Journal of Computational Physics 229 (2010) 7013–7029

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An unconditionally stable rotational velocity-correction scheme for incompressible flows

S. Dong [*,1], J. Shen [2]

Department of Mathematics, Purdue University, United States

ARTICLE INFO

Article history:
Received 11 December 2009
Received in revised form 31 March 2010
Accepted 25 May 2010
Available online 2 June 2010

Keywords:
Unconditional stability
Velocity-correction scheme
Spectral element method
Navier-Stokes equation

ABSTRACT

We present an unconditionally stable splitting scheme for incompressible Navier–Stokes equations based on the rotational velocity-correction formulation. The main advantages of the scheme are: (i) it allows the use of time step sizes considerably larger than the widely-used semi-implicit type schemes: the time step size is only constrained by accuracy; (ii) it does not require the velocity and pressure approximation spaces to satisfy the usual inf–sup condition: in particular, the equal-order finite element/spectral element approximation spaces can be used; (iii) it only requires solving a pressure Poisson equation and a linear convection–diffusion equation at each time step. Numerical tests indicate that the computational cost of the new scheme for each time step, under identical time step sizes, is even less expensive than the semi-implicit scheme with low element orders. Therefore, the total computational cost of the new scheme can be significantly less than the usual semi-implicit scheme.

1. Introduction

In practical flow simulations the time step size is usually chosen based on considerations of accuracy and stability. Oftentimes, the maximum time step size that can be used in a simulation is restricted by the stability concern, which is especially true for high Reynolds-number simulations of turbulent flows. With the often-used semi-implicit type schemes, in which the nonlinear term of the Navier–Stokes equations is treated explicitly and the viscous term is treated implicitly, the maximum allowable time step size, $\Delta t_{\max}$, is dictated by the CFL number. Consider a spectral-type spatial discretization, and let τ denote the Kolmogorov time scale, which is the smallest time scale in turbulence and needs to be resolved for accurate simulations. The ratio of the maximum allowable time step size to the Kolmogorov time scale scales as a function (see [38])

$$\frac{\Delta t_{\max}}{\tau} \propto Re^{1/2-3\alpha/4}, \tag{1}$$

where Re is the Reynolds number and α is a parameter related to the spectral discretization ($\alpha = 1$ for a Fourier discretization, $\alpha = 2$ for a Chebyshev discretization, and $\alpha \approx 3/2$ for a spectral/hp element discretization [38]). It is evident from this equation that at high Reynolds numbers the maximum allowable time step size dictated by stability can be orders of magnitude smaller than the Kolmogorov time scale (required for accuracy). Therefore, the time step sizes with semi-implicit type schemes widely employed in current incompressible flow simulations can be overly small to be computationally efficient.

[*] Corresponding author.
E-mail addresses: sdong@math.purdue.edu (S. Dong), shen@math.purdue.edu (J. Shen).

[1] The work of S.D. is partially supported by NSF DMS-0810929.

[2] The work of J.S. is partially supported by NSF DMS-0915066 and AFOSR FA9550-08-1-0416.

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.05.037

The time step size constraint can be alleviated by employing unconditionally stable schemes for Navier–Stokes equations. Indeed, the Navier–Stokes equations can be formulated in such a way that the discrete energy can be proven to be bounded at each time step [32,20]. The fully implicit versions of such algorithms are nonlinear, and entail the solution of nonlinear algebraic systems with Newton iterations, which renders the overall approach computationally inefficient. Particularly interesting are the linear versions of these algorithms, in the sense that the resulting boundary value problem is linear with respect to the velocity and pressure. As a result, only linear algebraic systems need to be solved at each time step. However, the velocity and the pressure are coupled in these formulations [20], even though the overall algebraic system is linear. This entails a nested iteration (or sub-iteration) when solving the coupled linear system [32], which is computationally not efficient.

The focus of this paper is a new splitting scheme that enables the use of large time step sizes and at the same time is computationally efficient by decoupling the linear solve of the velocity and the pressure. Splitting schemes, often referred to in the literature as projection methods, are attractive in terms of computational efficiency because only the decoupled elliptic equations for the velocity and the pressure need to be solved. The projection methods can be traced back to Chorin [4] and Temam [34] in the 1960s, and have since witnessed significant developments in various aspects. The main focus has revolved around improving the temporal accuracy of the velocity and pressure. The various techniques can be broadly classified into three categories [12]: pressure-correction type, velocity-correction type, and consistent splitting type. With the pressure-correction type method, an intermediate velocity is first computed from the momentum equation by ignoring the continuity requirement and treating the pressure explicitly, and subsequently it is projected to the space of divergence-free velocity field. The pressure correction idea was introduced in [7] with several subsequent improvements on the temporal accuracy [36,35], which have often been referred to as the standard and rotational forms of the pressure-correction schemes. The term ‘‘rotational form” here is with respect to the viscous term, not to be confused with the rotational form of the nonlinear term. The method of Kim and Moin [18] is equivalent to the rotational form of the pressure-correction scheme [12]. Analysis of the pressure-correction schemes can be found in [3,6,29,33]. The notion of velocity correction was introduced in [14]. With this type of schemes, the viscous term is first treated explicitly, and a correction to the velocity is made subsequently. Standard and rotational forms of the velocity-correction schemes have been identified with differing accuracy in the pressure approximation. The scheme of Karniadakis et al. [16] is equivalent to the velocity-correction scheme in rotational form [14]. The consistent splitting scheme was introduced in [13]. The main idea in computing the pressure is to directly test the momentum equation against a velocity gradient, and the incompressibility of the velocity is enforced only in the weak sense. Full second-order accuracy in pressure is realized with this scheme. It is shown in [12] that the consistent splitting scheme is equivalent to the gauge method [5]. The scheme of [15] in which the pressure is computed explicitly bears a conceptual similarity to the consistent splitting scheme. All three categories of the above schemes are based on a splitting of the Navier–Stokes equations, sometimes referred to as differential splitting in the literature. There is still another class of methods based on inexact factorization of the discretized system (sometimes referred to as algebraic splitting) after the velocity boundary conditions have been applied [8,26,28,21]. These schemes appear attractive because they do not explicitly involve the pressure boundary condition. However, it has been shown recently [12] that the inexact factorization methods weakly enforce an artificial boundary condition of the pressure and do not provide a better accuracy than the pressure-correction schemes.

In this paper we present a new unconditionally stable splitting scheme which does not require the velocity and pressure approximation spaces to satisfy the usual inf–sup condition. We note that it is very easy to construct an unconditionally stable splitting scheme based on the pressure-correction formulation, a simple example is the scheme (5.1) and (5.2) in [30]. However, it is well-known that pressure-correction schemes require the velocity and pressure approximation spaces to satisfy the usual inf–sup condition [12]. Therefore, they are not very convenient to use in practical applications.

Our new scheme is based on the rotational velocity-correction formulation. Previous implementations of the rotational velocity-correction scheme (cf., for instance, [14,16]) all treat the nonlinear term explicitly, leading to a CFL type stability constraint. In the new scheme, the nonlinear term is treated explicitly in the first substep but corrected with a linearly implicit treatment in the second substep.

To implement this scheme, a high-order spectral element approach employing Jacobi-polynomial based shape functions [17] has been used to discretize the Navier–Stokes equations in space. We compare the new splitting scheme, and the semi-implicit scheme of [16] implemented in the spectral element package $N\varepsilon\kappa T\alpha\mathbf{r}$ (see [17]), for test problems with analytic solutions as well as the flow past a circular cylinder at several Reynolds numbers. These tests demonstrate that accurate solutions can be obtained using the new splitting scheme at time step sizes considerably larger than the maximum allowable time step size of the semi-implicit scheme. It is also shown that the new scheme is very competitive in terms of computational cost compared to the semi-implicit scheme.

The rest of this paper is organized as follows. In Section 2 we introduce the new splitting scheme for the incompressible Navier–Stokes equations, and discuss related implementation issues, in particular, the efficient computation of the velocity coefficient matrix. In Section 3 the temporal second-order accuracy of the new scheme is demonstrated using an analytic solution of the Navier–Stokes equations. In Section 4 we compare the new splitting scheme and the semi-implicit scheme for the Kovasznay flow and the flow past a circular cylinder. It is shown that accurate solutions can be obtained using the new scheme at considerably larger time step sizes compared to the semi-implicit scheme; Computational costs between the new splitting scheme and the semi-implicit scheme have also been compared. Finally, Section 5 provides some concluding remarks.

2. A new splitting scheme

2.1. Description of the scheme

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be an open bounded domain, and set $\Gamma = \partial\Omega$. We consider the incompressible Navier–Stokes equations on this domain,

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\nabla^2\mathbf{u} + \nabla p = \mathbf{f}, \\ \nabla\cdot\mathbf{u} = 0, \\ \mathbf{u}|_\Gamma = \mathbf{w}, \end{cases} \tag{2}$$

where t denotes time; $\mathbf{u}$ and p are respectively the velocity and pressure (divided by density); $\mathbf{f}$ represents the external body force; ν is the fluid kinematic viscosity; $\mathbf{w}$ is the velocity Dirichlet boundary condition. The above equation is supplemented by the initial condition $\mathbf{u} = \mathbf{u}_0(\mathbf{x})$ ($\mathbf{x} \in \Omega$) at $t = 0$, which is assumed to be divergence free and compatible with the boundary velocity $\mathbf{w}$.

Given $\mathbf{u}^0 = \mathbf{u}_0$, the new velocity-correction scheme for (2) consists of two sub-steps: substep one

$$\frac{1}{\Delta t}\left(\gamma\tilde{\mathbf{u}}^{k+1} - \sum_{m=0}^{J-1}\alpha_m\mathbf{u}^{k-m}\right) + \nabla p^{k+1} + \mathbf{u}^k\cdot\nabla\mathbf{u}^k + \nu\nabla\times\nabla\times\mathbf{u}^k = \mathbf{f}^{k+1}, \tag{3a}$$

$$\nabla\cdot\tilde{\mathbf{u}}^{k+1} = 0, \tag{3b}$$

$$\mathbf{n}\cdot\tilde{\mathbf{u}}^{k+1}|_\Gamma = \mathbf{n}\cdot\mathbf{w}^{k+1}, \tag{3c}$$

and substep two

$$\frac{\gamma}{\Delta t}(\mathbf{u}^{k+1} - \tilde{\mathbf{u}}^{k+1}) - \nu\nabla^2\mathbf{u}^{k+1} + \tilde{\mathbf{u}}^{k+1}\cdot\nabla\mathbf{u}^{k+1} - \mathbf{u}^k\cdot\nabla\mathbf{u}^k - \nu\nabla\times\nabla\times\mathbf{u}^k = 0, \tag{4a}$$

$$\mathbf{u}^{k+1}|_\Gamma = \mathbf{w}^{k+1}. \tag{4b}$$

In the above, k represents the time level; Δt is the time step size; $\mathbf{n}$ is the outward-pointing directional vector on Γ. The coefficients γ and α_m ($m = 0,\ldots,J-1$; $J = 1$ or 2) are chosen such that $\frac{1}{\Delta t}D v^{k+1} = \frac{1}{\Delta t}\left(\gamma v^{k+1} - \sum_{m=0}^{J-1}\alpha_m v^{k-m}\right)$ represents the Jth order backward differentiation formula (BDF) that approximates $\frac{\partial v}{\partial t}$ at time step $k + 1$, where v is a generic variable. More precisely,

$$D v^{k+1} = \begin{cases} v^{k+1} - v^k, & \text{if } J = 1, \\ \frac{3}{2}v^{k+1} - 2v^k + \frac{1}{2}v^{k-1}, & \text{if } J = 2. \end{cases} \tag{5}$$

We note that the new feature of this scheme is the treatment of the nonlinear term. It is treated explicitly in the first substep, with a correction term in the second substep which is linearly implicit. We note that in existing implementations of velocity-correction schemes, the nonlinear term is generally treated explicitly as a forcing function in the first substep and it does not appear in the second substep. We shall refer to these schemes as *semi-implicit velocity-correction schemes*, and it is clear that a CFL type condition is needed for these schemes to be stable. However, our numerical results and a heuristic argument below show that the new scheme is unconditionally stable despite the explicit treatment of the nonlinear term in the first substep.

