



Numerical approximation of incompressible Navier-Stokes equations based on an auxiliary energy variable

Lianlei Lin ^{a,b}, Zhiguo Yang ^a, Suchuan Dong ^{a,*}

^a Center for Computational and Applied Mathematics, Department of Mathematics, Purdue University, USA

^b School of Electronics and Information Engineering, Harbin Institute of Technology, China



ARTICLE INFO

Article history:

Received 29 April 2018

Received in revised form 12 March 2019

Accepted 13 March 2019

Available online 20 March 2019

Keywords:

Energy stability

Unconditional stability

Navier-Stokes equations

Incompressible flows

Auxiliary variable

ABSTRACT

We present a numerical scheme for approximating the incompressible Navier-Stokes equations based on an auxiliary variable associated with the total system energy. By introducing a dynamic equation for the auxiliary variable and reformulating the Navier-Stokes equations into an equivalent system, the scheme satisfies a discrete energy stability property in terms of a modified energy and it allows for an efficient solution algorithm and implementation. Within each time step, the algorithm involves the computations of two pressure fields and two velocity fields by solving several de-coupled individual linear algebraic systems with constant coefficient matrices, together with the solution of a nonlinear algebraic equation about a *scalar number* involving a negligible cost. A number of numerical experiments are presented to demonstrate the accuracy and the performance of the presented algorithm.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

We focus on the numerical approximation of the incompressible Navier-Stokes equations, which form the basis for simulations of single-phase incompressible flows and turbulence [23,26,10,6]. The momentum equations for two-phase and multiphase flow problems [36,1,12,9] often take a form analogous to the incompressible Navier-Stokes equations with a similar mathematical structure. Developing efficient and effective algorithms for incompressible Navier-Stokes equations therefore can have implications in fields far beyond incompressible flows.

An essential tradeoff confronting a practitioner of computational fluid dynamics (CFD) is between the desire to be able to use larger time step sizes (permitted by accuracy/stability) and the computational cost. On one end of the spectrum, unconditionally energy-stable schemes (see e.g. [29,32,34,24,11,27,18,3], among others) can alleviate the time step size constraint, and one can concentrate on the accuracy requirement when choosing a time step size in a simulation task. This however comes with a downside. Within each time step energy-stable schemes typically involve the solution of a system of nonlinear algebraic equations or a system of linear algebraic equations with a variable and time-dependent coefficient matrix, inducing a high computational cost due to the Newton nonlinear iterations and the need for frequent re-computations of the coefficient matrices. This can render the overall approach inefficient, and therefore such schemes are not often used in large-scale production simulations in practice, especially for dynamic problems. On the other end of the spectrum, semi-implicit splitting type (or fractional-step) schemes (see e.g. [5,33,21,19,2,35,16,25,17,28], the review article [15], and the references

* Corresponding author.

E-mail address: sdong@purdue.edu (S. Dong).

therein) de-couple the computations for the velocity and the pressure fields, and all the coefficient matrices are constant and can be pre-computed. Such schemes have a low computational cost per time step. But they are only conditionally stable, and the time step size that can be used is restricted by e.g. CFL type conditions. Thanks to their low cost, semi-implicit schemes are very popular in large-scale production simulations for flow physics investigations (see e.g. [26,10,7,14]).

We present in this work a numerical scheme for the incompressible Navier-Stokes equations that in a sense resides somewhere between the two extremes on the spectrum of methods. The salient feature of this scheme is the introduction of an auxiliary variable (which is a scalar number) associated with the total energy of the Navier-Stokes system. This idea is inspired by a recent work [30] for gradient-type dynamical systems. The incompressible Navier-Stokes equations are then reformulated into an equivalent system employing the auxiliary energy variable, and a numerical scheme is devised to approximate the reformulated system. Within a time step, the algorithm requires the computations of two pressure fields and two velocity fields, as well as the solution of a nonlinear algebraic equation about a scalar number. The computation for each of the pressure/velocity fields involves a linear algebraic system with a constant coefficient matrix that can be pre-computed. Solving the nonlinear algebraic equation requires Newton iterations, but its cost is negligible (accounting for about 4% of the total cost per time step) because this nonlinear equation is about a *scalar number*, not a field function. This scheme can be shown to satisfy a discrete energy stability property in terms of a modified energy. Numerical experiments show that the algorithm allows the use of large (or fairly large) time step sizes and produces stable computation results in flow simulations. The amount of operations (and the computational cost) of this algorithm per time step is approximately twice that of the semi-implicit scheme from [8], which is only conditionally stable.

The novelties of this work lie in several aspects: (i) the introduction of the auxiliary energy variable into and the resultant reformulation of the Navier-Stokes system; (ii) the numerical scheme for approximating the reformulated system of equations; and (iii) the efficient solution procedure for overcoming the difficulty caused by the unknown auxiliary variable in the implementation of the scheme.

The rest of this paper is structured as follows. In Section 2 we introduce an auxiliary energy variable and reformulate the Navier-Stokes equations based on this variable. A numerical scheme is presented for approximating the reformulated equivalent system. We then discuss how to implement the scheme, and in particular how to overcome the challenge caused by the unknown auxiliary variable in the implementation. In Section 3 we present several representative numerical simulations to test the accuracy and performance of the algorithm. Section 4 concludes the presentation with some closing remarks.

2. Auxiliary variable-based algorithm for incompressible Navier-Stokes equations

2.1. Reformulated equations and numerical scheme formulation

Consider an incompressible flow contained in some domain Ω in two or three dimensions, whose boundary is denoted by $\partial\Omega$. The dynamics of the flow is described by the incompressible Navier-Stokes equations, given in a non-dimensional form as follows,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{N}(\mathbf{u}) + \nabla p - \nu \nabla^2 \mathbf{u} = \mathbf{f}, \quad (1a)$$

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

where \mathbf{x} and t denote the spatial coordinate and time, $\mathbf{u}(\mathbf{x}, t)$ and $p(\mathbf{x}, t)$ are respectively the normalized velocity and pressure, $\mathbf{f}(\mathbf{x}, t)$ is an external body force, and $\mathbf{N}(\mathbf{u})$ is the convection term, $\mathbf{N}(\mathbf{u}) = \mathbf{u} \cdot \nabla \mathbf{u}$. ν denotes the inverse of the Reynolds number Re ,

$$\nu = \frac{1}{Re} = \frac{\nu_f}{U_0 L} \quad (2)$$

where U_0 is the characteristic velocity scale, L is the characteristic length scale, and ν_f is the kinematic viscosity of the fluid. On the domain boundary $\partial\Omega$ we assume that the velocity is known

$$\mathbf{u} = \mathbf{w}(\mathbf{x}, t), \quad \text{on } \partial\Omega \quad (3)$$

where \mathbf{w} denotes the boundary velocity. The system is supplemented by the initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_{in}(\mathbf{x}) \quad (4)$$

where the initial velocity distribution \mathbf{u}_{in} is assumed to be compatible with the boundary condition (3) on $\partial\Omega$ and satisfies the equation (1b). In the governing equations (1a)–(1b), only the pressure gradient ∇p is physically meaningful, and the absolute pressure value is not fixed (pressure can be shifted by an arbitrary constant). In order to fix the pressure values in the numerical solution we impose the following often-used condition

$$\int_{\Omega} p = 0. \quad (5)$$

Consider a shifted total energy of the system, in non-dimensional form,

$$E(t) = E[\mathbf{u}] = C_0 + \int_{\Omega} \frac{1}{2} |\mathbf{u}|^2 \quad (6)$$

where C_0 is a chosen non-dimensional constant such that $E(t) > 0$ for all $t \geq 0$. In dimensional form, C_0 is a chosen energy constant, to be non-dimensionalized by $\rho U_0^2 L^d$ (ρ denotes the fluid density, and $d = 2$ or 3 is the dimension in space). The goal of C_0 is to shift the energy value so that the term $\frac{1}{\sqrt{E(t)}}$ (to be used subsequently) is well defined for all $t \geq 0$. We include the constant C_0 in (6) to ensure that the term $\frac{1}{\sqrt{E(t)}}$ is well defined even in certain extreme cases, e.g. zero initial velocity field.

Define an auxiliary variable $R(t)$ by

$$R(t) = \sqrt{E(t)}. \quad (7)$$

Then

$$2R \frac{dR}{dt} = \frac{dE}{dt} = \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{u} = \int_{\Omega} \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) \cdot \mathbf{u} - \int_{\partial \Omega} (\mathbf{n} \cdot \mathbf{u}) \frac{1}{2} |\mathbf{u}|^2 \quad (8)$$

where \mathbf{n} is the outward-pointing unit vector normal to the boundary $\partial \Omega$, and we have used integration by part, the equation (1b), and the divergence theorem. It should be emphasized that both $R(t)$ and $E(t)$ are scalar variables, not field functions. At $t = 0$,

$$R(0) = \left(C_0 + \int_{\Omega} \frac{1}{2} |\mathbf{u}_{in}|^2 \right)^{\frac{1}{2}}. \quad (9)$$

In light of the definition (7), we re-write equation (1a) into an equivalent form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{R(t)}{\sqrt{E(t)}} \mathbf{N}(\mathbf{u}) + \nabla p - \nu \nabla^2 \mathbf{u} = \mathbf{f}. \quad (10)$$

We also re-write equation (8) into an equivalent form

$$2R \frac{dR}{dt} = \int_{\Omega} \left[\frac{\partial \mathbf{u}}{\partial t} + \frac{R(t)}{\sqrt{E(t)}} \mathbf{N}(\mathbf{u}) \right] \cdot \mathbf{u} - \int_{\partial \Omega} (\mathbf{n} \cdot \mathbf{u}) \frac{1}{2} |\mathbf{u}|^2. \quad (11)$$

The original system consisting of the equations (1a)–(1b), (3) and (4) is equivalent to the reformulated system consisting of equations (10), (1b), (11), together with the boundary condition (3) and the initial conditions (4) and (9), in which $E(t)$ is given by (6). We next focus on this reformulated equivalent system of equations, and present a numerical scheme for approximating this system.

Let $n \geq 0$ denote the time step index, and $(\cdot)^n$ denote the variable (\cdot) at time step n . Let J ($J = 1$ or 2) denote the temporal order of accuracy of the scheme. We set

$$\mathbf{u}^0 = \mathbf{u}_{in}(\mathbf{x}), \quad R^0 = \sqrt{E(0)} = \sqrt{C_0 + \int_{\Omega} \frac{1}{2} |\mathbf{u}^0|^2}. \quad (12)$$

Then given (\mathbf{u}^n, p^n, R^n) , we compute $(\mathbf{u}^{n+1}, p^{n+1}, R^{n+1})$ through the following scheme

$$\frac{\gamma_0 \mathbf{u}^{n+1} - \hat{\mathbf{u}}}{\Delta t} + \frac{R^{n+1}}{\sqrt{E^{n+1}}} \mathbf{N}(\bar{\mathbf{u}}^{n+1}) + \nabla p^{n+1} - \nu \nabla^2 \mathbf{u}^{n+1} = \mathbf{f}^{n+1}, \quad (13a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (13b)$$

$$2R^{n+1} \frac{\gamma_0 R^{n+1} - \hat{R}}{\Delta t} = \int_{\Omega} \left[\frac{\gamma_0 \mathbf{u}^{n+1} - \hat{\mathbf{u}}}{\Delta t} + \frac{R^{n+1}}{\sqrt{E^{n+1}}} \mathbf{N}(\bar{\mathbf{u}}^{n+1}) \right] \cdot \mathbf{u}^{n+1} - \int_{\partial \Omega} (\mathbf{n} \cdot \mathbf{u}^{n+1}) \frac{1}{2} |\mathbf{u}^{n+1}|^2, \quad (13c)$$

$$\mathbf{u}^{n+1} = \mathbf{w}^{n+1}, \quad \text{on } \partial \Omega, \quad (13d)$$

$$E^{n+1} = \int_{\Omega} \frac{1}{2} |\mathbf{u}^{n+1}|^2 + C_0, \quad (13e)$$

$$\int_{\Omega} p^{n+1} = 0. \quad (13f)$$

Δt is the time step size in the above equations. If χ denotes a generic variable, then in the above equations $\frac{1}{\Delta t}(\gamma_0 \chi^{n+1} - \hat{\chi})$ is the approximation of $\left. \frac{\partial \chi}{\partial t} \right|^{n+1}$ based on the J -th order backward differentiation formula (BDF), with γ_0 and $\hat{\chi}$ defined specifically by

$$\hat{\chi} = \begin{cases} \chi^n, & J = 1, \\ 2\chi^n - \frac{1}{2}\chi^{n-1}, & J = 2; \end{cases} \quad \gamma_0 = \begin{cases} 1, & J = 1, \\ 3/2, & J = 2. \end{cases} \quad (14)$$

$\bar{\mathbf{u}}^{n+1}$ denotes a J -th order explicit approximation of \mathbf{u}^{n+1} , given by

$$\bar{\mathbf{u}}^{n+1} = \begin{cases} \mathbf{u}^n, & J = 1, \\ 2\mathbf{u}^n - \mathbf{u}^{n-1}, & J = 2. \end{cases} \quad (15)$$

