On efficient high-order semi-implicit time-stepping schemes for unsteady incompressible Navier–Stokes Equations

K.C. Loy\textsuperscript{a,b,1}, Y. Bourgault\textsuperscript{a,*}

\textsuperscript{a}Department of Mathematics & Statistics, 585 King Edward Avenue, Ottawa, ON, K1N 6N5, CANADA
\textsuperscript{b}School of Informatics \\& Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, MALAYSIA

Abstract

We focus on the development of higher-order semi-implicit time-stepping methods to solve the incompressible Navier–Stokes equations. The Taylor–Hood mixed finite elements method \((P_2-P_1)\) is used for space approximation of velocity and pressure. Semi-implicit time discretizations can provide very accurate approximations for nonstationary flow while treating the nonlinear terms explicitly and avoiding the need for a fancy nonlinear iterative solver. We first introduced three variants of higher-order semi-implicit time-stepping schemes: a multistep Semi-Implicit Backward Difference Formulae (SBDF) which is the easiest to implement; a one-step defect correction (DC) which produce better approximations than SBDF methods in terms of numerical errors on velocity and pressure (in time) for a given order; the one proposed by Guermond and Minev (GM), an uncoupled method which is based upon defect correction and artificial compressibility methods. SBDF and DC methods produce saddle point linear system while GM generates two linear systems with symmetric and positive definite matrix. We propose a modi-

\textsuperscript{*}This work is supported by NSERC Discovery Grant
\textsuperscript{*}Corresponding author
\textit{Email address:} ybourg@uottawa.ca (Y. Bourgault)
\textit{URL:} https://science.uottawa.ca/mathstat/en/people/bourgault-yves (Y. Bourgault)
\textsuperscript{1}Author is a PhD scholarship recipient given by Ministry of Education Malaysia and Universiti Malaysia Terengganu.
fication on GM denoted as GM-SRM methods—which implement sequential regularization method of Navier–Stokes equations. GM-SRM methods produce more rapid convergence on pressure during the start-up while requiring smaller stabilization parameter than that for GM methods. We showcase the numerical accuracy, stability and efficiency of these methods through numerical test cases: two manufactured solutions; the flow past a circular cylinder; and the lid-driven cavity flow, all in 2D settings. All methods show an agreement with the theoretical convergence rate. SBDF methods are the most CPU-efficient, followed by DC, GM and GM-SRM methods. We observe that the presence of $\text{grad-div}$ term in both GM and GM-SRM improves the numerical stability in terms of producing a higher CFL bound. Furthermore, relevant parameter, such as the Strouhal numbers, lift and drag coefficients are found to be very close to published values.

**Keywords:** high-order, time-stepping, semi-implicit, defect correction, Navier–Stokes equations, sequential regularization method

1. **Introduction: Incompressible Navier–Stokes Equations**

The time evolution of incompressible flows can be generally modeled by Navier–Stokes equations, which are expressed as follows

$$
\begin{align*}
\mathbf{u}_t - \nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f} \quad \text{on } \Omega \times (0, T), \\
\nabla \cdot \mathbf{u} &= 0 \quad \text{on } \Omega \times (0, T), \\
\mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0 \quad \text{in } \Omega, \\
\mathbf{u}(\mathbf{x}, t) &= \mathbf{g} \quad \text{on } \Gamma \times (0, T), 
\end{align*}
$$

(1)

where $\mathbf{u}$ is the velocity field, $p$ the pressure, $\nu$ the kinematic viscosity, $\mathbf{f}$ any external force, $\mathbf{u}_0$ the initial velocity and $\mathbf{g}$ the velocity field imposed on the boundary $\Gamma$ of the domain, $\Omega$. The space approximations of Eq.(1) can be done using various numerical techniques, e.g., finite difference, finite element, finite volume and pseudospectral methods. Independently from the space approximations, time-stepping schemes are used to solve the semi-discrete problem (e.g. backward-Euler, trapezoidal- or midpoint rules, Runge-Kutta, etc).

Fully-implicit schemes, such as backward-difference formulae (BDF), are popular time-stepping strategies to solve the semi-discrete Navier–Stokes equation. These fully implicit methods require a fixed point method (e.g.
Newton’s method, Picard iteration) for solving the nonlinear algebraic system resulting from the discretization. Splitting approaches, which include the \( \theta \)-methods (operator-splitting) and projection schemes etc, has an equal success to solve nonstationary Navier–Stokes equations \([11, 22, 27]\). These splitting schemes advocate predictor-corrector steps, i.e., advection-diffusion and Stokes subproblems for fractional time steps; and pressure correction algorithm. However, several of these methods are known to generate non-physical boundary layers that may further induce sub-optimality of the convergence rate known as the order barrier \([22]\). Moreover, a fixed point method is required to solve the nonlinear advection-diffusion sub-step impacting the efficiency of the method. Eq. (1) can also be resolved using semi-Lagrangian approach such as the Characteristic Galerkin method \([18]\). This can be done by discretizing the material derivative along the ‘characteristic curves’.

However, the numerical efficiency and accuracy of this method is limited by the efficiency of nodal search algorithm for the tip of characteristic curves and by the accuracy of numerical quadrature on mismatched elements. All of the above methods are unconditionally stable, making them attractive methods for solving stationary Navier–Stokes problems. At the expense of losing the unconditional stability, the nonlinear advection terms can be taken explicitly with so-called semi-implicit methods, hence nonlinear fixed point iterations can be totally avoided \([13]\).

Semi-implicit approaches make it possible to factorize the ‘constant’ matrix only once which gives a significant reduction in CPU time. Although time steps may be restricted by stability, this is justifiable when smaller temporal change is crucial to study non-stationary flows with higher accuracy. On the other hand, semi-implicit methods are less suitable for computing steady flows where implicit schemes with large time steps can lead to significant reduction in computing cost. An earliest convergence analysis of higher-order semi-implicit schemes for Navier-Stokes equation was done by Baker et al. \([7]\) but no numerical test cases were presented. Kress and Lötstedt \([29]\) provided linear stability analysis for similar semi-implicit schemes solving Navier–Stokes equations using finite difference approach for space approximation. A similar idea was introduced with so-called, ‘IMplicit-EXplicit’ schemes, shortened as IMEX, which were used to solved advection-diffusion or reaction-diffusion problems \([1, 3, 12, 44]\). The 2nd-order Crank–Nicolson Adam–Bashforth schemes (CNAB) is a relatively popular semi-implicit method. This method requires a special treatment to filter parasitic oscillations due to its sensitivity to the initial condition, which is a non-
trivial procedure. Numerical experiment shows that if these oscillations are not properly handled, the theoretical rate of convergence cannot be reproduced. There is work dedicated to make CNAB a more robust scheme but it works only for 2nd-order accuracy in time [33]. In this paper, we propose semi-implicit methods which are based on backward-forward Euler. These methods can be easily extended to arbitrary orders and have a great potential in terms of numerical accuracy and efficiency, particularly for unsteady flows.

The main objective of this paper is to conduct a thorough numerical assessment of time-stepping methods for the nonstationary, incompressible Navier–Stokes equations. Here we focus on higher-order semi-implicit methods for time-stepping (i.e. 2nd- and 3rd-order). We found that the numerical assessment on these methods in the framework of incompressible Navier–Stokes equations are still lacking. This paper is organized as follows: In Section 2, we first present the Semi-Implicit Backward Differentiation Formulae (SBDF), semi-implicit methods using deferred correction (DC), and a variant of this method proposed by Guermond and Minev (GM). All these schemes are 2nd- or 3rd-order in time. We also proposed a further modification of GM using sequential regularization methods (SRM) of Navier–Stokes equations. The combination of GM and SRM will be called the GM-SRM scheme. A short discussion on \text{grad-div} terms which arise naturally in GM and GM-SRM and the choice of the stabilization parameter will be presented in Section 3. Section 4 presents numerical test cases to highlight issues regarding the performance of these time-stepping schemes. These test cases involve nonstationary 2D flows, two of which are manufactured solutions with analytical solutions. Further, more challenging test cases are presented featuring a 2D pulsating flow around the cylinder (von Kármán alley) and a periodic lid-driven cavity flow to investigate the numerical stability and efficiency of semi-implicit methods. For these two flows, the unsteadiness results from a bifurcation, not from the periodic forcing term or periodic boundary conditions. Our numerical results are compared to the results from the literature. In Section 5, we present a short discussion on the strengths and weaknesses of these higher-order methods.

2. Time-stepping schemes

As mentioned earlier, the time-discretization of any PDE can be treated independently from its space approximation. Using the finite element method,
we first write down the semi-discrete problem for the 2D incompressible Navier–Stokes equations. This gives: Find $u_h(t) \in V_h$ and $p(t) \in M_h$ such that $\forall t \in (0, T)$,

$$
\begin{align*}
\int_{\Omega} \frac{d}{dt} u_h(t) \cdot v_h \, dx + \int_{\Omega} \nu \nabla u_h(t) : \nabla v_h \, dx + \\
\int_{\Omega} (u_h(t) \cdot \nabla u_h(t)) \cdot v_h \, dx - \int_{\Omega} p_h(t) \cdot v_h \, dx = \int_{\Omega} f_h(t) \cdot v_h \, dx, \quad \forall v_h \in V_h,
\end{align*}
$$

$$
\begin{align*}
- \int_{\Omega} q_h \nabla \cdot u_h(t) \, dx = 0, \quad \forall q_h \in M_h,
\end{align*}
$$

$$
\begin{align*}
u_{h0} = u_{h0},
\end{align*}
$$

(2)

where $u_{h0}$ is an approximation of $u_0$. We use the Taylor–Hood finite element discretization in space (continuous piecewise quadratic polynomial for velocity and linear polynomial for pressure):

$$
\begin{align*}
V_h = X_h \cap [H^1_0(\Omega)]^2, & \quad X_h = \{v_h \in [C^0(\overline{\Omega})]^2 \mid v_h|_K \in P_2, \forall K \in T_h\}, \\
M_h = Y_h \cap L^2_0(\Omega), & \quad Y_h = \{q_h \in C^0(\overline{\Omega}) \mid q_h|_K \in P_1, \forall K \in T_h\}.
\end{align*}
$$

Here $T_h$ denotes a regular triangulation of the domain $\Omega$ and $P_k$ the space of Lagrange polynomial of degree $k$ on the triangles $K$. The Taylor–Hood element satisfies the well-known inf-sup condition [5, 17]. For finite element discretization of the Stokes and Navier–Stokes equations, the pressure $p$ is taken in $L^2_0(\Omega)$, the space of $L^2(\Omega)$ function with zero-average on $\Omega$, as pressure is unique only up to a constant when the velocity has Dirichlet boundary conditions on all $\partial \Omega$. In practice, the pressure is penalized to control the value of this constant and obtain a nonsingular algebraic system. The solution $(u_\epsilon, p_\epsilon)$ of the penalized system converge to the solution $(u, p)$ of the nonpenalized system in $O(\epsilon)$, $\epsilon > 0$ being the penalization parameter [49].

