (1 - \partial_{xx})(1 - \partial_{yy})$ with appropriate boundary conditions also satisfies (2.4).

At this point we have made a significant step away from the projection paradigm. The presence of the operator A in (2.3) breaks the L^2 -orthogonality property mentioned above. Although, (2.3) can no longer be interpreted in terms of projection, the solutions to (2.2) and (2.3) have similar convergence properties.

Proposition 2.1. *Assume that the solution to (2.1) is smooth enough with respect to time and space and that (2.4) holds, then the solutions to (2.2) and (2.3) both satisfy the following estimates*

$$\|\mathbf{u} - \mathbf{u}_\epsilon\|_{L^2((0,T);H^1(\Omega))} \leq c \Delta t^{\frac{1}{2}}. \quad (2.6)$$

Proof. Repeat the arguments from [22].

2.3. Formulation of the scheme

We now construct a fractional step technique approximating (2.3) by using alternating direction strategies and the Crank–Nicolson time stepping.

2.3.1. Pressure predictor

The first step of the algorithm consists of computing a pressure predictor as any other fractional time stepping technique for the Navier–Stokes equations. The algorithm is initialized by setting $p^{-\frac{1}{2}} = 0$ and for $n \geq 0$ we set

$$p^{*,n+\frac{1}{2}} = p^{n-\frac{1}{2}}. \quad (2.7)$$

2.3.2. Velocity update

The second step consist of updating the velocity field. We propose to update the velocity by using a direction splitting technique proposed by Douglas [16]. The algorithm is initialized by setting $\mathbf{u}^0 = \mathbf{u}_0$, and the velocity is updated as follows for $n \geq 0$:

$$\begin{aligned} \frac{\xi^{n+1} - \mathbf{u}^n}{\Delta t} - \nu \Delta \mathbf{u}^n + \nabla p^{*,n+\frac{1}{2}} &= \mathbf{f}(t^{n+\frac{1}{2}}), \quad \mathbf{u}^n|_{\partial\Omega} = 0, \\ \frac{\boldsymbol{\eta}^{n+1} - \xi^{n+1}}{\Delta t} - \frac{\nu}{2} \partial_{xx}(\boldsymbol{\eta}^{n+1} - \mathbf{u}^n) &= 0, \quad \boldsymbol{\eta}^{n+1}|_{x=0,1} = 0, \\ \frac{\zeta^{n+1} - \boldsymbol{\eta}^{n+1}}{\Delta t} - \frac{\nu}{2} \partial_{yy}(\zeta^{n+1} - \mathbf{u}^n) &= 0, \quad \zeta^{n+1}|_{y=0,1} = 0, \\ \frac{\mathbf{u}^{n+1} - \zeta^{n+1}}{\Delta t} - \frac{\nu}{2} \partial_{zz}(\mathbf{u}^{n+1} - \mathbf{u}^n) &= 0, \quad \mathbf{u}^{n+1}|_{z=0,1} = 0. \end{aligned} \quad (2.8)$$

The two-dimensional version of the algorithm is obtained by omitting the last substep and setting $\mathbf{u}^{n+1} = \zeta^{n+1}$.

Remark 2.1. The Alternating Direction Implicit (ADI) strategy was first proposed in the seminal article of Peaceman and Rachford [20]. It is a Crank–Nicolson-like two-stage time integration scheme for two-dimensional parabolic problems in which the second-order derivative with respect to each space variable is treated implicitly while the other is made explicit at each time sub-step. Overall, the ADI strategy is very efficient for parabolic problems since the resulting schemes have the same computational complexity as explicit schemes while being unconditionally stable. The Douglas scheme (2.8) has similar stability and convergence properties as ADI but contrary to ADI it can be applied to parabolic problems in any space dimension.

2.3.3. Pressure update

The pressure is updated by solving $Ap^{n+\frac{1}{2}} = -\frac{1}{\Delta t} \nabla \cdot \mathbf{u}^{n+1}$ with the direction splitting operator $A := (1 - \partial_{xx})(1 - \partial_{yy})(1 - \partial_{zz})$ supplemented with appropriate boundary conditions. This is done as follows:

$$\begin{aligned} \psi - \partial_{xx}\psi &= -\frac{1}{\Delta t} \nabla \cdot \mathbf{u}^{n+1}, & \partial_x \psi|_{x=0,1} &= 0, \\ \varphi - \partial_{yy}\varphi &= \psi, & \partial_y \varphi|_{y=0,1} &= 0, \\ p^{n+\frac{1}{2}} - \partial_{zz}p^{n+\frac{1}{2}} &= \varphi, & \partial_z p^{n+\frac{1}{2}}|_{z=0,1} &= 0. \end{aligned} \tag{2.9}$$

2.4. Stability and convergence analysis

Probably the most remarkable feature of the scheme (2.7)–(2.9) is that it only requires solving one-dimensional boundary-value problems. We are going to show that it provides a solution which as accurate as that given by the classical Chorin–Temam projection method at a fraction of its complexity.

Before going through the stability analysis, we establish a preliminary result concerning the operator A which is used in (2.9). Henceforth we denote

$$A := (1 - \partial_{xx})(1 - \partial_{yy})(1 - \partial_{zz}), \tag{2.10}$$

and the domain of A is defined to be

$$\begin{aligned} D(A) := \{ & p \in L^2(\Omega); ((1 - \partial_{zz})p, (1 - \partial_{yy})(1 - \partial_{zz})p, Ap) \\ & \in [L^2(\Omega)]^3, \partial_z p|_{z=0,1} = 0, \partial_y(1 - \partial_{zz})p|_{y=0,1} \\ & = 0, \partial_x(1 - \partial_{yy})(1 - \partial_{zz})p|_{x=0,1} = 0 \}. \end{aligned} \tag{2.11}$$

We first prove that (2.9) indeed amounts to solving $Ap^{n+\frac{1}{2}} = -\frac{\nabla \cdot \mathbf{u}^{n+1}}{\Delta t}$ and that A has the properties specified in (2.4).

Lemma 2.2. Let $f \in L^2(\Omega)$. Let ψ , φ , and p solve

$$\begin{aligned} \psi - \partial_{xx}\psi &= f, & \partial_x \psi|_{x=0,1} &= 0, \\ \varphi - \partial_{yy}\varphi &= \psi, & \partial_y \varphi|_{y=0,1} &= 0, \\ p - \partial_{zz}p &= \varphi, & \partial_z p|_{z=0,1} &= 0, \end{aligned}$$

then $Ap = f$. Moreover the bilinear form $a : D(A) \times D(A) \ni (p, q) \mapsto \int_{\Omega} qAp \, dx \in \mathbb{R}$ satisfies (2.4).

Proof. The existence and uniqueness of ψ , φ , and p in $\{r \in L^2(\Omega), \partial_x r|_{x=0,1} = 0\}$, $\{r \in L^2(\Omega), \partial_y r|_{y=0,1} = 0\}$, and $\{r \in L^2(\Omega), \partial_z r|_{z=0,1} = 0\}$, respectively, is a simple consequence of the Lax–Milgram lemma. By proceeding by elimination, it is clear also that $Ap = f$ and p is a member of $D(A)$.

Let us now prove (2.4). Let q be a member of $D(A)$, then

$$\begin{aligned} \int_{\Omega} qAp \, dx &= \int_{\Omega} q(1 - \partial_{xx})\psi \, dx = \int_{\Omega} (q\psi + \partial_x q \partial_x \psi) \, dx \\ &= \int_{\Omega} (q(1 - \partial_{yy})\varphi + \partial_x q \partial_x (1 - \partial_{yy})\varphi) \, dx \\ &= \int_{\Omega} (q\varphi + \partial_y q \partial_y \varphi + \partial_x q \partial_x \varphi - \partial_x q \partial_{xy} \varphi) \, dx \\ &= \int_{\Omega} (q\varphi + \partial_y q \partial_y \varphi + \partial_x q \partial_x \varphi + \partial_{xy} q \partial_{xy} \varphi) \, dx \\ &\quad - \int_{\{y=1\}} \partial_x q(x, 1, z) \partial_{xy} \varphi(x, 1, z) \, dx \, dz \\ &\quad + \int_{\{y=0\}} \partial_x q(x, 0, z) \partial_{xy} \varphi(x, 0, z) \, dx \, dz. \end{aligned}$$

