New splitting methods for convection-dominated diffusion problems and Navier-Stokes equations

Feng Shi[*]   Guoping Liang[†]   Yubo Zhao[‡]   Jun Zou [§]

June 11, 2014

Abstract

We present a new splitting method for time-dependent convention-dominated diffusion problems. The original convention diffusion system is split into two sub-systems: a pure convection system and a diffusion system. At each time step, a convection problem and a diffusion problem are solved successively. A few important features of the scheme lie in the facts that the convection subproblem is solved explicitly and multistep techniques can be used to essentially enlarge the stability region so that the resulting scheme behaves like a unconditionally stable scheme; while the diffusion subproblem is always self-adjoint and coercive so that they can be solved efficiently using many existing optimal preconditioned iterative solvers. The scheme can be extended for solving the Navier-Stokes equations, where the nonlinearity is resolved by a linear explicit multistep scheme at the convection step, while only a generalized Stokes problem is needed to solve at the diffusion step and the major stiffness matrix stays invariant in the time marching process. Numerical simulations are presented to demonstrate the stability, convergence and performance of the single-step and multistep variants of the new scheme.

Key Words. Convention-dominated diffusion problems, Navier-Stokes equations, operator splitting, finite elements, multistep scheme.

AMS Classification. 65M12, 65M60, 76D05

1   Introduction

In this work we shall first propose a new fully discrete splitting scheme for solving the convention-dominated diffusion problems of the following general form

$$u_t + \nabla \cdot (\mathbf{b}u) - \nabla \cdot (\varepsilon \nabla u) + cu = F \quad \text{in} \quad \Omega \times (0, T) \tag{1}$$

with the boundary and initial conditions

$$u = u_b \quad \text{on} \quad \partial\Omega \times (0, T) ; \quad u(0, \mathbf{x}) = u_0(\mathbf{x}) \quad \text{in} \quad \Omega \tag{2}$$

where Ω is an open bounded polyhedral domain in $\mathbb{R}^d$ $(d = 1, 2, 3)$ with boundary $\Gamma = \partial\Omega$, and $[0, T]$ is the time interval. Functions $\mathbf{b}$ and c in (1) are the convective field and reactive coefficients respectively, and $\varepsilon > 0$ is a constant diffusion coefficient, while F, u_b and u_0 are the specified source term, the boundary and initial data respectively. As we are mainly interested in the construction of numerical schemes, we will not specify some detailed regularity conditions on all these coefficients to ensure the well-posedness of the initial-boundary value problem (1)-(2).

Then the new fully discrete splitting scheme will be extended for solving the Navier-Stokes equations

$$\begin{cases} \mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} - Re^{-1}\triangle\mathbf{u} + \nabla p = \mathbf{F} & \text{in} \quad \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in} \quad \Omega \times (0, T) \end{cases} \tag{3}$$

with the boundary and initial conditions

$$\mathbf{u} = \mathbf{u}_b \quad \text{on} \quad \partial\Omega \times (0, T) ; \quad \mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}) \quad \text{in} \quad \Omega \tag{4}$$

where $\mathbf{u}$, p, $\mathbf{F}$ and Re are respectively the velocity, the pressure, the body force and the Reynolds number, while $\mathbf{u}_b$ and $\mathbf{u}_0$ are the given boundary and initial data.

The numerical solution of a time-dependent problem requires a discretization in both time and space, and possibly some linearization if the problem is nonlinear. A great variety of time marching schemes are available in the literature, such as the classical methods like the forward and backward Euler schemes, the Crank-Nicolson scheme, the Adams-Bashforth method etc. Operator splitting is also a popular technique for time discretization, such as the Yanenko method, the Peaceman-Rachford method, the Douglas-Rachford method and the θ scheme; see [1, 2, 3] and references therein.

In solving the convection-dominated diffusion equations and the Navier-Stokes equations with large Reynolds numbers, it is well known that standard finite element methods perform poorly and may exhibit nonphysical oscillations. Many spatial stabilization techniques have been proposed and studied. The streamline-upwind Petrov-Galerkin method was originally developed in [4, 5] for convective transport problems, and its basic idea is to modify the standard Petrov-Galerkin formulation by adding a streamline upwind perturbation, which acts only in the flow direction and is solely defined in the interiors of elements. The Galerkin least-squares method [6] is a conceptual simplification of the streamline-upwind Petrov-Galerkin method, and adds a stabilization that involves an element-by-element weighted least-squares of the residual to the original differential equation. The efficiency of these two stabilization techniques are affected by the choices of stabilization parameters involved. There are still no precise general formulas to help select optimal parameters in numerical simulations; see, e.g., [7, Remark 10.4]. These stabilization parameters may depend possibly also on time step size for time-dependent problems, so their choices become more tricky in practice as we have to balance between temporal and spatial errors, which are usually given by the orders of the methods [8].

By changing the sign of the convective term in the weighted least-squares formulation, the unusual stabilized finite element method (USFEM) can achieve the absolute stability for any positive stabilization parameter involved in the scheme, but it is still a tricky and inconclusive technical issue of how to choose this parameter in order to obtain good accuracy [9, 10, 11, 12].

The variational multiscale method was developed based on the inherent multiscale structure of the solutions [13, 14, 15, 16]. This method defines the large scales by a projection into an appropriate subspace, but also involves the technical issue of how to select a stabilization parameter to balance the stability and accuracy.

As it is known [5], explicit Galerkin solutions for flow problems could be quite under-diffusive, effectively increasing the Peclet or Reynolds number. Furthermore, explicit methods are generally conditionally stable. But explicit schemes have their own advantages, e.g., they may not need to solve systems of algebraic equations [17] or the resulting stiff matrices stay the same in the time marching process.

The characteristic-based-split (CBS) method has been widely studied for fluid and solid dynamic problems [18, 19, 20, 21], and we refer to the monograph [22] and the references therein for its detailed introduction and various applications. This method is based on the splitting of the convection and diffusion parts. The convection part is formally handled by the standard characteristics method, where the numerical solutions at the current time are updated by the approximations at the previous time. But the computed spatial points, which is determined by characteristics, are likely no longer grid points of the spatial discretization. Therefore we need to compute the solutions at those spatial points using the solutions and other quantities at grid points. Two natural ways for the purpose are by updating the meshes and standard interpolation. An alternative technique, used in the CBS method, is to approximate numerical solutions at computed spatial points by the solutions and other quantities at grid points by Taylor expansion. In addition, the CBS method needs to approximate the average convective field, for which different treatments may lead to different schemes, such as fully explicit, semi-implicit or implicit ones, and also different stabilization effects [22].

In the derivation of our new scheme, we shall use the same operator splitting as the CBS method did, to split the convection diffusion system into a purely convective part and a diffusion part. The diffusion part is discretized by the standard backward scheme. But the central difference from the CBS method lies in our new treatment of the convection part, which is completely independent of the characteristic curves and any spatial grid points used, unlike the CBS method.

Another novel idea of our new method is the flexibility in its special explicit treatment of the convection part: we can recursively execute the explicit convection step up to a finite number of times with smaller local time step size during one diffusion correction. This can essentially improve the stability of the resulting scheme.

The rest of the paper is arranged as follows. The single-step scheme is first derived for the convection diffusion equation in Section 2.1, and its multistep variant in Section 2.2. The new scheme is then extended in Section 3 for the Navier-Stokes equations. Numerical experiments are carried out in Section 4 to check the accuracy, stability and performance of the new schemes, as well as to investigate how the stability condition can be improved by the multistep scheme compared with the single-step one. At the end of this numerical section, the driven cavity flow problem is tested with the new scheme and compared with the benchmark results to demonstrate the validity of the new method. Some concluding remarks are given in Section 5.

2 Derivation of algorithms

In this section we shall derive a new method for solving the convection-dominated diffusion equation (1). For this we introduce some notations. We first partition the time interval $[0, T]$: $0 = t_0 < t_1 < \cdots < t_N = T$, with $t_n = n\Delta t$ and $\Delta t = \frac{T}{N}$. We will use u^n and $u^{n+\frac{1}{2}}$ respectively for the approximate values of $u(\cdot, t)$ at $t = t_n$ and $t_n + \frac{\Delta t}{2}$. But when $u(\cdot, t)$ is given, u^n and $u^{n+\frac{1}{2}}$ will stand for its exact values at $t = t_n$ and $t_n + \frac{\Delta t}{2}$, e.g., $f^n = f(\cdot, t_n)$, and $\mathbf{b}^n = \mathbf{b}(\cdot, t_n)$.