We now comment on the accuracy of the scheme (3a)–(4b). Summing up the two sub-steps (3a)–(4b) we find:

$$\begin{cases} \frac{1}{\Delta t}\left(\gamma\mathbf{u}^{k+1} - \sum_{m=0}^{J-1}\alpha_m\mathbf{u}^{k-m}\right) + \nabla p^{k+1} + \tilde{\mathbf{u}}^{k+1}\cdot\nabla\mathbf{u}^{k+1} - \nu\nabla^2\mathbf{u}^{k+1} = \mathbf{f}^{k+1}, \\ \nabla\cdot\tilde{\mathbf{u}}^{k+1} = 0, \\ \mathbf{n}\cdot\tilde{\mathbf{u}}^{k+1}|_\Gamma = \mathbf{n}\cdot\mathbf{w}^{k+1}, \end{cases} \tag{6}$$

In the absence of the nonlinear term, it is shown in [14] that for $J = 2$, the scheme (3a)–(4b) is of second order for the velocity in the L^2 norm but of order $\frac{3}{2}$ for the pressure in L^2 norm and velocity in H^1 norm. Moreover, $\tilde{\mathbf{u}}^{k+1}$ and $\mathbf{u}^{k+1}$ are of the same order accuracy as approximation of $u(t_{k+1})$. Therefore, it is reasonable to expect from (6) that the scheme (3a)–(4b) is of the same accuracy as the corresponding scheme without the nonlinear term.

2.2. Implementation of the scheme

It is not convenient to implement the scheme (3a)–(4b) in its current form. First of all, (3a)–(3c) is a coupled system for $(\tilde{\mathbf{u}}^{k+1}, p^{k+1})$ which needs to be properly decoupled. Secondly, the term $\nu\nabla\times\nabla\times\mathbf{u}^k$ cannot be computed directly with a C_0 finite element discretization. Therefore, we shall first reformulate the scheme (3a)–(4b) into a weak form which is more convenient for implementation.

Given $(\mathbf{u}^k, p^k)$, we determine $(\mathbf{u}^{k+1}, p^{k+1})$ as follows. We set

$$\hat{\mathbf{u}} = \sum_{m=0}^{J-1} \alpha_m \mathbf{u}^{k-m} + \Delta t\left(\mathbf{f}^{k+1} - \mathbf{u}^k \cdot \nabla \mathbf{u}^k\right). \tag{7}$$

Let us denote the vorticity by $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. Taking the L^2-inner product of Eq. (3a) with ∇q and using the identity $\nabla \times \boldsymbol{\omega} \cdot \nabla q = \nabla \cdot (\boldsymbol{\omega} \times \nabla q)$, we obtain the following Poisson equation in the weak form for p^{k+1}:

$$\int_\Omega \nabla p^{k+1} \cdot \nabla q = \frac{1}{\Delta t}\int_\Omega \hat{\mathbf{u}} \cdot \nabla q - \int_\Gamma \mathbf{n} \cdot \left(\frac{\gamma}{\Delta t}\mathbf{w}^{k+1} q + \nu \boldsymbol{\omega}^k \times \nabla q\right), \quad \forall q \in H^1(\Omega). \tag{8}$$

Then, the intermediate velocity $\tilde{\mathbf{u}}^{k+1}$ can be expressed as

$$\tilde{\mathbf{u}}^{k+1} = \hat{\hat{\mathbf{u}}} - \frac{\nu \Delta t}{\gamma}\nabla \times \boldsymbol{\omega}^k \quad \text{with } \hat{\hat{\mathbf{u}}} = \frac{1}{\gamma}\left(\hat{\mathbf{u}} - \Delta t \nabla p^{k+1}\right). \tag{9}$$

We can then reduce Eq. (4a) to

$$\frac{\gamma}{\nu \Delta t}\mathbf{u}^{k+1} - \nabla^2 \mathbf{u}^{k+1} + \frac{1}{\nu}\hat{\hat{\mathbf{u}}} \cdot \nabla \mathbf{u}^{k+1} - \frac{\Delta t}{\gamma}\left(\nabla \times \boldsymbol{\omega}^k\right) \cdot \nabla \mathbf{u}^{k+1} = \frac{\gamma}{\nu \Delta t}\hat{\hat{\mathbf{u}}} + \frac{1}{\nu}\mathbf{u}^k \cdot \nabla \mathbf{u}^k. \tag{10}$$

Using the identity,

$$(\nabla \times \boldsymbol{\omega}) \cdot (\phi \nabla v) = \nabla \cdot (\phi \boldsymbol{\omega} \times \nabla v) - (\nabla \phi \times \boldsymbol{\omega}) \cdot \nabla v$$

and integration by parts, we obtain the weak form of the above equation for $\mathbf{u}^{k+1}$:

$$\begin{aligned}&\frac{\gamma}{\nu \Delta t}\int_\Omega \boldsymbol{\phi} \mathbf{u}^{k+1} + \int_\Omega \nabla \boldsymbol{\phi} \cdot \nabla \mathbf{u}^{k+1} + \frac{1}{\nu}\int_\Omega \boldsymbol{\phi} \hat{\hat{\mathbf{u}}} \cdot \nabla \mathbf{u}^{k+1} + \frac{\Delta t}{\gamma}\int_\Omega (\nabla \boldsymbol{\phi} \times \boldsymbol{\omega}^k) \cdot \nabla \mathbf{u}^{k+1} = \int_\Omega \phi\left(\frac{\gamma}{\nu \Delta t}\hat{\hat{\mathbf{u}}} + \frac{1}{\nu}\mathbf{u}^k \cdot \nabla \mathbf{u}^k\right),\\ &\quad \forall \boldsymbol{\phi} \in H_0^1(\Omega)^d, \mathbf{u}^{k+1}|_\Gamma = \mathbf{w}^{k+1},\end{aligned} \tag{11}$$

where $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_\Gamma = 0\}$.

Therefore, the reformulated scheme consists of only a Poisson Eq. (8) for p^{k+1} and a linear convection–diffusion Eq. (11) for each component of $\mathbf{u}^{k+1}$. The intermediate velocity $\tilde{\mathbf{u}}^{k+1}$ is not explicitly needed. These equations are in weak form, and so can be easily adopted to Galerkin finite-element or spectral element discretizations.

We now consider the spatial discretization of (8) and (11). Let Ω_h denote the partitioned domain Ω using a spectral element mesh, and Γ_h denote the boundary of Ω_h. Let $X_h \subset H^1(\Omega_h)^d$ and $M_h \subset H^1(\Omega_h)$ respectively denote the approximation spaces of the velocity, $\mathbf{u}_h^{k+1}$, and pressure, p_h^{k+1}. Then the fully discretized version of the system (8) and (11) is:

Find $\mathbf{u}_h^{k+1} \in X_h$ and $p^{k+1} \in M_h$ such that

$$\int_{\Omega_h} \nabla p_h^{k+1} \cdot \nabla q_h = \frac{1}{\Delta t}\int_{\Omega_h} \hat{\mathbf{u}}_h \cdot \nabla q_h - \int_{\Gamma_h} \mathbf{n} \cdot \left(\frac{\gamma}{\Delta t}\mathbf{w}_h^{k+1} q_h + \nu \boldsymbol{\omega}_h^k \times \nabla q_h\right), \quad \forall q_h \in M_h, \tag{12}$$

and

$$\begin{aligned}&\frac{\gamma}{\nu \Delta t}\int_{\Omega_h} \boldsymbol{\phi}_h \mathbf{u}_h^{k+1} + \int_{\Omega_h} \nabla \boldsymbol{\phi}_h \cdot \nabla \mathbf{u}_h^{k+1} + \frac{1}{\nu}\int_{\Omega_h} \boldsymbol{\phi}_h \hat{\hat{\mathbf{u}}}_h \cdot \nabla \mathbf{u}_h^{k+1} + \frac{\Delta t}{\gamma}\int_{\Omega_h} (\nabla \boldsymbol{\phi}_h \times \boldsymbol{\omega}_h^k) \cdot \nabla \mathbf{u}_h^{k+1}\\ &\quad = \int_{\Omega_h} \boldsymbol{\phi}_h\left(\frac{\gamma}{\nu \Delta t}\hat{\hat{\mathbf{u}}}_h + \frac{1}{\nu}\mathbf{u}_h^k \cdot \nabla \mathbf{u}_h^k\right), \quad \forall \boldsymbol{\phi}_h \in X_{h0}, \mathbf{u}_h^{k+1}|_\Gamma = \mathbf{w}_h^{k+1},\end{aligned} \tag{13}$$

where the subscript in $(\cdot)_h$ denotes the discretized version of $(\cdot)$, and $X_{h0} = \{v \in X_h : v|_{\Gamma_h} = 0\}$.

It is well-known that the discrete approximation spaces for the velocity and pressure should satisfy an inf-sup condition for compatibility, otherwise spurious modes for the pressure may result. On the other hand, ample evidences have shown that several types of schemes can work properly with approximation spaces that do not satisfy the usual inf-sup condition (such as with equal-order approximation for the velocity and pressure); see e.g. [16,13,17,12,23,22] for the velocity-correction scheme, consistent splitting scheme, and the scheme with explicit treatment of pressure. Our numerical experiments show that the splitting scheme represented by Eqs. (12) and (13) with spectral element discretizations can work properly with equal-order approximations for the velocity and the pressure. No spurious pressure modes are observed. In our implementation and the flow tests in Section 4, the same orders of expansion polynomials have been used to approximate the velocity and the pressure in the spectral element discretization. We note also that another advantage of the velocity-correction schemes, compared with the pressure-correction schemes, is that no initial pressure approximation is required to achieve second-order accuracy.

Note that a different semi-implicit treatment of the nonlinear term was suggested in (6.1) and (6.2) of [14] but was never implemented. This scheme is also unconditionally stable but the suggested implementation (3.9), (3.7) and (6.4) in [14] requires that the inf-sup condition be satisfied due to (3.7). Another difference is that a second-order extrapolation is used as the convective velocity in (6.1) and (6.2) of [14] while an updated intermediate velocity is used as the convective velocity in our scheme.

We note that a consistent full discretization of the velocity-correction scheme for the linear Stokes equations is presented and analyzed in [10]. However, the formulation in [10] requires the inf–sup condition be satisfied by the velocity–pressure approximation pair.