The boundary terms are zero since the two boundary conditions $\partial_y \varphi(x, 1, z) = 0$ and $\partial_y \varphi(x, 0, z) = 0$ for all $(x, z) \in (0, 1)^2$ imply that $\partial_{xy} \varphi(x, 1, z) = \partial_x(\partial_y \varphi(x, 1, z)) = 0$, and $\partial_{xy} \varphi(x, 0, z) = \partial_x(\partial_y \varphi(x, 0, z)) = 0$ for all $(x, z) \in (0, 1)^2$. As a result,

$$\begin{aligned} \int_{\Omega} qAp \, dx &= \int_{\Omega} (q\varphi + \partial_y q \partial_y \varphi + \partial_x q \partial_x \varphi + \partial_{xy} q \partial_{xy} \varphi) \, dx \\ &= \int_{\Omega} (q(p - \partial_{zz}p) + \partial_y q \partial_y (p - \partial_{zz}p) + \partial_x q \partial_x (p - \partial_{zz}p) \\ &\quad + \partial_{xy} q \partial_{xy} (p - \partial_{zz}p)) \, dx \\ &= \int_{\Omega} (qp + \partial_z q \partial_z p + \partial_y q \partial_y p + \partial_{yz} q \partial_{yz} p + \partial_x q \partial_x p \\ &\quad + \partial_{xz} \partial_{xz} p + \partial_{xy} q \partial_{xy} p + \partial_{xyz} \partial_{xyz} p) \, dx. \end{aligned}$$

The boundary terms resulting from the last integration by parts are zero since the following identities hold for all $\xi \in \{0, 1\}$ and all $(x, y) \in (0, 1)^2$:

$$\begin{aligned} \partial_{xz} p(x, y, \xi) &= \partial_x(\partial_z p(x, y, \xi)) = 0, \\ \partial_{yz} p(x, y, \xi) &= \partial_y(\partial_z p(x, y, \xi)) = 0, \\ \partial_{xyz} p(x, y, \xi) &= \partial_{xy}(\partial_z p(x, y, \xi)) = 0. \end{aligned}$$

The symmetry and the H^1 -coercivity of the bilinear form a are now evident. \square

Let $N := T/\Delta t$, and for any norm $\|\cdot\|_E$ let us introduce the following notation:

$$\|g\|_{\ell^2(0,T;E)}^2 := \Delta t \sum_{n=1}^N \|g(t^n, \cdot)\|_E^2, \quad \|g\|_{\ell^\infty(0,T;E)} := \max_{0 \leq n \leq N} \|g(t^n, \cdot)\|_E. \tag{2.12}$$

We define the negative norm $\|\cdot\|_{H^{-1}}$ as follows:

$$\|f\|_{H^{-1}} := \sup_{0 \neq g \in H_0^1(\Omega)} \frac{(f, g)}{\|\nabla g\|_{L^2}}. \tag{2.13}$$

We are now in position to prove the main result of this section:

Proposition 2.3. Assume that the solution to (2.1) is smooth enough with respect to time and space and that (2.4) holds. Then the solution to (2.7), (2.16), (2.9), with $p^{-\frac{1}{2}} = 0$, satisfies the following stability estimate for all $T > 0$:

$$\begin{aligned} \|\mathbf{u}\|_{\ell^\infty(0,T;L^2)}^2 + v \|\nabla \mathbf{u}\|_{\ell^2(0,T;L^2)}^2 + \Delta t^2 \|p\|_{\ell^2(-\frac{\Delta t}{2}, T - \frac{\Delta t}{2}; D(A))}^2 \\ \leq \|\mathbf{u}^0\|_{L^2}^2 + v^{-1} \|f\|_{\ell^2(\frac{\Delta t}{2}, T - \frac{\Delta t}{2}; H^{-1})}^2. \end{aligned} \tag{2.14}$$

For all T there is c , uniform in Δt , so that

$$\|\mathbf{u} - \mathbf{u}\|_{\ell^\infty(0,T;L^2)} + \sqrt{v} \|\nabla(\mathbf{u} - \mathbf{u})\|_{\ell^2(0,T;L^2)} \leq c \Delta t^{\frac{1}{2}}. \tag{2.15}$$

Proof. The direction splitting technique which is used to solve the momentum Eq. (2.8) is non-essential and certainly not original. The original part of the present work is the introduction of the direction splitting for computing the pressure update (2.9). To avoid mixing the two issues and to simplify the presentation, we are going to make the stability analysis assuming that the momentum equation is solved without direction splitting, i.e., we replace (2.8) by the Euler time stepping:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} - \nu \Delta \mathbf{u}^{n+1} + \nabla p^{*,n+\frac{1}{2}} = \mathbf{f}, \quad \mathbf{u}|_{\partial\Omega} = 0. \quad (2.16)$$

The error and stability analysis of the fully split algorithm is done in details in [11].

We multiply (2.16) by $2\Delta t \mathbf{u}^{n+1}$ and integrate over Ω , then using the identity $2(a-b, a) = \|a\|^2 + \|a-b\|^2 - \|b\|^2$ we obtain

$$\begin{aligned} \|\mathbf{u}^{n+1}\|_{L^2}^2 + \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2}^2 - \|\mathbf{u}^n\|_{L^2}^2 + 2\Delta t \nu \|\nabla \mathbf{u}^{n+1}\|_{L^2}^2 \\ + 2\Delta t (\nabla p^{n-\frac{1}{2}}, \mathbf{u}^{n+1}) = 2\Delta t \int_{\Omega} \mathbf{f}^{n+\frac{1}{2}} \cdot \mathbf{u}^{n+1} \, d\mathbf{x}. \end{aligned} \quad (2.17)$$

Now we use Lemma 2.2 to deduce that $p^{n+\frac{1}{2}} \in D(A)$ solves the following problem:

$$a(p^{n+\frac{1}{2}}, q) = -\Delta t^{-1} (\nabla \cdot \mathbf{u}^{n+1}, q), \quad \forall q \in D(A). \quad (2.18)$$

Using $2\Delta t^2 p^{n-\frac{1}{2}}$ as a test function together with the fact that $a(\cdot, \cdot)$ is symmetric and coercive in $H^1(\Omega)$, we infer that

$$\begin{aligned} -2\Delta t (\nabla \cdot \mathbf{u}^{n+1}, p^{n-\frac{1}{2}}) &= 2\Delta t^2 a(p^{n+\frac{1}{2}}, p^{n-\frac{1}{2}}) \\ &= \Delta t^2 \left(\|p^{n+\frac{1}{2}}\|_A^2 + \|p^{n-\frac{1}{2}}\|_A^2 - \|p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}\|_A^2 \right). \end{aligned} \quad (2.19)$$

We now seek a control on $\|p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}\|_A^2$ by subtracting (2.18) at time t^n to (2.18) at time t^{n+1} and by testing the result with $\Delta t (p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}})$,

$$\begin{aligned} \Delta t \|p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}\|_A^2 &= -(\nabla \cdot (\mathbf{u}^{n+1} - \mathbf{u}^n), p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ &= (\mathbf{u}^{n+1} - \mathbf{u}^n, \nabla (p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}})) \\ &\leq \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2} \|\nabla (p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}})\|_{L^2}. \end{aligned}$$

(Note that the above operation is legitimate also for $n=0$ provided $p^{-\frac{1}{2}}=0$ and $\mathbf{u}^0 = \mathbf{u}_0$.) Then using the coercivity property of the bilinear form a we infer that

$$\begin{aligned} \Delta t \|\nabla (p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}})\|_{L^2} \|p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}\|_A \\ \leq \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2} \|\nabla (p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}})\|_{L^2}, \end{aligned}$$

which then implies

$$\Delta t^2 \|p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}\|_A^2 \leq \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2}^2.$$

Inserting this bound into (2.19), we obtain

$$\Delta t^2 \left(\|p^{n+\frac{1}{2}}\|_A^2 + \|p^{n-\frac{1}{2}}\|_A^2 \right) \leq -2\Delta t (\nabla \cdot \mathbf{u}^{n+1}, p^{n-\frac{1}{2}}) + \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2}^2.$$

Combining this estimate with (2.17) gives

$$\begin{aligned} \|\mathbf{u}^{n+1}\|_{L^2}^2 + \Delta t^2 \left(\|p^{n+\frac{1}{2}}\|_A^2 + \|p^{n-\frac{1}{2}}\|_A^2 \right) + \Delta t \nu \|\nabla \mathbf{u}^{n+1}\|_{L^2}^2 \\ \leq \|\mathbf{u}^n\|_{L^2}^2 + \nu^{-1} \Delta t \|\mathbf{f}\|_{\mathbf{H}^{-1}}^2. \end{aligned} \quad (2.20)$$

The desired result is obtained by summing (2.20) over the time levels from 0 to $N-1$. \square

Remark 2.2. Lemma 2.2 has an obvious two-dimensional counterpart.

Remark 2.3. The advection part of the momentum equation can be discretized by various explicit methods. We use a second order Adams–Bashforth method in the numerical tests reported at the end of the paper. Another possible option could be to use a semi-implicit procedure which can be combined with the direction splitting of the momentum equation. We conjecture that this procedure would probably enhance the stability of the overall algorithm when solving the Navier–Stokes equations.