2.1 Single-step scheme for the convection diffusion equation

We first adopt the standard operator splitting technique [3] and split the convection diffusion equation (1) into a pure convection equation and a diffusion equation. Then we approximate the two equations in time by the central difference and backward Euler schemes respectively to obtain

$$\frac{u_*^{n+1} - u^n}{\Delta t} + \nabla \cdot (\mathbf{b}^{n+\frac{1}{2}} u^{n+\frac{1}{2}}) = f^{n+\frac{1}{2}}, \tag{5}$$

$$\frac{u^{n+1} - u_*^{n+1}}{\Delta t} - \nabla \cdot (\varepsilon \nabla u^{n+1}) + c^{n+1} u^{n+1} = g^{n+1}, \tag{6}$$

where f and g can be any functions such that $F = f + g$. However in order to have a unified principle for the selection of the components f and g for both the convection diffusion equation and Navier-Stokes equations, we will suggest some special selection of f and g later on; see Remark 3.1.

We shall use finite element methods to solve (5) and (6) respectively for the solutions u_*^{n+1} and u^{n+1}. For this purpose we need the variational formulations of these two equations. It is straightforward to derive the variational form of (6):

Find $u^{n+1} \in H^1(\Omega)$ such that $u^{n+1} = u_b^{n+1}$ on Γ and solves

$$(u^{n+1}, v) + \Delta t(\varepsilon \nabla u^{n+1}, \nabla v) + \Delta t(c^{n+1} u^{n+1}, v) = (u_*^{n+1}, v) + \Delta t(g^{n+1}, v) \quad \forall v \in H_0^1(\Omega). \tag{7}$$

On the other hand, the solution of the convection step (5) is more tricky. Clearly the scheme is implicit and involves the solution of a linear convection equation. The main idea of this work is to propose an explicit scheme to solve this linear convection equation. To do so, we apply the Taylor's expansion to compute $u^{n+\frac{1}{2}}$ by the values at previous times. We can write

$$u^{n+\frac{1}{2}} \approx u(\mathbf{x}, t_n + \frac{\Delta t}{2}) = u(\mathbf{x}, t_n) + \frac{\Delta t}{2} u_t(\mathbf{x}, t_n) + O(\Delta t^2),$$

then using the convection equation

$$u_t + \nabla \cdot (\mathbf{b} \, u) = f \tag{8}$$

we deduce

$$u^{n+\frac{1}{2}} \approx u^n + \frac{\Delta t}{2} (f^n - \nabla \cdot (\mathbf{b}^n u^n)) =: \xi^n. \tag{9}$$

Using this relation, we can rewrite (5) as

$$\frac{u_*^{n+1} - u^n}{\Delta t} + \nabla \cdot \left(\mathbf{b}^{n+\frac{1}{2}} \xi^n\right) = f^{n+\frac{1}{2}}. \tag{10}$$

Noting that (8) is a pure convective equation, only partial boundary condition on the inflow boundary

$$\Gamma_t^- := \{\mathbf{x} \in \Gamma;\ \mathbf{b}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) < 0\} \tag{11}$$

should be imposed. Accordingly we should set similar conditions on the inflow boundary associated with the scheme (10). So for any positive integer n, we define

$$\Gamma_n^- := \{\mathbf{x} \in \Gamma;\ \mathbf{b}^n(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) < 0\}. \tag{12}$$

As the exact solution is specified on the entire boundary (cf. (1)), it is natural for us to assume the values for the solution u_*^{n+1} to (10) on the inflow boundary Γ_{n+1}^-:

$$u_*^{n+1} = u_b^{n+1} \quad \text{on} \quad \Gamma_{n+1}^-. \tag{13}$$

This induces the following test space for the scheme (10):

$$H^1_{\Gamma_{n+1}^-}(\Omega) = \left\{ w \in H^1(\Omega);\ w = 0 \ \text{on} \ \Gamma_{n+1}^- \right\}.$$

Now multiplying a test function $v \in H^1_{\Gamma_{n+1}^-}(\Omega)$ on both sides of (10), and integrating over Ω and using the integration by parts we obtain

$$
\begin{aligned}
(u_*^{n+1}, v) &= (u^n, v) + \Delta t(f^{n+\frac{1}{2}}, v) \\
&\quad + \Delta t(\xi^n, \mathbf{b}^{n+\frac{1}{2}} \cdot \nabla v) - \Delta t < \xi^n, v\mathbf{b}^{n+\frac{1}{2}} \cdot \mathbf{n} >_{\Gamma \backslash \Gamma_{n+1}^-} \\
&= (u^n, v) + \Delta t(f^{n+\frac{1}{2}}, v) \\
&\quad + \Delta t \left(u^n + \frac{\Delta t}{2} \left(f^n - \nabla \cdot (\mathbf{b}^n u^n) \right), \mathbf{b}^{n+\frac{1}{2}} \cdot \nabla v \right) \\
&\quad - \Delta t \langle u^n + \frac{\Delta t}{2} \left(f^n - \nabla \cdot (\mathbf{b}^n u^n) \right), v\mathbf{b}^{n+\frac{1}{2}} \cdot \mathbf{n} \rangle_{\Gamma \backslash \Gamma_{n+1}^-}.
\end{aligned}
\tag{14}
$$

Now we move to the spatial discretization by some finite element method. Assume that V_h is a finite element space approximating the Sobolev space $H^1(\Omega)$, and I_h is the interpolation operator of $H^1(\Omega)$ into V_h. Then based on the variational formulations (14) and (7), we propose the following single-step scheme for solving the convection-dominated diffusion problem (1).

Algorithm 1 (Single-step scheme).

Step 0. Compute the initial value $u_h^0 = I_h u_0$. For each $n = 0, 1, \cdots, N - 1$, do the following.

Step 1. Find $u_{h,*}^{n+1} \in V_h$ such that $u_{h,*}^{n+1} = I_h u_b^{n+1}$ on Γ_{n+1}^- and solves

$$
\begin{aligned}
(u_{h,*}^{n+1}, v_h) &= (u_h^n, v_h) + \Delta t(f^{n+\frac{1}{2}}, v_h) \\
&\quad + \Delta t \left(u_h^n + \frac{\Delta t}{2} \left(f^n - \nabla \cdot (\mathbf{b}^n u_h^n) \right), \mathbf{b}^{n+\frac{1}{2}} \cdot \nabla v_h \right) \\
&\quad - \Delta t \left\langle u_h^n + \frac{\Delta t}{2} \left(f^n - \nabla \cdot (\mathbf{b}^n u_h^n) \right), v_h \mathbf{b}^{n+\frac{1}{2}} \cdot \mathbf{n} \right\rangle_{\Gamma \backslash \Gamma_{n+1}^-} \quad \forall\, v_h \in V_h \cap H^1_{\Gamma_{n+1}^-}(\Omega).
\end{aligned}
$$

Step 2. Find $u_h^{n+1} \in V_h$ such that $u_h^{n+1} = I_h u_b^{n+1}$ on Γ and solves

$$(u_h^{n+1}, v_h) + \Delta t(\varepsilon \nabla u_h^{n+1}, \nabla v_h) + \Delta t(c^{n+1} u_h^{n+1}, v_h) = (u_{h,*}^{n+1}, v_h) + \Delta t(g^{n+1}, v_h) \quad \forall\, v_h \in V_h \cap H_0^1(\Omega).$$

Remark 2.1. *By computing the term $(u_{h,*}^{n+1}, v_h)$ in Step 1 using the standard mass-lumping technique [17], $u_{h,*}^{n+1}$ can be computed explicitly without solving a linear system.*

2.2 Multistep scheme for the convection diffusion equation

As we are interested mainly in the convection-dominated case in the system (1), the single-step scheme (Algorithm 1) may not be stable enough as it is explicit. To improve the stability, we may execute the convection step (Step 1) a few times for each diffusion correction (Step 2) so that we can use much smaller time step size for the convection part. To do so, we write the result $u_{h,*}^{n+1}$ of Step 1 formally as

$$u_{h,*}^{n+1} = F_{conv}^{CD}\left(\Delta t, f^n, f^{n+1}, \mathbf{b}^n, \mathbf{b}^{n+1}, u_h^n, u_b^{n+1}\right). \tag{15}$$

Then the multistep scheme is to run this convection step m times with smaller time step size $\frac{\Delta t}{m}$ in order to get $u_{h,*}^{n+1}$, namely we compute

$$u_{h,*}^{n+\frac{i}{m}} = F_{conv}^{CD}\left(\frac{\Delta t}{m}, f^{n+\frac{i-1}{m}}, f^{n+\frac{i}{m}}, \mathbf{b}^{n+\frac{i-1}{m}}, \mathbf{b}^{n+\frac{i}{m}}, u_{h,*}^{n+\frac{i-1}{m}}, u_b^{n+\frac{i}{m}}\right), \tag{16}$$

recursively for $i = 1, 2, \cdots, m$, with $u_{h,*}^n = u_h^n$.