In our implementation we employ the basis functions based on Jacobi polynomials [17] to take advantage of their tensor-product nature for efficient evaluation of the velocity coefficient matrix. Fast computation of the velocity coefficient matrix is crucial for achieving numerical efficiency with the algorithm. Let Ω_e denote the domain occupied by an element. According to Eq. (13), a generic entry of the elemental velocity coefficient matrix, M_{mn}, predominantly involves the computation of the following terms:

$$\int_{\Omega_e}\left(\frac{\gamma}{\nu\Delta t}\psi_n+\frac{1}{\nu}\hat{\tilde{\mathbf{u}}}\cdot\nabla\psi_n\right)\psi_m+\int_{\Omega_e}\left(\nabla\psi_n+\frac{\Delta t}{\gamma}\boldsymbol{\omega}^k\times\nabla\psi_n\right)\cdot\nabla\psi_m=\int_{\Omega_e}(F_n\psi_m+\mathbf{G}_n\cdot\nabla\psi_m),\tag{14}$$

where ψ_i ($i = 1,\ldots,N_m$) denotes the modal expansion basis function, and N_m is the total number of modes in Ω_e; In the above equation, $F_n=\frac{\gamma}{\nu\Delta t}\psi_n+\frac{1}{\nu}\hat{\tilde{\mathbf{u}}}\cdot\nabla\psi_n$and $\mathbf{G}_n=\nabla\psi_n+\frac{\Delta t}{\gamma}\boldsymbol{\omega}^k\times\nabla\psi_n$. These terms can be transformed such that their evaluation can efficiently take advantage of the tensor-product nature of the expansion basis functions and the sum factorization technique [25]. We illustrate below using two-dimensional quadrilateral elements how to transform the right-hand-side (RHS) of Eq. (14) into a form amenable to sum factorization and to minimize the number of evaluations. With other types of elements the generalized tensor-product nature of the basis functions allows for the handling in a similar fashion. We will compute the elemental coefficient matrix, M_{mn}, one column at a time (i.e. $m = 1,\ldots,N_m$, and n is fixed). The index n in Eq. (14) will therefore be dropped below. The RHS of Eq. (14) can be written as

$$\int_{\Omega_e}(F\psi_m+\mathbf{G}\cdot\nabla\psi_m)=\int_{\Omega_{st}}J_a\left(F\psi_m+\mathbf{G}\cdot\nabla\xi_1\frac{\partial\psi_m}{\partial\xi_1}+\mathbf{G}\cdot\nabla\xi_2\frac{\partial\psi_m}{\partial\xi_2}\right)=\sum_{i,j}\left(f\psi_m+G_1\frac{\partial\psi_m}{\partial\xi_1}+G_2\frac{\partial\psi_m}{\partial\xi_2}\right)_{(\xi_{1i},\xi_{2j})},\tag{15}$$

where (ξ_1,ξ_2) are coordinates of the standard element Ω_{st}, and (ξ_{1i},ξ_{2j}) are the quadrature points; J_a is the Jacobian; Both J_a and $\nabla\xi_i$ ($i = 1,2$) are determined by the mapping between Ω_e and Ω_{st}, and are pre-computed; $f = FJ_aW$ and $G_i = (\mathbf{G}\cdot\nabla\xi_i)J_aW$ ($i = 1,2$), where W is the weight function. Thanks to the tensor-product nature, the basis function can be written as $\psi_m(\xi_1,\xi_2) = \varphi_r(\xi_1)\varphi_s(\xi_2)$, where φ are the basis functions in one direction, $m = 1,\ldots,N_m$, $r = 1,\ldots,n_r$, $s = 1,\ldots,n_s$, and n_r and n_s are the number of modes in ξ_1 and ξ_2 directions. Let $\varphi^D=\frac{d\varphi}{d\xi}$, which are also pre-computed. Eq. (15) can then be transformed to

$$\sum_{i,j}\left[f(\xi_{1i},\xi_{2j})\varphi_r(\xi_{1i})+G_1(\xi_{1i},\xi_{2j})\varphi_r^D(\xi_{1i})\right]\varphi_s(\xi_{2j})+\sum_{i,j}G_2(\xi_{1i},\xi_{2j})\varphi_r(\xi_{1i})\varphi_s^D(\xi_{2j}).\tag{16}$$

The sum factorization technique can be used to efficiently evaluate the expression in the above equation, which amounts to matrix–matrix multiplications involving level-3 BLAS calls.

The discretized pressure Eq. (12) leads to a linear algebraic system with a symmetric coefficient matrix, and is solved with the conjugate gradient (CG) solver. The velocity Eq. (13) results in a linear algebraic system with a non-symmetric coefficient matrix, and in our implementation is solved using the conjugate gradient squared (CGS) solver or the bi-conjugate gradient stabilized (BiCGSTAB) solver. All linear equation solvers are preconditioned with diagonal scaling. Two-dimensional numerical examples will be used in subsequent sections to demonstrate the performance of the scheme.

2.3. A stability result

In this subsection, we shall provide some theoretical justification for the stability of the scheme (3a)–(4b). Unfortunately, we are unable to prove the stability for the scheme (3a)–(4b) itself, so we will prove a stability result for a slightly modified scheme with a similar explicit treatment of the nonlinear term in the first substep. Since it is very technical to deal with the rotational form and the second-order BDF term, and our main concern is the effect of explicit treatment of the nonlinear term in the first substep, we shall consider the first-order standard velocity-correction form of a slightly modified scheme:

$$\begin{cases}\frac{1}{\Delta t}\left(\tilde{\mathbf{u}}^{k+1}-\mathbf{u}^k\right)-\nu\nabla^2\mathbf{u}^k+\tilde{\mathbf{u}}^k\cdot\nabla\mathbf{u}^k+\nabla p^{k+1}=\mathbf{f}^{k+1},\\ \nabla\cdot\tilde{\mathbf{u}}^{k+1}=0,\\ \mathbf{n}\cdot\tilde{\mathbf{u}}^{k+1}|_\Gamma=0;\end{cases}\tag{17}$$

$$\begin{cases}\frac{1}{\Delta t}(\mathbf{u}^{k+1}-\tilde{\mathbf{u}}^{k+1})-\nu\nabla^2(\mathbf{u}^{k+1}-\mathbf{u}^k)+\tilde{\mathbf{u}}^{k+1}\cdot\nabla\mathbf{u}^{k+1}-\tilde{\mathbf{u}}^k\cdot\nabla\mathbf{u}^k=0,\\ \mathbf{u}^{k+1}|_\Gamma=0.\end{cases}\tag{18}$$

Note that the only difference, compared with the first-order standard velocity-correction form of the scheme (3a)–(4b) is that the explicit nonlinear term $\mathbf{u}^k\cdot\nabla\mathbf{u}^k$ is replaced by $\tilde{\mathbf{u}}^k\cdot\nabla\mathbf{u}^k$. For the sake of simplicity, we assumed in the above a homogeneous Dirichlet boundary condition for the velocity.

Theorem 2.1. *The scheme (17) and (18) is unconditionally stable in the sense that for all $n \geqslant 0$ we have*

$$\|\mathbf{u}^{n+1}\|^2_{0,\Omega} + \nu\Delta t\sum_{k=0}^{n}\|\nabla\mathbf{u}^{k+1}\|^2_{0,\Omega} \leqslant \|\mathbf{u}_0\|^2_{0,\Omega} + (\Delta t)^2\| - \nu\nabla^2\mathbf{u}_0 + \tilde{\mathbf{u}}_0\cdot\nabla\mathbf{u}_0\|^2_{0,\Omega} + C\Delta t\sum_{k=0}^{n}\|\mathbf{f}^{k+1}\|^2_{0,\Omega},$$

where $\|v\|^2_{0,\Omega} = \int_\Omega |v|^2$ and C is a constant.

Proof. Let us denote $\mathbf{w}^k = -\nu\nabla^2\mathbf{u}^k + \tilde{\mathbf{u}}^k\cdot\nabla\mathbf{u}^k$. Take the inner product of (17) with $2\Delta t\tilde{\mathbf{u}}^{k+1}$ and integrate over Ω, we obtain

$$\|\tilde{\mathbf{u}}^{k+1}\|^2_{0,\Omega} - \|\mathbf{u}^k\|^2_{0,\Omega} + \|\tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k\|^2_{0,\Omega} + 2\Delta t(\mathbf{w}^k, \tilde{\mathbf{u}}^{k+1}) = 2\Delta t(\mathbf{f}^{k+1}, \tilde{\mathbf{u}}^{k+1}), \tag{19}$$

On the other hand, we rearrange (18) as

$$\mathbf{u}^{k+1} + \Delta t\mathbf{w}^{k+1} = \tilde{\mathbf{u}}^{k+1} + \Delta t\mathbf{w}^k. \tag{20}$$

We recall that if $\nabla\cdot\mathbf{u} = 0$ and $\mathbf{u}\cdot\mathbf{n}|_\Gamma = 0$, then

$$\int_\Omega(\mathbf{u}\cdot\nabla\mathbf{v})\cdot\mathbf{v} = 0 \quad \forall v \in H^1(\Omega).$$

We derive from the above that

$$(\mathbf{w}^{k+1}, \mathbf{u}^{k+1}) = \nu\|\nabla\mathbf{u}^{k+1}\|^2_{0,\Omega}.$$

Now taking the inner product of (20) with itself on both sides and using the above relation, we obtain

$$\|\mathbf{u}^{k+1}\|^2_{0,\Omega} + (\Delta t)^2\|\mathbf{w}^{k+1}\|^2_{0,\Omega} + 2\nu\Delta t\|\nabla\mathbf{u}^{k+1}\|^2_{0,\Omega} = \|\tilde{\mathbf{u}}^{k+1}\|^2_{0,\Omega} + (\Delta t)^2\|\mathbf{w}^k\|^2_{0,\Omega} + 2\Delta t(\mathbf{w}^k, \tilde{\mathbf{u}}^{k+1}). \tag{21}$$

Summing up (19) and (21), we obtain

$$\begin{aligned}&\|\mathbf{u}^{k+1}\|^2_{0,\Omega} - \|\mathbf{u}^k\|^2_{0,\Omega} + \|\tilde{\mathbf{u}}^{k+1} - \mathbf{u}^k\|^2_{0,\Omega} + (\Delta t)^2(\|\mathbf{w}^{k+1}\|^2_{0,\Omega} - \|\mathbf{w}^k\|^2_{0,\Omega}) + 2\nu\Delta t\|\nabla\mathbf{u}^{k+1}\|^2_{0,\Omega} = 2\Delta t(\mathbf{f}^{k+1}, \tilde{\mathbf{u}}^{k+1})\\ &\leqslant \nu\Delta t\|\nabla\mathbf{u}^{k+1}\|^2_{0,\Omega} + C\Delta t\|\mathbf{f}^{k+1}\|^2_{0,\Omega}.\end{aligned}$$

The last inequality is derived using the Cauchy–Schwarz inequality and Poincaré inequality. We can then conclude the desired result by summing up the above inequality for $k = 0,1,\ldots,n$. □

While we only provided a rigorous proof for the first-order standard velocity-correction scheme (17) and (18), but based on this proof and the stability proof for the second-order rotational velocity-correction scheme for linear Stokes equations in [14], one can speculate that similar result holds for the second-order rotational velocity-correction scheme (3a)–(4b) with $J = 2$, since the second-order and rotational treatments only add technical difficulties unrelated to the treatment of nonlinear terms.