Taking the L^2 inner product between \mathbf{u}^{n+1} and equation (13a) leads to

$$\begin{aligned} \int_{\Omega} \left[\frac{\gamma_0 \mathbf{u}^{n+1} - \hat{\mathbf{u}}}{\Delta t} + \frac{R^{n+1}}{\sqrt{E^{n+1}}} \mathbf{N}(\bar{\mathbf{u}}^{n+1}) \right] \cdot \mathbf{u}^{n+1} + \int_{\partial\Omega} (\mathbf{n} \cdot \mathbf{u}^{n+1}) p^{n+1} \\ - \nu \int_{\partial\Omega} (\mathbf{n} \cdot \nabla \mathbf{u}^{n+1}) \cdot \mathbf{u}^{n+1} + \nu \int_{\Omega} \|\nabla \mathbf{u}^{n+1}\|^2 = \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{u}^{n+1} \end{aligned} \quad (16)$$

where we have used integration by part, the divergence theorem, and the equation (13b). Sum up equations (13c) and (16), and we have

$$2R^{n+1} \frac{\gamma_0 R^{n+1} - \hat{R}}{\Delta t} = -\nu \int_{\Omega} \|\nabla \mathbf{u}^{n+1}\|^2 + \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{u}^{n+1} + \int_{\partial\Omega} \left(-p^{n+1} \mathbf{n} + \nu \mathbf{n} \cdot \nabla \mathbf{u}^{n+1} - \frac{1}{2} |\mathbf{w}^{n+1}|^2 \mathbf{n} \right) \cdot \mathbf{w}^{n+1} \quad (17)$$

where we have used equation (13d), and

$$\|\nabla \mathbf{u}\|^2 = \sum_{i,j=1}^d \partial_i u_j \partial_i u_j \quad (d \text{ denoting the dimension in space}). \quad (18)$$

Note the following relations

$$2R^{n+1}(\gamma_0 R^{n+1} - \hat{R}) = 2R^{n+1}(R^{n+1} - R^n) = |R^{n+1}|^2 - |R^n|^2 + |R^{n+1} - R^n|^2, \quad \text{if } J = 1; \quad (19a)$$

$$\begin{aligned} 2R^{n+1}(\gamma_0 R^{n+1} - \hat{R}) &= 2R^{n+1} \left(\frac{3}{2} R^{n+1} - 2R^n + \frac{1}{2} R^n \right) \\ &= \frac{1}{2} (|R^{n+1}|^2 - |R^n|^2) + \frac{1}{2} (|2R^{n+1} - R^n|^2 - |2R^n - R^{n-1}|^2) \\ &\quad + \frac{1}{2} |R^{n+1} - 2R^n + R^{n-1}|^2, \quad \text{if } J = 2. \end{aligned} \quad (19b)$$

Combining the above equations, we obtain the following stability result about the scheme:

Theorem 2.1. *In the absence of the external force \mathbf{f} and with zero boundary velocity \mathbf{w} , the scheme represented by equations (13a)–(13f) satisfies the following property:*

$$Q^{n+1} - Q^n = -|D^{n+1}|^2 - \nu \Delta t \int_{\Omega} \|\nabla \mathbf{u}^{n+1}\|^2 \quad (20)$$

where

$$Q^n = \begin{cases} |R^n|^2, & J = 1, \\ \frac{1}{2} (|R^n|^2 + |2R^n - R^{n-1}|^2), & J = 2; \end{cases} \quad D^{n+1} = \begin{cases} R^{n+1} - R^n, & J = 1, \\ \frac{1}{\sqrt{2}} (R^{n+1} - 2R^n + R^{n-1}), & J = 2. \end{cases} \quad (21)$$

2.2. Solution algorithm and implementation

We next consider how to implement the algorithm represented by the equations (13a)–(13f). Even though R^{n+1} and E^{n+1} are both implicit and E^{n+1} involves the integral of the unknown field function \mathbf{u}^{n+1} over the domain, the scheme can be implemented in an efficient way, thanks to the fact that R^{n+1} and E^{n+1} are scalar numbers, not field functions. We employ C^0 spectral elements [31,20,37,4] for spatial discretizations in the current work.

Let

$$S = \frac{R^{n+1}}{\sqrt{E^{n+1}}}, \quad E(S) = E^{n+1} = \int_{\Omega} \frac{1}{2} |\mathbf{u}^{n+1}|^2 + C_0. \quad (22)$$

Then equation (13a) can be written as

$$\frac{\gamma_0}{\Delta t} \mathbf{u}^{n+1} + \nabla p^{n+1} - \nu \nabla^2 \mathbf{u}^{n+1} = \mathbf{G}^{n+1} - \mathbf{SN}(\bar{\mathbf{u}}^{n+1}) \quad (23)$$

where $\mathbf{G}^{n+1} = \mathbf{f}^{n+1} + \frac{\hat{\mathbf{u}}}{\Delta t}$. In light of (13b), equation (23) can be transformed into

$$\frac{\gamma_0}{\Delta t} \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{G}^{n+1} - \mathbf{SN}(\bar{\mathbf{u}}^{n+1}) - \nu \nabla \times \nabla \times \mathbf{u}^{n+1}. \quad (24)$$

We would like to derive the weak forms for the pressure and velocity in the spatially continuous space first. The discrete function spaces for these variables will be specified later. Let $q(\mathbf{x})$ denote an arbitrary test function in the continuous space. Taking the L^2 inner product between ∇q and equation (24), we get

$$\int_{\Omega} \nabla p^{n+1} \cdot \nabla q = \int_{\Omega} [\mathbf{G}^{n+1} - \mathbf{SN}(\bar{\mathbf{u}}^{n+1})] \cdot \nabla q - \nu \int_{\partial\Omega} \mathbf{n} \times \boldsymbol{\omega}^{n+1} \cdot \nabla q - \frac{\gamma_0}{\Delta t} \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{w}^{n+1} q, \quad \forall q, \quad (25)$$

where $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity, and we have used the integration by part, the divergence theorem, equations (13b) and (13d), and the identity $\int_{\Omega} \nabla \times \boldsymbol{\omega} \cdot \nabla q = \int_{\partial\Omega} \mathbf{n} \times \boldsymbol{\omega} \cdot \nabla q$. Equation (25) couples the pressure and the velocity because of the vorticity $\boldsymbol{\omega}^{n+1}$ (tangent component) on the boundary. In order to simplify the implementation, we will make the following approximation, $\mathbf{n} \times \boldsymbol{\omega}^{n+1}|_{\partial\Omega} \approx \mathbf{n} \times \bar{\boldsymbol{\omega}}^{n+1}|_{\partial\Omega}$, where $\bar{\boldsymbol{\omega}}^{n+1} = \nabla \times \bar{\mathbf{u}}^{n+1}$ is the vorticity based on the explicitly approximated velocity defined in (15). This only slightly reduces the robustness in terms of stability, but significantly simplifies the implementation and reduces the computations. Employing this approximation, we transform (25) into

$$\int_{\Omega} \nabla p^{n+1} \cdot \nabla q = \int_{\Omega} [\mathbf{G}^{n+1} - \mathbf{SN}(\bar{\mathbf{u}}^{n+1})] \cdot \nabla q - \nu \int_{\partial\Omega} \mathbf{n} \times \bar{\boldsymbol{\omega}}^{n+1} \cdot \nabla q - \frac{\gamma_0}{\Delta t} \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{w}^{n+1} q, \quad \forall q. \quad (26)$$

This equation, together with equation (13f), can be solved for p^{n+1} , provided that the unknown scalar value $S = \frac{R^{n+1}}{\sqrt{E^{n+1}}}$ is given.

Exploiting the fact that S is a scalar number (not a field function), we solve equations (26) and (13f) for p^{n+1} as follows. Define two field variables p_1^{n+1} and p_2^{n+1} as solutions to the following two problems:

For p_1^{n+1} :

$$\int_{\Omega} \nabla p_1^{n+1} \cdot \nabla q = \int_{\Omega} \mathbf{G}^{n+1} \cdot \nabla q - \nu \int_{\partial\Omega} \mathbf{n} \times \bar{\boldsymbol{\omega}}^{n+1} \cdot \nabla q - \frac{\gamma_0}{\Delta t} \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{w}^{n+1} q, \quad \forall q. \quad (27a)$$

$$\int_{\Omega} p_1^{n+1} = 0. \quad (27b)$$

For p_2^{n+1} :

$$\int_{\Omega} \nabla p_2^{n+1} \cdot \nabla q = - \int_{\Omega} \mathbf{N}(\bar{\mathbf{u}}^{n+1}) \cdot \nabla q, \quad \forall q. \quad (28a)$$

$$\int_{\Omega} p_2^{n+1} = 0. \quad (28b)$$

Then it is straightforward to verify that the solution to equations (26) and (13f) is given by

$$p^{n+1} = p_1^{n+1} + S p_2^{n+1} \quad (29)$$

where S is to be determined later.

In light of (29), we re-write equation (23) as

$$\frac{\gamma_0}{\nu \Delta t} \mathbf{u}^{n+1} - \nabla^2 \mathbf{u}^{n+1} = \frac{1}{\nu} (\mathbf{G}^{n+1} - \nabla p_1^{n+1}) - \frac{S}{\nu} [\mathbf{N}(\bar{\mathbf{u}}^{n+1}) + \nabla p_2^{n+1}]. \quad (30)$$

Let $\varphi(\mathbf{x})$ denote an arbitrary test function that vanishes on $\partial\Omega$, i.e. $\varphi|_{\partial\Omega} = 0$. Taking the L^2 inner product between φ and the equation (30), we obtain the weak form about \mathbf{u}^{n+1} ,

$$\frac{\gamma_0}{\nu \Delta t} \int_{\Omega} \mathbf{u}^{n+1} \varphi + \int_{\Omega} \nabla \varphi \cdot \nabla \mathbf{u}^{n+1} = \frac{1}{\nu} \int_{\Omega} (\mathbf{G}^{n+1} - \nabla p_1^{n+1}) \varphi - \frac{S}{\nu} \int_{\Omega} [\mathbf{N}(\bar{\mathbf{u}}^{n+1}) + \nabla p_2^{n+1}] \varphi, \quad \forall \varphi \text{ with } \varphi|_{\partial\Omega} = 0, \quad (31)$$

where we have used integration by part, the divergence theorem, and the fact that $\varphi|_{\partial\Omega} = 0$. This equation, together with equation (13d), can be solved for \mathbf{u}^{n+1} , provided that S is known.

We again exploit the fact that S is a scalar number, and solve these equations for \mathbf{u}^{n+1} as follows. Define two field variables \mathbf{u}_1^{n+1} and \mathbf{u}_2^{n+1} as solutions to the following two problems:

For \mathbf{u}_1^{n+1} :

$$\frac{\gamma_0}{\nu \Delta t} \int_{\Omega} \mathbf{u}_1^{n+1} \varphi + \int_{\Omega} \nabla \varphi \cdot \nabla \mathbf{u}_1^{n+1} = \frac{1}{\nu} \int_{\Omega} (\mathbf{G}^{n+1} - \nabla p_1^{n+1}) \varphi, \quad \forall \varphi \text{ with } \varphi|_{\partial\Omega} = 0. \quad (32a)$$

$$\mathbf{u}_1^{n+1} = \mathbf{w}^{n+1}, \quad \text{on } \partial\Omega. \quad (32b)$$

For \mathbf{u}_2^{n+1} :

$$\frac{\gamma_0}{\nu \Delta t} \int_{\Omega} \mathbf{u}_2^{n+1} \varphi + \int_{\Omega} \nabla \varphi \cdot \nabla \mathbf{u}_2^{n+1} = -\frac{1}{\nu} \int_{\Omega} [\mathbf{N}(\bar{\mathbf{u}}^{n+1}) + \nabla p_2^{n+1}] \varphi, \quad \forall \varphi \text{ with } \varphi|_{\partial\Omega} = 0. \quad (33a)$$

$$\mathbf{u}_2^{n+1} = 0, \quad \text{on } \partial\Omega. \quad (33b)$$

Then the solution \mathbf{u}^{n+1} to equations (31) and (13d) is given by

$$\mathbf{u}^{n+1} = \mathbf{u}_1^{n+1} + S \mathbf{u}_2^{n+1}, \quad (34)$$

where S is to be determined.

Note that equation (13c) is equivalent to equation (17). We rewrite equation (17) into

$$\begin{aligned} \frac{2}{\Delta t} R^{n+1} (\gamma_0 R^{n+1} - \hat{R}) \frac{R^{n+1}}{\sqrt{E^{n+1}}} + \frac{\nu R^{n+1}}{\sqrt{E^{n+1}}} \int_{\Omega} \|\nabla \mathbf{u}^{n+1}\|^2 - \frac{R^{n+1}}{\sqrt{E^{n+1}}} \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{u}^{n+1} \\ - \frac{R^{n+1}}{\sqrt{E^{n+1}}} \int_{\partial\Omega} \left(-p^{n+1} \mathbf{n} + \nu \mathbf{n} \cdot \nabla \mathbf{u}^{n+1} - \frac{1}{2} |\mathbf{w}^{n+1}|^2 \mathbf{n} \right) \cdot \mathbf{w}^{n+1} = 0 \end{aligned} \quad (35)$$

where we have multiplied both sides of equation (17) by $\frac{R^{n+1}}{\sqrt{E^{n+1}}}$. We observe that this can improve the robustness of the scheme when the time step size Δt becomes large. This is a scalar nonlinear equation, and it will be solved for S .