   We shall call $\delta t = \frac{\Delta t}{m}$ and Δt as the local time step size and the global time step size respectively. Replacing Step 1 by the multistep iteration (16), we propose the following multistep scheme for the convection diffusion equation (1).

   Algorithm 2 (Multistep scheme with index m).

Step 0. Compute the initial value $u_h^0 = I_h u_0$. For each $n = 0, 1, \cdots, N-1$, do the following.

Step 1. Set $u_{h,*}^n = u_h^n$. For $i = 1, 2, \cdots, m$, compute $u_{h,*}^{n+\frac{i}{m}} \in V_h$ such that $u_{h,*}^{n+\frac{i}{m}} = I_h u_b^{n+\frac{i}{m}}$ on $\Gamma_{n+i/m}^-$ and solves for all $v_h \in V_h \cap H^1_{\Gamma_{n+i/m}^-}(\Omega)$,

$$(u_{h,*}^{n+\frac{i}{m}}, v_h) = (u_{h,*}^{n+\frac{i-1}{m}}, v) + \delta t(f^{n+\frac{2i-1}{2m}}, v_h)$$
$$+ \delta t\left(u_{h,*}^{n+\frac{i-1}{m}} + \frac{\delta t}{2}(f^{n+\frac{i-1}{m}} - \nabla\cdot(\mathbf{b}^{n+\frac{i-1}{m}}u_{h,*}^{n+\frac{i-1}{m}})), \mathbf{b}^{n+\frac{2i-1}{2m}}\cdot\nabla v_h\right)$$
$$- \delta t\left\langle u_{h,*}^{n+\frac{i-1}{m}} + \frac{\delta t}{2}\left(f^{n+\frac{i-1}{m}} - \nabla\cdot(\mathbf{b}^{n+\frac{i-1}{m}}u_{h,*}^{n+\frac{i-1}{m}})\right), v_h\mathbf{b}^{n+\frac{2i-1}{2m}}\cdot\mathbf{n}\right\rangle_{\Gamma\backslash\Gamma_{n+i/m}^-}.$$

Step 2. Compute $u_h^{n+1} \in V_h$ such that $u_h^{n+1} = I_h u_b^{n+1}$ on Γ and solves for all $v_h \in V_h \cap H_0^1(\Omega)$,

$$(u_h^{n+1}, v_h) + \Delta t(\varepsilon\nabla u_h^{n+1}, \nabla v_h) + \Delta t(c^{n+1}u_h^{n+1}, v_h) = (u_{h,*}^{n+1}, v_h) + \Delta t(g^{n+1}, v_h).$$

3 Single-step and multistep schemes for Navier-Stokes equations

We are now going to extend the new schemes proposed in sections 2.1-2.2 for the convection-dominated diffusion equation to the Navier-Stokes equations (3). For the purpose, we split the

system (3) into a pure convection system and a diffusion system (the generalized Stokes problem) as follows:

$$\frac{\mathbf{u}_*^{n+1} - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^{n+\frac{1}{2}} \cdot \nabla)\mathbf{u}^{n+\frac{1}{2}} = \mathbf{f}^{n+\frac{1}{2}}, \tag{17}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}_*^{n+1}}{\Delta t} - Re^{-1}\triangle\mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{g}^{n+1}, \tag{18}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \tag{19}$$

It is straightforward to derive the variational form of the generalized Stokes system (18)-(19): Find $\mathbf{u}^{n+1} \in H^1(\Omega)^d$ and $p \in L_0^2(\Omega)$ such that $\mathbf{u}^{n+1} = \mathbf{u}_b^{n+1}$ on Γ and solve

$$(\Delta t)^{-1}(\mathbf{u}^{n+1}, \mathbf{v}) + Re^{-1}(\nabla\mathbf{u}^{n+1}, \nabla\mathbf{v}) - (p^{n+1}, \nabla \cdot \mathbf{v}) = (\Delta t)^{-1}(\mathbf{u}_*^{n+1}, \mathbf{v}) + (\mathbf{g}^{n+1}, \mathbf{v}) \tag{20}$$

$$(\nabla \cdot \mathbf{u}^{n+1}, q) = 0 \tag{21}$$

for any $\mathbf{v} \in H_0^1(\Omega)^d$ and $q \in L_0^2(\Omega)$.

Next we will do the same as we did in section 2.1 to propose an explicit scheme for solving the convection system (17). To do so, we first handle the convection term involving $\mathbf{u}^{n+\frac{1}{2}}$. In fact, combining the Taylor's expansion

$$\mathbf{u}^{n+\frac{1}{2}} \approx \mathbf{u}(\mathbf{x}, t_n + \frac{\Delta t}{2}) = \mathbf{u}(\mathbf{x}, t_n) + \frac{\Delta t}{2}\mathbf{u}_t(\mathbf{x}, t_n) + O(\Delta t^2),$$

and the pure convection equation

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{f}, \tag{22}$$

we can obtain a similar approximation to (9) but in a vector-valued form:

$$\mathbf{u}^{n+\frac{1}{2}} \approx \mathbf{u}^n + \frac{\Delta t}{2}\left(\mathbf{f}^n - (\mathbf{u}^n \cdot \nabla)\mathbf{u}^n\right) =: \eta^n. \tag{23}$$

We introduce the inflow boundary

$$\Gamma_{n+1}^- = \left\{\mathbf{x} \in \Omega; \mathbf{u}_b^{n+1} \cdot \mathbf{n}(\mathbf{x}) < 0\right\}.$$

Then we can write by using integration by parts for any $\mathbf{v} \in \mathbf{H}^1(\Omega)$ with $\mathbf{v}|_{\Gamma_{n+1}^-} = 0$ that

$$((\eta^n \cdot \nabla)\eta^n, \mathbf{v}) = \langle\eta^n, \eta^n \cdot \mathbf{n}\,\mathbf{v}\rangle_{\Gamma\backslash\Gamma_{n+1}^-} - (\eta^n, \nabla \cdot \eta^n\mathbf{v}) - (\eta^n, (\eta^n \cdot \nabla)\mathbf{v}), \tag{24}$$

using this relation and plugging (23) in (17) we derive the variational form of (17):

$$\begin{aligned}(\mathbf{u}_*^{n+1}, \mathbf{v}) = (\mathbf{u}^n, \mathbf{v}) + \Delta t(\mathbf{f}^{n+\frac{1}{2}}, \mathbf{v}) \\ - \Delta t\,\langle\eta^n, \eta^n \cdot \mathbf{n}\,\mathbf{v}\rangle_{\Gamma\backslash\Gamma_{n+1}^-} + \Delta t\,(\eta^n, \nabla \cdot \eta^n\mathbf{v}) + \Delta t\,(\eta^n, (\eta^n \cdot \nabla)\mathbf{v}).\end{aligned} \tag{25}$$

Remark 3.1. *We observe from the formulation (25) that $\nabla \cdot \mathbf{f}^n$ is needed in the term $\nabla \cdot \eta^n$, hence it adds some extra regularity on the source component $\mathbf{f}$. This suggests us to better choose $\mathbf{f} \equiv \mathbf{0}$ in the decomposition $\mathbf{F} = \mathbf{f} + \mathbf{g}$ for the Navier-Stokes equations so that the new scheme does not need the evaluation of $\nabla \cdot \mathbf{f}^n$. For the unification of the numerical schemes for both the convection diffusion equation and Navier-Stokes equations, we shall always select $f \equiv 0$ in Algorithms 1 and 2 from now on for the convection diffusion equation.*

Let $\mathbf{V}_h$ and M_h be two finite element spaces approximating the Sobolev space $\mathbf{H}^1(\Omega)$ and $L_0^2(\Omega)$ respectively and satisfy the standard *inf-sup* condition, and $\mathbf{I}_h$ be a interpolation operator of $\mathbf{H}^1(\Omega)$ into $\mathbf{V}_h$. By virtue of the variational formulations (25) and (20), we propose the following single-step scheme for solving the Navier-Stokes equations (3).

Algorithm 3 (Single-step scheme).

Step 0. Compute the initial value $\mathbf{u}_h^0 = \mathbf{I}_h \mathbf{u}_0$. For each $n = 0, 1, \cdots, N-1$, do the following.