Remark 2.4. The operators $A = \left(\frac{1}{\beta_x} - \beta_y \partial_{xx}\right) \left(\frac{1}{\beta_y} - \beta_x \partial_{yy}\right)$ in two space dimensions and $A = \left(\frac{1}{\beta_x} - \beta_y \beta_z \partial_{xx}\right) \left(\frac{1}{\beta_y} - \beta_z \beta_x \partial_{yy}\right) \left(\frac{1}{\beta_z} - \beta_x \beta_y \partial_{zz}\right)$ in three space dimensions satisfy the property (2.4) for all positive real numbers $\beta_x, \beta_y, \beta_z$. The stability and error analysis developed in the present paper holds true for this class of operators as well. There may exist sets of coefficients that make the method optimal in some sense, but we have not explored this possibility yet. All the numerical results reported in Section 5 have been obtained with $\beta_x = \beta_y = \beta_z = 1$.

3. Direction splitting/preconditioning/FFT

In this section we discuss the merits of the present approach compared to various preconditioning techniques for solving the Poisson equation.

3.1. Connection with previous works

Starting with the 1955 seminal article of Peaceman and Rachford [20], direction splitting methods have since then been used extensively for solving parabolic equations. The reader is referred to the book of Marchuk [19] for a review.

Applying direction splitting to the Navier–Stokes equations is not a new idea either. For instance, in [25, Section 3.7.2], Temam studies a projection method where the solution of the momentum equation is obtained using direction splitting and the incompressibility constraint is enforced by means of a Poisson equation. Stability and convergence of the scheme are proved therein but no error estimates are provided. Lu et al. show in [17,18] that the scheme proposed by Temam is indeed $\mathcal{O}(\tau^{\frac{1}{2}})$ accurate, τ being the time-step.

Our work differs from that of Temam and Lu et al. mainly in two directions. First, we abandon the projection paradigm and adopt instead the direction splitting strategy for the computation of the pressure-correction as well. This renders the method extremely fast and massively parallelizable. Second, we provide higher-order error estimates for the proposed scheme, and we show that the so-called standard version of the scheme is $\mathcal{O}(\tau)$ -accurate in all quantities irrespective of the space dimension. We show also that the rotational version is $\mathcal{O}(\tau^{\frac{3}{2}})$ -accurate in two space dimensions. Numerical experiments show that the result holds true in three space dimensions and the actual convergence rates are higher than those inferred theoretically.

3.2. Direction splitting vs. preconditioning

The computational complexity per time step of a traditional projection method (either in pressure or velocity correction) is that of solving one vector-valued advection–diffusion equation plus one scalar-valued Poisson equation equipped with Neumann boundary conditions. For large size three-dimensional problems and large Reynolds numbers, the cost of solving the Poisson equation becomes dominant. One possibility to address this issue consists of designing preconditioners for the Poisson equation. The multigrid approach is probably one of the most optimal in this respect. The

downside of most preconditioners though, including multigrid, is that they are not easy to parallelize, which is not the case of the present splitting method. The proposed technique requires the solution of only two (in 2D) or three (in 3D) one-dimensional boundary-value problems for the pressure. If the one-dimensional problems are discretized by means of second-order finite differences, the resulting linear systems are three-diagonal and can be solved very efficiently on very large parallel clusters. If higher-order discretizations are employed, the resulting linear systems have more diagonals but these diagonals are next to each other and therefore the solution of such systems can be obtained very efficiently with a parallel direct method based on a Schur complement technique.

The method that we propose in this paper bears similarities to some preconditioning techniques of the Poisson equation that are based on direction splitting. It is indeed possible to solve the Poisson equation as the steady limit of the heat equation which in turn can be solved by a direction splitting procedure (see [28,19] for details). A recent parallel procedure for the Navier–Stokes equations based on such an approach for solving the Poisson problem is described in [1], see also [3]. In the present paper we go one step further and instead of using the direction splitting to precondition the Pressure Poisson equation, we approximate the pressure by just one application of a direction splitting operator and we prove that the resulting method is indeed stable and convergent.

3.3. Direction splitting vs. Poisson + FFT

The pressure update defined in (2.9) is significantly easier to compute than that given by the usual Poisson equation arising at the projection step of any projection schemes. Of course, one can argue that the pressure Poisson equation can be very efficiently solved using a Fast Fourier Transform (FFT) and therefore we now compare the two techniques.

The convergence analysis presented below (see Theorem 4.2) and in [11] and the numerical results reported in Section 5 show that, under mild regularity conditions, the time accuracy of the direction splitting algorithm is of the same order as that of the equivalent projection scheme. In conclusion, the present approach is not less accurate than the usual projection scheme.

Denoting N the total number of grid points in Ω and assuming that (2.9) is solved by using second-order finite differences on a Cartesian grid, the discrete version of (2.9) can be solved in $O(N)$ operations per time step whereas using FFT requires $O(N \log_2 N)$ operations per time step. In conclusion, the complexity of solving (2.9) with second-order finite differences is smaller than solving the Poisson equation with FFT.

The most significant advantage of the proposed direction splitting approach over the Poisson equation solved with FFT becomes evident when one compares the scalability of each algorithm when implemented on a distributed memory cluster, which seems to be the current trend in parallel computing. For simplicity let us assume that we have a cubic domain containing N grid points and that we employ P processors. A natural way of distributing the grid points when solving (2.9) consists of dividing the processors into a cubic Cartesian set of $P^{1/3}$ processors in each direction so that each processor contains n^3 grid points, where $n = (N/P)^{1/3}$. The amount of data exchange per processor (per linear solve) for the direction splitting algorithm is $6(N/P)^{2/3}$ (see Section 5.4). The situation is slightly different if one solves the Poisson equations by using FFT. To the best of our knowledge, the current parallel FFT methods distribute the data in a two-dimensional fashion, that is to say the data is distributed in parallelepipeds each containing $N^{1/3} \times m^2$ grid points where $m = N^{1/3}/P^{1/2}$ (see e.g., [23]). Then each processor needs to exchange $3(N^{1/3} - m)m^2 = 3(1 - 1/\sqrt{P})N/P$

values. The quantity $3(1 - 1/\sqrt{P})N/P$ is significantly larger than $6(N/P)^{2/3}$ when N/P is large. For instance if $N/P = 10^6$, then $\frac{3}{6}(N/P)^{2/3} = 50$; in this case the FFT involves 50 times more communications than solving (2.9) with a Schur complement technique. This implies that solving (2.9) instead of solving the Poisson equation with FFT allows to utilize a larger number of processors and to solve larger problems than with FFT since the processors can be arranged in a 3D rather than in a 2D Cartesian fashion. In conclusion, the weak scalability of solving (2.9) is better than that of solving the Poisson equation with FFT.

4. Higher-order variants

It is well known that the time accuracy of the Chorin–Temam scheme is limited and yields error estimates similar to those stated in Proposition 2.1. The purpose of this section is to introduce higher-order versions of the method by using the arsenal of the incremental schemes (see e.g., [26] for the so-called rotational variants and [12] for a complete review).

4.1. Singular perturbation analysis

Virtually all currently known incremental pressure-correction schemes are more or less semi-discrete versions of the following singular perturbation of (2.1):

$$\begin{cases} \partial_t \mathbf{u}_\epsilon - \nu \Delta \mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f} & \text{in } \Omega \times [0, T], & \mathbf{u}_\epsilon|_{\partial\Omega \times [0, T]} = 0, & \mathbf{u}_\epsilon|_{t=0} = \mathbf{u}_0, \\ -\Delta t \Delta \phi_\epsilon + \nabla \cdot \mathbf{u}_\epsilon = 0 & \text{in } \Omega \times [0, T], & \partial_n \phi_\epsilon|_{\partial\Omega \times [0, T]} = 0, \\ \Delta t \partial_t p_\epsilon = \phi_\epsilon - \chi \nu \nabla \cdot \mathbf{u}_\epsilon & & p_\epsilon|_{t=0} = p_0, \end{cases} \quad (4.1)$$

where Δt is the perturbation parameter (i.e., $\epsilon := \Delta t$) and $\chi \in [0, 1]$ is an adjustable parameter. This problem has been analyzed in [15, Section 3.3] and [14, Section 3.1] and \mathbf{u}_ϵ has been shown therein to be a $\mathcal{O}(\Delta t^2)$ perturbation of \mathbf{u} in the L^2 -norm and a $\mathcal{O}(\Delta t^{\frac{3}{2}})$ perturbation for all $0 < \chi \leq 1$.