2.1. High-order Semi–Implicit Backward Difference Formulae (SBDF)

Semi-implicit time-stepping methods of $k$th-order accuracy (in time) for solving Navier–Stokes equations can be written as follows: Given values $\{u^j\}_{j=0}^{k-1}$ at the first $k$-time steps, for all $n \geq 0$ find $(u^{n+k}, p^{n+k})$ solution
of discretization of the time-derivative

\[
\frac{1}{\tau}(\alpha u^{n+k} + \sum_{j=1}^{k} \beta_j u^{n+k-j}) - \nu \Delta u^{n+k} + 
\]

nonlinear advection extrapolated at \( t = t^{n+k} \)

\[
\sum_{j=1}^{k} \gamma_j u^{n+k-j} \cdot \nabla u^{n+k-j} + \nabla p^{n+k} = f^{n+k} \quad (3)
\]

\[
\nabla \cdot u^{n+k} = 0 \quad (4)
\]

where a constant time step \( \tau \) is considered. The set of coefficients \( \alpha, \{\beta_j\}_{j=1}^{k} \in \mathbb{R} \) arises from the backward differentiation formula for the time-derivative with \( \alpha + \sum_{j=1}^{k} \beta_j = 0 \). The coefficients \( \{\gamma_j\}_{j=1}^{k} \in \mathbb{R} \) produce the extrapolation formula for the nonlinear term satisfying \( \sum_{j=1}^{k} \gamma_j = 1 \). The diffusion term \( -\Delta u^{n+k} \) is taken implicitly to avoid stringent stability condition in \( O(\nu^{-1}h^2) \) on the time step. The terms \( \nabla p^{n+k} \) and \( \nabla \cdot u^{n+k} \) for the velocity-pressure coupling are taken implicitly to strictly enforce the discrete incompressibility condition. SBDF schemes are multistep methods which are not self-starting, therefore requiring proper initialization for \( \{u^j\}_{j=0}^{k-1} \). Taking the 3rd-order methods as an illustration, the initialization can be conveniently done as follows: 1st-order method (SBDF-1) is used to get \( u^1 \) from \( u^0 \); then 2nd-order method (SBDF-2) is used to get \( u^2 \) from \( u^0 \) and \( u^1 \). Then, the intended 3rd-order method (SBDF-3) can be used i.e. at \( t^n = n\tau \) for all \( n \geq 3 \) until the final simulation time \( T \) is reached. We noticed that initial data generated by one-step method of an equal order of accuracy has potential to reduce the numerical error (in time) by one to two orders of magnitude. For instance, we proposed initialization using a one-step 3rd-order method (e.g. DC-3) to generate \( u^1 \) and \( u^2 \) for SBDF-3; while a one-step 2nd-order method (e.g., Crank-Nicolson, DC-2, etc) is used to generate \( u^1 \) for SBDF-2. DC methods is based on defect correction strategy which we will introduce shortly. Although SBDF methods are well known for their simplicity, the critical time step for numerical stability is reduced as the order of the method increases. We will illustrate this fact through numerical test cases. In this work, we focus only on the 1st-, 2nd- and 3rd-order semi-implicit backward difference schemes for Navier–Stokes equations, referred to as SBDF-1, SBDF-2 and
SBDF-3, respectively. The time discretization of the momentum equations using these methods can be expressed as follows:

(a) **SBDF-1.** Given the initial solution \( u_h^0 \), for all \( n \in \mathbb{N} \), find \((u_h^{n+1}, p_h^{n+1})\) solution of

\[
\frac{u_h^{n+1} - u_h^n}{\tau} - \nu \Delta u_h^{n+1} + \nabla p_h^{n+1} = f_h^{n+1} - B(u_h^n),
\]

(b) **SBDF-2.** Given the initial solution \( u_h^0 \) and a proper initialization for \( u_h^1 \), for all \( n \in \mathbb{N} \), find \((u_h^{n+2}, p_h^{n+2})\) solution of

\[
\frac{3u_h^{n+2} - 4u_h^{n+1} + u_h^n}{2\tau} - \nu \Delta u_h^{n+2} + \nabla p_h^{n+2} = f_h^{n+2} - 2B(u_h^{n+1}) + B(u_h^n),
\]

(c) **SBDF-3.** Given the initial solution \( u_h^0 \) and a proper initialization for \( \{u_h^1, u_h^2\} \), for all \( n \in \mathbb{N} \), find \((u_h^{n+3}, p_h^{n+3})\) solution of

\[
\frac{11u_h^{n+3} - 18u_h^{n+2} + 9u_h^{n+1} - 2u_h^n}{6\tau} - \nu \Delta u_h^{n+3} + \nabla p_h^{n+3} = f_h^{n+3}
\]

\[
-3B(u_h^{n+2}) + 3B(u_h^{n+1}) - B(u_h^n),
\]

with these three methods sharing the same discretization of the continuity equation \( \nabla \cdot u_h^{n+k} = 0 \), \( k = 1, 2 \) and 3, respectively. For simplicity, we denote the nonlinear advection term by \( B(u_h) = u_h \cdot \nabla u_h \).

2.2. Defect Correction Method (DC)

Defect or ‘deferred’ correction methods were first proposed to solve ordinary differential equations (ODEs) [48] and more recent work can be found in [28, 43]. The defect correction method was applied to the time-dependent Navier–Stokes equations to produce more accurate solution [31] while more practical construction of the defect correction method which motivates our work is given in [21]. To begin with, time-stepping scheme with arbitrary high-order of accuracy can be constructed using only SBDF-1 with the defect correction approach. Assuming that the discrete solution (in time) for velocity and pressure can be expressed in terms of asymptotic expansions in
\[ u^{n+1} := u_0^{n+1} + \tau u_1^{n+1} + \tau^2 u_2^{n+1} + \cdots + \tau^k u_k^{n+1} + \mathcal{O}(\tau^{k+1}), \]
\[ p^{n+1} := p_0^{n+1} + \tau p_1^{n+1} + \tau^2 p_2^{n+1} + \cdots + \tau^k p_k^{n+1} + \mathcal{O}(\tau^{k+1}), \] (8)

where \( k \) also denotes the degree of the expansion. Similar expansions can be defined for \( u^n \) and \( p^n \). We illustrate the construction of a DC-\( k \)-method where \( k = 3 \) is fixed. Using Taylor's expansion, we first expand \( u(t) \) around \( t = t^{n+1} \) to produce the following

\[ u(t^n) = u(t^{n+1}) - \tau \frac{d}{dT} u(t^{n+1}) + \frac{\tau^2}{2!} \frac{d^2}{dT^2} u(t^{n+1}) - \frac{\tau^3}{3!} \frac{d^3}{dT^3} u(t^{n+1}) + \frac{\tau^4}{4!} \frac{d^4}{dT^4} u(\xi) \]
(9)

for some \( \xi = \xi(x) \in [t^n, t^{n+1}] \), assuming that the velocity is sufficiently smooth in time. Let us define further \( u^{n+1} := u(t^{n+1}), u^n := u(t^n), \) \( \nu(t) :\) \( d^j u^{n+1} := \frac{d^j}{dT^j} u(t^{n+1}) \). Combining Eq.(8) and (9) and keeping terms up to \( \tau^3 \) give the following:

\[ u_0^{n+1} + \tau u_1^n + \tau^2 u_2^n + \tau^3 u_3^n = u_0^{n+1} + \tau u_1^{n+1} + \tau^2 u_2^{n+1} + \tau^3 u_3^{n+1} \]

\[ - \tau \frac{d}{dT} u^{n+1} + \frac{\tau^2}{2} (d^2 u_0^{n+1} + \tau d^2 u_1^{n+1}) - \frac{\tau^3}{6} d^3 u_0^{n+1} \]

\[ + \mathcal{O}(\tau^4) \] (10)

Here, we observe that the first derivative of velocity, \( \frac{d^j}{dT^j} u^{n+1} \), can be replaced by the momentum equation from Navier-Stokes equations (Eq.(1)) to give:

**Asymptotic expansion for the momentum equation**

\[ u_0^{n+1} - u_0^n + \tau u_1^{n+1} - \tau u_1^n + \tau^2 u_2^{n+1} - \tau^2 u_2^n + \tau^3 u_3^{n+1} - \tau^3 u_3^n = \]
\[ \tau \nu \Delta u_0^{n+1} + \tau^2 \nu \Delta u_1^{n+1} + \tau^3 \nu \Delta u_2^{n+1} \]
\[ - \tau \nabla p_0^{n+1} - \tau^2 \nabla p_1^{n+1} - \tau^3 \nabla p_2^{n+1} \]
\[ - \tau B(u^{n+1}) - \frac{\tau^2}{2} (d^2 u_0^{n+1} + \tau d^2 u_1^{n+1}) + \frac{\tau^3}{6} d^3 u_0^{n+1} + \mathcal{O}(\tau^4). \] (11)
Further, we define approximations of high-order derivative terms using the difference schemes as follows:

\[
d^2 u_0^{n+1} = \frac{u_0^{n+1} - 2u_0^n + u_0^{n-1}}{\tau^2} + O(\tau), \quad \text{for } n \geq 0, \quad (12)
\]

\[
d^2 u_1^{n+1} = \frac{u_1^{n+1} - 2u_1^n + u_1^{n-1}}{\tau^2} + O(\tau), \quad \text{for } n \geq 1, \quad (13)
\]

\[
d^3 u_0^{n+1} = \frac{u_0^{n+1} - 3u_0^n + 3u_0^{n-1} - u_0^{n-2}}{\tau^3} + O(\tau), \quad \text{for } n \geq 2. \quad (14)
\]