Step 1. Compute $\eta_h^n = \mathbf{u}_h^n - \frac{\Delta t}{2}(\mathbf{u}_h^n \cdot \nabla)\mathbf{u}_h^n$;

Find $\mathbf{u}_{h,*}^{n+1} \in \mathbf{V}_h$ such that $\mathbf{u}_{h,*}^{n+1} = \mathbf{I}_h \mathbf{u}_b^{n+1}$ on Γ_{n+1}^- and solves

$$(\mathbf{u}_{h,*}^{n+1}, \mathbf{v}) = (\mathbf{u}_h^n, \mathbf{v}_h) - \Delta t \, \langle \eta_h^n, \eta_h^n \cdot \mathbf{n} \, \mathbf{v}_h \rangle_{\Gamma \backslash \Gamma_{n+1}^-}$$
$$+ \Delta t \, (\eta_h^n, \nabla \cdot \eta_h^n \, \mathbf{v}_h) + \Delta t \, (\eta_h^n, (\eta_h^n \cdot \nabla)\mathbf{v}_h) \quad \forall \, \mathbf{v}_h \in \mathbf{V}_h \cap \mathbf{H}^1_{\Gamma_{n+1}^-}(\Omega).$$

Step 2. Find $\mathbf{u}_h^{n+1} \in \mathbf{V}_h$ and $p_h \in M_h$, such that $\mathbf{u}_h^{n+1} = \mathbf{I}_h \mathbf{u}_b^{n+1}$ on Γ and solves

$$(\Delta t)^{-1}(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + Re^{-1}(\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) = (\Delta t)^{-1}(\mathbf{u}_{h,*}^{n+1}, \mathbf{v}_h) + (\mathbf{g}^{n+1}, \mathbf{v}_h),$$
$$(\nabla \cdot \mathbf{u}_h^{n+1}, q_h) = 0,$$

for any $\mathbf{v}_h \in \mathbf{V}_h \cap \mathbf{H}^1_0(\Omega)$ and $q_h \in M_h$.

We observe from Algorithm 3 that the nonlinear convection term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ in Navier-Stokes equations has been treated explicitly in the time marching process, which may severely restrict the time step size in order to ensure the stability of the scheme. To improve the stability, we may apply Step 1 several times with a smaller time step size during one diffusion correction (Step 2). For this purpose we write the result of Step 1 formally as

$$\mathbf{u}_{h,*}^{n+1} = \mathbf{F}_{conv}^{NS}\left(\Delta t, \mathbf{u}_h^n, \mathbf{u}_b^{n+1}\right). \tag{26}$$

Then a multistep variant of this scheme is to execute this Step 1 m times with a smaller time step size $\frac{\Delta t}{m}$ to derive $u_{h,*}^{n+1}$:

$$\mathbf{u}_{h,*}^{n+\frac{i}{m}} = \mathbf{F}_{conv}^{NS}\left(\frac{\Delta t}{m}, \mathbf{u}_{h,*}^{n+\frac{i-1}{m}}, \mathbf{u}_b^{n+\frac{i}{m}}\right) \tag{27}$$

for $i = 1, 2, \cdots, m$, with $\mathbf{u}_{h,*}^n = \mathbf{u}_h^n$. This leads to the following multistep scheme for the Navier-Stokes equations.

Algorithm 4 (Multistep scheme with index m).

Step 0. Compute the initial value $\mathbf{u}_h^0 = \mathbf{I}_h \mathbf{u}_0$. For each $n = 0, 1, \cdots, N-1$, do the following.

Step 1. Set $\mathbf{u}_{h,*}^n = \mathbf{u}_h^n$; For $i = 1, 2, \cdots, m$, calculate

$$\eta_{h,*}^{n+\frac{i-1}{m}} = \mathbf{u}_{h,*}^{n+\frac{i-1}{m}} - \frac{\Delta t}{2}(\mathbf{u}_{h,*}^{n+\frac{i-1}{m}} \cdot \nabla)\mathbf{u}_{h,*}^{n+\frac{i-1}{m}},$$

and compute $\mathbf{u}_{h,*}^{n+\frac{i}{m}} \in \mathbf{V}_h$ such that $\mathbf{u}_{h,*}^{n+\frac{i}{m}} = \mathbf{I}_h \mathbf{u}_b^{n+\frac{i}{m}}$ on $\Gamma_{n+i/m}^-$ and solves

$$(\mathbf{u}_{h,*}^{n+\frac{i}{m}}, \mathbf{v}) = (\mathbf{u}_{h,*}^{n+\frac{i-1}{m}}, \mathbf{v}_h) - \delta t \left\langle \eta_{h,*}^{n+\frac{i-1}{m}}, \eta_{h,*}^{n+\frac{i-1}{m}} \cdot \mathbf{n}\, \mathbf{v}_h \right\rangle_{\Gamma \backslash \Gamma_{n+i/m}^-}$$
$$+ \delta t \left(\eta_{h,*}^{n+\frac{i-1}{m}}, \nabla \cdot \eta_{h,*}^{n+\frac{i-1}{m}} \mathbf{v}_h \right) + \delta t \left(\eta_{h,*}^{n+\frac{i-1}{m}}, (\eta_{h,*}^{n+\frac{i-1}{m}} \cdot \nabla) \mathbf{v}_h \right).$$

Step 2. Compute $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in \mathbf{V}_h \times M_h$ such that $\mathbf{u}_h^{n+1} = \mathbf{I}_h \mathbf{u}_b^{n+1}$ on Γ and solves

$$(\Delta t)^{-1}(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + Re^{-1}(\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) = (\Delta t)^{-1}(\mathbf{u}_{h,*}^{n+1}, \mathbf{v}_h) + (\mathbf{g}^{n+1}, \mathbf{v}_h),$$
$$(\nabla \cdot \mathbf{u}_h^{n+1}, q_h) = 0$$

for any $(\mathbf{v}_h, q_h) \in (\mathbf{V}_h \cap \mathbf{H}_0^1(\Omega)) \times M_h$.

Remark 3.2. *The second steps in Algorithms 3 and 4 can be replaced by the projection-type methods so that the pair of finite element spaces for approximating the velocity and pressure does not need to meet the LBB condition and only Poisson problems are needed to solve for updating both the velocity and pressure. For the projection method, we refer to the pioneering work by Chorin [23] and Temam [24].*

Remark 3.3. *The newly proposed Algorithms 1-4 are only first-order accurate in time. In order to work out numerical schemes that are of similar nature to Algorithms 1-4 but with second-order accuracy in time, we should combine our current derivations of the first-order algorithms with the existing second-order time-marching schemes [25] [26] for the diffusion steps.*

4   Numerical experiments

In this section we shall carry out two sets of numerical tests to check the actual convergence orders of the single-step and multistep schemes proposed in the previous two sections and how the multistep scheme improves the stability region of the single-step scheme.

Let $\mathcal{T}_h$ be a regular triangulation of Ω, with $h_K = \mathrm{diam}(K)$ for $K \in \mathcal{T}_h$, and $h = \max_{K \in \mathcal{T}_h} h_K$. We shall use the following linear finite element space $V_h \subset H^1(\Omega)$:

$$V_h = \{w_h \in H^1(\Omega);\ w_h|_K \in P_1(K) \quad \forall K \in \mathcal{T}_h\} \tag{28}$$

for the solution of the convection diffusion equation (1), and the following Taylor-Hood finite element spaces [27]

$$\begin{cases} \mathbf{V}_h = \{\mathbf{v}_h \in H^1(\Omega)^2;\ \mathbf{v}_h|_K \in P_2(K)^2 \quad \forall K \in \mathcal{T}_h\}, \\ M_h = \{q_h \in H^1(\Omega);\ q_h|_K \in P_1(K) \quad \forall K \in \mathcal{T}_h\} \end{cases} \tag{29}$$

for the solution of the Navier-Stokes Equations (3).

We recall that we have used the central finite difference scheme for the convection diffusion equation and the backward Euler scheme for the diffusion equation in time discretization. Therefore it is natural for us to expect the following numerical convergence orders when the finite element spaces in (28) and (29) are used:

$$\|u^N - u_h^N\|_{L^2(\Omega)} \leq C(h^2 + \Delta t)$$

for the convection diffusion equation (1), and

$$\|\mathbf{u}^N - \mathbf{u}_h^N\|_{L^2(\Omega)} \le C(h^3 + \Delta t) \quad \text{and} \quad \|p^N - p_h^N\|_{L^2(\Omega)} \le C(h^2 + \Delta t)$$

respectively for the velocity and pressure of the Navier-Stokes equations (3).

We remark that all the errors shown in this section are the L^2-norm errors at the terminal time $t = T$ unless specified otherwise.

4.1 Tests for the convection diffusion equation

We first apply the new single-step and multistep schemes to the following two examples which are taken from references [8] and [16].

Example 1. *The coefficients and domain in equation (1) are taken to be the following:*

$$d = 2, \quad T = 1, \quad \varepsilon = 10^{-8}, \quad \mathbf{b} = (1, -1)^T, \quad c = 1, \quad \Omega = (0, 1)^2$$

with the exact solution given by $u(x, y, t) = e^{2\pi t} \sin(2\pi x) \sin(2\pi y)$.