We now introduce a generalization of (4.1) which allows for direction splitting by considering the following alternative $\mathcal{O}(\Delta t^2)$ -perturbation of (2.1):

$$\begin{cases} \partial_t \mathbf{u}_\epsilon - \nu \Delta \mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f} & \text{in } \Omega \times [0, T], & \mathbf{u}_\epsilon|_{\partial\Omega \times [0, T]} = 0, & \mathbf{u}_\epsilon|_{t=0} = \mathbf{u}_0, \\ \Delta t A \phi_\epsilon + \nabla \cdot \mathbf{u}_\epsilon = 0 & \text{in } \Omega \times [0, T], & \phi_\epsilon \in D(A), \\ \Delta t \partial_t p_\epsilon = \phi_\epsilon - \chi \nu \nabla \cdot \mathbf{u}_\epsilon & & p_\epsilon|_{t=0} = p_0. \end{cases} \quad (4.2)$$

Proposition 4.1. *Assume that the solution to (2.1) is smooth enough with respect to time and space and that (2.4) holds, then the solutions to (4.1) and (4.2) both satisfy the following estimates*

$$\|\mathbf{u} - \mathbf{u}_\epsilon\|_{L^2((0, T); \mathbf{H}^1(\Omega))} \leq c \Delta t, \quad \text{if } \chi = 0. \quad (4.3)$$

$$\|\mathbf{u} - \mathbf{u}_\epsilon\|_{L^2((0, T); \mathbf{H}^1(\Omega))} \leq c \Delta t^{\frac{3}{2}}, \quad \text{if } \chi \in (0, 1]. \quad (4.4)$$

Remark 4.1. When $A = -\Delta_N$ and $\chi \in (0, 1]$ it is known that u_ϵ is a $\mathcal{O}(\Delta t^2)$ perturbation of u in the L^2 -norm. Whether the $\mathcal{O}(\Delta t^2)$ perturbation estimate holds in general (i.e., when A induces a norm which is not equivalent to the H^1 -norm) is an open question, although numerical tests seem to show that it should hold. The main problem one encounters when trying to prove this estimate is that the duality argument which is usually used to prove second-order accuracy on the velocity in the L^2 -norm (see [14]) does not easily generalize when the bilinear form a is not bounded in H^1 , see [11].

4.2. Formulation of the scheme

We now describe an algorithm based on (4.2) that uses the direction splitting operator A defined in (2.10) and (2.11).

4.2.1. Pressure predictor

Denoting p_0 the pressure field at $t = 0$, the algorithm is initialized by setting $p^{-\frac{1}{2}} = p^{-\frac{3}{2}} = p_0$. Then for all $n > 0$ a pressure predictor is computed as follows:

$$p^{*,n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n-\frac{1}{2}}. \tag{4.5}$$

4.2.2. Velocity update

The velocity is updated by proceeding exactly as in the non-incremental version of the algorithm described in Section 2.3.2. The velocity field is initialized by setting $\mathbf{u}^0 = \mathbf{u}_0$, and for all $n > 0$ the velocity update is computed by solving the following series of one-dimensional problems:

$$\begin{aligned} \frac{\xi^{n+1} - \mathbf{u}^n}{\Delta t} - v\Delta \mathbf{u}^n + \nabla p^{*,n+\frac{1}{2}} &= \mathbf{f}(t^{n+\frac{1}{2}}), \quad \mathbf{u}^n|_{\partial\Omega} = 0, \\ \frac{\boldsymbol{\eta}^{n+1} - \xi^{n+1}}{\Delta t} - \frac{v}{2}\partial_{xx}(\boldsymbol{\eta}^{n+1} - \mathbf{u}^n) &= 0, \quad \boldsymbol{\eta}^{n+1}|_{x=0,1} = 0, \\ \frac{\zeta^{n+1} - \boldsymbol{\eta}^{n+1}}{\Delta t} - \frac{v}{2}\partial_{yy}(\zeta^{n+1} - \mathbf{u}^n) &= 0, \quad \zeta^{n+1}|_{y=0,1} = 0, \\ \frac{\mathbf{u}^{n+1} - \zeta^{n+1}}{\Delta t} - \frac{v}{2}\partial_{zz}(\mathbf{u}^{n+1} - \mathbf{u}^n) &= 0, \quad \mathbf{u}^{n+1}|_{z=0,1} = 0. \end{aligned} \tag{4.6}$$

4.2.3. Penalty step

The intermediate parameter ϕ_ϵ introduced in (4.2) is approximated by solving $A\phi^{n+\frac{1}{2}} = -\frac{1}{\Delta t}\nabla \cdot \mathbf{u}^{n+1}$. Owing to the definition of the direction splitting operator A , this is done by solving the following series of one-dimensional problems:

$$\begin{aligned} \psi - \partial_{xx}\psi &= -\frac{1}{\Delta t}\nabla \cdot \mathbf{u}^{n+1}, \quad \partial_x\psi|_{x=0,1} = 0, \\ \varphi - \partial_{yy}\varphi &= \psi, \quad \partial_y\varphi|_{y=0,1} = 0, \\ \phi^{n+\frac{1}{2}} - \partial_{zz}\phi^{n+\frac{1}{2}} &= \varphi, \quad \partial_z\phi^{n+\frac{1}{2}}|_{z=0,1} = 0. \end{aligned} \tag{4.7}$$

4.2.4. Pressure update

The last sub-step of the algorithm consists of updating the pressure as follows:

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n+\frac{1}{2}} - \chi v \nabla \cdot \left(\frac{1}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n) \right). \tag{4.8}$$

We say that the algorithm is in standard incremental form when we choose $\chi = 0$ and the algorithm is in rotational incremental form when we choose $\chi \in (0, 1]$.

4.3. Stability and error analysis

To simplify the notation we now define the following time-increment operator:

$$\delta p^{n+\frac{1}{2}} := p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}, \quad \delta^2 p^{n+\frac{1}{2}} := p^{n+\frac{1}{2}} - 2p^{n-\frac{1}{2}} + p^{n-\frac{3}{2}}. \tag{4.9}$$

We also denote $\bar{\mathbf{u}}$ the sequence whose generic term is $\bar{\mathbf{u}}^{n+\frac{1}{2}} = \frac{1}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n)$. The main result of this section is the following:

Theorem 4.2. Assume that the solution to (2.1) is smooth enough with respect to time and space and that (2.4) holds. There exist c_1 , uniform in Δt , so that for all T and all $\chi \in [0, 1]$ the solution to (4.5)–(4.8), with $p^{-\frac{1}{2}} = p^{-\frac{3}{2}} = p_0$, satisfies the following stability estimate:

$$\begin{aligned} &\|\mathbf{u}\|_{L^\infty(0,T;L^2)}^2 + v\|\nabla \bar{\mathbf{u}}\|_{L^2(\frac{\Delta t}{2},T-\frac{\Delta t}{2};L^2)}^2 + \Delta t^2\|p\|_{L^\infty(\frac{\Delta t}{2},T-\frac{\Delta t}{2};D(A))} \\ &\leq c_1 \left(\|\mathbf{u}^0\|_{L^2}^2 + \Delta t^2\|p^{-\frac{1}{2}}\|_A^2 + \Delta tv\|\nabla \mathbf{u}_0\|_{L^2}^2 + v^{-1}\|\mathbf{f}\|_{L^2(\frac{\Delta t}{2},T-\frac{\Delta t}{2};H^{-1})} \right). \end{aligned} \tag{4.10}$$

For all T there is c_2 , uniform in Δt , so that

$$\|\mathbf{u} - \bar{\mathbf{u}}\|_{L^\infty(0,T;L^2)} + \sqrt{v}\|\nabla(\mathbf{u} - \bar{\mathbf{u}})\|_{L^2(0,T;L^2)} \leq c_2\Delta t, \quad \text{if } \chi \in [0, 1], \tag{4.11}$$

$$\|\mathbf{u} - \bar{\mathbf{u}}\|_{L^2(0,T;L^2)} + \sqrt{v}\|\nabla \cdot \mathbf{u}\|_{L^\infty(0,T;L^2)} \leq c_2\Delta t^{\frac{3}{2}}, \quad \text{if } \chi \in (0, 1]. \tag{4.12}$$

Proof. We are going to prove (4.10) for $\chi = 0$ and $f = 0$ only. The full proof of the theorem is given in [11]. To avoid mixing issues and to avoid technicalities we assume again that the momentum equation is not split, i.e., we replace (4.6) by

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} - \frac{1}{2}v\Delta(\mathbf{u}^{n+1} + \mathbf{u}^n) + \nabla \tilde{p}^{*,n+\frac{1}{2}} = 0, \quad \mathbf{u}|_{\partial\Omega} = 0. \tag{4.13}$$

We multiply (4.13) by $2\Delta t\mathbf{u}^{n+1}$, integrate over Ω , and use the identity $2(a - b, a) = \|a\|^2 + \|a - b\|^2 - \|b\|^2$ to obtain

$$\begin{aligned} &\|\mathbf{u}^{n+1}\|_{L^2}^2 + \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2}^2 + \frac{1}{2}\Delta tv \left(\|\nabla \mathbf{u}^{n+1}\|_{L^2}^2 + 4\|\nabla \bar{\mathbf{u}}^{n+\frac{1}{2}}\|_{L^2}^2 \right) \\ &+ 2\Delta t \left(\nabla p^{*,n+\frac{1}{2}}, \mathbf{u}^{n+1} \right) = \|\mathbf{u}^n\|_{L^2}^2 + \frac{1}{2}\Delta tv \|\nabla \mathbf{u}^n\|_{L^2}^2. \end{aligned} \tag{4.14}$$