By the linearity of both diffusion and pressure gradient terms and by using a simple ansatz for the nonlinear terms, these terms are expanded in increasing power of \(\tau\) to produce the following:

\[
\frac{u_0^{n+1} - u_0^n}{\tau} - \nu \Delta u_0^{n+1} + B(u_0^n) + \nabla p_0^{n+1} - f^{n+1} + O(\tau)
\]

\[
\frac{u_1^{n+1} - u_1^n}{\tau} - \nu \Delta u_1^{n+1} + \frac{B(u_0^{n+1} + \tau u_1^n) - B(u_0^n)}{\tau} + \nabla p_1^{n+1} + \frac{1}{2} d^2 u_0^{n+1} + O(\tau)
\]

\[
\frac{u_2^{n+1} - u_2^n}{\tau} - \nu \Delta u_2^{n+1} + \frac{B(u_0^{n+1} + \tau u_1^n + \tau^2 u_2^n) - B(u_0^n + \tau u_1^n)}{\tau^2} + \nabla p_2^{n+1} + \frac{1}{2} d^2 u_1^{n+1} - \frac{1}{6} d^3 u_0^{n+1} + O(\tau^3) = 0, \quad (15)
\]

with the approximation \(B(u_0^{n+1}) \approx B(u_0^n + \tau u_1^n + \tau^2 u_2^n)\) since taking \(u_2^n\) instead of \(u_2^{n+1}\) in the nonlinear term gives rise to a semi-implicit method. It can be shown that the global approximation of the nonlinear term has a 3rd-order truncation error if and only if the condition

\[
\|u_2^{n+1} - u_2^n\|_{L^2(\Omega)} \leq c\tau \quad (16)
\]

is satisfied (the approximation \(u_2^n\) converges at 1st-order in time) for some real constant \(c > 0\) and for all \(n \geq 0\). We omit the proof here. This results in Eq.(15) being an approximation to the momentum equation with 3rd-order accuracy in time. A similar approach can be applied to the continuity equation to ensure 3rd-order accuracy.
Asymptotic expansion for the continuity equation

$$\nabla \cdot \mathbf{u}^{n+1}_0 + \tau \nabla \cdot \mathbf{u}^{n+1}_1 + \tau^2 \nabla \cdot \mathbf{u}^{n+1}_2 + O(\tau^3) = 0.$$  \hspace{1cm} (17)

Matching the coefficients of $\tau$ and using ‘divide and conquer’ strategy, Eq.(15) and (17) are recasted into subproblems (18)-(20), which are solved in a multi-stage manner. For brevity, we provide only the 3rd-order DC schemes (DC-3). By initializing $\mathbf{u}^0_0 = \mathbf{u}(0)$, $\mathbf{u}^0_1 = 0$, $\mathbf{u}^0_2 = 0$, the velocity and pressure are obtained using the following algorithm: Find $(\mathbf{u}^{n+1}_0, p^{n+1}_0)$ solution of Eq.(18), $(\mathbf{u}^n_1, p^n_1)$ solution of Eq.(19), $(\mathbf{u}^{n-1}_2, p^{n-1}_2)$ solution of Eq.(20) with the final
global update \((u^{n-1}, p^{n-1})\):

For \(n \geq 0\),

\[
\begin{align*}
\begin{aligned}
\text{nl}^{n+1}_0 &= \begin{cases} B(u^n_0), & \text{for } n = 0, 1, \\ B(u^n_0 + \tau u^{n-1}_1), & \text{for } n \geq 2, \end{cases} \\
\frac{u^{n+1}_0 - u^n_0}{\tau} - \nu \Delta u^{n+1}_0 + \nabla p^{n+1}_0 &= f^{n+1} - \text{nl}^{n+1}_0, \\
\nabla \cdot u^{n+1}_0 &= 0, \\
\text{du}^{n+1}_0 &= (u^{n+1}_0 - u^n_0)/\tau.
\end{aligned}
\end{align*}
\]

For \(n \geq 1\),

\[
\begin{align*}
\begin{aligned}
\text{nl}^n_1 &= \begin{cases} B(u^n_0 + \tau u^{n-1}_1), & \text{for } n = 1, \\ B(u^n_0 + \tau u^{n-1}_1 + \tau^2 u^{n-2}_2), & \text{for } n \geq 2, \end{cases} \\
\frac{u^n_1 - u^{n-1}_1}{\tau} - \nu \Delta u^n_1 + \nabla p^n_1 &= -\frac{1}{2} \frac{d^2 u^{n+1}_0}{\tau} - \frac{1}{6} \frac{d^3 u^{n+1}_0}{\tau}, \\
\nabla \cdot u^n_1 &= 0, \\
\text{du}^n_1 &= (u^n_1 - u^{n-1}_1)/\tau.
\end{aligned}
\end{align*}
\]

For \(n \geq 2\),

\[
\begin{align*}
\begin{aligned}
\text{nl}^{n-1}_2 &= B(u^{n-1}_0 + \tau u^{n-1}_1 + \tau^2 u^{n-2}_2), \\
\frac{u^{n-1}_2 - u^{n-2}_2}{\tau} - \nu \Delta u^{n-1}_2 + \nabla p^{n-1}_2 &= -\frac{1}{2} \frac{d^2 u^n_1}{\tau} + \frac{1}{6} \frac{d^3 u^n_1}{\tau}, \\
\nabla \cdot u^{n-1}_2 &= 0, \\
\text{u}^{n-1} &= u^{n-1}_0 + \tau u^{n-1}_1 + \tau^2 u^{n-2}_2, \\
\text{p}^{n-1} &= p^{n-1}_0 + \tau p^{n-1}_1 + \tau^2 p^{n-2}_2.
\end{aligned}
\end{align*}
\]

We propose new approximations to handle the nonlinear terms, i.e., \(\text{nl}^{n+1}_0\) and \(\text{nl}^n_1\) for DC-3 method which are different from the one proposed in [21]. We found that these new approximations result in a significant reduction of the numerical error while still being 3rd-order accurate on both velocity and pressure. Unlike higher-order SBDF methods which are multistep schemes, DC schemes are self-starting methods. In a hierarchical manner, the high-order derivatives, \(d^2 u^{n+1}_0\), \(d^2 u^n_1\) and \(d^3 u^n_1\), for \(n \geq 0\), \(n \geq 1\) and \(n \geq 2\), respectively, are evaluated explicitly using Eq.-(12)-(14), to produce a
method with \(k\)-th-order accuracy in time for both velocity and pressure. For any \(k\)-th-order methods, the structure of the resulting linear system for all subproblems is similar to that of SBDF-1 with a different right hand side. The downside of DC methods arises from the need to solve \(k\) saddle point problems at each time step compared to only one for SBDF methods. For a small scale problem, DC may still be efficient since the same matrix is used for all subproblems and can thus be factorized once during the first time step.

2.3. Guermond–Minev Method (GM)

The artificial compressibility technique was first introduced by Chorin [10] to eliminate both the undetermined constant in the pressure and complications that arise when solving saddle point systems. This can be done by modifying the continuity equation in Eq.(1) such that we have \(\epsilon p_t + \nabla \cdot \mathbf{u} = 0\) where \(\epsilon > 0\) is known as the penalization parameter. For illustration, we discretized the time-derivative term for velocity \(\mathbf{u}_t\) and pressure \(p_t\) using 1st-order backward Euler formula and by fixing \(r = \tau \epsilon\), the iteration parameter. Meanwhile, the nonlinear term is treated explicitly. Using the discretized continuity equation, we eliminate the pressure gradient \(\nabla p_{n+1}^{s+1}\) in the momentum equation. It results an augmented Lagrangian method (ALM) for solving the unsteady Navier–Stokes equations, which reads: Given the initial pressure \(p_0^0 = p(0)\) and suitable \(r > 0\), we iterate the problem for \(s = 1, 2, \ldots\)

\[
\frac{\mathbf{u}_{s+1}^{n+1} - \mathbf{u}_s^n}{\tau} - \nu \Delta \mathbf{u}_{s+1}^{n+1} - r \nabla \nabla \cdot \mathbf{u}_{s+1}^{n+1} + \mathbf{u}_{s+1}^{n+1} \cdot \nabla \mathbf{u}_s^n + \nabla p_s^n = \mathbf{f}, \text{ in } \Omega \times (0, T),
\]

\[
p_{s+1}^{n+1} = p_{s}^{n+1} - r \nabla \cdot \mathbf{u}_{s+1}^{n+1}, \text{ in } \Omega \times (0, T).
\]

(21)

while the stopping criterion \(\|p_{s+1}^{n+1} - p_{s}^{n+1}\|_{L^2(\Omega)} < tol\) is met each time step or \(t_{n+1}\), where \(tol > 0\) is chosen small. The convergence of this method is only proven for Stokes equations in [15]. For semi-implicit schemes, this idea is remarkable since only two linear systems with symmetric and positive definite matrices are required to be solved and iterative methods (e.g. conjugate gradient method) work well for such linear systems. For unsteady problem, however, one may predict that the large number of iteration required for the convergence of the pressure at every time step makes the method impractical. Recently, Guermond and Minev [21] proposed a high-order time-stepping schemes based upon artificial compressibility for Navier–Stokes equations.
Using defect correction methods and the bootstrapping technique, the resulting uncoupled scheme requires only one update of the pressure at every time step, which produces a significant reduction in CPU time compared to the native implementation of ALM extension. Without providing technical details on the derivation of this method, the Guermond-Minev scheme with 3rd-order accuracy (GM-3) is given as follows [21]: For a given initial data \((u^0_0, p^0_0) = (u(0), p(0))\) and setting \((u^0_1, p^0_1) = (0, 0),\) \((u^0_2, p^0_2) = (0, 0),\) we solve for \((u^{n+1}, p^{n+1})\) through the computation of \((u^{n+1}_0, p^{n+1}_0), (u^{n+1}_1, p^{n+1}_1)\) and
The stability and convergence analysis of the method is done only for the linear problem (i.e $B(u) = 0$) in [21]. In later section, we will illustrate through our numerical results that the method reaches the expected rate of convergence and is stable with a suitable choice of time step on a given mesh.
2.4. Guermond-Minev with sequential regularization method (GM-SRM)

We propose a modification of the GM method using the so-called sequential regularization method (SRM) which was first investigated by [32, 34] considering that Navier–Stokes equations is a system of differential-algebraic equations (DAE) of index 2. This gives rise to the new formulation of Navier-Stokes equations which reads

\[
\begin{aligned}
\mathbf{u}_t^{s+1} - \nu \Delta \mathbf{u}^{s+1} - \alpha_1 \nabla \nabla \cdot \mathbf{u}_t^{s+1} - \alpha_2 \nabla \nabla \cdot \mathbf{u}^{s+1} + \mathbf{u}^{s+1} \cdot \nabla \mathbf{u}^{s+1} \\
+ \nabla p^s = f, \quad \text{in } \Omega \times (0, T), \\
p^{s+1} = p^s - \alpha_1 \nabla \cdot \mathbf{u}_t^{s+1} - \alpha_2 \nabla \cdot \mathbf{u}^{s+1}, \quad \text{in } \Omega \times (0, T).
\end{aligned}
\] (25)