This example is a slight modification of the one in [8], where $e^{\sin(2\pi t)}$ is used. Instead we use $e^{2\pi t}$, which makes the solution vary in a much larger range, namely in the interval $[-e^{2\pi}, e^{2\pi}]$, and have a much larger norm i.e., $\|u(\cdot, 1)\| = \frac{1}{2}e^{2\pi} \approx 267.7458$.

Example 2. *The coefficients and domain in equation (1) are taken to be the following:*

$$d = 2, \quad T = 1, \quad \varepsilon = 10^{-8}, \quad \mathbf{b} = (2, -1)^T, \quad c = 1, \quad \Omega = (0, 1)^2$$

with the exact solution given by $u(x, y, t) = t^2 \cos(xy^2)$.

To compute the actual convergence orders of the numerical schemes, we shall use the uniform triangulations of domain Ω with triangular elements in all our numerical simulations.

4.1.1 Convergence Tests for the single-step scheme

In order to find the actual convergence order of the single-step scheme (Algorithm 1) in time, we choose a very small mesh size and then observe the changes of the errors when the time step size is halved. Similarly we will do the other way around when we try to find the actual convergence order of the single-step scheme (Algorithm 1) in space.

Tables 1 and 2 show the L^2-norm errors with different mesh sizes when the time step size is fixed for Examples 1 and 2 respectively. Clearly we see the second order spatial convergence of the single-step scheme (Algorithm 1).

Now we fix the uniform mesh size at $h = 1/128$, and run the single-step scheme (Algorithm 1) for Examples 1 and 2 with the following sequence of time step sizes

$$\Delta t = 0.1/2^k, \quad k = -1, 0, 1, 2, \cdots \tag{30}$$

to find out the stability region of the numerical scheme. The numerical results are listed in Tables 3 and 4, from which we observe that Algorithm 1 does not converge till $k = 6$ and 7 respectively for Examples 1 and 2, corresponding to two rather small time step sizes of $\Delta t = 1/640$ and $1/1280$. Such restrictions on time step size are natural, required by the stability condition for the explicit time marching scheme we have used. As we shall see in the next subsection, the new multistep scheme can essentially improve the stability condition.

Table 1: Convergence results of Algorithm 1 for Example 1 with fixed $\Delta t = 1/2^{16}$

h	$\|u - u_h\|$	order
1/4	9.18526(+1)	-
1/8	1.84780(+1)	2.3135
1/16	4.24054	2.1235
1/32	1.03466	2.0351
1/64	2.54797(-1)	2.0217
1/128	6.36669(-2)	2.0007

Table 2: Convergence results of Algorithm 1 for Example 2 with fixed $\Delta t = 1/2^{16}$

h	$\|u - u_h\|$	order
1/4	9.76826(-3)	-
1/8	2.41756(-3)	2.0145
1/16	6.02478(-4)	2.0046
1/32	1.49729(-4)	2.0086
1/64	3.69132(-5)	2.0201
1/128	8.70186(-6)	1.9687

Table 3: Convergence results of Algorithm 1 for Example 1 with fixed $h = 1/128$

Δt	$\|u - u_h\|$	order
$0.1/2^5$	divergence	-
$0.1/2^6$	2.02151	-
$0.1/2^7$	1.01181	0.9985

Table 4: Convergence results of Algorithm 1 for Example 2 with fixed $h = 1/128$

Δt	$\|u - u_h\|$	order
$0.1/2^6$	divergence	-
$0.1/2^7$	1.71484(-4)	-

4.1.2 Stability improvement by the multistep scheme

We can observe from the previous subsection that the single-step scheme (Algorithm 1) may provide the expected convergence and preserve the accurate convergence orders when it converges. However, this scheme requires sufficiently small time step size as shown in Tables 3 and 4, hence may restrict its applications in practice. The multistep scheme (Algorithm 2) is proposed to improve the stability of the single-step scheme. This section is to test how the multistep scheme can improve the stability region.

We note that Δt is the global time step size, which is used for the diffusion correction. As we are interested mainly in the convection-dominated diffusion problems, the time step size required for the convection is usually much smaller than the one for the diffusion.

In our numerical tests, for each fixed $\Delta t = 0.1/2^k$ ($k = -1, 0, 1, 2, \cdots$), we run the multistep scheme with index $m = 1, 2^1, 2^2, \cdots$ until we observe the convergence of the scheme, and then record the corresponding index m; see Tables 5 and 6 for the recorded index m corresponding to each fixed Δt and the resulting relative L^2-norm error of the approximate solution.

As we see from Table 5, when we take $\Delta t = 0.1$, which is too large for the stability of the explicit scheme involved in the convection step, but we can still achieve the convergence of the multistep scheme with index $m \geq 64$. Tables 5 and 6 have demonstrated that though the single-step scheme does not converge for a fixed Δt, the multistep scheme always converges when the index m is appropriately large. So we can conclude that if we take an appropriately large index m, say $m = 30$, the multistep scheme can be viewed as an unconditionally stable scheme.

Furthermore, we have also computed the convergence orders of the multistep scheme in terms of the global time step size for Examples 1 and 2 with a fixed index m and mesh size h. The results are shown in Tables 7 and 8. Combining these results with the ones for the single-step scheme (cf. Table 3), we can clearly observe the first order temporal convergence for both examples.

Table 5: Stability of Algorithm 2 for Example 1 with index m and fixed $h = 1/128$

Δt	m	$\frac{\|u-u_h\|}{\|u\|}$
$0.1/2^6$	1	7.55011(-3)
$0.1/2^5$	2	1.54379(-2)
$0.1/2^4$	4	3.15569(-2)
$0.1/2^3$	8	6.41671(-2)
$0.1/2^2$	16	1.30693(-1)
$0.1/2^1$	32	2.69788(-1)
0.1	64	5.68483(-1)
0.2	128	1.26837

Next, we carry out some numerical tests to check how the multistep scheme can improve the stability region quantitatively. For each fixed mesh size h, we increase the index m gradually and record the largest global time step size Δt that can ensure the convergence of the entire algorithm. And the largest time step size will be written as the critical time step size Δt_{crit} for the stability of the algorithm. The results are shown in Tables 9 and 10, from which we can see that the stability region is nearly doubled when the index m of the multistep scheme is

Table 6: Stability of Algorithm 2 for Example 2 with index m and fixed $h = 1/128$

Δt	m	$\frac{\|u-u_h\|}{\|u\|}$
$0.1/2^7$	1	1.65924(-4)
$0.1/2^6$	2	7.46950(-4)
$0.1/2^5$	4	1.96066(-3)
$0.1/2^4$	8	4.39337(-3)
$0.1/2^3$	16	9.28328(-3)
$0.1/2^2$	32	1.96240(-2)
$0.1/2^1$	64	4.11452(-2)
0.1	128	8.41251(-2)
0.2	256	1.70957(-1)

Table 7: Convergence order of Algorithm 2 for Example 1 with fixed index $m = 64$ and $h = \frac{1}{128}$

Δt	$\|u - u_h\|$	order
0.1	1.52209(+2)	-
$0.1/2^1$	7.23099(+1)	1.0738
$0.1/2^2$	3.51127(+1)	1.0422
$0.1/2^3$	1.73718(+1)	1.0152
$0.1/2^4$	8.66679	1.0032
$0.1/2^5$	4.34772	0.9952
$0.1/2^6$	2.18766	0.9909

Table 8: Convergence order of Algorithm 2 for Example 2 with fixed index $m = 128$ and $h = \frac{1}{128}$

Δt	$\|u - u_h\|$	order
0.1	8.69440(-2)	-
$0.1/2^1$	4.27948(-2)	1.0227
$0.1/2^2$	2.07747(-2)	1.0426
$0.1/2^3$	1.00067(-2)	1.0538
$0.1/2^4$	5.01775(-3)	0.9959
$0.1/2^5$	2.54242(-3)	0.9808
$0.1/2^6$	1.31725(-3)	0.9487

doubled. So the multistep scheme can indeed clearly and essentially enlarge the stability of the entire algorithm.

Table 9: Critical global time step size Δt_{crit} of Algorithm 2 for Example 1 in terms of index m

m	1	2	10	20	40	80
$h = 1/64$						
Δt_{crit}	0.0049	0.0093	0.046	0.093	0.18	0.37
$h = 1/128$						
Δt_{crit}	0.0024	0.0045	0.022	0.045	0.091	0.18

Table 10: Critical global time step size Δt_{crit} of Algorithm 2 for Example 2 in terms of index m

m	1	2	10	20	40	80
$h = 1/64$						
Δt_{crit}	0.0032	0.0060	0.029	0.058	0.11	0.23
$h = 1/128$						
Δt_{crit}	0.0015	0.0030	0.014	0.028	0.057	0.11

We remark that we have done many more numerical experiments for Examples 1 and 2, but with the diffusion coefficients ε varying in a wide range, from 10^{-3} to 10^{-15}, and many different convective vectors $\mathbf{b}$, and observed similar convergence and stability behaviors for the single-step and multistep schemes as we have shown above.