Now we use Lemma 2.2 to deduce that the pressure correction $(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \in D(A)$ solves the following problem for $n \geq 0$ and $\chi = 0$:

$$a(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}, q) = -\Delta t^{-1}(\nabla \cdot \mathbf{u}^{n+1}, q), \quad \forall q \in D(A). \tag{4.15}$$

Using $2\Delta t^2 p^{*,n+\frac{1}{2}} := 2\Delta t^2(2p^{n-\frac{1}{2}} - p^{n-\frac{3}{2}})$ as a test function together with the fact that $a(\cdot, \cdot)$ is symmetric and coercive in $H^1(\Omega)$, we infer that

$$\begin{aligned} &-2\Delta t(\nabla \cdot \mathbf{u}^{n+1}, p^{*,n+\frac{1}{2}}) \\ &= 2\Delta t^2 a(\delta p^{n+\frac{1}{2}}, 2p^{n-\frac{1}{2}} - p^{n-\frac{3}{2}}) = 2\Delta t^2 a(\delta p^{n+\frac{1}{2}}, p^{n+\frac{1}{2}}) \\ &+ 2\Delta t^2 a(\delta p^{n+\frac{1}{2}}, -\delta p^{n+\frac{1}{2}} + \delta p^{n-\frac{1}{2}}) \\ &= \Delta t^2 \left(\|p^{n+\frac{1}{2}}\|_A^2 + \|\delta p^{n+\frac{1}{2}}\|_A^2 - \|p^{n-\frac{1}{2}}\|_A^2 - \|\delta p^{n+\frac{1}{2}}\|_A^2 - \|\delta^2 p^{n+\frac{1}{2}}\|_A^2 + \|\delta p^{n-\frac{1}{2}}\|_A^2 \right). \end{aligned}$$

This then implies that

$$\begin{aligned} &-2\Delta t(\nabla \cdot \mathbf{u}^{n+1}, p^{*,n+\frac{1}{2}}) \\ &= \Delta t^2 \left(\|p^{n+\frac{1}{2}}\|_A^2 - \|p^{n-\frac{1}{2}}\|_A^2 + \|\delta p^{n-\frac{1}{2}}\|_A^2 - \|\delta^2 p^{n+\frac{1}{2}}\|_A^2 \right). \end{aligned} \tag{4.16}$$

We now seek a control on $\|\delta^2 p^{n+\frac{1}{2}}\|_A^2$ by subtracting (4.15) at time t^n from (4.15) at time t^{n+1} and by testing the result with $\Delta t\delta^2 p^{n+\frac{1}{2}}$,

$$\begin{aligned} \Delta t\|\delta^2 p^{n+\frac{1}{2}}\|_A^2 &= -(\nabla \cdot (\mathbf{u}^{n+1} - \mathbf{u}^n), \delta^2 p^{n+\frac{1}{2}}) = (\mathbf{u}^{n+1} - \mathbf{u}^n, \nabla \delta^2 p^{n+\frac{1}{2}}) \\ &\leq \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2} \|\nabla \delta^2 p^{n+\frac{1}{2}}\|_{L^2}. \end{aligned}$$

(Note that the above operation is legitimate also for $n = 0$ since (4.15) holds for $n = -1$ owing to the definition $p^{-\frac{1}{2}} = p^{-\frac{3}{2}}$ and $\nabla \cdot \mathbf{u}^0 = \nabla \cdot \mathbf{u}_0 = 0$.) Then using the coercivity property of the bilinear form a we infer that

$$\Delta t\|\nabla \delta^2 p^{n+\frac{1}{2}}\|_{L^2} \|\delta^2 p^{n+\frac{1}{2}}\|_A \leq \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2} \|\nabla \delta^2 p^{n+\frac{1}{2}}\|_{L^2},$$

which then implies that $\Delta t^2\|\delta^2 p^{n+\frac{1}{2}}\|_A^2 \leq \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2}^2$. Inserting this bound into (4.16), we obtain

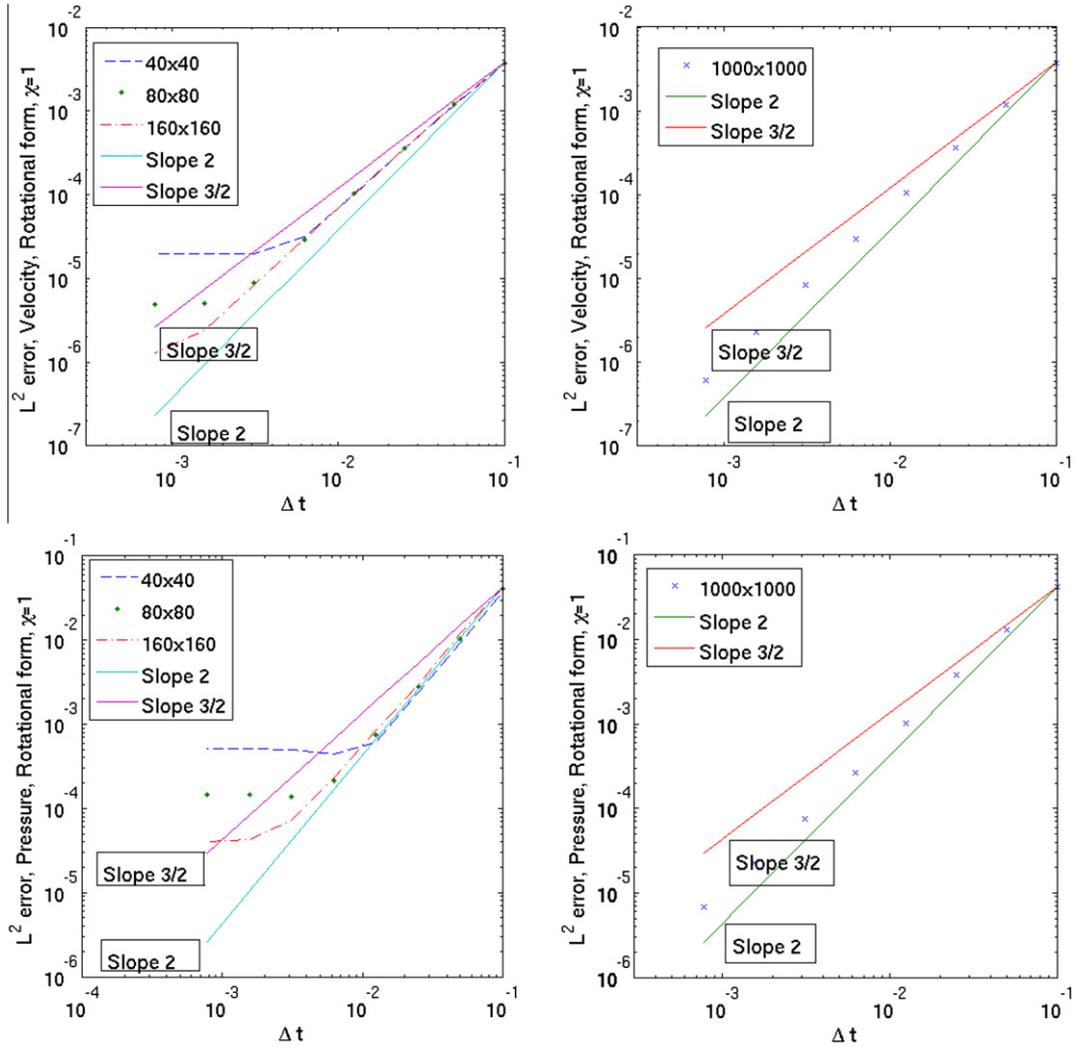


Fig. 5.1. Rotational form ($\chi = 1$). L^2 -norm of the error at $T = 2$ on uniform grids. Top: velocity. Bottom: pressure. Left: 40×40 (dashed line), 80×80 (symbols) and 160×160 (dash-dotted line). Right: 1000×1000 uniform grid (symbols).

$$\Delta t^2 \left(\|p^{n+\frac{1}{2}}\|_A^2 + \|\delta p^{n-\frac{1}{2}}\|_A^2 - \|p^{n-\frac{1}{2}}\|_A^2 \right) \leq -2\Delta t (\nabla \cdot \mathbf{u}^{n+1}, p^{n-\frac{1}{2}}) + \|\mathbf{u}^{n+1} - \mathbf{u}^n\|_{L^2}^2.$$

Combining this estimate with (4.14) gives

$$\|\mathbf{u}^{n+1}\|_{L^2}^2 + \Delta t^2 \|p^{n+\frac{1}{2}}\|_A^2 + \frac{1}{2} \Delta t v \left(\|\nabla \mathbf{u}^{n+1}\|_{L^2}^2 + 4 \|\nabla \mathbf{u}^{n+\frac{1}{2}}\|_{L^2}^2 \right) \leq \|\mathbf{u}^n\|_{L^2}^2 + \Delta t^2 \|p^{n-\frac{1}{2}}\|_A^2 + \frac{1}{2} \Delta t v \|\nabla \mathbf{u}^n\|_{L^2}^2. \quad (4.17)$$

The desired result is obtained by summing (4.17) over the time levels from 0 to $N - 1$. \square

Remark 4.2. The stability analysis of the above algorithm with the pressure predictor $2p^{n-\frac{1}{2}} - 2p^{n-\frac{3}{2}}$ (as announced in [9]) reveals that the algorithm is stable for $\chi \in [0, \chi_0)$ for some χ_0 small enough. Our numerical tests have revealed that χ_0 depends on the space discretization and $\chi_0 \approx 0.6$ for the MAC scheme. The stability analysis of the algorithm using the pressure predictor $p^{n-\frac{1}{2}} + \phi^{n+\frac{1}{2}}$ shows that the algorithm is stable for all $\chi \in [0, 1]$ (see [11]). This shows that the pressure predictor $p^{n-\frac{1}{2}} + \phi^{n+\frac{1}{2}}$ should be preferred to $2p^{n-\frac{1}{2}} - 2p^{n-\frac{3}{2}}$.