This problem can be solved using any suitable time-stepping scheme and stopping criteria as for ALM. Both \(\alpha_1, \alpha_2 > 0\) are user-defined stabilization parameters for SRM. By choosing \(\alpha_1 = 0\) and \(\alpha_2 = \lambda\), we fall back on the GM methods. The time-derivative of divergence of \(\mathbf{u}\), i.e., \(\nabla \cdot \mathbf{u}_t\), is discretized using similar Taylor series expansion as in Eq.(9). We retain the high-order derivative terms for defect correction purposes, as was done for the other terms in the PDE. The method GM-SRM with 3rd-order accuracy is given as follows: For a given initial data \((\mathbf{u}_0^0, p_0^0) = (\mathbf{u}(0), p(0))\) and setting \((\mathbf{u}_1^0, p_1^0) = (0, 0), (\mathbf{u}_2^0, p_2^0) = (0, 0)\), we solve for \((\mathbf{u}_n^{n+1}, p_n^{n+1})\) through the computation of \((\mathbf{u}_0^{n+1}, p_0^{n+1}), (\mathbf{u}_1^n, p_1^n)\) and \((\mathbf{u}_2^{n-1}, p_2^{n-1})\) using Eq.(26), (27)
and (28), respectively:

\[
\begin{align*}
\mathbf{n}_0^{n+1} &= \begin{cases} 
B(u_0^n), & \text{for } 0 \leq n \leq 1, \\
B(u_0^n + \tau u_1^{n-1}), & \text{for } n \geq 2,
\end{cases} \\
\frac{u_0^{n+1} - u_0^n}{\tau} - \nu \Delta u_0^{n+1} - \mu_1 \nabla \cdot u_0^{n+1} + \mu_2 \nabla \cdot u_0^n + \nabla p_0^n & = f^{n+1} - \mathbf{n}_1^{n+1}, \\
p_0^{n+1} & = p_0^n - \mu_1 \nabla \cdot u_0^{n+1} + \mu_2 \nabla \cdot u_0^n, \\
du_0^{n+1} & = (u_0^{n+1} - u_0^n)/\tau, \quad dp_0^{n+1} = (p_0^{n+1} - p_0^n)/\tau.
\end{align*}
\]

For \( n \geq 0 \),

\[
\begin{align*}
d^2 u_0^{n+1} &= (d u_0^{n+1} - d u_0^n)/\tau, \\
\mathbf{n}_1^n &= \begin{cases} 
B(u_0^n + \tau u_1^{n-1}), & \text{for } n = 1, \\
B(u_0^n + \tau u_1^{n-1} + \tau^2 u_2^{n-2}), & \text{for } n \geq 2,
\end{cases} \\
\frac{u_1^n - u_1^{n-1}}{\tau} - \nu \Delta u_1^n - \mu_1 \nabla \cdot u_1^n + \mu_2 \nabla \cdot u_1^{n-1} & + \nabla (p_1^{n-1} + dp_0^n) = -\frac{1}{2} (d^2 u_0^{n+1} - \mu_2 \nabla \cdot d^2 u_0^{n+1}) \\
- \frac{\mathbf{n}_1^n}{\mathbf{n}_2^n}, \\
p_1^n & = p_1^{n-1} + dp_0^n - \mu_1 \nabla \cdot u_1^n + \mu_2 \nabla \cdot u_1^{n-1} - \frac{\mu_2 \tau}{2} \nabla \cdot d^2 u_0^{n+1}, \\
du_1^n & = (u_1^n - u_1^{n-1})/\tau, \quad dp_1^n = (p_1^n - p_1^{n-1})/\tau.
\end{align*}
\]

For \( n \geq 1 \),

\[
\begin{align*}
d^2 u_1^n &= (d u_1^n - d u_1^{n-1})/\tau, \quad d^3 u_0^{n+1} = (d^3 u_0^{n+1} - d^3 u_0^n)/\tau, \\
\mathbf{n}_2^{n-1} &= B(u_0^{n-1} + \tau u_1^{n-1} + \tau^2 u_2^{n-2}), \\
\frac{u_2^{n+1} - u_2^{n-2}}{\tau} - \nu \Delta u_2^{n-1} - \mu_1 \nabla \cdot u_2^{n-1} + \mu_2 \nabla \cdot u_2^{n-2} & + \nabla (p_2^{n-2} + dp_1^{n-1}) = -\frac{1}{2} (d^2 u_1^n - \mu_2 \nabla \cdot d^2 u_1^n) \\
& + \frac{1}{6} (d^3 u_0^{n+1} - \mu_2 \nabla \cdot d^3 u_0^{n+1}) - \frac{\mathbf{n}_2^{n-1} - \mathbf{n}_1^{n-1}}{\tau^2}, \\
p_2^{n-1} & = p_2^{n-2} + dp_1^{n-1} - \mu_1 \nabla \cdot u_2^{n-1} + \mu_2 \nabla \cdot u_2^{n-2} - \frac{\mu_2 \tau}{2} \nabla \cdot d^2 u_1^n \\
& + \frac{\mu_2 \tau}{6} \nabla \cdot d^3 u_0^{n+1}, \\
u_2^{n-1} & = u_0^{n-1} - \tau u_1^{n-1} \times \tau^2 u_2^{n-1}, \\
p_2^{n-1} & = p_2^{n-1} + \tau p_1^{n-1} + \tau^2 p_2^{n-1}.
\end{align*}
\]
where $\mu_1 = \frac{\alpha_1}{\tau} + \alpha_2$ and $\mu_2 = \frac{\alpha_1}{\tau}$. The existence and uniqueness of the solution of SRM formulation for time-dependent Navier–Stokes equations was studied in [34] for 1st-order approximations in time. There are very few numerical test cases available in the literature for SRM applied to unsteady flows, not to mention when this involves high-order methods in time. In this paper, we intend to fill this gap.

3. Stabilization Parameter for GM and GM-SRM

The $\text{grad}$-$\text{div}$ stabilization term arises naturally in GM and GM-SRM methods since the Navier–Stokes equations are solved by uncoupling the velocity and pressure. In the continuous setting of the Navier–Stokes equations, the presence of $\text{grad}$-$\text{div}$ term in the momentum does not change the solution. In discrete setting, however, $\text{grad}$-$\text{div}$ term does not change the solution only under certain conditions, for instance when finite elements lie within the divergence free subspace or when $\nabla \cdot V_h \subset M_h$. In particular, when $\mathbb{P}_2-\mathbb{P}_1$ finite elements are used, we have $\nabla \cdot V_h \subseteq Z_h \not\subset M_h$, where

$$Z_h = \{ v_h \in L^2(\Omega) \mid v_{h|K} \in \mathbb{P}_1, \forall K \in T_h \}. \quad (29)$$

For the discrete weak formulation of GM and GM-SRM, the $\text{grad}$-$\text{div}$ term in the momentum equation cannot be eliminated as is done from the continuity equation at the continuous level. This explains that for both GM and GM-SRM methods, the computed solution may differ from that with SBDF and DC methods even when we let $\tau \to 0$, with $h$ fixed. When computing flow problem with $\mathbb{P}_2-\mathbb{P}_1$ elements on a coarser grid, it is known that the $\text{grad}$-$\text{div}$ term helps to penalize the incompressibility condition at the element level hence improving the overall quality of the solutions. Several other advantages of adding the $\text{grad}$-$\text{div}$ stabilization term for solving Stokes equations have been addressed by Olshanskii et al. [37, 38] and references therein. Our work will only provide a comparison in terms of the solution and numerical behaviour between our methods with $\text{grad}$-$\text{div}$ and without $\text{grad}$-$\text{div}$ on a given mesh. The numerical improvement of $\mathbb{P}_2-\mathbb{P}_1$ elements by the addition of $\text{grad}$-$\text{div}$ term will be addressed in a separate work.

A challenging issue when using GM and GM-SRM is the choice of stabilization parameter, i.e., $\lambda$ for GM, $\alpha_1$ and $\alpha_2$ for GM-SRM, which remains a controversial issue. For instance, by picking a large $\lambda$, it helps to enforce the incompressibility condition rapidly (in very few time steps). However,
the choice of such large stabilization parameter may result in the poor con-
ditioning of the linear system due to the “locking” or very stiff behaviour
imposed by the \textbf{grad-div} term in the momentum equation \cite{ref21}. A similar
problem was studied by Glowinski and Fortin \cite{ref15} but involving only Stokes
equation with ALM. Our linear system for the momentum equation (in GM
and GM-SRM) is almost identical with theirs except for the presence of an
additional diagonal block with the mass matrix from the time discretization
in our case. They had proven that a large “iterative parameter” akin to our
stabilization parameter results in poor convergence behaviour (ill-conditioned
matrix) when iterative solvers are used to solve this linear system. Since we
use a direct solver for the linear system, this is not a major issue. Large sta-
bilization parameter is also related to over-stabilization effects which induces
numerical dissipation and lost of accuracy \cite{ref37}. Small $\lambda$ on the other hand,
may complicate the convergence of pressure between momentum and contin-
uity equation. This results in parasitic oscillations on both pressure and
velocity due to poor mass conservation. GM and GM-SRM methods do not
have additional iterations between subproblems to ensure such convergence
is satisfied, as is done with ALM. The incompressibility condition is enforced
only with bootstrapping technique which critically depends on a well-chosen
stabilization parameter. As similar problems arise for GM-SRM methods,
the choice of $\alpha_1$ and $\alpha_2$ remains an open problem.

At this moment, we are not able to establish a mathematical theory to
determine an optimal value of these parameters in terms of producing the
least numerical error at a given time step. In our work, best values of the
parameters, $\lambda, \alpha_1$ and $\alpha_2$ can be estimated numerically (trial and error).
For instance in GM, we start by picking very large $\lambda$ and observing the
velocity and pressure during first few time steps. If no oscillations are found,
computations are rerun with smaller $\lambda$. The ‘optimal’ $\lambda$ is chosen by picking
the one that starts to induce oscillations but will be progressively dampened
the oscillations from one time step to the next. For the GM-SRM method,
a good combination of $\alpha_1$ and $\alpha_2$ is harder to identify. For simplicity, we
propose $\alpha_1 = \tau \alpha_2$ for test case which requires larger $\alpha_2$ (e.g., $\alpha_2 \gg 1$) while
$\alpha_1 = 1$ is typically chosen when only smaller $\alpha_2$ is required (e.g., $\alpha_2 = 1$).
We only need to determine the value of $\alpha_2$ using an approach that is similar
to what is done to obtain $\lambda$ in the GM method. Nonetheless, this approach
delivers good results in our computations. The nearly optimal values of $\lambda$ in
GM and $\alpha_1$ and $\alpha_2$ in GM-SRM are provided for each of the test cases.
4. Numerical Results

Numerical validation and benchmarking are done for the time-stepping schemes SBDF, DC, GM and GM-SRM, all with 2nd- and 3rd-order accuracy. All our tests are done with 2D flows. We start with two manufactured solutions, one involving Dirichlet–Neumann boundary conditions adapted from [21] and the Taylor’s Vortex flow with Dirichlet boundary conditions [40]. For test cases which are deemed to be more challenging, the flow around a cylinder (von Kármán alley) (see for instance [20, 40, 51]) and the lid-driven cavity flow (see [19, 39, 46]) are chosen. Since exact solutions do not exist for these last two test cases, we made comparisons with reference solutions and the results found in the literature.