In light of equation (34), we have

$$E^{n+1} = E(S) = C_0 + \int_{\Omega} \frac{1}{2} |\mathbf{u}^{n+1}|^2 = A_0 + A_1 S + A_2 S^2, \quad (36)$$

where

$$A_0 = C_0 + \int_{\Omega} \frac{1}{2} |\mathbf{u}_1^{n+1}|^2, \quad A_1 = \int_{\Omega} \mathbf{u}_1^{n+1} \cdot \mathbf{u}_2^{n+1}, \quad A_2 = \int_{\Omega} \frac{1}{2} |\mathbf{u}_2^{n+1}|^2. \quad (37)$$

In light of equations (22), (13d), (29) and (34), we can transform equation (35) into

$$F(S) = \frac{2\gamma_0}{\Delta t} S^3 E(S) - \frac{2\hat{R}}{\Delta t} S^2 \sqrt{E(S)} + B_0 S + B_1 S^2 + B_2 S^3 = 0 \quad (38)$$

where \hat{R} is defined by (14), and

$$\begin{cases} B_0 = \nu \int_{\Omega} \|\nabla \mathbf{u}_1^{n+1}\|^2 - \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{u}_1^{n+1} - \int_{\partial\Omega} \left(-p_1^{n+1} \mathbf{n} + \nu \mathbf{n} \cdot \nabla \mathbf{u}_1^{n+1} - \frac{1}{2} |\mathbf{w}^{n+1}|^2 \mathbf{n} \right) \cdot \mathbf{w}^{n+1} \\ B_1 = 2\nu \int_{\Omega} \nabla \mathbf{u}_1^{n+1} : \nabla \mathbf{u}_2^{n+1} - \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{u}_2^{n+1} - \int_{\partial\Omega} \left(-p_2^{n+1} \mathbf{n} + \nu \mathbf{n} \cdot \nabla \mathbf{u}_2^{n+1} \right) \cdot \mathbf{w}^{n+1} \\ B_2 = \nu \int_{\Omega} \|\nabla \mathbf{u}_2^{n+1}\|^2, \end{cases} \quad (39)$$

and $\|\mathbf{u}\|^2$ is defined by equation (18) and $\nabla \mathbf{u} : \nabla \mathbf{v} = \sum_{i,j=1}^d \partial_i u_j \partial_i v_j$. Equation (38) is a nonlinear *scalar* equation about S . It can be solved for S using the Newton's method with an initial guess $S = 1$. The cost of this computation is very small and is essentially negligible compared to the total cost within a time step, which will be shown by the numerical experiments in Section 3. With S known, E^{n+1} can be computed based on equation (36), and R^{n+1} can be computed using equation (22). The velocity \mathbf{u}^{n+1} and pressure p^{n+1} are then given by equations (34) and (29).

Remark 1. It is important to note that when computing S (or R^{n+1}) we have employed equation (35) in the implementation, which is based on the transformed equation (17) rather than the original equation (13c). While the equations (13c) and (17) are mathematically equivalent, we find that the scheme is much more robust using the transformed equation (17), and stable simulations can be achieved with large time step sizes for both steady-state and unsteady (dynamic) flow problems. On the other hand, with the implementation based on equation (13c) stable results can be obtained with large time step sizes only for steady-state problems, and it encounters difficulties for unsteady flows with large time step sizes in practice.

Let us now consider the spatial discretization of the equations (27a)–(28b) and (32a)–(33b). We discretize the domain Ω with a mesh consisting of N_{el} conforming spectral elements. Let the positive integer K denote the element order, which represents a measure of the highest polynomial degree in the polynomial expansions of the field variables within an element. Let Ω_h denote the discretized domain, $\partial\Omega_h$ denote the boundary of Ω_h , and Ω_h^e ($1 \leq e \leq N_{el}$) denote the element e . Define two function spaces

$$\begin{cases} X = \{ v \in H^1(\Omega_h) : v \text{ is a polynomial characterized by } K \text{ on } \Omega_h^e, \text{ for } 1 \leq e \leq N_{el} \}, \\ X_0 = \{ v \in X : v|_{\partial\Omega_h} = 0 \}. \end{cases} \quad (40)$$

In the following let d ($d = 2$ or 3) denote the spatial dimension, and the subscript in $(\cdot)_h$ denote the discretized version of the variable (\cdot) . The fully discretized equations corresponding to (27a)–(28b) are:

For p_{1h}^{n+1} : find $p_{1h}^{n+1} \in X$ such that

$$\int_{\Omega_h} \nabla p_{1h}^{n+1} \cdot \nabla q_h = \int_{\Omega_h} \mathbf{G}_h^{n+1} \cdot \nabla q_h - \nu \int_{\partial\Omega_h} \mathbf{n}_h \times \bar{\omega}_h^{n+1} \cdot \nabla q_h - \frac{\gamma_0}{\Delta t} \int_{\partial\Omega_h} \mathbf{n}_h \cdot \mathbf{w}_h^{n+1} q_h, \quad \forall q_h \in X. \quad (41a)$$

$$\int_{\Omega_h} p_{1h}^{n+1} = 0. \quad (41b)$$

For p_{2h}^{n+1} : find $p_{2h}^{n+1} \in X$ such that

$$\int_{\Omega_h} \nabla p_{2h}^{n+1} \cdot \nabla q_h = - \int_{\Omega_h} \mathbf{N}(\bar{\mathbf{u}}_h^{n+1}) \cdot \nabla q_h, \quad \forall q_h \in X. \quad (42a)$$

$$\int_{\Omega_h} p_{2h}^{n+1} = 0. \quad (42b)$$

The fully discretized equations corresponding to (32a)–(33b) are:

For \mathbf{u}_{1h}^{n+1} : find $\mathbf{u}_{1h}^{n+1} \in [X]^d$ such that

$$\frac{\gamma_0}{\nu \Delta t} \int_{\Omega_h} \mathbf{u}_{1h}^{n+1} \varphi_h + \int_{\Omega_h} \nabla \varphi_h \cdot \nabla \mathbf{u}_{1h}^{n+1} = \frac{1}{\nu} \int_{\Omega_h} \left(\mathbf{G}_h^{n+1} - \nabla p_{1h}^{n+1} \right) \varphi_h, \quad \forall \varphi_h \in X_0. \quad (43a)$$

$$\mathbf{u}_{1h}^{n+1} = \mathbf{w}_h^{n+1}, \quad \text{on } \partial\Omega_h. \quad (43b)$$

For \mathbf{u}_{2h}^{n+1} : find $\mathbf{u}_{2h}^{n+1} \in [X]^d$ such that

$$\frac{\gamma_0}{\nu \Delta t} \int_{\Omega_h} \mathbf{u}_{2h}^{n+1} \varphi_h + \int_{\Omega_h} \nabla \varphi_h \cdot \nabla \mathbf{u}_{2h}^{n+1} = -\frac{1}{\nu} \int_{\Omega_h} [\mathbf{N}(\bar{\mathbf{u}}_h^{n+1}) + \nabla p_{2h}^{n+1}] \varphi_h, \quad \forall \varphi_h \in X_0. \quad (44a)$$

$$\mathbf{u}_{2h}^{n+1} = 0, \quad \text{on } \partial\Omega_h. \quad (44b)$$

Combining the above discussions, we arrive at the final solution algorithm. It involves the following steps:

- (i) Solve equations (41a)–(41b) for p_1^{n+1} ;
Solve equations (43a)–(43b) for \mathbf{u}_1^{n+1} .
- (ii) Solve equations (42a)–(42b) for p_2^{n+1} ;
Solve equations (44a)–(44b) for \mathbf{u}_2^{n+1} .
- (iii) Compute the coefficients A_0, A_1 and A_2 based on (37);
Compute the coefficients B_0, B_1 and B_2 based on (39).
- (iv) Solve equation (38) for S using the Newton's method with an initial guess $S = 1$.
- (v) Compute \mathbf{u}^{n+1} from equation (34);
Compute p^{n+1} from equation (29);
Compute E^{n+1} from equation (36);
Compute R^{n+1} by $R^{n+1} = S\sqrt{E^{n+1}}$.

It is noted that the linear algebraic systems for p_1^{n+1} , p_2^{n+1} , \mathbf{u}_1^{n+1} , and \mathbf{u}_2^{n+1} all involve constant and time-independent coefficient matrices, which can be pre-computed during pre-processing.

3. Representative numerical examples

We next use several numerical examples in two dimensions to test the performance and the accuracy of the algorithm presented in the previous section. The spatial and temporal convergence rates of the scheme will first be demonstrated using a contrived analytic solution to the incompressible Navier-Stokes equations. Then we will employ the method to study a steady-flow problem (Kovasznay flow) and the flow past a circular cylinder in a periodic channel for a range of Reynolds numbers with various time step sizes.

3.1. Convergence rates

In this subsection we employ a manufactured analytic solution to the incompressible Navier-Stokes equations to investigate the spatial and temporal convergence rates of the algorithm developed herein.

Consider a rectangular domain $0 \leq x \leq 2$ and $-1 \leq y \leq 1$, and the following solution to the Navier-Stokes equations (1a)–(1b) and (5) on this domain:

$$\begin{cases} u = 2 \cos(\pi y) \sin(\pi x) \sin t, \\ v = -2 \sin(\pi y) \cos(\pi x) \sin t, \\ p = 2 \sin(\pi y) \sin(\pi x) \cos t, \end{cases} \quad (45)$$

where u and v are the two components of the velocity \mathbf{u} . In equation (1a) the external body force \mathbf{f} is chosen such that the expressions in (45) satisfy this equation.

We simulate this problem using the algorithm from Section 2. The domain is discretized with two uniform elements along the x direction. On the domain boundary we impose the Dirichlet boundary condition (3), in which the boundary velocity \mathbf{w} is chosen based on the analytic expressions in (45). The initial velocity \mathbf{u}_m is chosen according to the analytic expressions in (45) by setting $t = 0$.

We integrate the Navier-Stokes equations in time from $t = 0$ to $t = t_f$ (t_f to be specified next). Then we compare the numerical solutions of different flow variables at $t = t_f$ with the analytic expressions from (45), and the errors in different norms are computed and recorded. The element order and the time step size Δt are varied systematically in the simulations in order to study their effects on the numerical errors. We employ a non-dimensional viscosity $\nu = 0.01$ for this problem, and the constant C_0 in equation (6) is fixed at $C_0 = 1.0$ in the simulations.

In the spatial convergence tests, we employ a fixed $t_f = 0.1$ and $\Delta t = 0.001$, and vary the element order systematically between 2 and 20. The errors at $t = t_f$ between the numerical solution and the analytic solution in L^∞ and L^2 norms have been computed corresponding to all these element orders. Fig. 1(a) shows these numerical errors as a function of the element order for this group of tests. For element orders 12 and below we observe an exponential decrease in the numerical errors as the element order increases. As the element order increases beyond 14, we observe a saturation in the numerical errors due to the temporal truncation error. The error curves for different flow variables level off approximately at a level around 10^{-7} . These results are indicative of an exponential convergence rate in space for the current method.

In the temporal convergence tests, we fix the integration time at $t_f = 0.2$ and the element order at a large value 16, and then vary the time step size systematically between $\Delta t = 0.1$ and $\Delta t = 1.953125 \times 10^{-4}$. Fig. 1(b) shows the L^∞

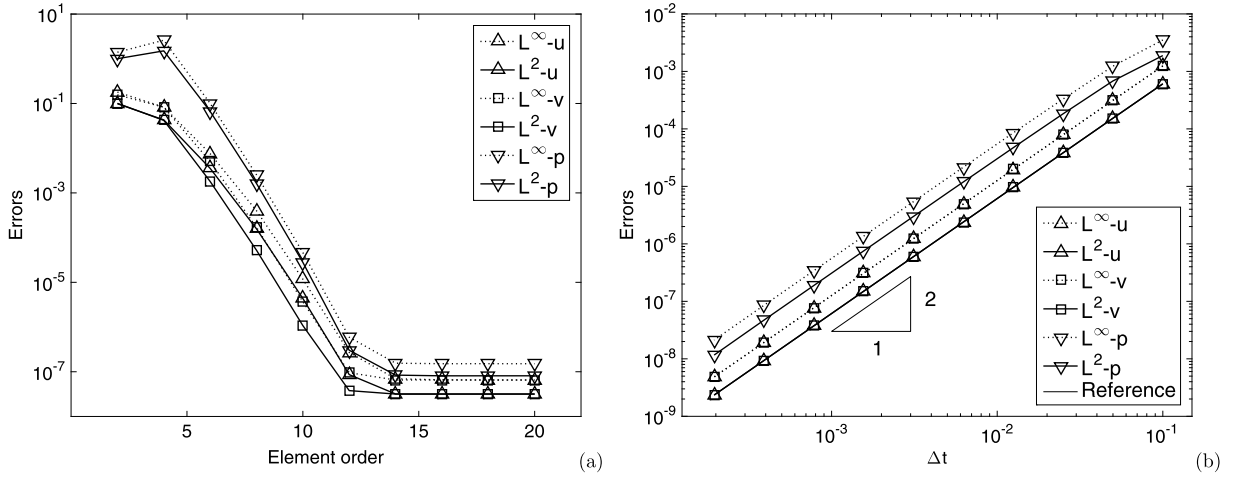


Fig. 1. Convergence tests: (a) Numerical errors vs. element order (fixed $\Delta t = 0.001$ and $t_f = 0.1$). (b) Numerical errors vs. Δt (fixed element order 16 and $t_f = 0.2$).

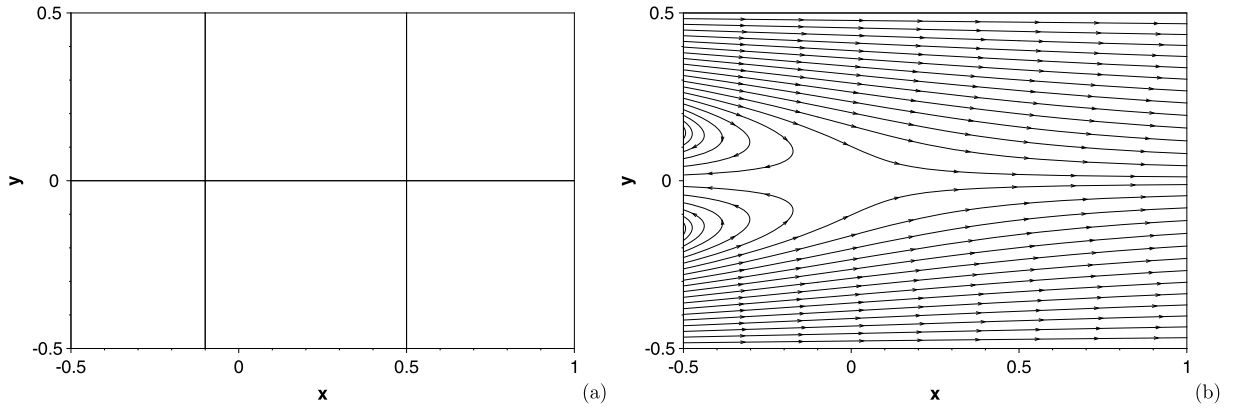


Fig. 2. Kovasznay flow: (a) Flow domain and mesh of 6 quadrilateral elements; (b) Streamlines.

and L^2 errors of the velocity and pressure as a function of Δt , plotted in logarithmic scales for both axes. A second-order convergence rate in time is clearly observed with the method developed herein.