4.2 Tests for the Navier-Stokes Equations

Now we will apply our new single-step and multistep schemes (Algorithms 3 and 4) to two examples of Navier-Stokes equations with analytical solutions to check the actual convergence orders of the schemes and how the multistep scheme improves the stability region of the single-step scheme. Then we will apply these schemes to the benchmark problem of the lid-driven cavity flow to verify their validity.

Example 3. *Consider the Navier-Stokes equations (3) with the following parameters:*

$$\Omega = [0,1]^2, \quad T = 1, \quad Re = 5000 \quad and \quad 10000$$

with the exact solution $(\mathbf{u}, p) = (u_1, u_2, p)$ *given by* $p = (x^2 - y^2)\cos(t)$ *and*

$$u_1 = 10x^2(x-1)^2y(y-1)(2y-1)\cos(t), \quad u_2 = -10x(x-1)(2x-1)y^2(y-1)^2\cos(t).$$

Example 4. *Consider the Navier-Stokes equation (3) with the same parameters as in Example 3, but the exact solution* $(\mathbf{u}, p) = (u_1, u_2, p)$ *given by*

$$u_1 = t^3 y^2, \quad u_2 = t^2 x, \quad p = tx + y - (t+1)/2.$$

This is an example where only a discretization error in time occurs [28].

4.2.1 Convergence Tests for the single-step scheme

We first verify the convergence orders of the single-step scheme (Algorithm 3) in both space and time for Example 3. Tables 11-12 present the convergence results in time for the Reynolds numbers $Re = 5000$ and 10000 respectively, with a fixed uniform mesh of size $h = 1/128$, and Tables 13-14 give the convergence results in space for the Reynolds numbers $Re = 5000$ and 10000 respectively, with a fixed $\Delta t = 10^{-6}$. From these tables we can clearly see the optimal first order convergence of the single-step scheme in time and the optimal third and second order convergence in space respectively for the velocity and pressure.

For Example 4, we have tested the single-step scheme (Algorithm 3) with the Reynolds numbers $Re = 5000$ and 10000, and the uniform mesh of size $h = 1/48$ and $1/64$, and the sequence of time step sizes as listed in (30). The results have shown that the scheme converges only when the time step size $\Delta t = 0.1/2^k$ is sufficiently small, namely when k takes at least 4 ($\Delta t = 1/160$) and 5 ($\Delta t = 1/320$) respectively for $h = 1/48$ and $1/64$. This test indicates that the single-step scheme may require sufficiently small time step size to ensure its convergence. In the next section 4.2.2 we will show the multistep scheme (Algorithm 4) can essentially improve the stability of the single-step scheme.

Table 11: Convergence of Algorithm 3 for Example 3 with $h = 1/128$ and $Re = 5000$

Δt	$\|\mathbf{u} - \mathbf{u}_h\|$	order	$\|p - p_h\|$	order
0.2	3.28203(-3)	-	1.00222(-4)	-
0.1	1.65607(-3)	0.9868	4.79084(-5)	1.0648
$0.1/2^1$	8.31889(-4)	0.9933	2.35900(-5)	1.0221
$0.1/2^2$	4.16919(-4)	0.9966	1.20411(-5)	0.9702

Table 12: Convergence of Algorithm 3 for Example 3 with $h = 1/128$ and $Re = 10000$

Δt	$\|\mathbf{u} - \mathbf{u}_h\|$	order	$\|p - p_h\|$	order
0.2	3.29348(−3)	-	9.98106(-5)	-
0.1	1.66141(−3)	0.9872	4.77006(-5)	1.0652
$0.1/2^1$	8.34455(−4)	0.9935	2.34866(-5)	1.0222
$0.1/2^2$	4.18176(−4)	0.9967	1.19910(-5)	0.9699

4.2.2 Stability improvement by the multistep scheme

As shown in the previous subsection, the convergence of the single-step scheme (Algorithm 3) for Example 4 requires a sufficiently small global time step size for a fixed mesh size h.

In order to improve this severe restriction on time step size by the single-step scheme, we now show how we can achieve the convergence for large global time step size by the multistep scheme. For each fixed $\Delta t = 0.1/2^k$ ($k = -1, 0, 1, 2, \cdots$), we run the multistep scheme with index $m = 1, 2^1, 2^2, \cdots$ until we observe the convergence of the scheme, and then record the

Table 13: Convergence of Algorithm 3 for Example 3 with $\Delta t = 10^{-6}$, $Re = 5000$ and $T = 0.2$

h	$\|\mathbf{u} - \mathbf{u}_h\|$	order	$\|p - p_h\|$	order
1/4	1.31468(-3)	-	6.45683(-3)	-
1/8	1.81020(-4)	2.8607	1.61419(-3)	2.000016
1/16	2.38018(-5)	2.9270	4.03547(-4)	2.000002
1/32	3.00134(-6)	2.9874	1.00887(-4)	1.999996
1/48	8.71038(-7)	3.0511	4.48386(-5)	2.000004

Table 14: Convergence of Algorithm 3 for Example 3 with $\Delta t = 10^{-6}$, $Re = 10000$ and $T = 0.2$

h	$\|\mathbf{u} - \mathbf{u}_h\|$	order	$\|p - p_h\|$	order
1/4	1.31577(-3)	-	6.45683(-3)	-
1/8	1.81589(-4)	2.8572	1.61419(-3)	2.000016
1/16	2.42194(-5)	2.9064	4.03547(-4)	2.000002
1/32	3.20061(-6)	2.9197	1.00887(-4)	1.999996
1/48	9.28064(-7)	3.0533	4.48386(-5)	2.000004

corresponding index m; see Tables 15 and 16 for the recorded index m corresponding to each fixed Δt and the resulting relative L^2-norm errors of the approximate solutions for the velocity and pressure.

As we see from Table 15, when we take $\Delta t = 0.1$, which is too large for the stability of the explicit scheme involved in the convection step, but we can still achieve the convergence of the multistep scheme with index $m \geq 32$. Tables 15 and 16 have demonstrated that though the single-step scheme does not converge for a fixed Δt, the multistep scheme always converges when the index m is appropriately large. So we can conclude that if we take an appropriately large index m, say $m = 30$, the multistep scheme can be viewed as an unconditionally stable scheme.

Next we have tested the actual convergence orders of the multistep scheme when the index m is fixed at $m = 64$. Tables 17-18 have showed the computational results for $Re = 5000$ and 10000 with fixed $h = 1/48$ and 1/64 respectively. We can observe clearly the optimal first order convergence for both velocity and pressure in terms of the global time step size.

The last test we have carried out is to check how the multistep scheme can improve the stability region quantitatively. For each fixed mesh size h, we increase the index m gradually and record the largest global time step size Δt (the critical time step size Δt_{crit} as we called earlier) that can ensure the convergence of the entire algorithm. The results are shown in Table 19, from which we can see that the stability region is nearly doubled when the index m of the multistep scheme is doubled. So the multistep scheme can indeed clearly and essentially enlarge the stability of the entire algorithm.

We end this subsection with some concluding remarks on convergence and stability behaviors of the single-step and multistep schemes, based on our observations from the numerical tests in this and previous subsections.

- The single-step scheme (Algorithm 3) is generally conditionally stable, and requires sufficiently small time step size to ensure its convergence with a fixed mesh and larger Reynolds number.

- The multistep scheme (Algorithm 4) can essentially relax the restriction of the time step size (see Tables 15, 16 and 19).

- Comparing the results in Tables 15-16 with the ones in Tables 17-18, we can clearly see the stability and robustness of the multistep scheme (Algorithm 4). For example, for the global time step size $\Delta t = 0.1/2^4$, the multistep scheme with a small index like $m = 2$ and a large index like $m = 64$ provide about the same accuracies; see Tables 16 and 18.