Let us define $t^n = n\tau^*$ and $T = M\tau^*$ where $T$ is total simulation time, $\tau^*$ the time step for output purposes and $M \in \mathbb{N}$ given by the user. For convenience, we choose $\tau^*$ as a multiple of $\tau$. The error in time can be computed using the reference solutions $(u_{ref}, p_{ref})$ where $u_{ref} = u_{h,\tau_{ref}}$ and $p_{ref} = p_{h,\tau_{ref}}$ are the best approximants to the solution of the semi-discrete problem (in space). Very small time step, given as $\tau_{ref}$ is used to compute the reference solutions. The error from the time discretization, by fixing the discretization in space, can be computed as follows:

$$
\|u_{ref} - u_{h,\tau}\|_{L^2(0,T;L^2(\Omega))} = \left( \tau^* \sum_{n=1}^{M} \|u^n_{ref} - u^n_{h,\tau}\|_{[L^2(\Omega)]^2} \right)^{1/2},
$$

$$
\|p_{ref} - p_{h,\tau}\|_{L^2(0,T;L^2(\Omega))} = \left( \tau^* \sum_{n=1}^{M} \|p^n_{h,\tau} - p^n_{ref}\|_{L^2(\Omega)} \right)^{1/2}.
$$

(30)

This type of error is useful when exact solutions are not available and to study the order of convergence in time only. Due to the numerical discrepancies found between methods with \texttt{grad-div} and without \texttt{grad-div} term, we compare the errors produced by DC and SBDF methods with the reference solution that is generated by DC-3. On the other hand, we consider a reference solution that is generated by GM-3 in the assessment of the GM and GM-SRM methods. The formula to obtain the order of convergence in time, $k$, for velocity and pressure (respectively $w = u$ and $w = p$) reads:

$$
k = \ln \left( \frac{\|w_{ref} - w_{h,\tau}\|_{L^2(0,T;L^2(\Omega))}}{\|w_{ref} - w_{h,\tau/r}\|_{L^2(0,T;L^2(\Omega))}} \right) / \ln r,
$$

(31)
where \( r \in \mathbb{R}^+ \) is the refinement factor between consecutive time steps. The value \( r = 2 \) was chosen. For the finite element approximation, the mesh is assumed to be quasi-uniform, i.e., \( h_{\text{min}} \leq h \leq h_{\text{max}} \). Suppose that the space approximation is done using \( \mathbb{P}_2 - \mathbb{P}_1 \) mixed finite elements with sufficient regularity on \((u, p)\), we expect that the \( L^2 \)-error between numerical and exact solutions using \( k \)-th order time-stepping schemes to behave as

\[
\| (u - u_{h,\tau})(t_i) \|_{L^2(\Omega)} = \mathcal{O}(\tau^k + h^3), \quad 1 \leq i \leq N, \\
\| (p - p_{h,\tau})(t_i) \|_{L^2(\Omega)} = \mathcal{O}(\tau^k + h^2), \quad 1 \leq i \leq N.
\] (32)

where \( N = T/\tau \in \mathbb{N} \) is the total number of time step. Similarly, the \( L^2 \)-error between numerical and reference semi-discrete solutions obtained on the same mesh is expected to have the following order of accuracy

\[
\| (u_{\text{ref}} - u_{h,\tau})(t_i) \|_{L^2(\Omega)} = \mathcal{O}(\tau^k), \quad 1 \leq i \leq N, \\
\| (p_{\text{ref}} - p_{h,\tau})(t_i) \|_{L^2(\Omega)} = \mathcal{O}(\tau^k), \quad 1 \leq i \leq N.
\] (33)

The critical time step, \( \tau_{\text{crit}} \) for numerical stability of each time-stepping scheme is computed using a bisection method. For instance, the initial time step, \( \tau_0 \) is chosen large enough to result in a numerical blow-up after a few time steps. Next, same computation is carried out using the time step, \( \tau_1 = 0.5\tau_0 \). If the numerical stability is observed for a chosen total time \( T \), computation is restarted again with the larger time-step \( \tau_2 = 1.5\tau_1 \), otherwise we set \( \tau_2 = 0.5\tau_1 \) and the process is repeated. Typically, \( T = 10 \) is chosen to account nonlinear instability effects that may show-up in the computation. A fully developed periodic flow is used as initial condition. To obtain a converged critical time step \( \tau_{\text{crit}} \), the stopping criteria, i.e., 
\[
|\tau_{n+1} - \tau_n| < tol_{\tau},
\]

is used for some small \( tol_{\tau} > 0 \). All computations are done using FreeFEM++ [23]. We intend to focus only on the performance of the time-stepping schemes, hence the impact of linear solvers and preconditioners are not taken into consideration. We used MUMPS (Multifrontal Massively Parallel Sparse Direct Solver) which is one of the best direct solver available in FreeFEM++. MUMPS is chosen as it turns out to be twice as fast as the Unsymmetric MultiFrontal method (UMFPACK). All simulations are run under a Linux-platform PC with an Intel® Core™ i7-3770 CPU 3.40GHz and 32Gb of RAM. We compare the numerical efficiency of the time-stepping methods by looking at the CPU time (total runtime) required
to reach a desired level of error in time. Some of the methods may be very efficient when the error is large but surpassed by others when smaller error is required. Relative efficiency depends on several factors, namely the numerical complexity (i.e., number of subproblems and linear systems to be solved per time step), accuracy, stability condition (which determines the size of the maximal time step) and convergence behaviour of the methods. For 2D problems, the memory requirement of all these methods can be easily fulfilled by modern desktop computers. However, this may not be true for 3D problems.

4.1. Manufactured Solution I

This manufactured solution is an example where the discretization error in time is larger than that in space which was presented in [21]. For this test case, we use the square domain Ω = (0, 1) × (0, 1). The exact solutions for velocity and pressure are given by

\[
\mathbf{u} = (u_1, u_2) = (\sin(x) \sin(y + t), \cos(x) \cos(y + t)),
\]
\[
p = \cos(x) \sin(y + t),
\]

for \((x, y) \in \Omega, \ t \geq 0\). Homogeneous Neumann boundary conditions are prescribed at the left boundary while homogeneous Dirichlet boundary conditions (no-slip) are prescribed at bottom, right and upper boundaries. By fixing the viscosity constant, \(\nu = 1\), initial and boundary conditions, and \(f\) can be computed using Eq.(34). The domain is discretized using an unstructured mesh with uniform mesh size \(h = 0.01768\) (subdivision \(80 \times 80\) along the boundary). This gives a total of 6561 vertices, 12800 triangles and 58403 degrees of freedom for velocity and pressure. This test case is run for a total time \(T = 2\) with varying time steps, \(\tau = 0.02, 0.01, 0.005\) and \(6.25 \times 10^{-5}\). The smallest time step is used to generate the reference solution \((\mathbf{u}_{\text{ref}}, p_{\text{ref}})\) for computing the error (in time only) in the convergence analysis. For the stabilization parameters, we fix \(\lambda = 83.5\) in GM methods while \(\alpha_1 = 40/\tau\) and \(\alpha_2 = 40.0\) are used in GM-SRM methods.

Fig. 1 shows the time evolution of log-scaled \(L^2\)-error computed with Eq.(30) for all 2\textsuperscript{nd}- and 3\textsuperscript{rd}-order methods, both for velocity and pressure with \(\tau = 0.02, 0.01\) and 0.005. These plots confirm as expected, the following facts: the accuracy of the velocity obtained with all methods is better than that of pressure since \(P_2-P_1\) finite element is used; the 3\textsuperscript{rd}-order methods are more accurate than their 2\textsuperscript{nd}-order counterparts at any fixed time \(t \in [0, 2]\).
and the numerical errors for velocity and pressure decrease as the time step is refined. For $\tau = 0.02$, all graphs for 3rd-order methods (except SBDF-3 for velocity) are superposed with the lowest curve, where only the error from the space approximation remains. The error of velocity for SBDF-3 is still large with the time step $\tau = 0.02$. We suspect that this error is induced from a time step chosen too close to the critical value for SBDF-3 method. The critical time step for SBDF-3 is known to be smaller than that for other 3rd-order methods to fulfill the CFL stability condition. With the smallest time step, $\tau = 0.005$, both DC-3 and SBDF-3 produce the least error for both velocity and pressure, followed by GM-3 and GM-SRM-3. Here, GM-SRM produces very similar errors to that computed with GM for both 2nd- and 3rd-order accuracy as computations proceed further from the start-up. However, it can be observed that numerical oscillations on the solutions of GM-3 (and consequently on the error) at start-up require more time steps to attenuate completely compared to GM-SRM-3, which suggests that GM-SRM-3 may have better stability property than GM-3 in terms of enforcing divergence free condition at element level. For $\tau = 0.005$, we observed that the graph produced with all methods except GM-2, GM-SRM-2 and SBDF-2 methods collapse to the bottom graph with space error only except for minor discrepancies due to the presence of the $\text{grad-div}$ term in GM and GM-SRM. We note that the error produced by GM and GM-SRM can be reduced further and brought closer to minimal level by choosing a slightly smaller stabilization parameter, e.g., $\lambda < 83.5$ and $\alpha_2 < 40.0$. A disadvantage for this, however, is by having more severe oscillations generated at the beginning of the computations which require more time steps to be damped.
Figure 1: Manufactured solution I: Time history of $L^2$-error for velocity and pressure between numerical and exact solutions using 2nd- and 3rd-order SBDF, DC, GM, GM-SRM schemes (From left to right, the time steps $\tau = 0.02, 0.01$ and 0.005 are used).

We analyze the numerical efficiency and convergence in time of the scheme, i.e., by comparing the numerical and reference solutions for all methods (see Fig. 2). The numerical efficiency can be studied by considering first, the $L^2$-error versus CPU time and second, the $L^2$-error versus the time step, $\tau$ for both velocity and pressure. These errors are computed using Eq.(30). To disregard possible pressure oscillations generated by GM and GM-SRM methods at start-up, we evaluate the $L^2$-error using $\tau^* = 0.2$ only for $t \in [1, 2]$. Fig. 2 shows that all 3rd-order methods are more efficient than 2nd-order methods for both velocity and pressure as the accuracy reached for a given CPU time is always better with 3rd-order methods. Among all methods, the accuracy of SBDF is found to be lacking. SBDF methods converge to the solution of semi-discrete problem (in space) only if the time step is chosen sufficiently small. However, we noted that the accuracy of SBDF methods can be improved by implementing different discretizations of the nonlinear term. Further discussion on this will be provided in a separate paper.
any given level of error, DC-3 method is found to be the most efficient, for instance it can reach an error of about $10^{-8}$ for velocity and about $10^{-6}$ for pressure with shortest CPU time, closely followed by GM-3, GM-SRM-3 and SBDF-3 methods. It is noteworthy that DC-3 method requires to solve three saddle point problems at every time step, and still is the most efficient for this test case. With a small number of unknowns, the use of a direct solver (e.g., MUMPS) makes these computations efficient. For a fixed time step and order of the method, DC, GM and GM-SRM methods produce smaller error in time than that with SBDF methods since they are all based upon defect correction strategy. Fig. 2 (at the bottom) shows that all time-stepping schemes converge in time with the theoretical rate for both velocity and pressure.