3.2. Kovasznay flow

In this subsection we test the proposed method using a steady-state problem, the Kovasznay flow [22], for which the exact solution of the flow field is available.

Specifically, we consider the domain $-0.5 \leq x \leq 1$ and $-0.5 \leq y \leq 0.5$, as depicted in Fig. 2(a). The Kovasznay flow solution to the Navier-Stokes equations (1a)–(1b) (with $\mathbf{f} = 0$) is given by the following expressions for the velocity $\mathbf{u} = (u, v)$ and pressure:

$$\begin{cases} u = 1 - e^{\lambda x} \cos(2\pi y), \\ v = \frac{\lambda}{2\pi} e^{\lambda x} \sin(2\pi y), \\ p = \frac{1}{2}(1 - e^{2\lambda x}), \end{cases} \quad (46)$$

where the constant $\lambda = \frac{1}{2\nu} - \sqrt{\frac{1}{4\nu^2} + 4\pi^2}$. The flow pattern for this solution is illustrated by the streamlines shown in Fig. 2(b), which is similar to that behind an obstacle. We employ a non-dimensional viscosity $\nu = 0.025$ for this problem.

To simulate the problem, we discretize the domain using 6 quadrilateral elements as shown in Fig. 2(a). The element orders and the time step sizes are varied in the tests and will be specified subsequently. In the Navier-Stokes equation (1a) the external body force is set to $\mathbf{f} = 0$. On the four boundaries Dirichlet type conditions are imposed for the velocity according to the expression given in (46). We set a zero initial velocity, $\mathbf{u}_{in} = 0$, in the initial condition (4). The simulations

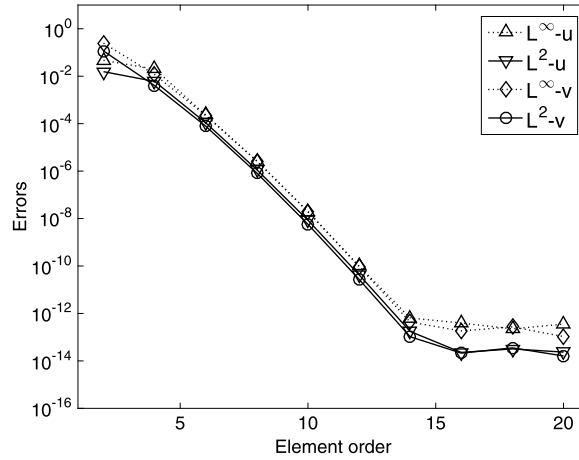


Fig. 3. Kovaszny flow: L^∞ and L^2 errors of the steady-state velocity as a function of the element order. Results are obtained with $\Delta t = 0.005$ and $C_0 = 0.01$.

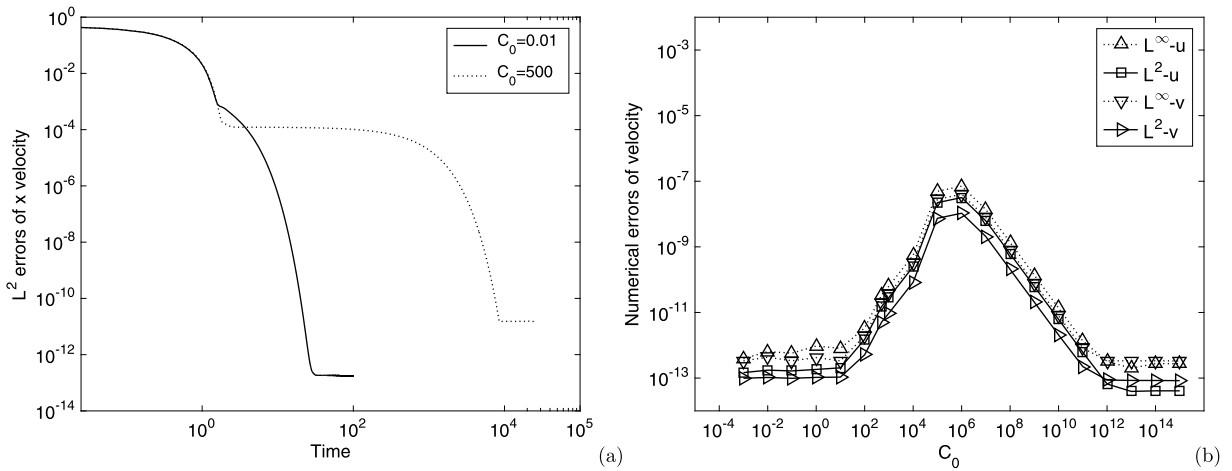


Fig. 4. Effect of C_0 value on the simulation errors of Kovaszny flow: (a) Time histories of L^2 errors of the x velocity obtained with $C_0 = 0.01$ and $C_0 = 500$. (b) Numerical errors of the steady-state velocities as a function of C_0 . Results are obtained using $\Delta t = 0.005$ and an element order 14.

have been performed for a sufficiently long time until the steady state is reached. Then the errors of the numerical solutions against the exact solution as given by (46) are computed, as well as the norms of the flow variables.

Fig. 3 shows the L^∞ and L^2 errors of the steady-state velocity from the simulations as a function of the element order. In this group of tests the time step size is fixed at $\Delta t = 0.005$, and the constant C_0 in equation (6) is $C_0 = 0.01$, while the element order has been varied systematically between 2 and 20. The errors of the steady-state velocity decrease exponentially with increasing element order, until they saturate at a level about 10^{-13} as the element order increases to 14 and beyond.

We observe that the value for the constant C_0 in (6) appears to have some influence on the error history and the error level of the steady-state solution to this problem. This is shown by the results in Fig. 4. Fig. 4(a) shows the L^2 errors of the x velocity component as a function of time as the simulation proceeds. These results are for two constant values, $C_0 = 0.01$ and $C_0 = 500$. The time step size is $\Delta t = 0.005$ and the element order is 14 in these simulations. Three stages can be observed from the error history curves: an initial stage when the error decreases rapidly ($t \lesssim 2$), followed by an intermediate stage when the error decreases at a slower rate, and then the level-off of the error curves. At the initial stage the history curves with $C_0 = 0.01$ and 500 essentially overlap with each other. This stage ends around $t \sim 2$. At the intermediate stage, the error histories for $C_0 = 0.01$ and 500 exhibit quite different characteristics. The error corresponding to $C_0 = 0.01$ decreases quickly and reaches its final level (between 10^{-14} and 10^{-13}) around $t \sim 30$. On the other hand, the error corresponding to $C_0 = 500$ decreases extremely slowly at this stage, and it appears to essentially stagnate for a finite window in time. The error for $C_0 = 500$ reaches its final level ($\sim 10^{-11}$) at about $t \approx 8500$. The error curves corresponding to other moderate C_0 values exhibit a similar behavior to that of $C_0 = 500$, with an extremely slow error decrease at the intermediate stage. On the other hand, the error histories corresponding to small or very large C_0 values are similar to that of $C_0 = 0.01$, and reach the final error level rather quickly. Fig. 4(b) is a study of the C_0 effect on the final error levels of the numerically-obtained steady-state solutions. We vary C_0 between 0.001 and 10^{15} , and for each C_0 value the errors of the

Table 1

Kovaszny flow: L^2 errors of x component of steady-state velocity versus Δt , computed with fixed $C_0 = 0.01$ and element orders 10 and 16.

Δt	Element order 10	Element order 16
0.001	$9.57e-9$	$1.08e-13$
0.002	$8.92e-9$	$5.52e-14$
0.003	$8.69e-9$	$3.78e-14$
0.004	$8.57e-9$	$1.88e-14$
0.005	$8.49e-9$	$2.28e-14$
0.006	$8.43e-9$	$5.25e-3$
0.007	$8.38e-9$	$1.55e-2$
0.008	$8.34e-9$	$2.39e-2$
0.009	$8.31e-9$	$3.06e-2$
0.01	$1.16e-2$	$3.57e-2$
0.03	$1.14e-1$	$1.12e-1$
0.05	$1.42e-1$	$1.39e-1$
0.1	$1.68e-1$	$1.85e-1$
0.5	$2.67e-1$	$2.58e-1$
1.0	$2.76e-1$	$2.91e-1$

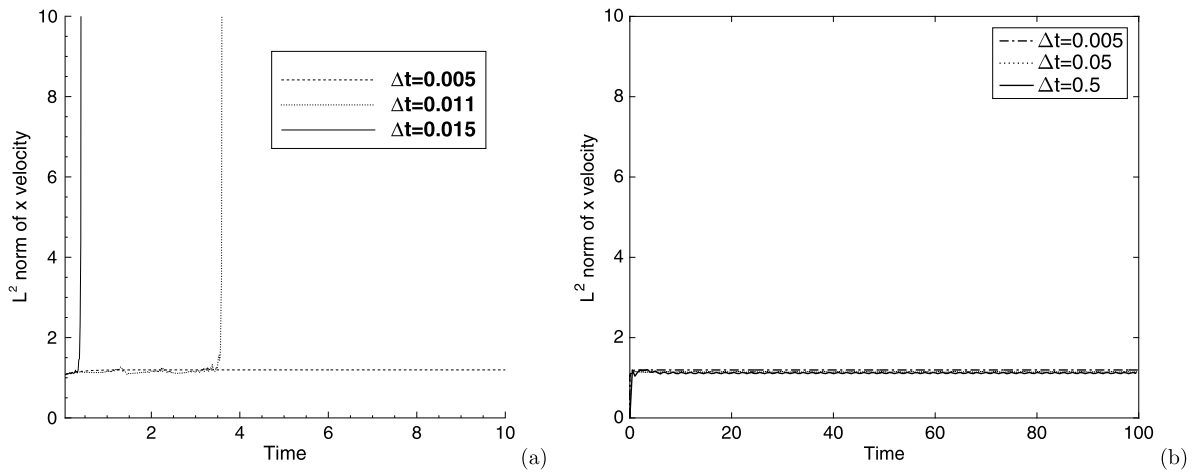


Fig. 5. Kovaszny flow: Time histories of the L^2 norm of the x velocity obtained using several Δt values using (a) the semi-implicit scheme of [8], and (b) the current scheme.

steady-state velocity is computed and recorded. In Fig. 4(b) we show the errors of the steady-state velocity as a function of C_0 . Fixed values of $\Delta t = 0.005$ and an element order 14 are employed in these tests. As C_0 increases from 10^{-3} to about 10, the numerical errors remain essentially the same (around 10^{-13}). Then as C_0 increases from 10 to about 10^6 there is a gradual increase in the numerical errors, reaching a peak level between 10^{-8} and 10^{-7} with $C_0 \approx 10^6$. Beyond $C_0 \approx 10^6$, the numerical errors decrease with increasing C_0 until around $C_0 = 10^{12}$. As C_0 increases beyond 10^{12} , the numerical errors remain essentially the same, at a level around 10^{-13} . These results suggest that using a small C_0 or a very large C_0 in the algorithm may be more favorable for the Kovaszny flow. We employ a constant value $C_0 = 0.01$ for the results reported subsequently in this subsection.

The energy stability property of the current scheme (see Theorem 2.1) is conducive to the stability of computations. We observe that this scheme allows the use of fairly large or large time step sizes (Δt) in the simulations. In Table 1 we have listed the L^2 errors of the x component of the steady-state velocity computed with different Δt values, ranging from $\Delta t = 0.001$ to $\Delta t = 1.0$. In this set of simulations $C_0 = 0.01$ and two element orders (10 and 16) have been used. We observe that for a given spatial resolution (element order) the computation result becomes less accurate or inaccurate when Δt becomes too large. But the current scheme produces stable computations with all these Δt values. In contrast, we observe that with the often-used semi-implicit type schemes, the computation will become unstable for moderately increased Δt values. In Fig. 5 we show the time histories of the L^2 norm of the x component of the velocity computed using the semi-implicit scheme of [8] (Fig. 5(a)) and the current scheme (Fig. 5(b)). These results correspond to an element order 10, and for the current scheme also $C_0 = 0.01$ in the simulations. The computation using the semi-implicit scheme blows up when Δt is beyond about 0.01, while the current scheme exhibits a different behavior and produces stable computations even with much larger Δt values.

Fig. 6 shows the time histories of the L^2 norm of the x velocity of the Kovaszny flow obtained using the current scheme with several large time step sizes ranging from $\Delta t = 1.0$ to $\Delta t = 1000$. The element order is 10 and $C_0 = 0.01$ in these

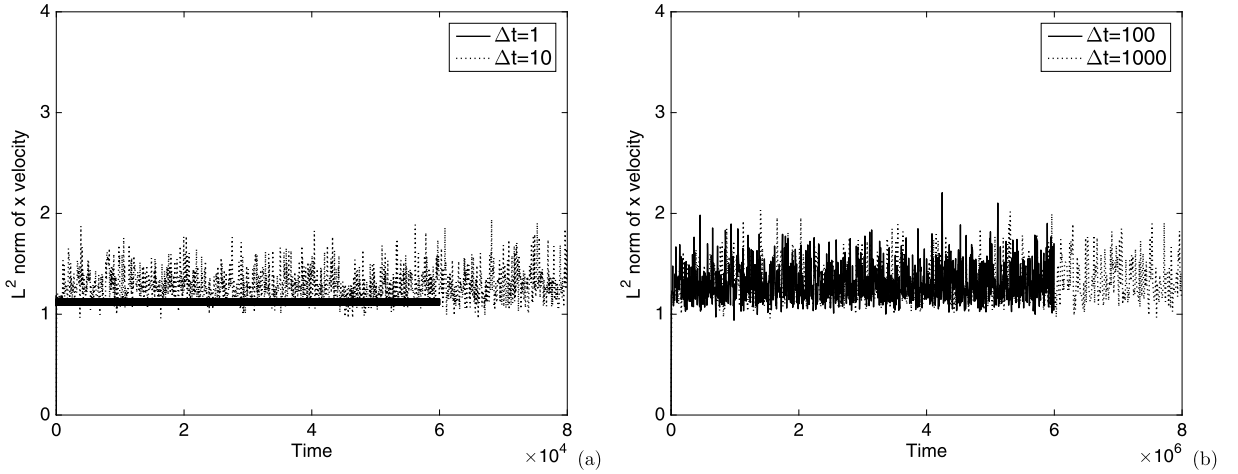


Fig. 6. Kovasznay flow: Time histories of the L^2 norm of the x velocity obtained using several large Δt values: (a) $\Delta t = 1$ and 10, (b) $\Delta t = 100$ and 1000. Results correspond to an element order 10 and $C_0 = 0.01$.

simulations. While we cannot expect accuracy in these results because of the large Δt values, the computations using the current scheme are nonetheless stable with these large Δt for the Kovasznay flow.