5. Numerical results

We report in this section numerical tests illustrating the performance of the algorithm described above. The tests are made in two and three space dimensions using second-order central differences on a MAC stencil. Weak scalability tests reported in Section 5.4 show that the method scales very well.

5.1. Convergence tests

The algorithm (4.5)–(4.8) is tested numerically on the following smooth solution of the unsteady Stokes equations (with the source term chosen properly):

$$\mathbf{u} = (\sin x \sin(y + t), \cos x \cos(y + t)), \quad p = \cos x \sin(y + t). \quad (5.1)$$

The problem is solved in $\Omega = (0, 1) \times (0, 1)$, for $0 \leq t \leq T := 2$ with Dirichlet boundary conditions (given by the pointwise values of the exact solution). The initial condition is the exact solution at $t = 0$. The accuracy of the incremental algorithm in standard and rotational forms is measured on four uniform meshes 40×40 , 80×80 , 160×160 , and 1000×1000 , with $\Delta t \in \{0.00625, 0.0125, 0.025, 0.05, 0.1\}$. The rotational version of the method is obtained by setting $\chi = 1$. The L^2 -norm of the error on the velocity and the

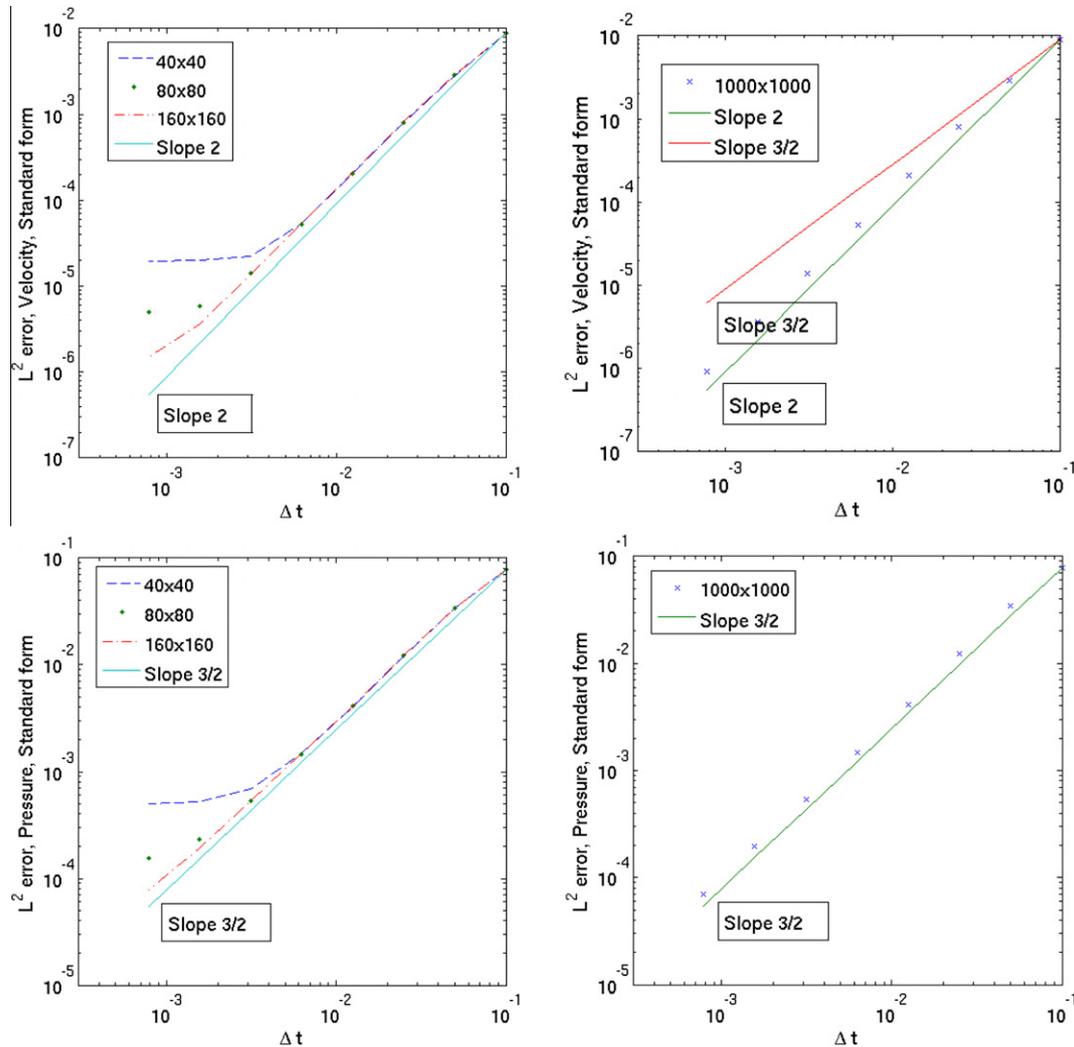


Fig. 5.2. Standard form ($\chi = 0$). Errors at $T = 2$ on uniform grids. Top: L^2 -norm on the velocity. Bottom: L^2 -norm on the pressure. Left: 40×40 (dashed line), 80×80 (symbols) and 160×160 (dash-dotted line). Right: 1000×1000 uniform grid (symbols).

L^2 -norm of error on the pressure at $T = 2$ are shown in Fig. 5.1 for the rotational form of the scheme and in Fig. 5.2 for the standard one.

Since the velocity is computed on the MAC grid and the grid points for the two Cartesian components are staggered, the error is computed by averaging the Cartesian components at the center of each square cell. This does not spoil the spatial error since the averaging is second-order accurate.

The left panels in Figs. 5.1 and 5.2 demonstrate second-order convergence in space on the L^2 -norm of velocity and the L^2 -norm of the pressure at small time steps ($\Delta t = 0.0007825$) and relatively coarse grids ($h = 1/40$ – $h = 1/160$) where the spatial error dominates. At larger time steps the convergence in time is very close to second-order.

In order to quantify the convergence rate with respect to Δt , we present in the right panels of Figs. 5.1 and 5.2 the L^2 -norm of the error on the velocity and the L^2 -norm of the error on pressure on a relatively fine grid ($h = 0.001$). Both the rotational and standard forms exhibit convergence rate in time that are close to second-order on the L^2 -norm of the velocity. The situation is slightly different for the pressure. The convergence rate in time on the L^2 -norm of the pressure is close to $\frac{3}{2}$ for the standard form of the algorithm and close to 2 for the rotational form. In all the cases the observed convergence rates in time are higher than that those stated in Theorem 4.2;

5.2. 2D driven cavity

We now compute the solution to the well known lid-driven cavity problem in $\Omega = (0, 1)^2$, using (4.5)–(4.8) with $\chi = \frac{1}{2}$. (The steady state is independent of χ .) Note that the purpose of this test is solely meant to show that long time integration of (4.5)–(4.8) is convergent. Accurate time-marching schemes are not CPU efficient to compute steady-state solutions when compared to well-tuned Newton-based steady-state solvers. In all the cases shown below the computation starts with zero initial data and the horizontal component of the velocity is set to one on the lid.

We show in Fig. 5.3 the horizontal velocity profile along the vertical line $x = \frac{1}{2}$ at $Re = 1000$ (left panel) and $Re = 5000$ (right panel). We compare our results obtained on a uniform grid composed of 5000×5000 grid points with the results of [6] which have been computed on a uniform grid of 600×600 grid points. Both sets of results are close to each other. Our results match those of [6] up to the fourth digit and those of [4] up to the fifth digit at $Re = 1000$.