4.2. Manufactured Solution II (Taylor Vortex)

The Taylor vortex is a manufactured solution modeling flow problems with Dirichlet boundary conditions [30] which is often used for numerical
validation. It models a decaying flow in time and exhibits a dominating error in space over the error in time. This test case mimics industrial fluid stirring mechanism using rotating and counter-rotating cylinder rods in a mixing tank. In our test case, the square domain \( \Omega = (0.25, 1.25) \times (0.5, 1.5) \) is used with the exact solutions for velocity and pressure given by

\[
\begin{align*}
\mathbf{u} &= (u_1, u_2) = (-\cos(2\pi x)\sin(2\pi y)e^{-8\pi^2\nu t}, \sin(2\pi x)\cos(2\pi y)e^{-8\pi^2\nu t}), \\
p &= -\frac{1}{4} (\cos(4\pi x) + \cos(4\pi y))e^{-16\pi^2\nu t},
\end{align*}
\]

(35)
in which the external force, \( \mathbf{f} \equiv 0 \). We set the kinematic viscosity, \( \nu \) to 0.01. Dirichlet boundary conditions which are prescribed on the four edges of the square and initial conditions can be obtained through the exact solution (Eq.(35)). The domain is discretized with an unstructured triangular mesh by subdividing each side of the square in 120 equal length edges. With \( \mathbb{P}_2-\mathbb{P}_1 \) finite elements, a total of 34210 elements, 17346 vertices and 155148 degrees of freedom for both velocity and pressure are produced. The element size, \( h_K \) lies in the range \( 0.00645 \leq h_K \leq 0.01518 \). Since this is an exponentially decaying solutions, we limit the simulation only to \( t \in [0, 1] \). We fix \( \lambda = 3.3 \) in GM methods and \( \alpha_1 = \alpha_2 = 1 \) in GM-SRM methods. The numerical comparisons are done in a similar fashion as in Section 4.1. For the error (in time) and convergence analysis, we only consider the solution generated at every step \( \tau^* = 0.2 \) for \( t \in [0.6, 1.0] \) (using Eq. 30).

Fig. 3 shows the time evolution of the \( L^2 \)-error on velocity and pressure comparing with the exact solutions, computed by various methods with time steps \( \tau = 0.02, 0.01, 0.005 \). Remarks are similar as for the previous test case, except for the observation that GM-SRM methods slightly outperform other methods of the same order since they have the smallest error on velocity. This suggests that the presence of a \textbf{grad-div} term with nearly optimal choice of the stabilization parameters helps to enforce the incompressibility condition while improving the accuracy on velocity (in space) by about one order of magnitude compared to other methods. We observe again that the accuracy of SBDF-3 is compromised when the time step is chosen too close to the critical time step for stability (see top left graph in Fig. 3). Fig. 4 illustrates the numerical efficiency of the methods by showing graphs of the \( L^2 \)-error (in time only) versus the required CPU time and chosen time step, respectively, for all methods. Here, we set \( T = 1 \) and a very small time step \( \tau = 6.25 \times 10^{-5} \) to compute the reference solution. The numerical rate of convergence of all
Figure 3: Manufactured solution II: Time history of $L^2$-error for velocity and pressure between numerical and exact solutions using 2nd- and 3rd-order SBDF, DC, GM, GM-SRM schemes (From left to right, the time steps $\tau = 0.02, 0.01$ and 0.005 are used).
time-stepping schemes shows a satisfactory agreement with the theoretical rate of convergence in time for both velocity and pressure. At larger time step (e.g., $\tau > 0.01$), SBDF-3 does not exhibit the expected order of convergence when the chosen time step is too close to the critical time step. However, the SBDF-3 method is found to quickly surpass DC-3, GM-3 and GM-SRM-3 methods for errors below $1 \times 10^{-6}$ and $3 \times 10^{-7}$ on velocity and pressure, respectively. At any given order of accuracy, SBDF methods are found to be the most efficient for both velocity and pressure at least when sufficiently small time steps are used. Interestingly, SBDF methods produce the smallest error on velocity in time at smaller time step, e.g., $\tau < 0.01$ for this test case. This is observed since the initialization procedure is done using high-order method, i.e., DC methods. For pressure, GM, GM-SRM and DC methods produced smaller error than SBDF methods for a given time step and order of accuracy.

Figure 4: Manufactured Solution II: Plot of the error on velocity (left) and pressure (right) as a function of CPU time (top) and time-step (bottom) for SBDF, DC, GM, GM-SRM time-stepping schemes.
4.3. Flow Past a Circular Object: von Kármán Alley

Figure 5: The computational domain and mesh for flow past a circular cylinder (top). A blow-up of the mesh with boundary layer near the cylinder (yellow represents higher flow speed and blue represents lower flow speed) (bottom)

The flow past a circular cylinder is one of the most famous flow experiments in fluid dynamics (see e.g. [25, 40, 45, 51]). In many numerical studies for Navier–Stokes equations, this test case has been used for benchmarking purposes, since only accurate numerical methods are able to reproduce the right flow properties. The categorization of vortex shedding behind the circular cylinder such as steady or unsteady laminar, subcritical and turbulent flows are strictly determined by the Reynolds numbers

\[ Re = \frac{U_\infty D}{\nu}, \]  

where \( D \) is the diameter of the cylinder, \( U_\infty \) the far-field speed and \( \nu \) the kinematic viscosity. The wake behind the cylinder gives rise to a periodic flow with symmetry breaking whenever \( 40 < Re < 150 \). The so-called laminar von Kármán vortex sheet is then fully developed. To quantify the periodicity of the flow, the Strouhal number is used, i.e., \( Str = \frac{fD}{U_\infty} \) where \( f \) is the frequency of the vortex shedding.
There are several ways to set the computational domain for this flow. In this paper, we use a conventional rectangular domain, \( \Omega = (-10, 25) \times (-10, 10) \) which is discretized by a total of 26936 non-uniform triangles (unstructured mesh) and 13648 vertices. The circular cylinder is located at the origin with \( D = 1 \). As \( \mathbb{P}_2-\mathbb{P}_1 \) finite elements are used for space discretization, there are 108474 unknowns for the velocity \( \mathbf{u} = (u, v) \) and 13648 unknowns for the pressure \( p \). The varying element size, \( h_K \), lies in the interval \([0.05129, 0.61148]\), with a sufficiently fine mesh generated near the cylinder to capture the boundary layer and coarsest element along the outer boundary (see Fig. 5). Setting \( U_\infty = 1 \) and \( \nu = 0.01 \), a Reynolds number, \( Re = 100 \) is obtained. For GM methods, the stabilization parameter \( \lambda = 10^4 \) is considered. For GM-SRM methods, typically smaller stabilization parameters \( \alpha_1 = 200 \) and \( \alpha_2 = 200 \) are chosen. The boundary conditions are prescribed as follows: along the left, upper and lower boundaries, Dirichlet boundary conditions \( \mathbf{u} = (u, v) = (1, 0) \) are set. Upper and lower boundaries in this paper are dealt differently from what is done in \([25, 35, 40, 51]\). In these papers, non-slip boundary \( \mathbf{u} = (0, 0) \) or \( \partial_x u = 0 \) with \( v = 0 \) were used along the upper and lower boundaries on much smaller computational domain. Neumann boundary conditions (free exit) are employed along the right outflow boundary, i.e., \( p - \nu \partial_x u|_{\Gamma_N} = 0 \) and \( -\nu \partial_x v|_{\Gamma_N} = 0 \) where \( \Gamma_N = \{(x, y) \in \Gamma \mid x = 25\} \). Non-slip boundary condition \( \mathbf{u} = (u, v) = (0, 0) \) is prescribed on the circular cylinder. Starting from an initial state at rest, a periodic motion is reached for about \( T = 80 \). This periodic flow then shows vortex shedding behind the cylinder propagating all the way to the outflow boundary. To reduce the computational time, we use the periodic solution that is computed with SBDF-2 as the initial state for all subsequent numerical tests. Since there is no analytical solution for this test case, we use a reference solution \((\mathbf{u}_{ref}, p_{ref})\) to assess all our methods.

In Fig. 6, the numerical efficiency is assessed by plotting the \( L^2 \)-error (in time) on velocity and pressure versus the CPU time for completing the computation for \( t \in [0, T] \) starting from the periodic solution mentioned above. We also illustrate the rate of convergence for each of the time-stepping schemes in the same figure. For a given time step, DC, GM and GM-SRM methods produce the least numerical errors for both velocity and pressure in time, then followed by SBDF methods. However, SBDF methods are the most CPU-efficient schemes, followed by DC, GM and GM-SRM methods at a given order of accuracy. All methods reproduce the theoretical rate of convergence. Table 1 summarizes the critical time steps \( \tau_{crit} \) for each of
Figure 6: Flow around the circular cylinder: Plot of the error on velocity (left) and pressure (right) as a function of CPU time (top) and time-step (bottom) for SBDF, DC, GM, GM-SRM time-stepping schemes.
the time-stepping schemes that are obtained using the bisection method. In addition, we provide the maximal CFL number or the $CFL_{\text{max}}$ attained in a local element which can be computed using the following:

$$CFL_{\text{max}} = \max_{K \in \mathcal{T}_h} \left\{ \frac{\tau_{\text{cri}}}{h_K} \| u_h \|_{L^2(K)} \right\},$$ (37)

where each $K$ is an element of the mesh $\mathcal{T}_h$. The critical time step (or equivalently the $CFL_{\text{max}}$) of GM-2/GM-SRM-2 is about 20% larger than that of both DC-2 and SBDF-2 methods, while the critical time step of GM-3/GM-SRM-3 is about 30% and 50% larger than DC-3 and SBDF-3 methods, respectively. These percentage are computed using the critical time step of GM/GM-SRM as the reference. We conjecture that the difference gained in terms of $\tau_{\text{cri}}$ in GM/GM-SRM methods compared to others will be larger as the order of the methods increases but this requires further validation. The stability of GM and GM-SRM methods is exceptional for such high-order semi-implicit schemes thanks to the presence of $\text{grad}$-$\text{div}$ terms. The so-called $\text{grad}$-$\text{div}$ stabilization term improves certain stability property akin to the streamline-upwinding Petrov Galerkin or the SUPG methods to solve highly nonlinear flow which had been addressed in [37, 38]. Interestingly, the critical time step for SBDF and DC methods are comparable. For instance, the critical time step of SBDF-2 method is slightly larger than that for DC-2 method while the critical time step of SBDF-3 method is slightly smaller than that for DC-3 method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Critical time step, $\tau_{\text{cri}}$</th>
<th>$CFL_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBDF-1</td>
<td>$1.7744 \times 10^{-2}$</td>
<td>$4.6319 \times 10^{-1}$</td>
</tr>
<tr>
<td>SBDF-2</td>
<td>$1.1951 \times 10^{-2}$</td>
<td>$4.1453 \times 10^{-1}$</td>
</tr>
<tr>
<td>SBDF-3</td>
<td>$7.0278 \times 10^{-3}$</td>
<td>$1.8320 \times 10^{-1}$</td>
</tr>
<tr>
<td>DC-2</td>
<td>$1.1551 \times 10^{-2}$</td>
<td>$3.0124 \times 10^{-1}$</td>
</tr>
<tr>
<td>DC-3</td>
<td>$8.1168 \times 10^{-3}$</td>
<td>$2.1156 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-2</td>
<td>$1.4529 \times 10^{-2}$</td>
<td>$3.7814 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-3</td>
<td>$1.0537 \times 10^{-2}$</td>
<td>$2.7438 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-SRM-2</td>
<td>$1.4522 \times 10^{-2}$</td>
<td>$3.7697 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-SRM-3</td>
<td>$1.0525 \times 10^{-2}$</td>
<td>$2.7407 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