In Fig. 7 we plot the time histories of two variables: $E(t)$ defined by (6), and $[R(t)]^2$ computed by solving (11) from the current scheme. These two variables are supposed to be equal due to equation (7). With small Δt values, the two quantities resulting from the computations are indeed the same, which is evident from the time histories obtained with $\Delta t = 0.005$ in Fig. 7(a). The initial difference between the curves for $E(t)$ and $R^2(t)$ in Fig. 7(a) is due to the initial condition, because the zero initial velocity field is not compatible with the Dirichlet boundary condition (non-zero velocity) on the domain boundary. When Δt becomes moderately large or very large, we can observe a difference between the computed $E(t)$ and $R^2(t)$, and the discrepancy between them becomes larger with increasing Δt (Figs. 7(b)–(d)). Both $E(t)$ and $R^2(t)$ fluctuate over time in the simulations using large Δt values. While the computed $E(t)$ approximately stays at a constant mean level, $R(t)$ appears to be driven toward zero with very large Δt in the simulations (Fig. 7(d)). These results suggest that when Δt becomes large the dynamical system consisting of equations (13a)–(13f) seems to be able to automatically adjust the $R(t)$ and the $\frac{R(t)}{\sqrt{E(t)}}$ levels. The term $\frac{R(t)}{\sqrt{E(t)}}$ places a control on the explicitly-treated nonlinear term in the Navier-Stokes equation (13a). With very large Δt values, the system drives $\frac{R(t)}{\sqrt{E(t)}}$ toward zero, thus making the computation more stable or stabilizing the computation. This seems to be the mechanism by which the current scheme produces stable simulations with large Δt for the Kovasznay flow.

3.3. Flow past a circular cylinder in a periodic channel

In this section we investigate the flow past a circular cylinder inside a periodic channel, which is driven by a constant pressure gradient along the channel direction. We employ this problem to test the algorithm developed herein.

Consider a circular cylinder of diameter D placed inside a horizontal channel occupying the domain $-2.5D \leq x \leq 6.5D$ and $-1.5D \leq y \leq 1.5D$, as depicted in Fig. 8(a). The top and bottom of the domain ($y = \pm 1.5D$) are the channel walls. A pressure gradient is imposed along the x direction to drive the flow. The flow domain and all the physical variables are assumed to be periodic in the horizontal direction at $x = -2.5D$ and $x = 6.5D$. The center of the cylinder coincides with the origin of the coordinate system. This setting is equivalent to the flow past an infinite sequence of circular cylinders inside an infinitely long horizontal channel.

We use the cylinder diameter D as the characteristic length scale. Let $\frac{\Delta P}{D}$ denote the pressure gradient (body force) that drives the flow along the x direction, and let g_0 denote a unit body force magnitude. Then the non-dimensional body force in equation (1a) has the magnitude $|\mathbf{f}| = \frac{\Delta P}{g_0 D}$ and points along the x direction. We use $U_0 = \sqrt{\frac{g_0 D}{\rho}}$, where ρ is the fluid density, as the characteristic velocity scale. All the length variables are normalized by D , and all velocity variables are normalized by U_0 .

Fig. 8(a) shows the spectral element mesh used to discretize the domain, which consists of 720 quadrilateral elements. The element order has been varied between 2 and 6 in the simulations to test the effect of spatial resolutions on the simulation results. On the top and bottom channel walls, as well as on the cylinder surface, a no-slip condition is imposed on the velocity, that is, the boundary condition (3) with $\mathbf{w} = 0$. In the horizontal direction (at $x/D = -2.5$ and 6.5) periodic conditions are imposed for all the flow variables. Long-time simulations have been performed using the algorithm from Section 2 for a range of Reynolds numbers (or ν) and the driving force $|\mathbf{f}|$. The simulation is started at a low Reynolds number with a zero initial velocity field at the very beginning. Then the Reynolds number is increased incrementally. A

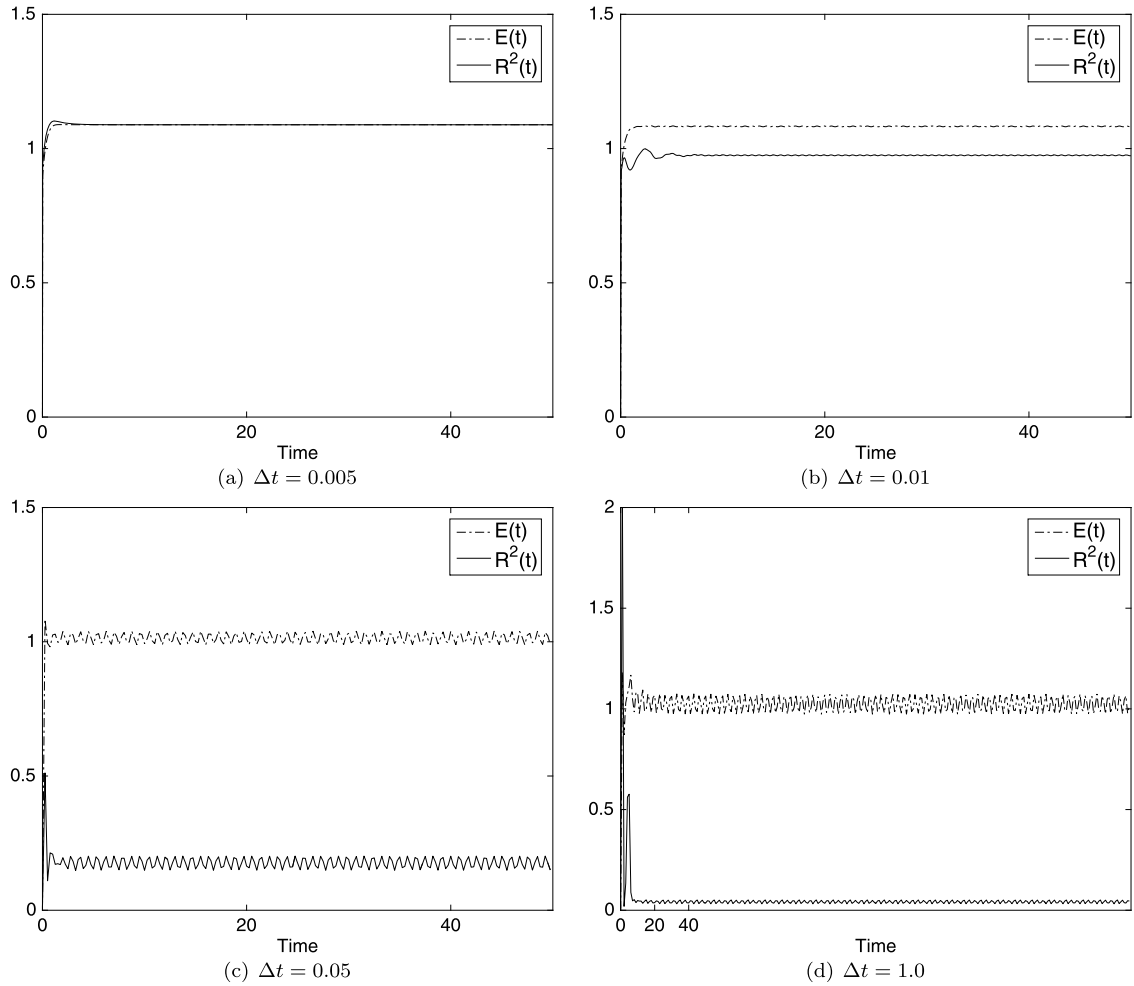


Fig. 7. Kovaszny flow: time histories of $E(t)$ and $R^2(t)$ obtained using several time step sizes: (a) $\Delta t = 0.005$, (b) $\Delta t = 0.01$, (c) $\Delta t = 0.05$, (d) $\Delta t = 1.0$. Results correspond to $C_0 = 0.01$ and an element order 10 in the simulations.

snapshot of the flow field from a lower Reynolds number is used as the initial condition in the simulation of the next larger Reynolds number. At each Reynolds number, a long-time simulation is performed (with typically 100 flow-through times), such that the flow has reached a statistically stationary state for that Reynolds number and the initial condition will have no effect on the flow. A range of values for Δt and C_0 have been tested in the simulations.

An overview of the flow features is provided by Figs. 8(b) and (c), in which the contours of the instantaneous vorticity have been shown for two Reynolds numbers corresponding to $\nu = 0.01$ and $\nu = 0.005$ and a non-dimensional driving pressure gradient $|\mathbf{f}| = 0.02$. The dashed curves denote negative vorticity values. With $\nu = 0.01$ the flow reaches a steady state eventually. The pattern of the vorticity contours around the cylinder is typical of the cylinder flow [10] at low Reynolds numbers. Due to presence of the channel, we can also observe a certain level of vorticity distribution near the upper/lower channel walls above or below the cylinder. With $\nu = 0.005$ an unsteady flow is observed, with periodic vortex shedding from the cylinder into the wake. Because of the periodicity in the horizontal direction, the vortices shed to the cylinder wake will re-enter the domain on the left side and influence the flow upstream of the cylinder. These upstream vortices interact with the cylinder, leading to more complicated dynamics and flow structures in the cylinder wake.

The effect of the spatial mesh resolution on the simulation results is demonstrated by Table 2. We have computed the total forces acting on the walls (sum of those on the cylinder surface and channel wall surfaces) from the simulations. This table lists the time-averaged mean drag (or drag if steady flow), root-mean-square (rms) drag, mean lift, and rms lift on the walls corresponding to a non-dimensionalized driving pressure gradient $\frac{\Delta P}{g_0 D} = 0.02$. Several element orders, ranging from 2 to 6, are tested in the simulations. The results for two Reynolds numbers corresponding to $\nu = 0.01$ and $\nu = 0.001$ have been included in the table, and they are obtained using $\Delta t = 0.01$ for $\nu = 0.01$ and $\Delta t = 0.001$ for $\nu = 0.001$ in the simulations. The total driving force on the flow is $\frac{\Delta P}{g_0 D} \times V_\Omega = 0.02 \times 26.2146 \approx 0.524$, where $V_\Omega = ((6.5 + 2.5) \times (1.5 + 1.5) - \pi/4) \approx 26.2146$ is the normalized volume (or area) of the flow domain. At steady state or statistically stationary state of the flow, this driving force will be balanced by the total force exerted on the flow by the walls. This will give

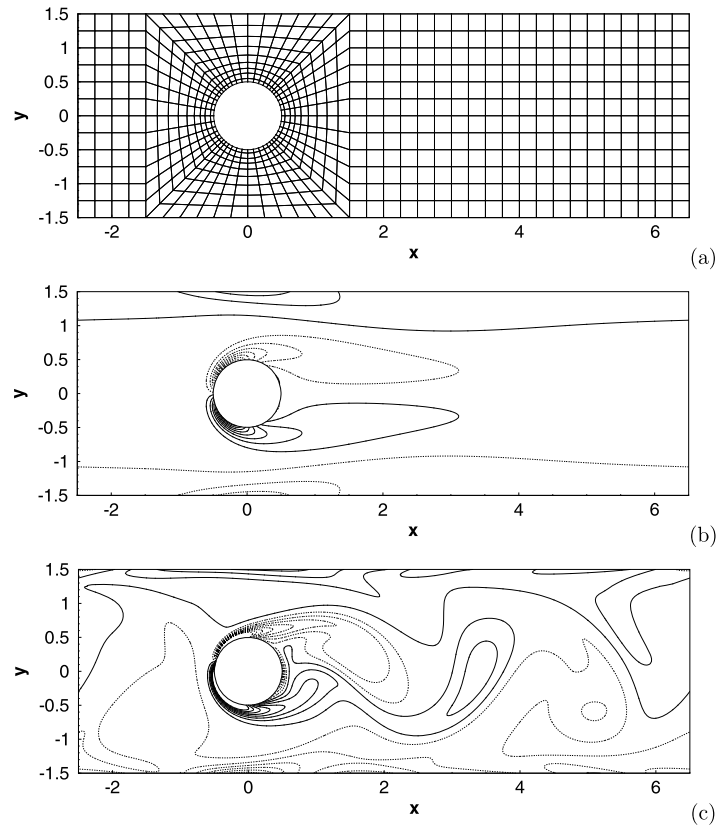


Fig. 8. Cylinder flow in a periodic channel: (a) Mesh of 720 quadrilateral elements. Vorticity contours at $\nu = 0.01$ (b) and $\nu = 0.005$ (c), with the flow driven by a normalized pressure gradient of 0.02.

Table 2

Effect of spatial resolution on the computed forces acting on channel/cylinder walls. Drag refers to the force in the x direction and lift refers to the force in the y direction. Driving force is the normalized total force driving the flow due to the imposed pressure gradient.