We note that the implementation of the modified scheme is not as convenient as the original scheme (3a)–(4b) due to the fact that we cannot avoid the intermediate velocity $\tilde{\mathbf{u}}^{k+1}$ whose computation requires an extra projection step onto the approximation space Y_h (which is often different from X_h for $\mathbf{u}^{k+1}$) for $\tilde{\mathbf{u}}^{k+1}$ (cf. [11,10]). In any event, our numerical results indicate that the modified scheme and the original scheme have essentially the same stability and accuracy characteristics.

3. Convergence characteristics

We next demonstrate the temporal convergence characteristics of the splitting scheme presented in Section 2 using an analytic solution to the incompressible Navier–Stokes equations. Consider a rectangular domain, $\Omega = \{(x,y): 0 \leqslant x \leqslant 2, -1 \leqslant y \leqslant 1\}$, and the incompressible flow on Ω. We employ the following unsteady analytic solution to the Navier–Stokes equations in the convergence tests,

$$\begin{cases} u = \frac{A\pi}{a}\cos ax\cos\pi y\sin bt,\\ v = A\sin ax\sin\pi y\sin bt,\\ p = A\sin ax\sin\pi y\cos bt,\end{cases} \tag{22}$$

where A, a and b are prescribed constants. The fluid density is assumed to be a unit value, $\rho = 1$. The flow experiences the following time-dependent body-force field, $\mathbf{f} = (f_x, f_y)$, which is determined in a way such that the solution given by Eq. (22) satisfies the Navier–Stokes equations,

$$\begin{cases} f_x = \frac{A\pi b}{a}\cos\pi y\cos ax\cos bt - \frac{A^2\pi^2}{a}\cos ax\sin ax\sin^2 bt + \nu A\pi a\left(1 + \frac{\pi^2}{a^2}\right)\cos\pi y\cos ax\sin bt\\ \qquad + Aa\sin\pi y\cos ax\cos bt,\\ f_y = Ab\sin\pi y\sin ax\cos bt + A^2\pi\cos\pi y\sin\pi y\sin^2 bt + \nu A(\pi^2 + a^2)\sin\pi y\sin ax\sin bt\\ \qquad + \pi A\cos\pi y\sin ax\cos bt.\end{cases} \tag{23}$$

Dirichlet boundary conditions based on the solution (22) are applied on the boundaries of Ω. Initial condition for the velocity field is given by setting $t = 0$ to the analytic solution.

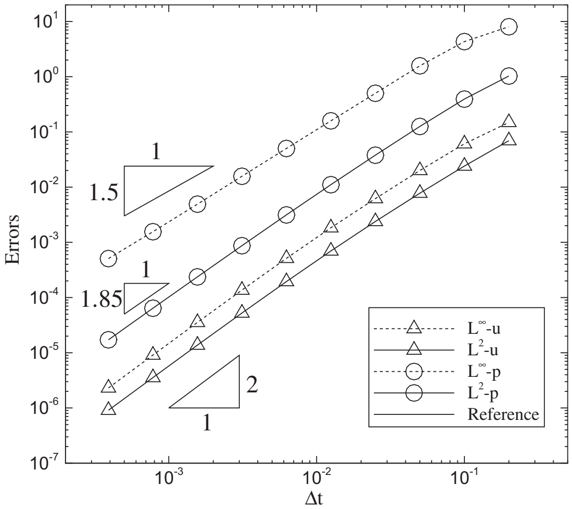


Fig. 1. Temporal convergence: L^∞ and L^2 errors of the velocity (x-component u) and pressure (p) as a function of time step size Δt, for an analytic solution. Computed using the new scheme with $J = 2$ in Eq. (3a).

We partition the domain along the x direction into two quadrilateral elements of equal size. The Navier–Stokes equations are integrated in time using the splitting scheme presented in Section 2, and discretized in space employing the spectral element approach. We use a sufficiently high fixed element order, 15, for each element so that the temporal error will be dominant. Then we systematically vary the time step size Δt. With each Δt value the flow field is simulated in time from $t = 0$ to t_f, and then at $t = t_f$ we compute the L^∞ and L^2 errors of the velocity and the pressure on Ω between the numerical solution and the exact solution given by Eq. (22). In Fig. 1 we plot the errors of the x-velocity component and the pressure as a function of Δt, computed using the new splitting scheme with $J = 2$ (see Eq. (3a)). The results correspond to a problem with the following parameters:

$$A = 2, \quad a = \pi, \quad b = 1, \quad t_f = 1.0, \quad \nu = 1. \tag{24}$$

Second-order temporal convergence has been observed for the velocity. The convergence rate for the pressure is slightly below second order; The L^∞ error of the pressure exhibits a convergence rate of about 1.5, and the L^2 error of the pressure shows a rate of about 1.85. These numerical results are consistent with the error estimates in [14] for the linear Stokes problem, and it indicates that the new scheme for the nonlinear Navier–Stokes equations has the same order of accuracy as the scheme in [14] for the linear Stokes problem.

4. Representative numerical tests

We test the new splitting scheme with two representative problems in this section. The first problem, Kovasznay flow, is a steady-state flow with an analytic solution; This problem allows us to quantify the errors of the scheme at large time step sizes. The second problem, flow past a circular cylinder, is a canonical test problem; We consider the cylinder flow at several Reynolds numbers, in both the steady-state regime and the unsteady regime with vortex shedding. We compare the second-order versions of the new splitting scheme and the semi-implicit scheme of [16], in terms of the effects of time step sizes and the computational cost.

4.1. Kovasznay flow

In the first test we consider the Kovasznay flow, which is a steady-state problem with an analytic expression. The velocity and pressure fields are given by the following equations [19],

$$\begin{cases} u = 1 - e^{\lambda x}\cos 2\pi y, \\ v = \frac{\lambda}{2\pi} e^{\lambda x}\sin 2\pi y, \\ p = \frac{1}{2}(1 - e^{2\lambda x}). \end{cases} \tag{25}$$

where $\lambda = \frac{1}{2\nu} - \sqrt{\frac{1}{4\nu^2} + 4\pi^2}$. We choose $\nu = 1/40$ for this test problem. The pattern of the flow is shown by the streamlines in Fig. 2(a).

We consider the Kovasznay flow on the domain $\Omega = \{(x,y): -0.5 \leqslant x \leqslant 1, -0.5 \leqslant y \leqslant 0.5\}$, which is discretized with four quadrilateral spectral elements (see Fig. 2(b)). Element orders of 10 and 16 have been used for all elements in the following

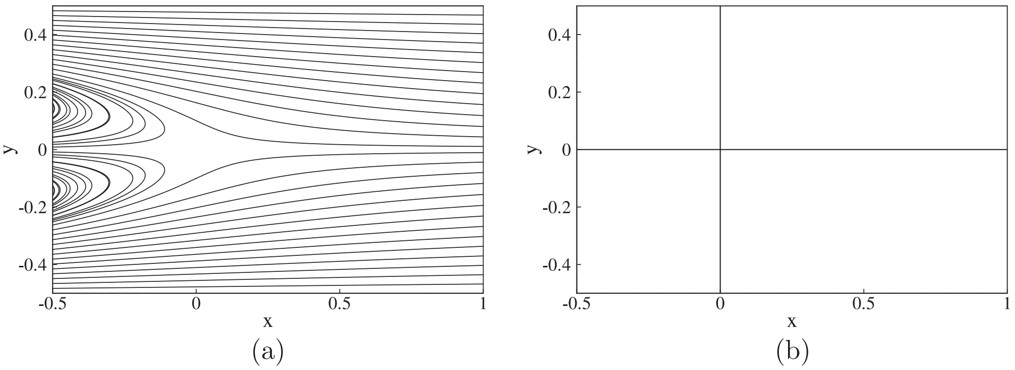


Fig. 2. Kovasznay flow: (a) streamlines and (b) domain discretized with four quadrilateral elements.

test results. Dirichlet conditions based on Eq. (25) are imposed on the boundaries of Ω for the velocity. The initial velocity field is set to be zero.

We solve the Navier–Stokes equations using the second-order versions of the new splitting scheme and the semi-implicit scheme [16]. With the semi-implicit scheme, the nonlinear term is treated explicitly with an extrapolation, and the viscous term is treated implicitly; The resultant linear algebraic systems are symmetric for both the velocity and the pressure, and are solved with the conjugate gradient solver preconditioned with a diagonal scaling.

With the new splitting scheme, the BiCGSTAB solver preconditioned with diagonal scaling has been used for the velocity linear system solve for this problem. We monitor the time histories of the errors between the numerical and exact solutions, as well as the norms of the numerical solution.

We vary the time step size Δt systematically, and with each Δt value integrate the Navier–Stokes equations in time until the flow reaches the steady state. The errors of the steady-state solution against the exact solution are then computed and compared. Table 1 summarizes the H^1 errors of the x-component of the velocity corresponding to different time step sizes for both the new splitting scheme and the semi-implicit scheme.

Let us first consider the computations with an element order 10. With the semi-implicit scheme, accurate steady-state solutions have been obtained with time step sizes $\Delta t = 0.01$ and lower. With $\Delta t > 0.01$ the computation with the semi-implicit scheme is unstable, and at large Δt values it blows up only a few time steps into the simulation. Fig. 3(a) shows time

Table 1
Kovasznay flow: H^1 errors of the x-velocity of the steady-state solution computed using the semi-implicit scheme and the new splitting scheme. Δt is the time step size.

Δt	Element order 10		Element order 16	
	Semi-implicit scheme	New scheme	Semi-implicit scheme	New scheme
0.004			2.382e−12	2.229e−12
0.005			4.703e−13	1.740e−12
0.006			(Unstable)	1.565e−12
0.007			(Unstable)	1.075e−12
0.008	2.813e−8	2.600e−8	(Unstable)	4.929e−13
0.01	2.965e−8	2.768e−8	(Unstable)	6.730e−13
0.02	(Unstable)	3.616e−8	(Unstable)	3.095e−13
0.03	(Unstable)	4.403e−8	(Unstable)	1.711e−13
0.04	(Unstable)	5.126e−8	(Unstable)	2.002e−13
0.05	(Unstable)	5.799e−8	(Unstable)	1.538e−13
0.06	(Unstable)	6.431e−8	(Unstable)	1.635e−13
0.07	(Unstable)	7.033e−8	(Unstable)	1.979e−13
0.08	(Unstable)	7.609e−8	(Unstable)	1.582e−13
0.1	(Unstable)	8.705e−8	(Unstable)	1.428e−13
0.12	(Unstable)	9.745e−8	(Unstable)	1.403e−13
0.14	(Unstable)	1.075e−7	(Unstable)	1.399e−13
0.16	(Unstable)	1.172e−7	(Unstable)	1.389e−13
0.18	(Unstable)	1.268e−7	(Unstable)	1.403e−13
0.2	(Unstable)	1.363e−7	(Unstable)	1.421e−13
0.25	(Unstable)	1.603e−7	(Unstable)	1.580e−13
0.3	(Unstable)	1.856e−7	(Unstable)	1.683e−13
0.35	(Unstable)	2.150e−7	(Unstable)	1.873e−13
0.4	(Unstable)	2.593e−7	(Unstable)	5.619e−1
0.45	(Unstable)	4.267e−7	(Unstable)	8.430e−1