5.3. 3D driven cavity

The final validation test consists of solving the three-dimensional flow in a lid-driven cavity in $\Omega = (0, 1) \times (0, 1) \times (0, 2)$. Denoting x, y, z the Cartesian coordinates, the driving lid is the side

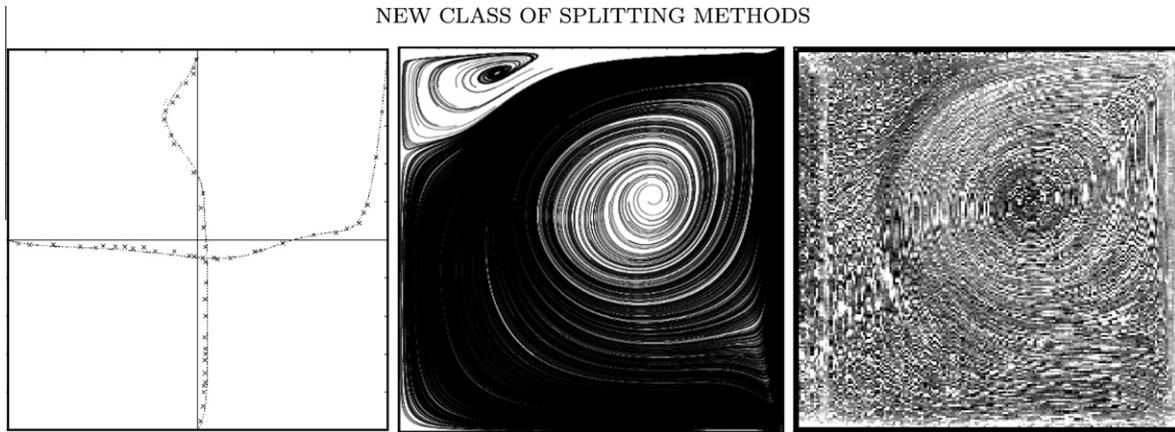


Fig. 5.4. Left: Horizontal component of the velocity along the vertical line through $x = 0.5, y = 0.5, z = 1$ and vertical component of the velocity along the horizontal line through the same point: results with (4.5)–(4.8) (dashed line), computational results from [8] (dotted line), and experimental results from [8] (symbols \times). Center and Right: Streamlines in the plane $z = 1$ at $t = 8$. Present computational results (left) and experimental results (right).

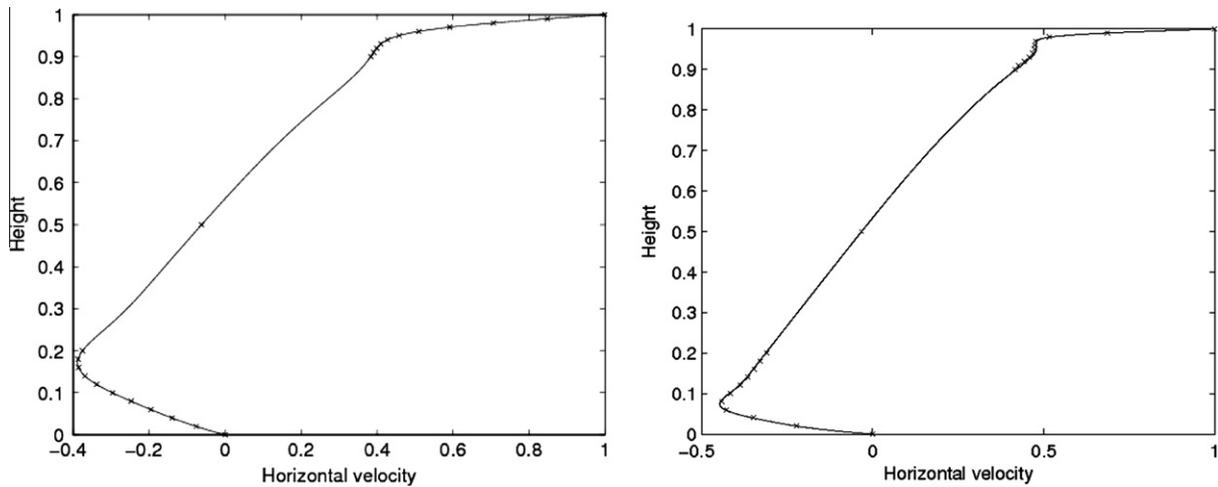


Fig. 5.3. Horizontal component of the velocity along the vertical line through the center of the cavity: results with the rotational version of the scheme on a 5000×5000 uniform grid (continuous line) and benchmark results of [6] (\times). Left graph: $Re = 1000$; Right graph: $Re = 5000$.

wall at $x = 1$. This flow has been studied by Guermond et al. [8] at $Re = 1000$. [8] presents both experimental and computational results at various times after the impulsive start.

We present in the left panel of Fig. 5.4 the graph of the horizontal component of the velocity along the vertical line in the plane $z = 1$ passing through $x = 0.5, y = 0.5, z = 1$ at $t = 4$ and the vertical component of the velocity along the horizontal line passing through the same point and in the same plane at $t = 4$ also. The results obtained with (4.5)–(4.8) are shown in dashed lines. The numerical results from [8] are shown in dotted lines, and the experimental results reported in [8] are shown with symbols \times . The results shown in this figure have been produced using the rotational version of the algorithm with $\chi = 0.5$, on a non-uniform cosine-type grid as defined in [8]. The grid consists of $80 \times 80 \times 160$ grid cells. The computational results from [8] are produced on a $\mathbb{P}_2 - \mathbb{P}_1$ finite element cosine-type grid and containing $39 \times 39 \times 60$ hexahedra each subdivided into five tetrahedra. The solver used in [8] is the standard version of the incremental pressure correction scheme with a semi-implicit treatment of the advection terms.

A very detailed benchmark solution of this flow using (4.5)–(4.8) with $\chi = 1$ and grids consisting of up to 2×10^9 grid

points will be presented in a forthcoming paper (see [10]); these computations reveal that the results (dashed lines) shown in the left panel of Fig. 5.4 are very close to the converged solution.

We show in center and right panels of Fig. 5.4 the flow streamlines in the plane $z = 1$ at $t = 8$. The computational solution is shown in the center panel and the experimental streamlines from [8] are shown in the right panel. We observe that the position of the centers of the vortices in the two pictures are in a very good agreement.

5.4. Parallel implementation

The algorithm (4.5)–(4.8) has been implemented in a parallel code using MPI. The space approximation is done using second-order central differences on a MAC stencil. The parallelization is based on a Cartesian block decomposition of Ω with equal number of grid points in each block.

Once approximated in space, all the one-dimensional linear problems give rise to tridiagonal linear systems. We have chosen to solve these systems by means of a Schur complement technique based on the unknowns located at the interfaces between the subdomains. The only communications between subdomains that are

Table 5.1

Weak scalability: CPU time (in second) per time step for (4.5)–(4.8) + explicit nonlinear terms.

# procs	2.710^4 nds/proc	2.1610^5 nds/proc	10^6 nds/proc
$1 \times 1 \times 1$	0.056 s	0.41 s	2.0 s
$8 \times 8 \times 8$	0.077 s	0.54 s	2.3 s
$8 \times 8 \times 16$	0.094 s	0.55 s	2.34 s

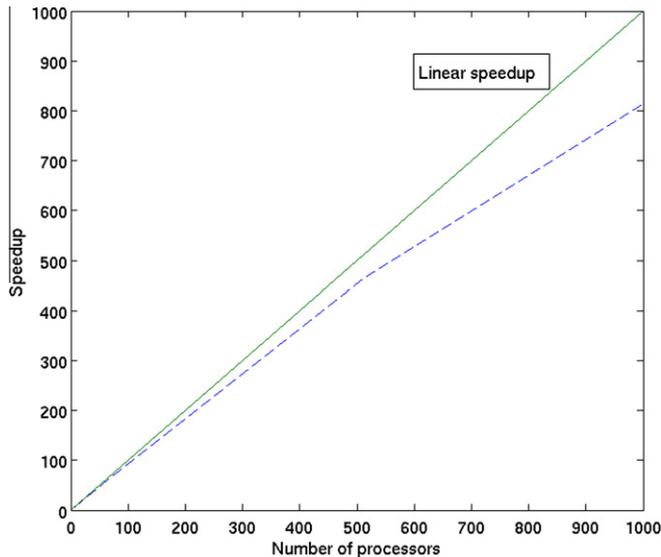


Fig. 5.5. Speedup (dashed line) of the code used for this paper on the Hurr HPC cluster of IAMCS at Texas A&M University with infiniband interconnect (algorithm (4.5)–(4.8)). The ideal linear speedup is plotted with a solid line.

required by (4.5)–(4.8) arise when solving the Schur complement systems and when assembling the corresponding right-hand sides. The Schur complement matrices are tridiagonal and are assembled only once at a pre-processing step. Denoting by P_0, \dots, P_s the processors involved in the assembling of a Schur matrix, the matrix is assembled on P_0 and all the processors communicate with P_0 . The Schur linear system is solved by means of the direct Thomas algorithm. This is extremely fast. For instance, on a cubic grid composed of 1000 processors, the Schur systems are 9×9 . Once the Schur complement linear systems are solved, the solutions are communicated back to the other processors P_1, \dots, P_s and the interior unknowns in each sub-domain are obtained by solving independent tridiagonal linear systems. The overall algorithm requires only one exchange of the interface data per solve in each direction and per time step. Therefore the parallel efficiency of the algorithm is comparable to that of a fully explicit method.