ν	Element order	Mean-drag	rms-drag	Mean-lift	rms-lift	Driving force
0.01	2	0.487	0	0	0	0.524
	3	0.527	0	0	0	0.524
	4	0.524	0	0	0	0.524
	5	0.524	0	0	0	0.524
	6	0.524	0	0	0	0.524
0.001	2	0.352	0.155	$4.13e-4$	0.272	0.524
	3	0.519	0.258	$6.41e-4$	$2.95e-2$	0.524
	4	0.529	0.273	$3.39e-4$	$1.40e-2$	0.524
	5	0.529	0.264	$-2.69e-5$	$3.75e-3$	0.524
	6	0.524	0.287	$-7.20e-5$	$1.69e-3$	0.524

rise to a total mean drag acting on the walls with the same value as the total driving force. Therefore, we expect that the time-averaged mean drag (or the drag if steady flow) obtained from the simulations should be equal or close to the total driving force in the flow domain. This can be used to check the accuracy of the simulations. With very low element orders, we observe a discrepancy between the computed drag from the simulations and the expected value. As the element order increases in the simulations, this discrepancy decreases and becomes negligible or completely vanishes. For $\nu = 0.01$, the computed drag becomes very close to the expected value for element order 3, and with element orders 4 and above the computed drag matches the expected value. For $\nu = 0.001$, the difference between the computed mean-drag and the expected value becomes very small with element orders 4 and above.

Fig. 9 shows a comparison of the profiles of the steady-state stream-wise velocity along the centerline of the domain ($y = 0$), as well as along the cross-flow direction at several downstream locations $x/D = 1, 3$, and 5 for the Reynolds number corresponding to $\nu = 0.01$ and computed with several element orders in the simulations. The total driving force in the domain is 0.524. While some difference can be observed between the profile corresponding to element order 2 and the other profiles, all the velocity profiles obtained with the element orders 3 and above basically overlap with one another,

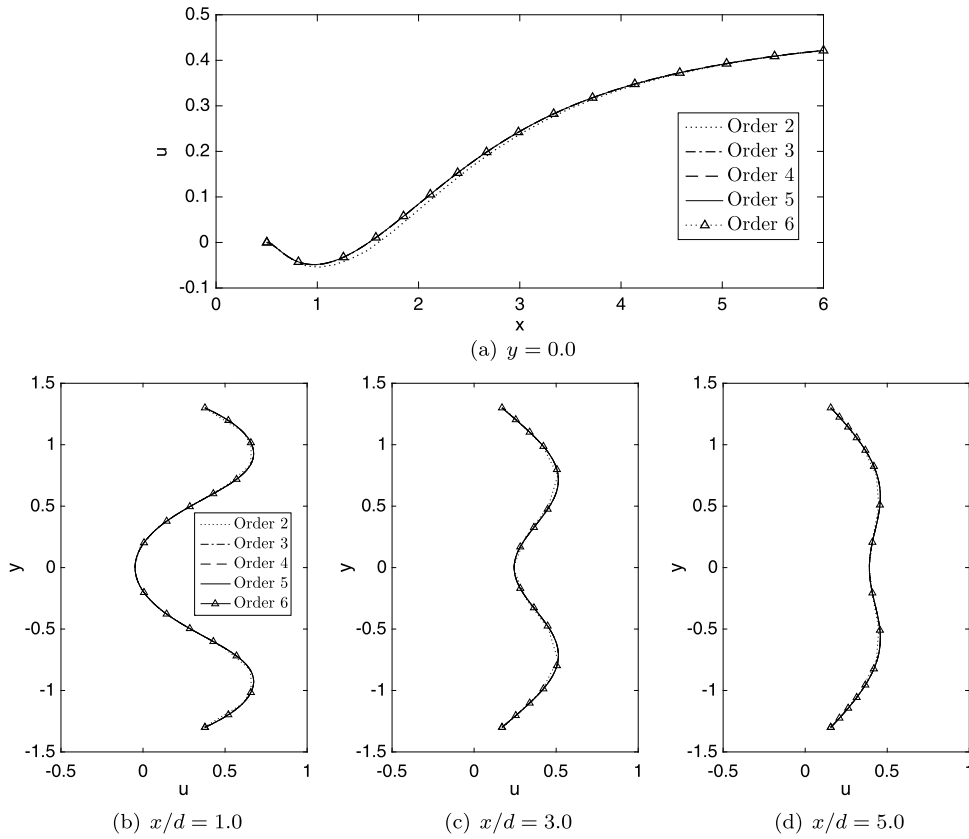


Fig. 9. Comparison of stream-wise velocity profiles ($\nu = 0.01$) computed using different element orders along the centerline of domain (a), and at downstream locations (b) $x/d = 1.0$, (c) $x/d = 3.0$, (d) $x/d = 5.0$. Results are obtained with $\Delta t = 0.01$ and $C_0 = 1000$ in the simulations.

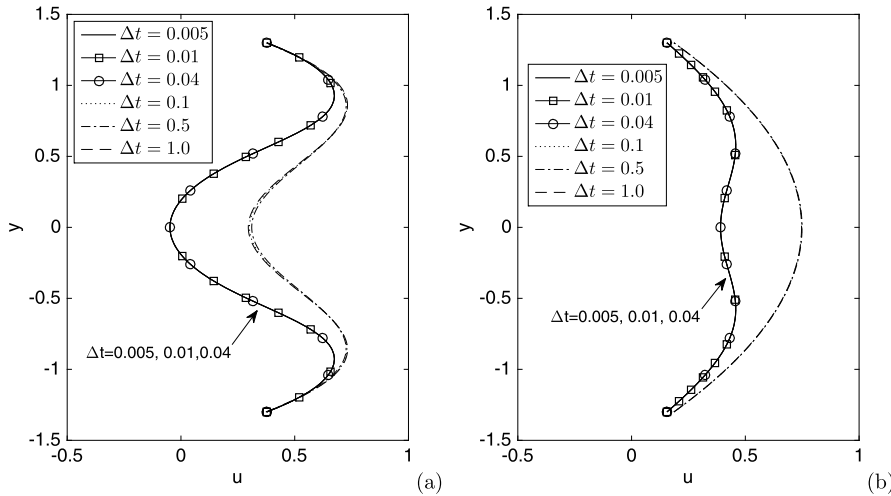


Fig. 10. Effect of the time step size Δt ($\nu = 0.01$): stream-wise velocity profiles at downstream locations (a) $x/D = 1.0$ and (b) $x/D = 5.0$ obtained with different time step sizes.

suggesting the independence with respect to the spatial resolutions. In light of these observations, the majority of simulations reported below are performed using an element order 4, and for higher Reynolds numbers the results corresponding to an element order 5 are also employed in the simulations.

The effect of the time step size Δt on the simulation results is illustrated by Fig. 10, in which we compare the stream-wise velocity profiles along the y direction at the downstream locations $x/D = 1$ and 5 obtained with time step sizes ranging from $\Delta t = 0.005$ to $\Delta t = 1.0$. These results are computed with an element order 4 for the Reynolds number cor-

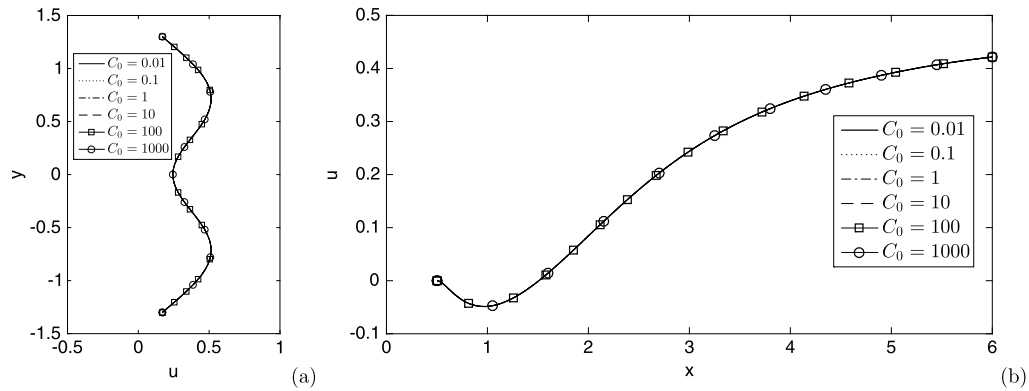


Fig. 11. C_0 effect: Comparison of stream-wise velocity profiles ($\nu = 0.01$) at (a) $x = 3.0$ and (b) along the centerline, obtained using a range of C_0 values in the algorithm.

Table 3

Comparison of the forces on the cylinder/channel walls from the simulations and the imposed pressure gradient and total driving force.

ν	Mean-drag	rms-drag	Mean-lift	rms-lift	Pressure gradient	Driving force
0.01	0.131	0	0	0	0.005	0.131
	0.262	0	0	0	0.01	0.262
	0.524	0	0	0	0.02	0.524
	0.787	0	0	0	0.03	0.786
	1.049	0	0	0	0.04	1.049
	1.311	0	0	0	0.05	1.311
0.002	0.131	0.0105	$9.65e-5$	$3.18e-4$	0.005	0.131
	0.264	0.0487	$-1.97e-6$	$9.14e-4$	0.01	0.262
	0.529	0.150	$-9.83e-5$	$4.22e-3$	0.02	0.524
	0.796	0.293	$2.24e-4$	$8.99e-3$	0.03	0.786
	1.060	0.425	$1.03e-3$	$1.48e-2$	0.04	1.049
	1.325	0.565	$3.57e-4$	$2.13e-2$	0.05	1.311

responding to $\nu = 0.01$, and the total driving force is 0.524 in the domain. The profiles obtained with time step sizes $\Delta t = 0.04$ and smaller all overlap with one another. With larger Δt values (e.g. $\Delta t = 0.1, 0.5$ and 1.0), our method is also able to produce stable simulations at this Reynolds number. But the obtained velocity profiles exhibit a marked difference when compared with those computed using smaller Δt values, indicating that these results are no longer accurate. From numerical experiments we observe that the current method is able to produce stable computations, even with very large Δt values in the simulations, as have been shown here and in Section 3.2. For the results reported subsequently, the simulations are performed mostly with $\Delta t = 0.01$ for lower Reynolds numbers ($\nu \geq 0.005$), and with $\Delta t = 0.001$ for higher Reynolds numbers ($\Delta t = 2.5e-4$ or varied for the Reynolds number corresponding to $\nu = 0.0002$).

The constant C_0 in the current algorithm has been observed to have some influence on the accuracy of the steady-state solution for the Kovasznay flow in the previous section. It is further noted that the error levels are all very low ($10^{-13} \sim 10^{-7}$) corresponding to the entire range of C_0 values there. For practical flow situations it seems very unlikely for these minuscule differences in the flow field to have any effect on the meaningful physical quantities. This is indeed what we have observed with the cylinder in channel problem. Whether C_0 affects the accuracy of results for the current problem has also been studied, and we observe no apparent effect of the C_0 value on the forces or the flow field distributions. For the Reynolds number corresponding to $\nu = 0.01$ and a total driving force 0.524, we have performed simulations with the C_0 constant ranging from $C_0 = 0.01$ to $C_0 = 1000$ in the algorithm. The computed forces on the cylinder/channel walls from all these simulations match the total force imposed to drive the flow. Fig. 11 compares the stream-wise velocity profiles at the downstream location $x/D = 3$ and along the centerline of the domain ($y = 0$) computed using different C_0 values in the current algorithm. The results correspond to an element order 4 and time step size $\Delta t = 0.01$ in the simulations. The velocity profiles for different cases exactly overlap with one another, suggesting that the computed velocity field is not sensitive to C_0 in the simulations. In the discussions of subsequent results, a value $C_0 = 1000$ has been employed in the simulations unless otherwise specified.

We have varied the magnitude of the driving pressure gradient systematically ranging from $\frac{\Delta P}{g_0 D} = 0.005$ to 0.05, and carried out simulations corresponding to each of the force values. In Table 3 we list the drag and lift on the walls obtained from the simulations for two fixed Reynolds numbers corresponding to $\nu = 0.01$ and $\nu = 0.002$. The imposed pressure gradient and the total driving force in the flow domain have also been shown. These results are obtained with an element order 4, and a time step size $\Delta t = 0.01$ for $\nu = 0.01$ and $\Delta t = 0.001$ for $\nu = 0.002$ in the simulations. Note that physically the mean drag from the simulations is expected to match the imposed total driving force. At the lower Reynolds number

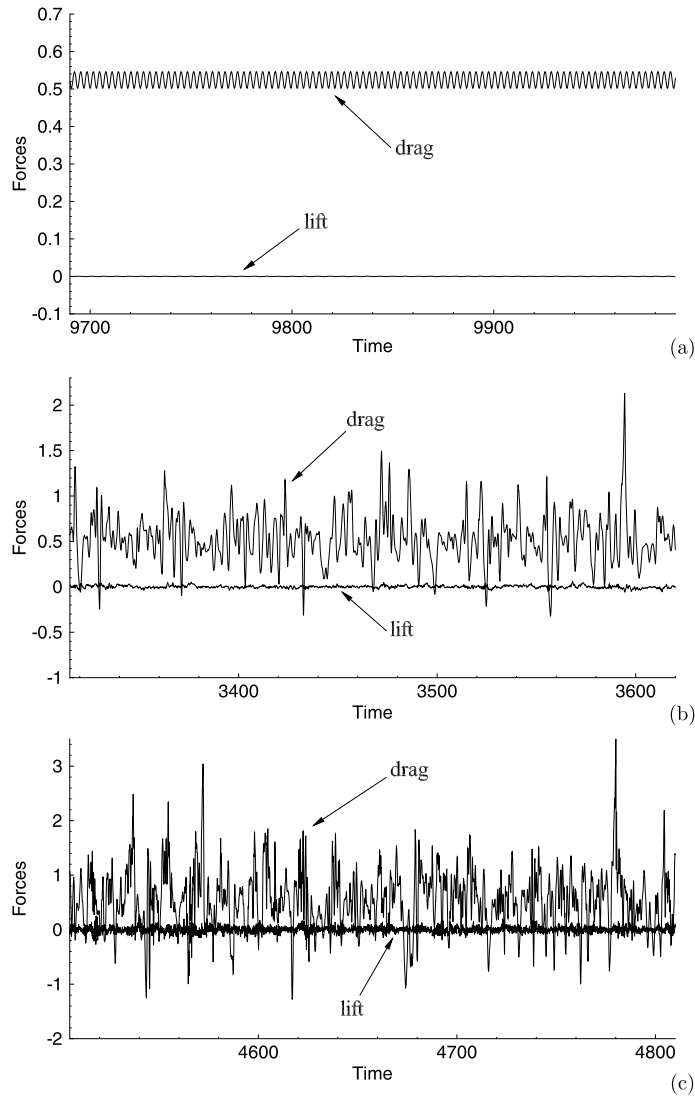


Fig. 12. Time histories of the drag and lift on the cylinder/channel walls for (a) $\nu = 0.005$, (b) $\nu = 0.001$, and (c) $\nu = 0.0002$. The total driving force in the domain is 0.524.