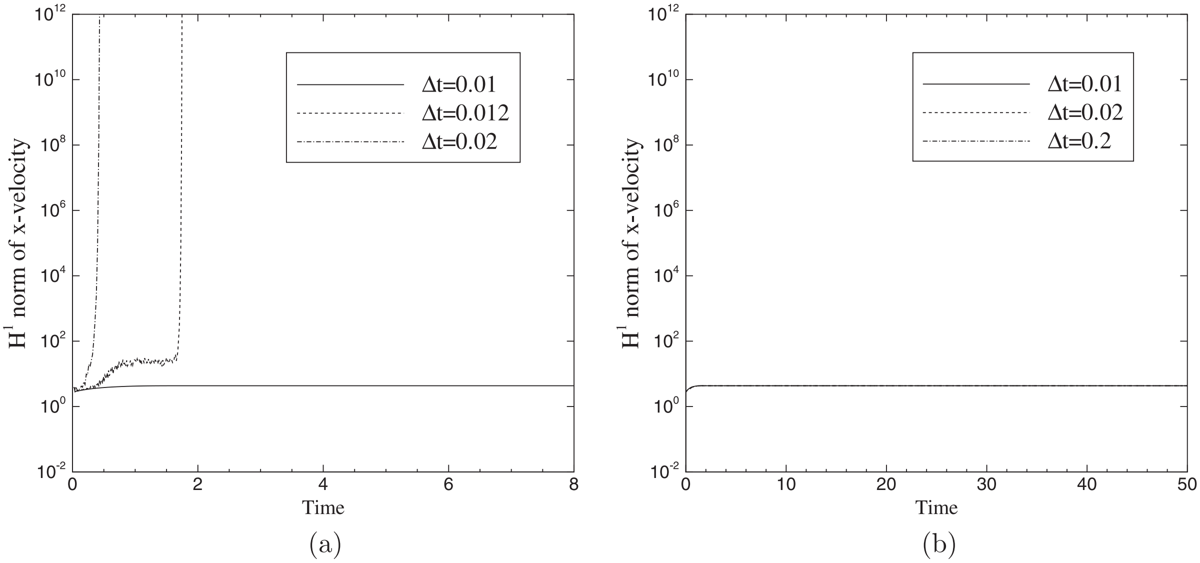


Fig. 3. Kovasznay flow: time histories of the solution norms with different time step sizes computed using (a) the semi-implicit scheme, and (b) the new splitting scheme. Element order is 10.

histories of the H^1 norms of the x-velocity obtained with time step sizes $\Delta t = 0.01$, 0.012 and 0.02 using the semi-implicit scheme. With Δt larger than 0.01, the solution norm rapidly grows unbounded.

On the other hand, stable and accurate solutions have been obtained employing the new splitting scheme with time step sizes up to $\Delta t = 0.45$ (see Table 1), which is over 40 times larger than the maximum allowable time step size with the semi-implicit scheme. Fig. 3(b) shows time histories of the H^1 norms of the x-velocity obtained with time step sizes $\Delta t = 0.01$, 0.02 and 0.2 using the new splitting scheme. It demonstrates the stability of the new scheme at a Δt value 10 times of those in Fig. 3(a) using the semi-implicit scheme. As the time step size increases to 0.5 and larger, the computation using the new splitting scheme becomes inaccurate, but it does not blow up. Fig. 4(a) shows time histories of the H^1 norms of the x-velocity computed with time step sizes $\Delta t = 1.0$ and 10.0, and Fig. 4(b) shows results with even larger time step sizes $\Delta t = 100.0$ and 1000.0. The solution norm becomes highly fluctuational at these large time step sizes, but is bounded throughout time. This characteristic is very different from that of the semi-implicit scheme.

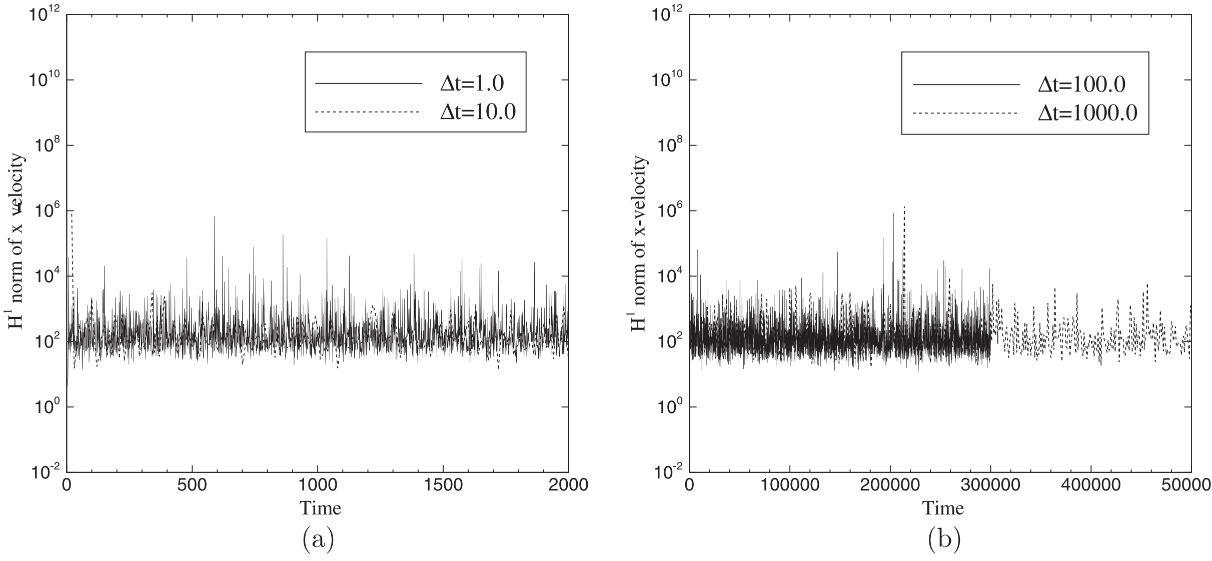


Fig. 4. Kovasznay flow: time histories of solution norms computed with the new splitting scheme at large time step sizes, (a) $\Delta t = 1.0$ and 10.0, (b) $\Delta t = 100.0$ and 1000.0. The computation becomes inaccurate, but does not blow up. Element order is 10.

The errors of the steady-state solutions computed with the new scheme that are discussed above, albeit small, exhibit values on the order of magnitude of 10^{-8}–10^{-7}. These are primarily errors due to the spatial discretization. In Table 1 we have also shown the H^1 errors of the steady-state solution of the x velocity computed using an increased element order, 16, in the spatial discretization with both the semi-implicit and the new splitting schemes. The trend demonstrated by the data is similar to that with the lower element order 10. However, the errors of the steady-state solution have dropped to nearly machine zero. Note that a tolerance of 10^{-14} has been used in the iterative solvers. The semi-implicit scheme works fine with Δt up to 0.005. With the new scheme, the solution has an error around 10^{-13} for a range of Δt values, and it becomes inaccurate with $\Delta t = 0.4$ and larger.

We next comment on the effect of the time step size on how fast the steady-state solution is approached. Fig. 5 shows the convergence histories of the semi-implicit and the new schemes for this problem. Plotted are the H^1 errors of the x velocity as a function of the time step index, computed using the semi-implicit scheme with $\Delta t = 0.005$ (the maximum allowable Δt) and the new scheme with several larger Δt values. The general observation is that, within a certain range (about 10 times the maximum allowable Δt of the semi-implicit scheme), as Δt increases the number of time steps its takes to reach the steady-state solution is decreased proportionately. For example, the semi-implicit scheme takes about 1300–1400 time steps to approach the steady-state solution with $\Delta t = 0.005$. Using the new splitting scheme, with a time step size twice as large, $\Delta t = 0.01$, it takes about 640 time steps to reach the steady-state solution; with a time step size 10 times as large, $\Delta t = 0.05$, this takes about 140 time steps. On the other hand, as Δt further increases beyond 10 times the maximum allowable Δt of the semi-implicit scheme, the number of time steps it takes to reach the steady-state solution stays approximately the same, with a slight increase with increasing Δt. For example, with $\Delta t = 0.1$ (20 times the maximum allowable Δt of semi-implicit scheme) it takes about 160 time steps to reach the steady-state solution, and with $\Delta t = 0.2$ this takes about 210 time steps. The actual time it takes to reach the steady-state solution will of course depend on the computational cost of each time step, which will be compared between the semi-implicit scheme and the new scheme systematically in the next section for the flow past a cylinder.

4.2. Flow past a circular cylinder

In the second test we consider the two-dimensional flow past a circular cylinder. Three Reynolds numbers have been studied in different regimes: $Re = 40$ is characterized by a steady-state flow with two re-circulation bubbles behind the cylinder; $Re = 200$ and $Re = 500$ are characterized by unsteady flows with periodic vortex shedding in the cylinder wake. Fig. 6 shows contours of the instantaneous vorticity at these three Reynolds numbers computed using the new splitting scheme, demonstrating the flow features to be simulated in the following tests.

The setting of problem is as follows. We consider the flow domain, $-10 \leqslant x \leqslant 40$, $-10 \leqslant y \leqslant 10$, in the x–y plane. The center of the cylinder coincides with the origin of the coordinate system, and its diameter is a unit value, $D = 1$. The incoming free-stream flow is assumed to be uniform with a unit magnitude; so the Dirichlet boundary condition $\mathbf{u} = (U_0, 0) = (1, 0)$ is imposed at the inlet ($x = -10$) for the velocity, where U_0 is the free-stream velocity. At the outlet ($x = 40$), zero flux boundary condition $\frac{\partial \mathbf{u}}{\partial n} = 0$ is imposed for the velocity, and the pressure is assumed to be zero. In the cross-flow direction the flow is assumed to be periodic at $y = 10$ and $y = -10$. No-slip boundary condition is imposed on the cylinder surface. The Reynolds number is defined based on the free-stream velocity and the cylinder diameter, $Re = U_0 D/\nu$.

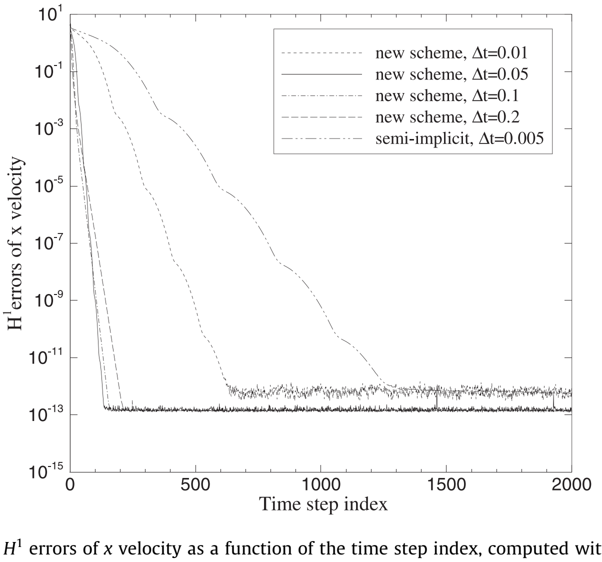


Fig. 5. Kovasznay flow: decrease of the H^1 errors of x velocity as a function of the time step index, computed with an element order 16 for all elements.