We show weak scalability tests of the code in Table 5.1. Weak scalability tests consists of fixing the number of grid points per processor and observing the CPU time as the number of processor increases. The tests have been performed while solving the 3D lid-driven cavity problem mentioned in Section 5.3. The time reported in each column corresponds to a fixed number of grid points per processor (2.7×10^4 in column 1; 2.16×10^5 in column 2; and 10^6 in column 3). The CPU times reported in each row of the table correspond to a fixed number of processors (1 processor in row 1; 512 processors in row 2; 1024 processors in row 3). For instance, with 10^6 grid points per processor, the CPU time per time step goes from 2.0 s on one processor to 2.34 s on 1024 processors (this was the maximum number of processors available per user on the Hurr HPC cluster of IAMCS at Texas A&M University at the time of this test). This shows that the weak scalability of the algorithm is rea-

sonable considering that no attempt had been made to hide the communications or to optimize the code.

The strong scalability performance of the algorithm is illustrated in Fig. 5.5. The speedup on p processors is defined as the ratio of the CPU time necessary to solve the problem on one processor divided by the CPU time needed to solve it on p processors. We show the speedup on up to 1000 processors for a Navier–Stokes problem discretized on grid composed of $400 \times 400 \times 400 = 64 \times 10^6$ nodes. The computations have been done over ten time steps with 1,64,512 and 1000 processors. Fig. 5.5 shows that our direction splitting code has a speedup close to the ideal speedup on up to 1000 processors.

6. Conclusions

We have proposed a new class of splitting schemes generalizing the pressure-correction methods for the incompressible Navier–Stokes equations. The main idea consists of replacing the Laplace operator in the pressure-correction step by a more general symmetric positive definite operator. This new idea allowed us to introduce a splitting technique based entirely on direction splitting. One striking feature of this new algorithm is that it has the same stability and convergence properties as the pressure-correction method, either in standard or rotational form. When combined with a second-order central difference scheme for the second derivatives, the algorithm only requires the solution of tridiagonal linear systems, independently of the dimensionality of the problem. Therefore, this new algorithm is computationally very efficient, with a computational complexity of the same order as that of an explicit scheme, and yet, unconditionally stable. This scheme is particularly convenient for parallel implementation and demonstrates excellent weak and strong scalability up to 1024 processors.

As described in this paper, the direction splitting algorithm can be applied only to problems posed in simple domains that are composed of the union of parallelepipeds whose faces are parallel to the coordinate axes. This setting is nevertheless appropriate for a large class of problems in science and engineering. We are thinking in particular of academic problems that can be solved in simple geometries e.g., simulation of turbulent flows in the atmosphere and in the ocean, stratified flows, variable density flows, combustion, solution of subgrid problems as part of an homogenization procedure, etc. We believe also that the algorithm can also be applied to problems posed in more complex geometries. This can be done for instance by combining it with a fictitious domain formulation as that described in [27] or and immersed boundary formulation like that of Peskin [21]. Another option could be to use penalty methods as in [2] or a boundary fitting technique using directional grid adjustment as in [7]. The authors have started exploring the possibility of using directional grid adjustment for the Dirichlet boundaries and have observed that the resulting scheme is unconditionally stable and convergent for the time-dependent Stokes problem. These results along with the stability and convergence analysis of this adaptive algorithm will be reported elsewhere.

References

- [1] I.I. Albarreal Núñez, M.C. Calzada Canalejo, J.L. Cruz Soto, E. Fernández Cara, J.R. Galo Sánchez, M. Marín Beltrán, Time and space parallelization of the Navier–Stokes equations, *Comput. Appl. Math.* 24 (3) (2005) 417–438.
- [2] Ph. Angot, Analysis of singular perturbations on the Brinkman problem for fictitious domain models of viscous flows, *Math. Methods Appl. Sci.* 22 (16) (1999) 1395–1412.
- [3] J. Blasco, M.C. Calzada, M. Marín, A fictitious domain, parallel numerical method for rigid particulate flows, *J. Comput. Phys.* 228 (20) (2009) 7596–7613.
- [4] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, *Comput. Fluids* 27 (4) (1998) 421–433.

- [5] A.J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* 22 (1968) 745–762.
- [6] E. Erturk, T.C. Corke, C. Gokcol, Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers, *Int. J. Numer. Methods Fluids* 48 (2005) 747–774.
- [7] J.R. Galo, I.I. Albarreal, M.C. Calzada, J.L. Cruz, E. Fernández-Cara, M. Marín, Convergence and optimization of the parallel method of simultaneous directions for the solution of elliptic problems, *J. Comput. Appl. Math.* 222 (2) (2008) 458–476.
- [8] J.-L. Guermond, C. Migeon, G. Pineau, L. Quartapelle, Start-up flows in a three-dimensional rectangular driven cavity of aspect ratio 1:1:2 at $Re = 1000$, *J. Fluid Mech.* 450 (2002) 169–199.
- [9] J.-L. Guermond, P.D. Mineev, A new class of fractional step techniques for the incompressible Navier–Stokes equations using direction splitting, *Comptes Rendus Mathématique* 348 (9–10) (2010) 581–585.
- [10] J.-L. Guermond, P.D. Mineev, Start-up flow in a three-dimensional lid-driven cavity by means of a massively parallel direction splitting algorithm, *Int. J. Numer. Methods Fluids*, in press.
- [11] J.-L. Guermond, P.D. Mineev, Abner Salgado, Convergence analysis of a class of massively parallel direction splitting algorithms for the Navier–Stokes equations, *Math. Comput.* submitted for publication.
- [12] J.-L. Guermond, P.D. Mineev, Jie Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Engrg.* 195 (2006) 6011–6054.
- [13] J.-L. Guermond, Abner Salgado, A splitting method for incompressible flows with variable density based on a pressure Poisson equation, *J. Comput. Phys.* 228 (8) (2009) 2834–2846.
- [14] J.L. Guermond, Jie Shen, On the error estimates for the rotational pressure-correction projection methods, *Math. Comput.* 73 (248) (2004) 1719–1737 (electronic).
- [15] J.L. Guermond, Jie Shen, Velocity-correction projection methods for incompressible flows, *SIAM J. Numer. Anal.* 41 (1) (2003) 112–134.
- [16] J. Douglas Jr., Alternating direction methods for three space variables, *Numer. Math.* 4 (1962) 41–63.
- [17] T. Lu, P. Neittaanmäki, X.-C. Tai, A parallel splitting up method and its application to Navier–Stokes equations, *Appl. Math. Lett.* 4 (2) (1991) 25–29.
- [18] T. Lu, P. Neittaanmäki, X.-C. Tai, A parallel splitting-up method for partial differential equations and its applications to Navier–Stokes equations, *RAIRO Modél. Math. Anal. Numér.* 26 (6) (1992) 673–708.
- [19] G.I. Marchuk, Splitting and alternating direction methods, in: P.G. Ciarlet, J.L. Lions (Eds.), *Handbook of Numerical Analysis*, vol. 1, North-Holland, Amsterdam, 1990, pp. 199–462.
- [20] D.W. Peaceman, H.H. Rachford Jr., The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. Appl. Math.* 3 (1) (1955) 28–41.
- [21] C. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220.
- [22] R. Rannacher, On Chorin’s projection method for the incompressible Navier–Stokes equations, *Lect. Notes Math.* 1530 (1991).
- [23] R. Schulz, 3D FFT with 2D decomposition, CS project report: <<http://cmb.ornl.gov/Members/z8g/csproject-report.pdf>>, 2008.
- [24] R. Temam, Sur l’approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires ii, *Arch. Rat. Mech. Anal.* 33 (1969) 377–385.
- [25] R. Temam, *Navier–Stokes equations*, AMS Chelsea Publishing, Providence, RI, 2001. Theory and numerical analysis, Reprint of the 1984 edition.
- [26] L.J.P. Timmermans, P.D. Mineev, F.N. Van De Vosse, An approximate projection scheme for incompressible flow using spectral elements, *Int. J. Numer. Methods Fluids* 22 (1996) 673–688.
- [27] C. Veeramani, P.D. Mineev, K. Nandakumar, A fictitious domain method for particle sedimentation, *Lect. Notes Comput. Sci.* 3743 (2005) 544.
- [28] N.N. Yanenko, *The Method of Fractional Steps, The Solution of Problems of Mathematical Physics in Several Variables*, Springer-Verlag, New York, 1971.