Table 15: Stability of Algorithm 4 for Example 4 with index m and fixed $h = 1/48$, $Re = 5000$

Δt	m	$\|\mathbf{u} - \mathbf{u}_h\|$	$\|p - p_h\|$
$0.1/2^4$	1	1.61352(-3)	9.22561(-3)
$0.1/2^3$	4	3.15065(-3)	1.82083(-2)
$0.1/2^2$	8	6.32378(-3)	3.52747(-2)
$0.1/2^1$	16	1.34098(-2)	6.64021(-2)
0.1	32	2.70632(-2)	1.18921(-1)
0.2	64	5.60465(-2)	1.97385(-1)

Table 16: Stability of Algorithm 4 for Example 4 with index m and fixed $h = 1/64$, $Re = 10000$

Δt	m	$\|\mathbf{u} - \mathbf{u}_h\|$	$\|p - p_h\|$
$0.1/2^5$	1	8.13177(-4)	4.65014(-3)
$0.1/2^4$	2	1.61134(-3)	9.24510(-3)
$0.1/2^3$	4	3.46105(-3)	1.82018(-2)
$0.1/2^2$	16	6.37609(-3)	3.52913(-2)
$0.1/2^1$	32	1.29496(-2)	6.64033(-2)
0.1	64	2.67569(-2)	1.18926(-1)
0.2	128	5.67472(-2)	1.97454(-1)

4.2.3 The lid-driven cavity flow

As our final numerical example we test a popular benchmark problem, i.e., the lid-driven cavity flow problem, where the fluid is enclosed in a unit square box, with an imposed velocity of unity in the horizontal direction on the top boundary, and a no-slip condition on the remaining walls. We shall compare our results with three benchmark results: Ghia et al. [29] with $h = 1/128$ for Reynolds numbers $Re = 100, 400, 1000$ and 3200; Erturk et al. [30] with $h = 1/128$ for Reynolds number $Re = 1000$; Botella et al. [31] for the Reynolds number $Re = 1000$.

Table 17: Convergence order of Algorithm 4 for Example 4 with fixed $h = 1/48$, $Re = 5000$ and fixed index $m = 64$

Δt	$\|\mathbf{u} - \mathbf{u}_h\|$	order	$\|p - p_h\|$	order
0.2	5.60465(-2)	-	1.97385(-1)	-
0.1	2.64298(-2)	1.0845	1.18888(-1)	0.7314
$0.1/2^1$	1.28200(-2)	1.0438	6.63946(-2)	0.8405
$0.1/2^2$	6.33073(-3)	1.0180	3.52954(-2)	0.9116
$0.1/2^3$	3.17694(-3)	0.9947	1.82354(-2)	0.9527
$0.1/2^4$	1.60997(-3)	0.9806	9.27332(-3)	0.9756
$0.1/2^5$	8.20838(-4)	0.9719	4.67745(-3)	0.9874

Table 18: Convergence order of Algorithm 4 for Example 4 with fixed $h = 1/64$, $Re = 10000$ and fixed index $m = 64$

Δt	$\|\mathbf{u} - \mathbf{u}_h\|$	order	$\|p - p_h\|$	order
0.1	2.67569(-2)	-	1.18926(-1)	-
$0.1/2^1$	1.29252(-2)	1.0497	6.64054(-2)	0.8407
$0.1/2^2$	6.35955(-3)	1.0232	3.52978(-2)	0.9117
$0.1/2^3$	3.17773(-3)	1.0009	1.82330(-2)	0.9530
$0.1/2^4$	1.60477(-3)	0.9856	9.27095(-3)	0.9758
$0.1/2^5$	8.16528(-4)	0.9748	4.67451(-3)	0.9879

Table 19: Critical global time step size Δt_{crit} of Algorithm 4 for Example 4 in terms of index m

m	1	5	10	20	40	80
$Re = 10000, h = 1/64$						
Δt_{crit}	0.0039	0.018	0.024	0.048	0.089	0.18

In all our computations for this example, we use the uniform mesh of size $h = 1/128$ and the Taylor-Hood element (29), and have tested the cases with Reynolds numbers $Re = 100, 400, 1000$ and 3200, and the global time step size $\Delta t = 0.005$. The stoping condition for time advancing, which is considered as the criterion of capturing the steady state solution, is chosen as

$$\frac{\|\mathbf{u}_h^{n+1} - \mathbf{u}_h^n\|}{\|\mathbf{u}_h^{n+1}\|} \leq 10^{-5},$$

where $\mathbf{u}_h^n$ is the finite element solution at time $t = t_n$. We have observed from our numerical results that the single-step scheme (Algorithm 3) works when the Reynolds number is relatively small, e.g., $Re = 100, 400$ and 1000, but it is unstable when Re is large, e.g., $Re \geq 3200$. But the multistep scheme may still work for larger Reynolds number, e.g., $Re = 3200$.

Tables 20-21 present the streamfunction values and the locations of the primary and secondary vortices for various Reynolds numbers. Figures 1, 2 and 3 show the computed velocity components and vorticity profiles along the horizonal and vertical lines compared with the results of Ghia et al. [29] and Botella et al. [31]. As one can see that the results by the new schemes confirm very well the ones by the benchmark schemes.

Table 20: Streamfunction values Ψ_{min}, Ψ_{max} and locations of the primary and secondary vortices

Vortex	property	Re=1000 Single-step scheme	Re=1000 Ghia et al. [29]	Re=1000 Erturk et al. [30]
Primary	Ψ_{min}	-0.114722	-0.117929	-0.118781
	Location (x, y)	(0.5313, 0.5625)	(0.5313, 0.5625)	(0.5300, 0.5650)
First BL	Ψ_{max}	2.12504E-4	2.31129E-4	2.3261E-4
	Location (x, y)	(0.0781, 0.0781)	(0.0859, 0.0781)	(0.0833, 0.0783)
First BR	Ψ_{max}	1.67313E-3	1.75102E-3	1.7281E-3
	Location (x, y)	(0.8672, 0.1094)	(0.8594,0.1094)	(0.8633, 0.1117)
Second BR	Ψ_{min}	-4.815059E-8	-9.31929E-8	5.4962E-8
	Location (x, y)	(0.9922, 0.0078)	(0.9922, 0.0078)	(0.9917, 0.0067)

5   Concluding remarks

We have proposed a new splitting method for solving the general time-dependent convection-dominated diffusion problem and the Navier-Stokes equations. A pure convection problem and a pure diffusion problem are solved successively at each iteration of the method. Explicit schemes are proposed for the time discretization of the convective problem. The explicitness of the scheme may cause a severe restriction on the time step sizes, which can be essentially improved by an explicit multistep scheme with smaller time step sizes so that the resulting method behaves like an unconditionally stable method. The diffusion problem involved at each iteration is always self-adjoint and coercive so that it can be solved efficiently using many existing optimal preconditioned iterative solvers. The optimal convergence orders have been confirmed by several numerical examples with smooth solutions. The schemes are then extended for the Navier-Stokes equations, where the nonlinearity is resolved by a linear explicit multistep scheme at

Table 21: Streamfunction values Ψ_{min}, Ψ_{max} and locations of the primary and secondary vortices

Number	property	Re=3200 Multistep scheme with index $m = 2$	Re=3200 Ghia et al. [29]
Primary	Ψ_{min}	-0.109962	-0.120377
	Location, x, y	(0.5156,0.5391)	(0.5165,0.5469)
First T	Ψ_{max}	5.759079E-4	7.27682E-4
	Location (x, y)	(0.0469,0.8984)	(0.0547,0.8984)
First BL	Ψ_{max}	1.09512E-3	9.7823E-4
	Location (x, y)	(0.0781,0.1250)	(0.0859,0.1094)
First BR	Ψ_{max}	2.70425E-3	3.13955E-3
	Location (x, y)	(0.8281,0.0859)	(0.8125,0.0859)
Second BL	Ψ_{min}	-1.04040E-8	-6.33001E-8
	Location (x, y)	(0.0078,0.0078)	(0.0078,0.0078)
Second BR	Ψ_{min}	-1.36461E-7	-2.51648E-7
	Location (x, y)	(0.9844,0.0078)	(0.9844,0.0078)