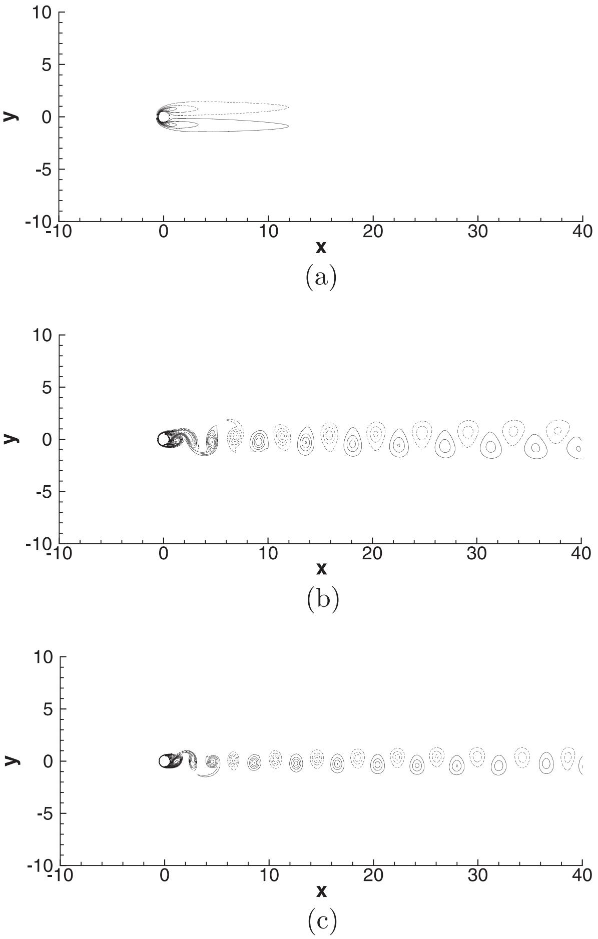


Fig. 6. Cylinder flow: instantaneous vorticity contours at Reynolds numbers *Re* = 40 (a), 200 (b), and 500 (c), computed with the new splitting scheme.

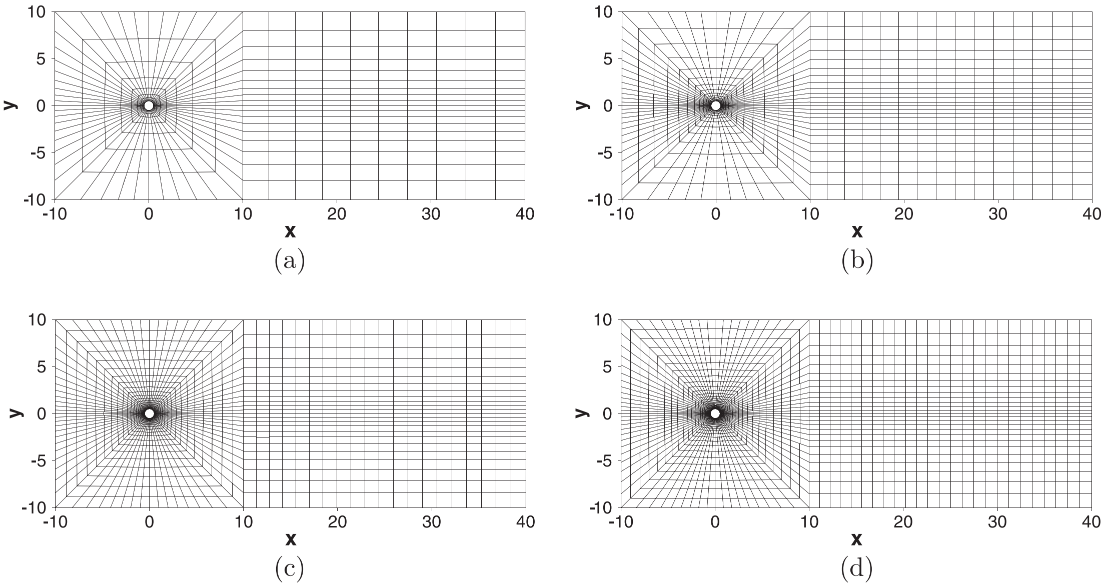
The incompressible Navier–Stokes equations are discretized in time using the new splitting scheme of Section 2 and the semi-implicit scheme [16], both of second-order temporal accuracy. The spectral element approach has been employed for spatial discretization. We first discretize the flow domain with a spectral element mesh. Four meshes have been considered in this test, respectively consisting of 740, 1380, 2240 and 2850 quadrilateral elements, which are shown in Fig. 7. The element order has been varied between 3 and 7 in the following tests. Long-time simulations have been performed at each Reynolds number, in order for the flow to reach a steady state (*Re* = 40) or a statistically stationary state (*Re* = 200 and 500). We then record the forces on the cylinder at the steady state or statistically stationary states, together with the wall time per step the computation takes for comparison.

4.2.1. Computational cost

Let us first consider the computational cost of the new splitting scheme against that of the semi-implicit scheme. With the new splitting scheme, because the velocity coefficient matrix involves the term $\tilde{\mathbf{u}}^{k+1} \cdot \nabla \mathbf{u}^{k+1}$ (Eq. (4a)), it needs to be updated or re-computed every time step, which is undesirable in terms of the increased computational cost. The algorithm can be easily modified to allow for the re-computation of coefficient matrix once every certain number of time steps to reduce the associated cost. For example, one can replace Eq. (4a) with the following modified equation,

$$\frac{\gamma}{\Delta t}(\mathbf{u}^{k+1} - \tilde{\mathbf{u}}^{k+1}) - \nu\nabla^2\mathbf{u}^{k+1} + \tilde{\mathbf{u}}^{k_0} \cdot \nabla\mathbf{u}^{k+1} - \mathbf{u}^k \cdot \nabla\mathbf{u}^k - \nu\nabla \times \nabla \times \mathbf{u}^k = \left(\tilde{\mathbf{u}}^{k_0} - \tilde{\mathbf{u}}^{k+1}\right) \cdot \nabla\mathbf{u}^{*,k+1}, \tag{26}$$

where $u^{*,k+1}$ is an explicit approximation of u^{k+1} such as $u^{*,k+1} = 2u^k - u^{k-1}$, and $\tilde{\mathbf{u}}^{k_0}$ corresponds to time step k_0. This allows k_0, and therefore the velocity coefficient matrix, to be updated once every specified number of time steps. However, we will not consider this cost improvement here. We will compare the cost of the new splitting scheme, with the velocity coefficient matrix re-computed every time step, with that of the semi-implicit scheme. Note that in the semi-implicit scheme, if the fluid viscosity is constant, the velocity coefficient matrix does not change over time and can therefore be pre-computed, as is considered in the current test. The re-computation of the velocity coefficient matrix is an extra cost induced by the



(a)

(b)

(c)

(d)

Fig. 7. Cylinder flow: Four spectral element meshes consisting of (a) 740, (b) 1380, (c) 2240, and (d) 2850 quadrilateral elements.

new splitting scheme. In some situations, the coefficient matrix in the semi-implicit scheme would also need to be updated or re-computed, for example, if the computational mesh moves over time such as in arbitrary Lagrangian Eulerian computations. In this case, the new scheme would induce no extra cost compared to the semi-implicit scheme.

Note that with both the new splitting scheme and the semi-implicit scheme, the coefficient matrix for the pressure equation stays the same over time, and is therefore pre-computed.

The extra cost of computing the velocity coefficient matrix in the new scheme would induce a higher overall cost compared to the semi-implicit scheme. If the problem size is not so large, a direct solver may be used for the resulting linear algebraic equations; the new scheme would entail extra factorizations of the velocity coefficient matrix. For large problem sizes (e.g. 3D problems in general, or 2D problems with a large number of elements), an iterative solver will be preferred; We will subsequently concentrate on this situation, anticipating that this will be the more common mode of usage. The question of interest is how more costly the new scheme is compared to the semi-implicit scheme under identical time step sizes.

We next compare the cost of the new splitting scheme, with the pressure coefficient matrix pre-computed and the velocity coefficient matrix re-computed every time step, with that of the semi-implicit scheme, with both the pressure and velocity coefficient matrices pre-computed. Iterative solvers will be used for the linear algebraic equations resulting from the discretization. Tables 2–4 summarize the wall time (in seconds) per time step of the new and semi-implicit schemes with various grid resolutions at Reynolds numbers $Re = 40$, 200 and 500.

All four spectral element meshes have been tested at each Reynolds number, and the element order ranges from 3 to 7. The same time step size has been used for both the new splitting scheme and the semi-implicit scheme for each grid resolution; the time step sizes are $\Delta t = 0.005$, 0.002 and 0.001 corresponding to the three Reynolds numbers $Re = 40$, 200 and 500, respectively. The computer time was collected on an Intel Xeon Linux workstation cluster (on a single processor), and the application codes for the new scheme and the semi-implicit scheme were compiled with identical optimization flags. The wall-time value listed in the table is an average over a number of time steps for each grid resolution.

The wall-time data in these three tables show some clear trends about the computational cost. The costs for both schemes increase with increasing element order, but the increase with the new scheme is more rapid. For a fixed mesh size, at low

Table 2
Cylinder flow (computational cost): comparison of wall time per time step (in seconds) between the new and semi-implicit schemes with various grid resolutions at $Re = 40$. "New" refers to the new splitting scheme, and "semi" refers to the semi-implicit scheme.

Order/elements	740		1380		2240		2850	
	New	Semi	New	Semi	New	Semi	New	Semi
3	0.0328	0.0364	0.0636	0.147	0.107	0.271	0.141	0.528
4	$\boxed{0.0975}$	$\boxed{0.108}$	0.186	0.313	0.305	0.804	0.397	1.165
5	0.218	0.205	$\boxed{0.408}$	$\boxed{0.579}$	0.671	1.324	0.854	1.724
6	0.515	0.325	0.962	0.856	$\boxed{1.573}$	$\boxed{1.665}$	$\boxed{2.011}$	$\boxed{2.281}$
7	0.970	0.458	1.804	1.248	2.983	2.238	3.744	3.004

Table 3
Cylinder flow (computational cost): comparison of wall time per time step (in seconds) at *Re* = 200.

Order/elements	740		1380		2240		2850	
	New	Semi	New	Semi	New	Semi	New	Semi
3	$\boxed{0.0839}$	$\boxed{0.0911}$	0.157	0.202	0.301	0.397	0.443	0.694
4	0.167	0.167	$\boxed{0.334}$	$\boxed{0.381}$	$\boxed{0.634}$	$\boxed{0.935}$	0.821	1.358
5	0.417	0.288	0.938	0.752	1.697	1.621	$\boxed{2.085}$	$\boxed{2.329}$
6	0.718	0.410	1.683	1.077	2.485	2.271	3.231	3.054
7	1.561	0.694	3.176	1.685	5.010	3.094	5.916	3.984

Table 4
Cylinder flow (computational cost): comparison of wall time per time step (in seconds) at *Re* = 500.