($\nu = 0.01$) the computed mean drag on the walls is essentially the same as the total driving force for different cases. At the higher Reynolds number ($\nu = 0.002$) the computed mean-drag values also agree well with the driving forces. The maximum discrepancy is a little over 1% for the range of driving forces considered here. The mean lift is zero or essentially zero, and the rms lift is also observed to be small.

Fig. 12 shows a window of the time histories of the drag and lift on the domain walls at three Reynolds numbers corresponding to $\nu = 0.005$, 0.001 and 0.0002 with a non-dimensional driving force 0.524 in the domain. Periodic vortex shedding behind the cylinder can be observed at $\nu = 0.005$ (see Fig. 8(c) for the vorticity distribution), resulting in a periodic drag signal fluctuating about a constant mean value. As the viscosity decreases, vortices shed to the cylinder wake persist downstream and can be observed to re-enter the domain through the left boundary. The interactions between these upstream vortices and the cylinder cause the vortex shedding from the cylinder and the forces acting on the walls to become highly irregular. At the higher Reynolds numbers corresponding to $\nu = 0.001$ and $\nu = 0.0002$ one can observe chaotic fluctuations in the time histories of the drag and lift on the channel/cylinder walls (Fig. 12(b,c)). Because of the confinement effect of the channel, the total lift acting on the domain walls appears quite insignificant, and it does not exhibit the large fluctuations as typically observed in the flow past a cylinder in an open domain (see e.g. [10,13]).

We have simulated this flow problem for a range of Reynolds numbers corresponding to viscosities ranging from $\nu = 0.02$ to $\nu = 0.0002$, with the total driving force fixed at 0.524 (corresponding to a non-dimensional pressure gradient 0.02). Table 4 lists the mean and rms forces acting on the walls obtained from the simulations corresponding to different Reynolds numbers. In the simulations the element order is 5 for the case $\nu = 0.0002$ and 4 for the other cases. As expected,

Table 4

Forces on cylinder/channel walls obtained with different fluid viscosities. Simulations are performed with an element order 5 for $\nu = 0.0002$ and 4 for the other ν values.

ν	Mean-drag	rms-drag	Mean-lift	rms-lift	Driving force
0.02	0.524	0.0	0.0	0.0	0.524
0.01	0.524	0.0	0.0	0.0	0.524
0.005	0.525	0.0157	$-1.94e-8$	$3.87e-4$	0.524
0.00333	0.526	0.0751	$2.71e-4$	$9.92e-4$	0.524
0.002	0.529	0.150	$-9.83e-5$	$4.22e-3$	0.524
0.001	0.529	0.273	$3.39e-4$	$1.40e-2$	0.524
0.0005	0.523	0.388	$2.24e-4$	$3.56e-2$	0.524
0.0002	0.528	0.538	$6.03e-4$	$5.59e-2$	0.524

Table 5

Forces on cylinder/channel walls computed with various Δt values for Reynolds numbers corresponding to $\nu = 0.001$ and 0.0002 . Driving force is 0.524, and the element order is 4 with $\nu = 0.001$ and 5 with $\nu = 0.0002$ in the simulations.

ν	Δt	Mean-drag	rms-drag	Mean-lift	rms-lift
0.001	5e-4	0.530	0.276	$-7.00e-4$	1.62e-2
	7.5e-4	0.528	0.265	$5.49e-4$	1.48e-2
	1e-3	0.529	0.273	$3.39e-4$	1.40e-2
	2.5e-3	0.529	0.121	$-1.32e-5$	2.69e-3
	5e-3	0.528	0.114	$2.26e-4$	2.29e-3
	1e-2	0.527	0.0923	$-7.32e-5$	1.33e-3
0.0002	5e-5	0.528	0.535	$2.94e-3$	6.14e-2
	1e-4	0.528	0.538	$6.03e-4$	5.59e-2
	2.5e-4	0.528	0.549	$6.09e-4$	4.69e-2
	5e-4	0.534	0.404	$1.14e-3$	1.89e-2
	1e-3	0.537	0.420	$-1.51e-4$	1.60e-2
	5e-3	0.524	0.233	$-2.02e-4$	4.70e-2
	1e-2	0.527	0.205	$1.61e-4$	8.93e-3

the mean drag values obtained from the simulations are close to or the same as the driving force for different Reynolds numbers, indicating that the method has captured the flow quite well. The rms drag is observed to increase significantly with increasing Reynolds number (decreasing ν), while the rms lift remains insignificant for the range of Reynolds numbers considered here.

The dynamics of this flow is illustrated by the temporal sequence of snapshots of the velocity fields shown in Fig. 13 for the Reynolds number corresponding to $\nu = 0.0002$. In addition to the multitude of vortices permeating the cylinder wake, a prominent feature of this flow lies in the upstream vortices and their interactions with the cylinder. Such interactions give rise to complicated dynamics of the flow. Some upstream vortices can simply squeeze through the gap between the cylinder and the channel wall and move downstream into the wake. However, at times this interaction can be more complicated and help spawn new vortices. This is illustrated by the vortex just in front of the cylinder and near the centerline in Figs. 13(a) and (b). This incoming vortex appears to collide with the cylinder (Figs. 13(b)), generating a strong shear layer near the lower front shoulder of the cylinder. This subsequently spawns a new vortex at the lower front shoulder of the cylinder, which is evident from Fig. 13(c). Other processes, e.g. the coalescence of vortices as evidenced by Figs. 13(a)–(d) with the two vortices near the rear upper shoulder of the cylinder, can also be observed in the cylinder wake.

Thanks to the discrete energy stability property (Theorem 2.1), our algorithm can achieve stable simulations with fairly large time step sizes. In Table 5 we have listed the mean and rms drag and lift forces on the cylinder/channel walls for two Reynolds numbers corresponding to $\nu = 0.001$ and $\nu = 0.0002$ obtained with a number of time step sizes ranging from $\Delta t = 5e-5$ to $\Delta t = 0.01$. The element orders are 4 and 5 for these two Reynolds numbers, respectively, and $C_0 = 1000$ in the simulations, with a total driving force 0.524 in the domain. The values of the time-averaged mean drag obtained from the simulations are close to the driving force. When the time step size is small, the computed mean and rms drags corresponding to different Δt values are close to one another, exhibiting a sense of convergence. As Δt becomes large, we observe a decrease in the computed rms drag when compared with those corresponding to small Δt values. These results indicate that, while the simulations with all these Δt values are stable, the simulation results can become less accurate or inaccurate when Δt is too large for a given Reynolds number.

Finally we look into the computational cost of the scheme developed in this work. For the cylinder flow problem in a periodic channel we have monitored the wall clock time per time step in the simulations. In Table 6 we list the typical wall time it takes the current scheme to compute one time step (in seconds) on a single processor for the Reynolds number corresponding to $\nu = 0.0002$, as well as the wall time spent in the Newton solver for solving the scalar equation (38). These wall-time numbers are collected on a Linux cluster in the authors' institution. The cost of the Newton solver is insignificant, accounting for about 4% of the total cost of the current scheme per time step. Essentially all this time is spent on computing the coefficients given in equation (39) in preparation for the Newton iteration, while the actual time spent in the Newton

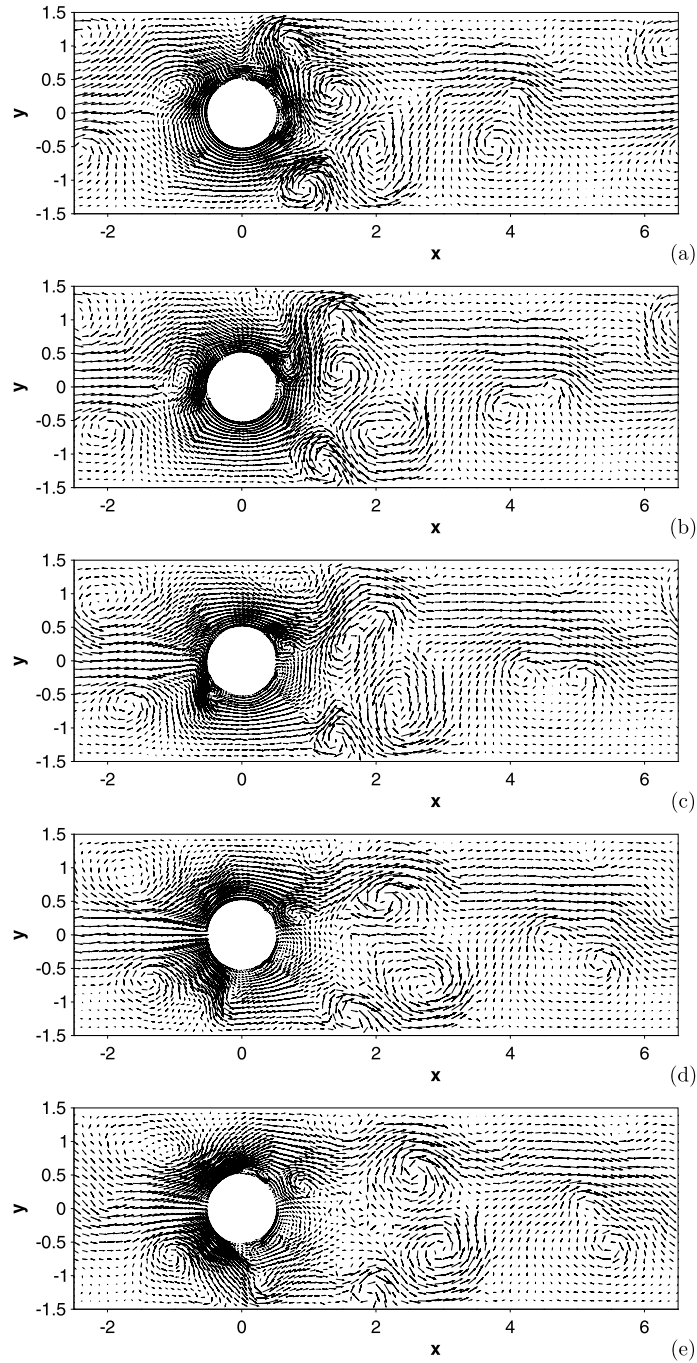


Fig. 13. Temporal sequence of snapshots of the velocity fields ($\nu = 0.0002$, total driving force 0.524) at time (a) $t = 4886$, (b) $t = 4886.5$, (c) $t = 4887$, (d) $t = 4887.5$, (e) $t = 4888$. Velocity vectors are plotted on every eighth quadrature points in each direction within each element.

Table 6

Computational cost: comparison of wall time per time step (single processor) between the current scheme and the semi-implicit scheme of [8] for the flow past a cylinder in a periodic channel with $\nu = 0.0002$ (element order 5, $\Delta t = 2.5e - 4$). The total wall time contains the Newton solver time for the current scheme.

	Newton-solver time/step (seconds)	Total wall time/step (seconds)
Current scheme	0.00654	0.161
Semi-implicit scheme [8]	–	0.0883

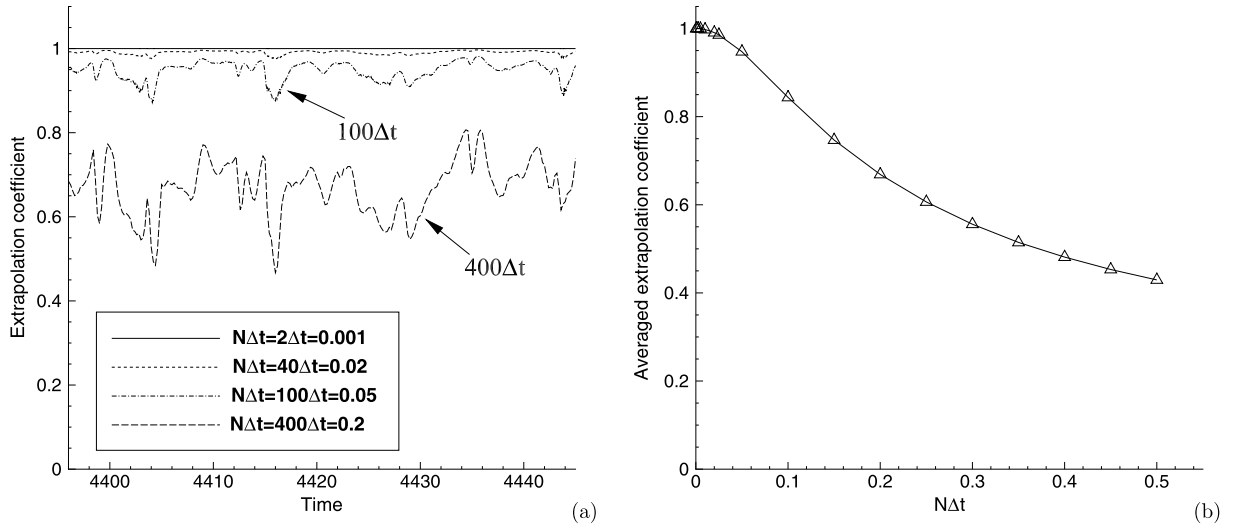


Fig. 14. Effect of velocity extrapolation on the nonlinear term ($\nu = 0.001$, $\Delta t = 0.0005$): (a) Time histories of the extrapolation coefficient, $\sqrt{\frac{\int_{\Omega} |\mathbf{u}^n \cdot \nabla \mathbf{u}^n|^2}{\int_{\Omega} |\bar{\mathbf{u}}^n \cdot \nabla \bar{\mathbf{u}}^n|^2}}$, where \mathbf{u}^n denotes the true velocity at time step n , and $\bar{\mathbf{u}}^n$ is the extrapolated velocity given by $\bar{\mathbf{u}}^n = 2\mathbf{u}^{n-N} - \mathbf{u}^{n-2N}$. (b) Time-averaged extrapolation coefficient as a function of $N\Delta t$.

iteration is negligible. For comparison the table also includes the wall clock time per time step of the semi-implicit scheme from [8]. The current scheme requires the solution of two pressure fields p_1^{n+1} and p_2^{n+1} and two velocity fields \mathbf{u}_1^{n+1} and \mathbf{u}_2^{n+1} , which involves approximately twice as many operations as the semi-implicit scheme. The computational cost of the current scheme is roughly twice that of the semi-implicit scheme, as is evident from Table 6.