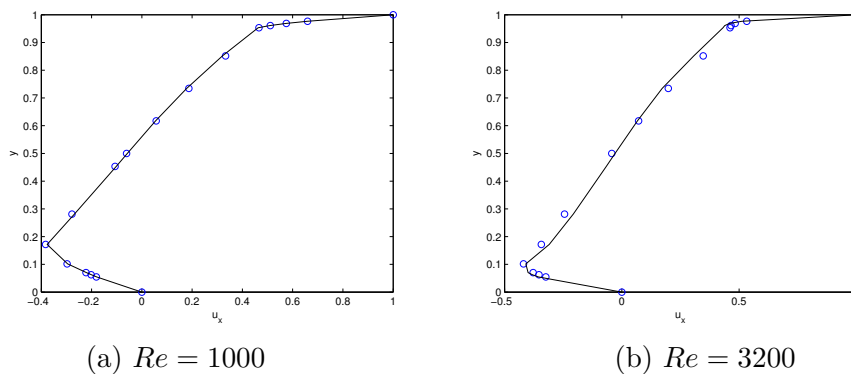(a) $Re = 1000$ (b) $Re = 3200$

Figure 1: Velocity (u_x) profiles along the vertical line passing through the geometric center of the cavity. Black solid lines: (a) single-step scheme, (b) multistep scheme with index $m = 2$; Blue circle lines: Ghia et al. [29]
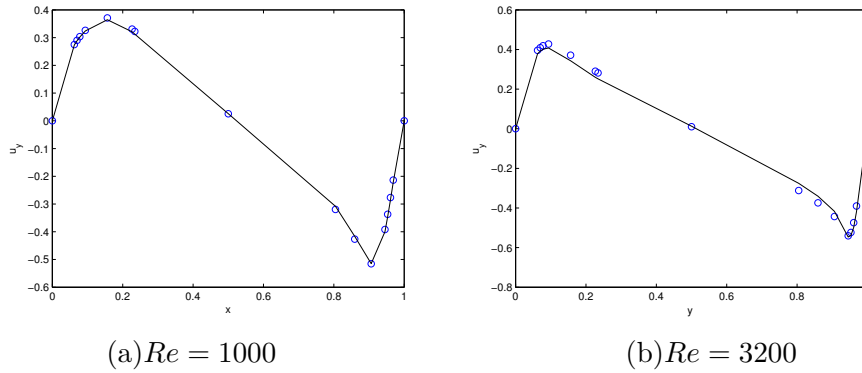
(a)$Re = 1000$ (b)$Re = 3200$

Figure 2: Velocity (u_y) profiles along the horizontal line passing through the geometric center of the cavity. Black solid lines: (a) single-step scheme, (b) multistep scheme with index $m = 2$; Blue circle lines: Ghia et al. [29]
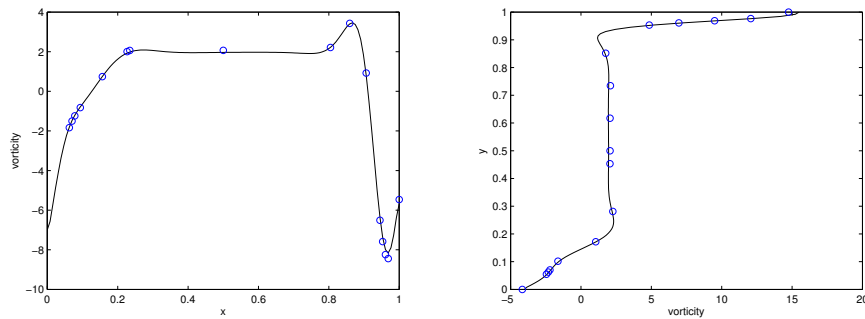

Figure 3: Vorticity values along the vertical line $x = 0.5$ (left) and the horizontal line $y = 0.5$ (right) passing through the geometric center of the cavity with $Re = 1000$. Black solid lines: single-step scheme; Blue circle lines: Botella et al. [31]

the convection step, while only a generalized Stokes problem is needed to solve at the diffusion step and the major stiffness matrix stays invariant in the time marching process. Numerical simulations are presented to demonstrate the stability, convergence and performance of the single-step and multistep variants of the new schemes. The effectiveness and robustness of the new schemes are finally well demonstrated by the benchmark lid-driven cavity flow problem.

References

[1] A. QUARTERONI AND A. VALLI, *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, Berlin, 1994.

[2] J. DONEA AND A. HUERTA, *Finite Element Methods for Flow Problems*, Wiley, New York, 2003.

[3] R. GLOWINSKI AND P. LE TALLEC, *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, 1989.

[4] T.J.R. Hughes and A.N. Brooks, *A multidimensional upwind scheme with no crosswind diffusion*, In T.J.R. Hughes (ed.) Finite Element Methods for Convection Dominated Flows (ASME, New York, 1979) 19-35.

[5] A.N. Brooks and T.J.R. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199-259.

[6] T.J.R. Hughes, L. P. Franca and G.M. Hulbert, *A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations*, Comput. Methods Appl. Mech. Engrg. 73 (1989) 173-189.

[7] M. Stynes, *Steady-state convection-diffusion problems*, Acta Numer. 14 (2005) 445-508.

[8] V. John and J. Novo, *Error Analysis of the SUPG Finite Element Discretization of Evolutionary Convection-Diffusion-Reaction Equations*, SIAM J. Numer. Anal. 49 (2011) 1149-1176.

[9] L.P. Franca, S.L. Frey and T.J.R. Hughes, *Stabilized finite element methods: I. Application to the advective-diffusive model*, Comput. Methods Appl. Mech. Engrg. 96 (1992) 253-276.

[10] L.P. Franca and S.L. Frey, *Stabilized finite element methods: II. The incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg. 99 (1992) 209-233.

[11] L.P. Franca and C. Farhatb, *Bubble functions prompt unusual stabilized finite element methods*, Comput. Methods Appl. Mech. Engrg. 123 (1995) 299-308.

[12] L.P. Franca and F. Valentin, *On an improved unusual stabilized finite element method for the advective-reactive-diffusive equation*, Comput. Methods Appl. Mech. Engrg. 190 (2000) 1785-1800.

[13] T.J.R. Hughes, *Multiscale phenomena: Greens functions, the Dirichlet-to-Neumann formulation, subgrid-scale models, bubbles and the origin of stabilized methods*, Comput. Methods Appl. Mech. Engrg. 127 (1992) 387-401.

[14] T.J.R. Hughes, G.R. Feijóo, L. Mazzei and J.-B. Quincy, *The variational multiscale methoda paradigm for computational mechanics*, Comput. Methods Appl. Mech. Engrg. 166 (1998) 3-24.

[15] T.J.R. Hughes, L. Mazzei and K.E. Jensen, *The large eddy simulation and the variational multiscale method*, Comput. Vis. Sci. 3 (2000) 47-59.

[16] V. John, S. Kaya and W. Layton, *A two-level variational multiscale method for convection-dominated convection-diffusion equations*, Comput. Methods Appl. Mech. Engrg. 195 (2006) 4594-4603.

[17] C.M. Chen and V. Thomée, *The lumped mass finite element method for a parabolic problem*, J. Austral. Math. Soc. Ser. B 26 (1985) 329-354.

[18] O.C. Zienkiewicz and R. Codina, *Search for a general fluid mechanics algorithm*, In: D.A. Caughey, M.M.Hafez (eds.) Frontiers of Computational Fluid Dynamics (Wiley, New York, 1995) 101-113.

[19] O.C. Zienkiewicz and R. Codina, *A general algorithm for compressible and incompressible flowpart I: The split, characteristic-based scheme*, Int. J. Numer. Meth. Fluids 20 (1995) 869-885.

[20] O.C. Zienkiewicz, P. Nithiarasu, R. Codina, M. Vázquez and P. Ortiz, *The characteristic-based-split procedure: an efficient and accurate algorithm for fluid problems*, Int. J. Numer. Meth. Fluids 31 (1999) 359-392.

[21] P. Nithiarasu, O.C. Zienkiewicz and R. Codina, *The Characteristic-Based Split (CBS) scheme–a unified approach to fluid dynamics*, Int. J. Numer. Meth. Engrg. 66 (2006) 1514-1546.

[22] O.C. Zienkiewicz, R.L. Taylor and P. Nithiarasu, *The finite element method for fluid dynamics* (6th Edition), Elsevier, Amsterdam, 2005.

[23] A.J. Chorin, *Numerical solution of the Navier-Stokes equations*, Math. Comp. 22 (1968) 745-762.

[24] R. Temam, *Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des fractionnarires II*, Arch. Rational Mech. Anal. 33 (1969) 377-385.

[25] Y. He, *Two-level method based on finite element and Crank-Nicolson extrapolation for the time-dependent Navier-Stokes equations*, SIAM J. Numer. Anal. 41 (2003), 1263-1285.

[26] Y. He and W. Sun, *Stability and convergence of the Crank-Nicolson/Adams-Bashforth scheme for the time-dependent Navier-Stokes equations*, SIAM J. Numer. Anal. 45 (2007), 837-869.

[27] C. Taylor and P. Hood,*A numerical solution of the Navier-Stokes equations using the finite element technique*, Computers & Fluids 1 (1973) 73-100.

[28] V. John, G. Matthies and J. Rang, *A comparison of time-discretization/linearization approaches for the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg. 195 (2006) 5995-6010.

[29] U. GHIA, K. N. GHIA AND C. T. SHIN, *High-resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys. 48 (1982) 387-411.

[30] E. ERTURK, T.C. CORKE AND C. GÖKÇÖL, *Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers*, Int. J. Numer. Meth. Fluids 48 (2005) 747-774.

[31] O. BOTELLA AND R. PEYRET, *Benchmark spectral results on the Lid-driven cavity flow*, Computers & Fluids, 27 (1998) 421-433.