Order/elements	740		1380		2240		2850	
	New	Semi	New	Semi	New	Semi	New	Semi
3	$\boxed{0.0842}$	$\boxed{0.0888}$	0.163	0.193	0.284	0.378	0.432	0.615
4	0.157	0.154	$\boxed{0.349}$	$\boxed{0.352}$	0.718	0.827	0.835	1.275
5	0.354	0.263	0.777	0.695	$\boxed{1.406}$	$\boxed{1.448}$	$\boxed{1.746}$	$\boxed{2.078}$
6	0.678	0.377	1.387	1.025	2.357	1.935	2.938	2.873
7	1.465	0.573	2.799	1.502	4.390	2.702	5.674	3.793

element orders the cost of the new scheme tends to be lower than that of the semi-implicit scheme. As the element order increases above a certain value, the new scheme becomes more expensive than the semi-implicit scheme. So for a given spectral element mesh there exists a cross-over element order, beyond which the new splitting scheme is more expensive. For example, with the 1380-element mesh the cross-over element orders are respectively 5, 4, and 4 for the Reynolds numbers *Re* = 40, 200 and 500. The locations of the cross-over element orders are marked by the boxed values in the three tables. We observe that the cross-over element order increases as the mesh size (i.e. the number of elements) increases. For example, at *Re* = 40 the cross-over element order increases from 4 to 6 as the number of elements in the mesh increases from 740 to 2850; At *Re* = 200 and 500, the cross-over element order increases from 3 to 5 as the number of elements increases from 740 to 2850. This indicates that with regard to computational cost the new splitting scheme can be more favorable with a large mesh size and low to moderate element orders when compared to the semi-implicit scheme. We note that in practical simulations of computational fluid dynamics moderate element orders will usually be used. Very high element orders are rarely used for large-scale problems because of the pronounced increase in computational cost associated with the p-type refinement (i.e. increase in element order) of spectral element computations. The common practice is that for large simulation problems one usually increases the number of elements in the mesh while keeping the element order approximately within a range of moderate values. This suggests that for large-scale problems the new splitting scheme will be very competitive, and will likely be more favorable than the semi-implicit scheme.

Note that the new splitting scheme involves a computation of the N_m^2 ($N_m \sim P^2$, P being the element order) entries of the velocity coefficient matrix on each element. For a given spectral element mesh (with a fixed number of elements), this cost increases with respect to the element order and will be dominant as the element order becomes large. This explains the existence of a cross-over element order for a given spectral element mesh.

It is interesting to note that for a given mesh size, even though the velocity coefficient matrix is re-computed every time step, the new splitting scheme is computationally less expensive than the semi-implicit scheme with low element orders. This results from the performance difference of the two schemes in the linear system solves. Examination of the linear equation solvers shows that with the semi-implicit scheme the pressure equation requires a substantially larger number of iterations for the CG solver to converge to a specified tolerance; For the velocity linear algebraic system, the number of CG iterations per time step in the semi-implicit scheme is also larger than the number of CGS iterations in the new splitting scheme, although the difference is not as substantial. For example, with the 2850-element mesh and element order 4, the pressure solve in the semi-implicit scheme requires approximately 300 and 270 CG iterations per time step for the Reynolds numbers *Re* = 200 and 500, respectively; On the other hand, the velocity solve requires about 20 CG iterations/step at both Reynolds numbers with the semi-implicit scheme. In contrast, with the same mesh and element order and the same time step size, the pressure solve in the new splitting scheme requires about 50 CG iterations per time step at both *Re* = 200 and 500, and the velocity solve requires about 13 CGS iterations/step at both Reynolds numbers. Note that the CG solver has been used to solve the pressure linear algebraic system in both the new and semi-implicit schemes, and it is preconditioned with diagonal scaling in both cases. For the velocity solves, both the CG solver in the semi-implicit scheme and the CGS solver in the new splitting scheme are preconditioned with diagonal scaling. The same tolerance (10^{-8}) has been used for both velocity and pressure in both the new and semi-implicit schemes. At Reynolds number *Re* = 40 (on the same 2850-element mesh), the semi-implicit scheme involves about 290 CG iterations/step for the pressure solve, and only one CG iteration/step for the velocity solve. On the other hand, the new splitting scheme involves one CG iteration/step for the pressure, and one CGS iteration/step for the velocity. We have monitored the time histories of the velocity and pressure at several points in the flow domain. With the semi-implicit scheme, at the steady state of $Re = 40$ we observe that while the numerical values of the velocity on the history points remain exactly the same over time, the numerical values of the pressure fluctuate around the fifth digit (10^{-5}) from time step to time step. This fluctuation seems to perpetuate forever; It does not change in intensity or disappear, even after a very long-time simulation. In contrast, with the new splitting scheme, at the steady state of $Re = 40$ the numerical values of the velocity and the pressure on the history points remain exactly the same over time.

4.2.2. Effect of time step size

We next investigate the effects of the time step size Δt on the accuracy and stability of the computations. For all three Reynolds numbers $Re = 40$, 200 and 500, we vary Δt systematically from small to fairly large values, and carry out a long-time simulation at each Δt. The spectral element mesh with 2850 quadrilateral elements (Fig. 7d) has been used in the following tests, and the element order is 4 for all elements. The fluid forces acting on the cylinder have been monitored in time, and compared in the following tests.

Table 5 summarizes the drag coefficient corresponding to various time step sizes computed with the new splitting scheme and the semi-implicit scheme at $Re = 40$. The drag coefficient is defined by $\frac{F_d}{\frac{1}{2}\rho U_0^2}$, where F_d is the steady-state total drag force (pressure and friction) on the cylinder and ρ is the fluid density. Stable flow solutions are obtained with $\Delta t \leqslant 0.03$ using the semi-implicit scheme; Beyond $\Delta t = 0.03$ the semi-implicit scheme becomes unstable and the computation blows up.

Table 5
Cylinder flow at $Re = 40$: comparison of drag coefficient between the semi-implicit and the new splitting schemes.

Δt	Semi-implicit scheme	New scheme
0.01	1.612	1.612
0.02	1.612	1.612
0.03	1.612	1.612
0.04	(Unstable)	1.612
0.05	(Unstable)	1.612
0.06	(Unstable)	1.612
0.08	(Unstable)	1.612
0.1	(Unstable)	1.612
0.15	(Unstable)	1.612
0.2	(Unstable)	1.612
0.25	(Unstable)	1.612
0.3	(Unstable)	1.612
0.35	(Unstable)	1.612
0.4	(Unstable)	1.620
0.45	(Unstable)	1.632

Table 6
Cylinder flow at $Re = 200$: comparison of mean drag coefficient ("drag-coeff") and r.m.s. lift coefficient ("lift-coeff") between the semi-implicit and the new splitting schemes.

Δt	Semi-implicit scheme		New scheme	
	Drag-coeff	Lift-coeff	Drag-coeff	Lift-coeff
0.004	1.379	0.498	1.378	0.497
0.005	1.379	0.498	1.378	0.497
0.006	1.348	0.548	1.378	0.497
0.007	(Unstable)	–	1.379	0.498
0.008	(Unstable)	–	1.379	0.498
0.009	(Unstable)	–	1.379	0.498
0.01	(Unstable)	–	1.379	0.497
0.02	(Unstable)	–	1.379	0.498
0.03	(Unstable)	–	1.380	0.498
0.04	(Unstable)	–	1.381	0.499
0.05	(Unstable)	–	1.382	0.500
0.06	(Unstable)	–	1.383	0.501
0.08	(Unstable)	–	1.385	0.503
0.1	(Unstable)	–	1.388	0.504
0.15	(Unstable)	–	1.393	0.505
0.2	(Unstable)	–	1.397	0.494
0.25	(Unstable)	–	1.404	0.486
0.3	(Unstable)	–	1.419	0.485
0.35	(Unstable)	–	1.459	0.506

With the new splitting scheme, stable solutions have been obtained at much larger Δt values (up to the maximum $\Delta t = 0.45$ considered in this test). The drag coefficient obtained using the new splitting scheme is identical to that from the semi-implicit scheme for time step sizes up to $\Delta t = 0.35$, which is over 10 times the maximum allowable Δt value with the semi-implicit scheme. At even larger time step sizes the flow solution obtained from the new splitting scheme is inaccurate. Beyond $\Delta t = 0.35$, the flow computed with the new splitting scheme at $Re = 40$ becomes unsteady, and the drag coefficient values at $\Delta t = 0.4$ and 0.45 in Table 5 are the mean values averaged over time. Although these values are very close to those computed with smaller Δt, the unsteady nature of the flow solution is unphysical for this Reynolds number.

Tables 6 and 7 summarize the mean drag coefficient and root-mean-square (r.m.s.) lift coefficient corresponding to various Δt at Reynolds numbers $Re = 200$ and 500. The r.m.s. lift coefficient is defined by $\frac{F_L}{\frac{1}{2}\rho U_0^2}$, where F_L is the r.m.s. lift force on the cylinder. At these Reynolds numbers the flow is unsteady and characterized by periodic vortex shedding behind the cylinder (Fig. 6). The unsteady drag and lift forces on the cylinder have been averaged over a long-time simulation to obtain the drag and lift coefficients.

At $Re = 200$ stable flow solutions have been obtained using the semi-implicit scheme with time step sizes up to $\Delta t = 0.006$; the semi-implicit scheme is unstable beyond $\Delta t = 0.006$. With the new splitting scheme stable solutions have been obtained at significantly larger Δt values. With time step sizes up to about 30 times the maximum allowable Δt of

Table 7
Cylinder flow at $Re = 500$: Comparison of mean drag coefficient ("drag-coeff") and r.m.s. lift coefficient ("lift-coeff") between the semi-implicit and the new splitting schemes.

Δt	Semi-implicit scheme		New scheme	
	Drag-coeff	Lift-coeff	Drag-coeff	Lift-coeff
0.002	1.466	0.851	1.466	0.851
0.003	1.467	0.851	1.466	0.850
0.004	(Unstable)	–	1.466	0.849
0.005	(Unstable)	–	1.466	0.850
0.006	(Unstable)	–	1.466	0.850
0.007	(Unstable)	–	1.466	0.850
0.008	(Unstable)	–	1.466	0.850
0.01	(Unstable)	–	1.467	0.850
0.02	(Unstable)	–	1.468	0.850
0.03	(Unstable)	–	1.469	0.850
0.04	(Unstable)	–	1.472	0.850
0.05	(Unstable)	–	1.474	0.850
0.06	(Unstable)	–	1.477	0.850
0.08	(Unstable)	–	1.479	0.847
0.1	(Unstable)	–	1.465	0.837
0.12	(Unstable)	–	1.458	0.825

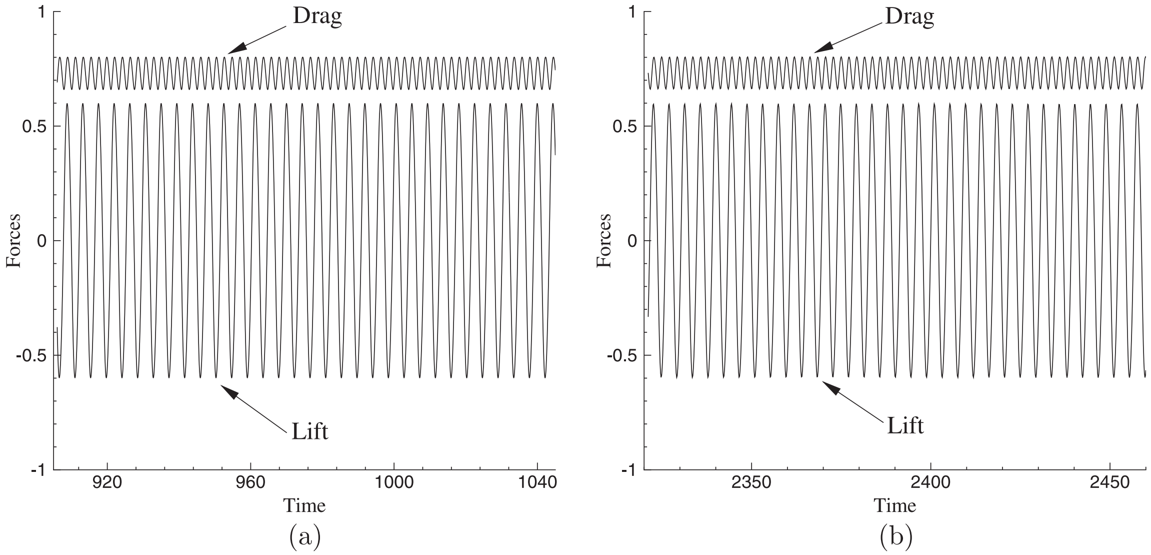


Fig. 8. Cylinder flow at $Re = 500$: Time histories of the drag and lift forces computed using the semi-implicit scheme with $\Delta t = 0.003$ (a), and the new splitting scheme with $\Delta t = 0.03$ (b).