Our next task is to compare the lift and drag coefficients which are computed by all our methods to the values in the literature. To achieve this goal, we repeat similar computations of the flow past the circular cylinder at $Re = 100$ but using a larger time interval, i.e., $t \in [0, 2000]$, to be sure
to reach a fully periodic state. The time step, $\tau = 0.005$ is chosen for all methods. The lift and drag coefficients, $c_l$ and $c_d$, respectively, are defined as follows:

$$
c_d(t) = \frac{2}{\rho U_\infty D} \int_S \left( \rho \nu \frac{\partial u_s(t)}{\partial n} - n_y p(t) n_x \right) dS,
$$

$$
c_l(t) = -\frac{2}{\rho U_\infty D} \int_S \left( \rho \nu \frac{\partial u_s(t)}{\partial n} - n_x p(t) n_y \right) dS,
$$

(38)

The parameter $\rho = 1$ is the density of the fluid, $S$ the boundary of the cylinder, $n = (n_x, n_y)^T$ is the unit normal vector on $S$ pointing inward $\Omega$, $t_s = (n_y, -n_x)^T$ is the tangential vector and $u_s$ is the tangential velocity. According to John [25], the computational of lift and drag using a volume integral formulation is more accurate and less sensitive to the mesh size around the cylinder than using the conventional line integral (Eq. 38). This method gives

$$
c_l(t) = -20 \int_\Omega \left[ u_t \cdot v_l + \nu \nabla u(t) : \nabla v_l + (u(t) \cdot \nabla) u(t) \cdot v_l - p(t) (\nabla v_l) \right] d\Omega \tag{39}
$$

for $v_l \in [H^1(\Omega)]^2$ solution of an auxiliary Stokes equations with boundary conditions $v_l|_S = (0, 1)$ and $v_l|_\Gamma = (0, 0)$ on all other boundaries. Similarly,

$$
c_d(t) = -20 \int_\Omega \left[ u_t \cdot v_d + \nu \nabla u(t) : \nabla v_d + (u(t) \cdot \nabla) u(t) \cdot v_d - p(t) (\nabla v_d) \right] d\Omega \tag{40}
$$

for $v_d \in [H^1(\Omega)]^2$ solution of an auxiliary Stokes equations with boundary conditions $v_d|_S = (1, 0)$ and $v_d|_\Gamma = (0, 0)$ on all other boundaries. Knowing a priori that the solution produces single periodic mode, the mean, amplitude and frequency for both lift and drag coefficients are computed with a simple approximation using the time history of these parameters in the last 50 time units. For instance, the period of the lift and drag can be approximated using the successive difference of the times taken when the maximal slope occurs. These slopes are computed using linear interpolation of two consecutive points generated at each time step. For flow with a single periodic mode, we found that this method delivers more accurate results than the one with the conventional Fast Fourier Transform (FFT).

Table 2 shows the mean, amplitude and frequency of the lift and drag
coefficients produced by our time-stepping methods. All methods produce a frequency for drag which is about twice the frequency for lift. This is in agreement with the literature on 2D laminar flows around the cylinder at $Re = 100$. The Strouhal number which is also the frequency of the lift coefficient in our present work is close to 0.17. Since the flow is still in laminar regime, the Strouhal number is less than 0.20—which is a value known for subcritical flows [4]. All methods produce a mean for lift coefficient close to zero, again as expected for such laminar flows (see, e.g., [25, 47]).

We observe that the mean, amplitude and frequency for the drag and lift coefficients computed using our methods are very close to each other. These values differ only in the second decimal digit, e.g., the lift amplitude shows a maximal difference of 4.7% (DC-3 versus GM-3 methods) and the mean drag, a maximal difference of 1.2% (DC-3 versus GM-SRM-3 methods). We observe that the most significant discrepancies occur for methods with $\text{grad}$-div term (GM/GM-SRM) versus methods without $\text{grad}$-div term (SBDF/DC). Still, we cannot assert which solver produces the most accurate result.

Table 2: The mean, amplitude and frequency of the lift and drag coefficients

<table>
<thead>
<tr>
<th>Method</th>
<th>$c_l$ mean</th>
<th>$c_l$ amplitude</th>
<th>$c_l$ frequency</th>
<th>$c_d$ mean</th>
<th>$c_d$ amplitude</th>
<th>$c_d$ frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBDF-2</td>
<td>$-1.78129 \times 10^{-3}$</td>
<td>$3.40513 \times 10^{-1}$</td>
<td>$1.70068 \times 10^{-1}$</td>
<td>$1.38565$</td>
<td>$9.79036 \times 10^{-3}$</td>
<td>$3.40136 \times 10^{-1}$</td>
</tr>
<tr>
<td>SBDF-3</td>
<td>$-1.76713 \times 10^{-5}$</td>
<td>$3.40500 \times 10^{-1}$</td>
<td>$1.70047 \times 10^{-1}$</td>
<td>$1.38565$</td>
<td>$9.78998 \times 10^{-3}$</td>
<td>$3.40099 \times 10^{-1}$</td>
</tr>
<tr>
<td>DC-2</td>
<td>$-1.77619 \times 10^{-5}$</td>
<td>$3.40506 \times 10^{-1}$</td>
<td>$1.70040 \times 10^{-1}$</td>
<td>$1.38564$</td>
<td>$9.78931 \times 10^{-3}$</td>
<td>$3.40092 \times 10^{-1}$</td>
</tr>
<tr>
<td>DC-3</td>
<td>$-9.92424 \times 10^{-6}$</td>
<td>$3.45476 \times 10^{-1}$</td>
<td>$1.69937 \times 10^{-1}$</td>
<td>$1.39060$</td>
<td>$1.00610 \times 10^{-2}$</td>
<td>$3.39936 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-2</td>
<td>$-8.87375 \times 10^{-5}$</td>
<td>$3.29150 \times 10^{-1}$</td>
<td>$1.70182 \times 10^{-1}$</td>
<td>$1.37285$</td>
<td>$9.60383 \times 10^{-3}$</td>
<td>$3.39443 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-3</td>
<td>$-7.41786 \times 10^{-5}$</td>
<td>$3.29204 \times 10^{-1}$</td>
<td>$1.70180 \times 10^{-1}$</td>
<td>$1.37299$</td>
<td>$9.20195 \times 10^{-3}$</td>
<td>$3.40362 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-SRM-2</td>
<td>$-6.50141 \times 10^{-5}$</td>
<td>$3.29261 \times 10^{-1}$</td>
<td>$1.70182 \times 10^{-1}$</td>
<td>$1.37304$</td>
<td>$9.20363 \times 10^{-3}$</td>
<td>$3.40367 \times 10^{-1}$</td>
</tr>
<tr>
<td>GM-SRM-3</td>
<td>$-6.50348 \times 10^{-5}$</td>
<td>$3.29256 \times 10^{-1}$</td>
<td>$1.70180 \times 10^{-1}$</td>
<td>$1.37295$</td>
<td>$9.11191 \times 10^{-3}$</td>
<td>$3.40362 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

We also make comparison with lift and drag coefficients obtained from the literature, both for numerical and experimental values. From Table 3, one can see that our computed Strouhal numbers = 0.17 is in a good agreement with published values except the one from Ranjani et. al [42] ($St = 0.1569$). For mean drag, our result differs by 0.8%-10.7% compared to values in the literature. Meanwhile, the differences on lift amplitude with published values are within 1.3%-15.6%. We cannot find sufficient data on the drag amplitude to provide comparison. Notice that none of the published values are computed by time-stepping schemes with 3rd- or higher-order accuracy. There are also differences in space approximations (e.g., finite volume, difference etc). These discrepancies may result also from the different methods used to compute the lift and drag (e.g., integration along the circular boundary [52] versus integration on volume [25]). The FFT analysis which is used by many may not be able to provide very accurate spectrum for the lift and drag
coefficients.

4.4. Lid-driven cavity flow

The lid-driven cavity is one of the most documented test cases in computational fluid dynamics. Cavity flows occur in many applications in modern industry. To name a few, lid-driven cavity can be used to study the efficiency of fluid mixing, stirring processes and fluid cooling mechanisms, where all of these take place either in an enclosed or semi-enclosed region. Although the experiment setting of lid-driven cavity is simple, the study of the resulting flow patterns and generated vortices, especially near corners, is very challenging. In dimensionless settings, the characterisation of flow patterns, the number of vortices and the stability of the flow in a square cavity is determined only by the Reynolds number. For Reynolds numbers, \( Re < 8000 \), the lid-driven cavity flow is known to be steady. However, for \( Re \geq 8000 \), the flow becomes unsteady, following a Hopf bifurcation leading to a periodic solution in both velocity and pressure. The critical Reynolds number, \( Re_{1,cri} \), where the first Hopf bifurcation takes place is given under 7500 by [19] and is reported very close to 8000 in more recent publications [2, 9, 14, 50].

In this paper, we investigate the performance of our methods using the lid-driven cavity flow at \( Re = 8500 \) [2, 39]. This Reynolds number is chosen for two reasons: to make sure that the flow is unsteady since \( Re > Re_{1,cri} \) with a sufficiently large gap, and simulations with \( Re = 8500 \) are known to be challenging. For instance, excessively long computational time is required beginning from a ‘cold’ start to go over a long transient and finally reach a fully periodic flow with a frequency around 0.44. Meanwhile, \( Re = 8500 \) is in the vicinity of a second Hopf bifurcation, which occurs around 8700 < \( Re_{2,cri} < 10000 \). This second Hopf bifurcation introduces a second periodic mode with a frequency around 0.61 [9, 50].
The numerical comparison is done both qualitatively and quantitatively. For qualitative analysis, we compare the formation of vortices using stream function plots with published results. While for quantitative analysis, we first establish a set of numerical results in terms of the frequency, amplitude and mean value of the $x$-velocity, $u$, at several monitoring points. Then the results computed by our methods are compared among themselves. Finally, we compare the frequency of the pulsating flow or the Strouhal number with similar values found in the literature.