4. Concluding remarks

We have presented an algorithm for approximating the incompressible Navier-Stokes equations based on an auxiliary variable associated with the total energy of the system. This auxiliary variable is a scalar number (not a field function), and a dynamic equation for this variable has been introduced. This leads to a reformulated equivalent system consisting of the modified incompressible Navier-Stokes equations and the dynamic equation for the auxiliary variable. The numerical scheme for the reformulated system satisfies a discrete energy stability property in terms of a modified energy. Within each time step, the algorithm requires the computations in a de-coupled fashion of (i) two pressure variables p_1^{n+1} and p_2^{n+1} , (ii) two velocity variables \mathbf{u}_1^{n+1} and \mathbf{u}_2^{n+1} , and (iii) the scalar auxiliary variable. Computing the pressure variables and the velocity variables involves only the usual Poisson equations and Helmholtz equations with constant coefficient matrices. Computing the auxiliary variable requires the solution of a nonlinear *scalar* algebraic equation based on the Newton's method. The cost for the Newton solution is insignificant and essentially negligible (accounting for a few percent of the total cost per time step), because the nonlinear equation is about a scalar number, not a field function.

The current algorithm employs a reformulated form of the nonlinear term,

$$\frac{R^{n+1}}{\sqrt{E^{n+1}}} \mathbf{N}(\bar{\mathbf{u}}^{n+1}) = \frac{R^{n+1}}{\sqrt{E^{n+1}}} (\bar{\mathbf{u}}^{n+1} \cdot \nabla \bar{\mathbf{u}}^{n+1}) = \left(\sqrt{\frac{R^{n+1}}{E^{n+1}}} \bar{\mathbf{u}}^{n+1} \right) \cdot \nabla \left(\sqrt{\frac{R^{n+1}}{E^{n+1}}} \bar{\mathbf{u}}^{n+1} \right) = \mathbf{u}^* \cdot \nabla \mathbf{u}^* \quad (47)$$

where $\bar{\mathbf{u}}^{n+1} = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ (for 2nd-order) is the extrapolated velocity, and $\mathbf{u}^* = \sqrt{\frac{R^{n+1}}{E^{n+1}}} \bar{\mathbf{u}}^{n+1}$ is a modified extrapolated velocity. The above equation indicates that the current scheme uses the modified extrapolated velocity \mathbf{u}^* , instead of the extrapolated velocity $\bar{\mathbf{u}}^{n+1}$, for the computation of the nonlinear term. Numerical experiments indicate that when Δt is small the modification coefficient $\sqrt{\frac{R^{n+1}}{E^{n+1}}}$ equals the unit value, and for large Δt the modification coefficient can be less than the unit value (see Section 3.2). This is consistent with one's intuitions. The extrapolated velocity $\bar{\mathbf{u}}^{n+1}$ only approximates \mathbf{u}^{n+1} well when Δt is small (2nd-order approximation). For large Δt , $\bar{\mathbf{u}}^{n+1}$ is a poor approximation of \mathbf{u}^{n+1} , and will introduce large errors when it is used to compute the nonlinear term. Therefore, for large Δt some suitable modification to $\bar{\mathbf{u}}^{n+1}$ will be expected if this is used for computing the nonlinear term. In the current algorithm this is modified by the coefficient $\sqrt{\frac{R^{n+1}}{E^{n+1}}}$, which is computed based on a dynamic equation.

The error in the nonlinear term computation and the need for a modification to the extrapolated velocity at large Δt when it is used to compute the nonlinear term are demonstrated by Fig. 14 with the cylinder flow in a periodic channel

problem (see Section 3.3). Here the flow field data for $\nu = 0.001$ with a total driving force 0.524 obtained using the semi-implicit scheme of [8] with a small time step size $\Delta t = 0.0005$ are post-processed to compute the nonlinear term using an extrapolated velocity and using the true velocity. More specifically, at a time step n we compute the nonlinear term in two ways when post-processing the flow data: (i) $\int_{\Omega} |\mathbf{u}^n \cdot \nabla \mathbf{u}^n|^2$, where \mathbf{u}^n is the velocity at step n from the simulation and is referred to as the “true” velocity; (ii) $\int_{\Omega} |\tilde{\mathbf{u}}^n \cdot \nabla \tilde{\mathbf{u}}^n|^2$, where $\tilde{\mathbf{u}}^n = 2\mathbf{u}^{n-N} - \mathbf{u}^{n-2N}$. Note that $\tilde{\mathbf{u}}^n$ is the velocity at step n obtained by extrapolating the velocity fields at steps $(n - N)$ and $(n - 2N)$, where N is a prescribed integer. $\tilde{\mathbf{u}}^n$ therefore mimics an extrapolated velocity corresponding to a “larger” time step size $N\Delta t$. We refer to the ratio $\sqrt{\frac{\int_{\Omega} |\mathbf{u}^n \cdot \nabla \mathbf{u}^n|^2}{\int_{\Omega} |\tilde{\mathbf{u}}^n \cdot \nabla \tilde{\mathbf{u}}^n|^2}}$ as the extrapolation coefficient at time step n . In a sense, the extrapolation coefficient reflects the accuracy of the computed nonlinear term using the extrapolated velocity corresponding to a nominal time step size “ $N\Delta t$ ”. A unit value for the extrapolation coefficient means that using such extrapolated velocity results in a perfectly accurate nonlinear term. If the extrapolation coefficient is significantly less than 1.0, it means that using such extrapolated velocity to compute the nonlinear term tends to produce a value substantially larger than the true value in magnitude. Fig. 14(a) shows time histories of the extrapolation coefficient corresponding to several $N\Delta t$ values, and Fig. 14(b) shows the time-averaged extrapolation coefficient as a function of $N\Delta t$. With $N\Delta t = 2\Delta t = 0.001$, the extrapolation coefficient is essentially the unit value, indicating that the nonlinear term computed using this extrapolated velocity accurately approximates the true nonlinear term. With increasing $N\Delta t$, we observe a generally smaller value (< 1) for the extrapolation coefficient. With large $N\Delta t$ the extrapolation coefficient is substantially smaller than the unit value. For example, with $N\Delta t = 400\Delta t = 0.2$ the extrapolation coefficient fluctuates about the value 0.67, suggesting that a modified extrapolated velocity for the nonlinear term computation, $0.67\tilde{\mathbf{u}}^n \cdot \nabla \tilde{\mathbf{u}}^n$, would likely result in a better result. The trend exhibited by these results is qualitatively consistent with the spirit of the current algorithm, and demonstrates the need for a modification of the extrapolated velocity if it is used to compute the nonlinear term at large time step sizes. However, it should be noted that the extrapolation coefficient value obtained above cannot be compared quantitatively with the quantity $\frac{R^{n+1}}{\sqrt{E^{n+1}}}$ from the current algorithm. This is because the above extrapolation coefficient is obtained by *post-processing* the simulation data and by always extrapolating the underlying stable simulated velocity fields, no matter how large $N\Delta t$ is. In this process, the extrapolated velocity field with step size $N\Delta t$ is never used to calculate the flow at a new step, as occurs in actual simulations when using the current algorithm.

The current algorithm has been implemented based on a C^0 spectral element technique in this work, and several numerical examples have been presented to test its accuracy and performance. The method is observed to exhibit a second-order convergence rate in time and an exponential convergence rate in space (for smooth field solutions). It can capture the flow field accurately when the time step size is not too large. The method also allows the use of large time step sizes in computations, and stable simulation results can be produced.

The presented scheme has an attractive energy stability property, and it can be implemented in an efficient fashion. The algorithm involves only constant and time-independent coefficient matrices in the resultant linear algebraic systems, which is unlike other energy-stable schemes for Navier-Stokes equations (see e.g. [32,24,11], among others). This algorithm can serve as an alternative to the semi-implicit schemes for production simulations of incompressible flows and flow physics studies.

Acknowledgement

This work was partially supported by National Science Foundation (DMS-1318820, DMS-1522537).

References

- [1] H. Abels, H. Garcke, G. Grün, Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities, *Math. Models Methods Appl. Sci.* 22 (2012) 1150013.
- [2] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 168 (2001) 464–499.
- [3] H. Chen, S. Sun, T. Zhang, Energy stability analysis of some fully discrete numerical schemes for incompressible Navier-Stokes equations on staggered grids, *J. Sci. Comput.* 75 (2018) 427–456.
- [4] L. Chen, J. Shen, C.J. Xu, A unstructured nodal spectral-element method for the Navier-Stokes equations, *Commun. Comput. Phys.* 12 (2012) 315–336.
- [5] A.J. Chorin, Numerical solution of the Navier-Stokes equations, *Math. Comput.* 22 (1968) 745–762.
- [6] S. Dong, Direct numerical simulation of turbulent Taylor-Couette flow, *J. Fluid Mech.* 587 (2007) 373–393.
- [7] S. Dong, Evidence for internal structures of spiral turbulence, *Phys. Rev. E* 80 (2009) 067301.
- [8] S. Dong, A convective-like energy-stable open boundary condition for simulations of incompressible flows, *J. Comput. Phys.* 302 (2015) 300–328.
- [9] S. Dong, Multiphase flows of N immiscible incompressible fluids: a reduction-consistent and thermodynamically-consistent formulation and associated algorithm, *J. Comput. Phys.* 361 (2018) 1–49.
- [10] S. Dong, G.E. Karniadakis, A. Ekmekci, D. Rockwell, A combined direct numerical simulation-particle image velocimetry study of the turbulent near wake, *J. Fluid Mech.* 569 (2006) 185–207.
- [11] S. Dong, J. Shen, An unconditionally stable rotational velocity-correction scheme for incompressible flows, *J. Comput. Phys.* 229 (2010) 7013–7029.
- [12] S. Dong, J. Shen, A time-stepping scheme involving constant coefficient matrices for phase field simulations of two-phase incompressible flows with large density ratios, *J. Comput. Phys.* 231 (2012) 5788–5804.
- [13] S. Dong, G.S. Triantafyllou, G.E. Karniadakis, Elimination of vortex streets in bluff body flows, *Phys. Rev. Lett.* 100 (2008) 204501.
- [14] S. Dong, X. Zheng, Direct numerical simulation of spiral turbulence, *J. Fluid Mech.* 668 (2011) 150–173.
- [15] J.L. Guermond, P. Mineev, J. Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 6011–6045.

- [16] J.L. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, *J. Comput. Phys.* 192 (2003) 262–276.
- [17] B. Hyounsu, G.E. Karniadakis, Subiteration leads to accuracy and stability enhancements of semi-implicit schemes for the Navier-Stokes equations, *J. Comput. Phys.* 230 (2011) 4384–4402.
- [18] N. Jiang, M. Mohebujjaman, L.G. Rebholz, C. Trechea, An optimally accurate discrete regularization for second order timestepping methods for Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.* 310 (2016) 388–405.
- [19] 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.
- [20] G.E. Karniadakis, S.J. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, 2nd ed., Oxford University Press, 2005.
- [21] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [22] L.I.G. Kovasznay, Laminar flow behind a two-dimensional grid, *Proc. Camb. Philol. Soc.* 44 (1948) 58.
- [23] A.G. Kravchenko, P. Moin, Numerical studies of flow over a circular cylinder at $Re_D = 3900$, *Phys. Fluids* 12 (2000) 403–417.
- [24] 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.
- [25] J.-G. Liu, J. Liu, R.L. Pego, Stability and convergence of efficient Navier-Stokes solvers via a commutator estimate, *Commun. Pure Appl. Math.* LX (2007) 1443–1487.
- [26] X. Ma, C.S. Karamanos, G.E. Karniadakis, Dynamics and low-dimensionality of a turbulent near wake, *J. Fluid Mech.* 410 (2000) 29–65.
- [27] B. Sanderse, Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 233 (2013) 100–131.
- [28] D. Serson, J.R. Meneghini, S.J. Sherwin, Velocity-correction schemes for the incompressible Navier-Stokes equations in general coordinate systems, *J. Comput. Phys.* 316 (2016) 243–254.
- [29] J. Shen, On error estimate of projection methods for Navier-Stokes equations: first-order schemes, *SIAM J. Numer. Anal.* 29 (1992) 57–77.
- [30] J. Shen, J. Xu, J. Yang, The scalar auxiliary variable (SAV) approach for gradient flows, *J. Comput. Phys.* 353 (2018) 407–416.
- [31] S.J. Sherwin, G.E. Karniadakis, A triangular spectral element method: applications to the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.* 123 (1995) 189–229.
- [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] 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.
- [34] R.W.C.P. Verstappen, A.E.P. Veldman, Symmetry-preserving discretization of turbulent flow, *J. Comput. Phys.* 187 (2003) 343–368.
- [35] C.J. Xu, R. Pasquetti, On the efficiency of semi-implicit and semi-Lagrangian spectral methods for the calculation of incompressible flows, *Int. J. Numer. Methods Fluids* 35 (2001) 319–340.
- [36] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex fluids, *J. Fluid Mech.* 515 (2004) 293–317.
- [37] X. Zheng, S. Dong, An eigen-based high-order expansion basis for structured spectral elements, *J. Comput. Phys.* 230 (2011) 8573–8602.