the semi-implicit scheme, the drag and lift coefficients obtained with the new scheme are essentially the same as those from the semi-implicit scheme (with much lower Δt). With even larger time step sizes, the drag and lift coefficients computed using the new splitting scheme become less accurate compared to those obtained with small Δt values; Their differences become more pronounced. To place these Δt values into the context of CFL numbers, at $Re = 200$ a time step size $\Delta t = 0.006$ corresponds to a CFL number approximately (a little less than) 2 and $\Delta t = 0.1$ corresponds to a CFL number about 33. The data exhibit a similar trend with regard to Δt for the higher Reynolds number $Re = 500$. At $Re = 500$ the semi-implicit scheme has a maximum allowable time step size $\Delta t = 0.003$. The new splitting scheme has produced drag and lift coefficients essentially the same as the semi-implicit scheme with time step sizes up to about 20 to 30 times larger than the maximum allowable Δt of the latter. With even larger Δt, more significant differences in the drag and lift coefficient values are observed when compared to those obtained with a small Δt. Fig. 8 shows time histories of the drag and lift forces on the cylinder at $Re = 500$ computed using the semi-implicit scheme with $\Delta t = 0.003$ (Fig. 8(a)), and using the new splitting scheme with a time step size 10 times larger, $\Delta t = 0.03$ (Fig. 8(b)). No apparent difference can be discerned from the force histories.

5. Concluding remarks

We have presented an unconditionally stable splitting scheme for incompressible Navier–Stokes equations based on the rotational velocity-correction formulation. The new scheme enjoys a number of features, including (i) it allows the use of time step sizes considerably larger than widely-used semi-implicit type schemes; (ii) it does not require the velocity and pressure approximation spaces to satisfy the usual inf-sup condition; (iii) it only requires solving a pressure Poisson equation and a linear convection–diffusion equation at each time step. Extensive numerical tests indicate that accurate flow solutions can be obtained using the new scheme with time step sizes 20 or 30 times larger than the maximum allowable time step size of the semi-implicit scheme; the time step size of the new scheme is only constrained by the accuracy not stability.

The new scheme is very competitive with regard to computational efficiency. It involves solving linear equations only, and the velocity and pressure computations are decoupled. This avoids the deficiencies of some known unconditionally stable schemes, e.g. the expensive nonlinear algebraic equations of fully implicit schemes, and the expensive coupling between the velocity and pressure of the linear-type schemes [20,32].

In terms of computational cost, numerical tests using the spectral element package *Nεκ*T*α***r** [17] indicate that the computational cost of the new scheme for each time step is less expensive than the semi-implicit scheme with low element orders, even though the velocity coefficient matrix is re-computed every time step in the new scheme. However, when the element order increases beyond a certain value, the extra cost for computing the velocity coefficient matrix will dominate, and the new scheme will become more costly than the semi-implicit scheme. Numerical tests indicate that the value of this cross-over element order increases as the mesh size increases. So the new scheme can be more favorable for large-scale problems, which are usually characterized by a large mesh size and moderate element orders. We plan in the future to explore to combine more advanced preconditioning techniques [9,2] and the new splitting scheme to further improve the computational efficiency.

An alternative approach for dealing with increased time step sizes in solving Navier–Stokes equations is the semi-Lagrangian method; see e.g. [27,1,38,31] and the references therein. The basic idea is to treat the material derivative (convection term) in the Lagrangian frame and the viscous term in the usual Eulerian frame. This requires the solution at the foot of the characteristic (departure point) from each discrete mesh point. This can be done either by a backward particle tracking or by solving an auxiliary advection equation, respectively referred to as the strong form and the auxiliary form of the semi-Lagrangian method in [37]. The strong form requires backward integrations of the characteristic equation, and interpolations, at every mesh point. It is capable of using increased time step sizes, but the cost of interpolations at every mesh point can be very substantial [39,38]; Due to the particular structure of the error term, the overall error of the method is not monotonic with respect to the time step size Δt, leading to the fact that the error does not approach zero and can grow as $\Delta t \to 0$ with a fixed spatial resolution. In the auxiliary form (sometimes referred to as the operator integration factor splitting scheme [24]), instead of backward particle tracking, the solution at the departure point is obtained by solving a pure advection equation in the Eulerian form in an explicit fashion. The time step size in the auxiliary form is subject to the CFL constraint due to the explicit integration of the advection problem. Its efficiency depends on the ratio of the computational costs of the advection step and the diffusion step. Techniques to reduce the cost of the advection step by using diagonal mass matrices have been discussed in several studies [24,31].

Acknowledgement

The work of S.D. is partially supported by NSF grant DMS-0810929 and the DOE PSAAP program, while the work of J.S. is partially supported by NSF grant DMS-0915066 and AFOSR grant FA9550-08-1-0416. Computer time was provided by the TeraGrid and the Rosen Center for Advanced Computing at Purdue University.

References

[1] Y. Achdou, J.L. Guermond, Convergence analysis of a finite element projection/Lagrang–Galerkin method for the incompressible Navier–Stokes equations, SIAM J. Numer. Anal. 37 (2000) 799.

[2] J.H. Bramble, J.E. Pasciak, A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems, Math. Comput. 50 (1988) 1–17.
[3] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[4] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[5] J.-G. Liu, Gauge method for viscous incompressible flows, Comm. Math. Sci. 1 (2003) 317–332. E.N.E.
[6] J.-G. Liu, Projection method I: convergence and numerical boundary layers, SIAM J. Numer. Anal. 32 (1995) 1017–1057. W.N.E.
[7] K. Goda, A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows, J. Comput. Phys. 30 (1979) 76–95.
[8] P.M. Gresho, S.T. Chan, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via finite element method that also introduces a nearly consistent mass matrix, Int. J. Numer. Meth. Fluids 11 (1990) 587–620.
[9] L. Grinberg, D. Pekurovsky, S.J. Sherwin, G.E. Karniadakis, Parallel performance of the coarse space linear vertex solver and low energy basis preconditioner for spectral/hp elements, Parallel Comput. 35 (2009) 284–304.
[10] J.L. Guermond, Jie Shen, Xiaofeng Yang, Error analysis of fully discrete velocity-correction methods for incompressible flows, Math. Comput. 77 (263) (2008) 1387–1405.
[11] Jean-Luc Guermond, Some implementations of projection methods for Navier–Stokes equations, RAIRO Modél. Math. Anal. Numér. 30 (5) (1996) 637–667.
[12] J.L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006) 6011–6045.
[13] J.L. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, J. Comput. Phys. 192 (2003) 262–276.
[14] J.L. Guermond, J. Shen, Velocity-correction projection methods for incompressible flows, SIAM J. Numer. Anal. 41 (2003) 112–134.
[15] H. Johnston, J.-G. Liu, Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term, J. Comput. Phys. 199 (2004) 221–259.
[16] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 97 (1991) 414–443.
[17] G.E. Karniadakis, S.J. Sherwin, Spectral/hp element methods for computational fluid dynamics, second ed., Oxford University Press, 2005.
[18] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[19] L.I.G. Kovasznay, Laminar flow behind a two-dimensional grid, Proc. Cambridge Philos. Soc. 44 (1948) 58.
[20] A. Labovsky, W.J. Layton, C.C. Manica, M. Neda, L.G. Rebholz, The stabilized extrapolated trapezoidal finite-element method for the Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 198 (2009) 958–974.
[21] M.J. Lee, B.D. Oh, Y.B. Kim, Canonical fractional-step methods and consistent boundary conditions for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 73–100.
[22] J. Liu, Open and traction boundary conditions for the incompressible Navier–Stokes equations, J. Comput. Phys. 228 (2009) 7250–7267.
[23] J.-G. Liu, J. Liu, R.L. Pego, Stability and convergence of efficient Navier–Stokes solvers via a commutator estimate, Comm. Pure Appl. Math. LX (2007) 1443–1487.
[24] Y. Maday, A.T. Patera, E.M. Ronquist, An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, J. Sci. Comput. 4 (1990) 263–292.
[25] S.A. Orszag, Spectral methods for problems in complex geometries, J. Comput. Phys. 37 (1980) 70.
[26] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[27] O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier–Stokes equations, Numer. Math. 38 (1982) 309.
[28] A. Quarteroni, F. Saleri, A. Veneziani, Facorization methods for the numerical approximation of Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 188 (2000) 505–526.
[29] J. Shen, On error estimate of the projection methods for the Navier–Stokes equations: second-order schemes, Math. Comput. 65 (1996) 1039–1065.
[30] Jie Shen, On error estimates of the projection methods for the Navier–Stokes equations: first-order schemes, SIAM J. Numer. Anal. 29 (1992) 57–77.
[31] S. Sherwin, A substepping Navier–Stokes splitting scheme for spectral/hp element discretizations, in: Parallel Computational Fluid Dynamics: New Frontiers and Multi-Disciplinary Applications, North-Holland, 2003, pp. 43–52.
[32] J.C. Simo, F. Armero, Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier–Stokes and Euler equations, Comput. Methods Appl. Mech. Eng. 111 (1994) 111–154.
[33] J.C. Strokwerda, Y.S. Lee, The accuracy of the fractional step method, SIAM J. Numer. Anal. 37 (1999) 37–47.
[34] R. Temam, Sur l'approximation de la solution des equations de Navier–Stokes par la methods des pas fractionnaires ii, Arch. Ration. Mech. Anal. 33 (1969) 377–385.
[35] L.J.P. Timmermans, P.D. Minev, F.N. van de Vosse, An approximate projection scheme for incompressible flow using spectral elements, Int. J. Numer. Meth. Fluids 22 (1996) 673–688.
[36] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Sci. Stat. Comput. 7 (1986) 870–891.
[37] D. Xiu, S. Sherwin, S. Dong, G.E. Karniadakis, Strong and auxiliary forms of the semi-Lagrangian method for incompressible flows, J. Sci. Comput. 25 (2005) 323–346.
[38] D.B. Xiu, G.E. Karniadakis, A semi-lagrangian high-order method for Navier–Stokes equations, J. Comput. Phys. (2001) 658–684.
[39] J. Xu, D. Xiu, G.E. Karniadakis, A semi-lagrangian method for turbulence simulations using mixed spectral discretizations, J. Sci. Comput. 17 (2002) 585.