We consider a unit square domain $\Omega = (0,1) \times (0,1)$ discretized using a non-uniform triangular mesh with 120 triangles along each boundary edge. This produces a mesh with element size $h_K$ within the range $[0.00628, 0.01660]$, with 34164 vertices, 17323 triangles and 154941 degrees of freedom with $\mathbb{P}_2-\mathbb{P}_1$ mixed finite elements. A constant velocity $\mathbf{u} = (1,0)$ is imposed on the top boundary (the lid), while for the remaining boundaries, we set $\mathbf{u} = (0,0)$. We prescribe the state of rest for velocity and pressure at $t = 0$. For this test case, we do not pursue an efficiency analysis similar to those carried out in above sections. Equally useful, the CPU time per time step (found as an average over 100 time steps) for each of the methods is given in Table 4. We choose a smaller time step, $\tau = 0.001$ for all 3rd-order methods to produce very accurate solution in time, which will serve as reference solutions for each method. In order to assess the quality of the 2nd-order methods, the following time step is set for each method: $\tau = 0.001$ in both SBDF-2 and DC-2 method and $\tau = 0.025$ in both GM-2 and GM-SRM-2 methods. We also include SBDF-2 method with a larger time step $\tau = 0.014$ which we call as SBDF-2*. For stabilization parameter, we fix $\lambda = 1$ in both GM-2 and GM-3 while prescribing $\alpha_1, \alpha_2 = 1$ in both GM-SRM-2 and GM-SRM-3.

Table 4 summarizes the critical time step of each method obtained using the bisection method and maximum CFL bound computed using Eq.(37). In terms of numerical stability, the maximum CFL number of each method decreases slightly from the 2nd- to the 3rd-order accuracy. Interestingly, the critical time-step for SBDF-3 is larger than that of SBDF-2 while the opposite is observed when comparing their CFL bounds. We expect that the critical time step has a similar trend to the CFL bound. This anomaly is explained in the following manner. The CFL bound of SBDF-2 methods must be larger than that of SBDF-3 methods as the stability region of a higher order methods becomes smaller. The maximum CFL should be attained within a same local element regardless of the methods used, provided the flow computed is
sufficiently accurate. By fixing the time steps near their critical values, the SBDF-2 method produces a larger error than SBDF-3 methods, causing the SBDF-2 method to attain its CFL bound in a different element than with the SBDF-3 method. We observe that both GM and GM-SRM methods have the largest maximum CFL bound (about 0.30) which are followed by SBDF and DC methods (about 0.22). Hence, GM/GM-SRM methods are the most stable for this lid-driven cavity flow.

The \(x\)-velocity \(u\) is sampled at three different points near the corners within the domain, i.e., bottom left at \((0.2, 0.3)\), bottom right at \((0.8, 0.3)\) and top right at \((0.8, 0.7)\). Fig. 7 shows the \(x\)-velocity \(u\) sampled over time at

<table>
<thead>
<tr>
<th>Method</th>
<th>Critical time step, (\tau_{\text{crit}})</th>
<th>CFL(_{\text{max}})</th>
<th>CPU time/time step</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBDF-2</td>
<td>(1.45608 \times 10^{-1})</td>
<td>(2.28314 \times 10^{-1})</td>
<td>0.9802</td>
</tr>
<tr>
<td>SBDF-3</td>
<td>(1.57227 \times 10^{-3})</td>
<td>(2.23712 \times 10^{-1})</td>
<td>1.3086</td>
</tr>
<tr>
<td>DC-2</td>
<td>(1.75781 \times 10^{-3})</td>
<td>(2.23344 \times 10^{-1})</td>
<td>1.9948</td>
</tr>
<tr>
<td>DC-3</td>
<td>(1.59180 \times 10^{-3})</td>
<td>(1.95099 \times 10^{-1})</td>
<td>3.6222</td>
</tr>
<tr>
<td>GM-2</td>
<td>(2.58789 \times 10^{-3})</td>
<td>(3.17185 \times 10^{-1})</td>
<td>2.4570</td>
</tr>
<tr>
<td>GM-3</td>
<td>(2.37305 \times 10^{-3})</td>
<td>(2.90853 \times 10^{-1})</td>
<td>4.1908</td>
</tr>
<tr>
<td>GM-SRM-2</td>
<td>(2.59766 \times 10^{-3})</td>
<td>(3.18392 \times 10^{-1})</td>
<td>2.8796</td>
</tr>
<tr>
<td>GM-SRM-3</td>
<td>(2.37305 \times 10^{-3})</td>
<td>(2.90853 \times 10^{-1})</td>
<td>4.8534</td>
</tr>
</tbody>
</table>

(0.2, 0.3) near the bottom left corner for all proposed time-stepping methods. To narrate the behaviour of the flow, we use the time-history of \(x\)-velocity \(u\) computed by DC-3 (in green, top left in Fig. 7). The \(x\)-velocity \(u\) shows three distinct transient phases. The first phase is the fast transient solution starting immediately from the state of rest that occurs for \(t \in [0, 250]\). During this phase, the constant flow along the lid induces a large circulation which quickly transforms into a major vortex at the center of the cavity. Since the advective term is more predominant than diffusion for \(Re = 8500\), the interaction of the center vortex and the walls generates several counter-rotating vortices at the four corners of the domain. The flow is gradually stabilized with well-developed secondary vortices appearing at three corners, i.e., two at bottom right, one at bottom left and another one at top left. Near the end of the first phase, there are several smaller blinking vortices that can be observed, two are formed just below the top left corner and one at the bottom left corner. The total number of vortices generated at this stage varies between 5 and 8. The second phase of the flow takes place for \(t \in [250, 500]\) where the \(x\)-velocity further decreases. During this phase, the blinking vortices at
Figure 7: The x-velocity, $u$ near bottom left corner $(0.2, 0.3)$ for SBDF, SBDF-2* and DC methods (top left graph), GM methods (top right graph), and GM-SRM methods (bottom left graph). A close-up view of x-velocity for all time-stepping methods for $t \in [1700, 1750]$ (bottom right graph).
the bottom left corner can still be observed while the two at the top left corner diminish. The number of vortices generated varies between 5 and 6. During the third phase of flow, the solution experiences a gradual growth of a periodic mode that can be easily observed beginning at $t = 600$. This is because the flow has lost its stability and the mode corresponding to the first Hopf bifurcation is taking over. For $t$ around 1100, the $x$-velocity becomes fully periodic (single mode).

From Fig. 7, we also observe that the total time of the transient solution depends on the time-stepping methods, the order of accuracy and time step used. SBDF-2, SBDF-3, DC-2 and DC-3 methods reach a full periodic solution beginning at $t = 1100$; GM methods around $t = 1150$; and GM-SRM methods around $t = 1000$ which have the shortest transient duration. SBDF-2* attains its full periodic solution beginning at $t = 2200$. In addition to the accuracy gained from the defect correction strategy, the sequential regularization method improves stability in connection with the incompressibility condition, hence reduces the time for transient solution of GM-SRM compared to other methods. In general, methods that produce shorter transient solutions help to reduce the total CPU time to reach time-periodic flows. We also plotted the $x$-velocity $u$, computed with all methods at point $(0.2, 0.3)$ about $t \in [1700, 1750]$. The phase for each $u$ is shifted for better illustration. The graph indicates that SBDF-2 methods may not produce a desirable results if the time step is chosen close to its the critical value.

Fig. 8 illustrates the evolution of the time-periodic flow over one cycle with snapshots of the streamlines every 0.23 time unit (for a period $T_f = 2.22$). During a full cycle, we observe one to two smaller pulsating vortices appearing below the secondary vortex near top left corner. Near the bottom left corner a more complex vortex formation—with as many as 4 pulsating vortices are observed, e.g., at $t = 1.84$. Compared to previous published results, our numerical solution captures two extra smaller vortices as shown in Fig. 9 (at right). The maximal number vortices for a complete cycle is 10. Using our current computational settings, similar streamlines can be reproduced with all methods except SBDF-2 with $\tau = 0.0014$ (SBDF-2*). To some extend, our results contradict with few existing results. For instance, Pan and Glowinski [39] produced only 2 vortices near the bottom left corner, while other appearances of vortex are identical to our results at $Re = 8500$. They used $P_1$-iso-$P_2$ finite elements and Chorin’s projection method for space and time discretizations, respectively on a mesh of size $256 \times 256$ mesh. Kufferman [26] produced results similar to Pan and Glowinski but
Figure 8: Snapshot of streamlines for 2D lid-driven cavity flow ($Re = 8500$) by increment of 0.23 time units over 2.30 time unit, a time interval slightly larger than one period of the flow, $T_f \approx 2.22$. 
using 2nd-order center-difference schemes for space and Crank-Nicolson for time stepping on 128 × 128 mesh. Our results were produced with higher-order methods, both in space and time, which explain the extra flow features observed.

Figure 9: Lid-Driven Cavity: Close-up view of vortices near the top left corner at $t = 1.38$ (left) and bottom left corner at $t = 1.84$ (right). The arrows labelled as (a), (b) and (c) indicate the vortices that were not captured by previous published results.

The quantitative results on $x$-velocity $u$ at various locations in terms of mean, amplitude, period and frequency are summarized in Table 5. It is observed that the accuracy of SBDF-2 is compromised when using a time step near its critical value. On the other hand, SBDF-2 method produces satisfactory result with $\tau = 0.001$, which can be accurate up to 4th decimal point compared to the results computed with 3rd-order SBDF and DC methods. The numerical results computed by DC-2 method are in agreement with the one computed with DC-3 and SBDF-3 using the same time step, $\tau = 0.001$. Both 2nd-order GM and GM-SRM methods with time steps near the critical value produce very close result to that produced by their 3rd-order counterparts with time step, $\tau = 0.001$ (e.g., maximum difference starts at the 3rd decimal point). This suggests that the defect correction algorithm produces an unmatched accuracy even with 2nd-order method at larger time step.

Fig. 10 shows the phase portraits of $x$- and $y$-velocity at bottom left (0.2, 0.3), which is a plot of $u(t)$ versus $u(t + \frac{T_f}{2})$ for $t \in [1700, 1750]$, where $T_f$ is the respective period of the $x$-velocity produced by each of the time-stepping methods. There are three distinct phase plots that can be observed in these diagrams. The first one is obtained with SBDF-2* (Phase Portrait I, PPI) shown as the red ellipse, second one with DC and SBDF methods
Figure 10: Phase plots for $x$- and $y$-velocity, $u$ and $v$ respectively, monitor at three different locations with SBDF, DC, GM and GM-SRM methods (2nd- and 3rd-order).
(PPII) shown as the overlapping pink ellipse and the third one with GM and GM-SRM methods (PPIII) shown as the black ellipse. In this numerical experiment, the phase plot with 2\textsuperscript{nd}-order methods are found to be indistinguishable from their respective 3\textsuperscript{rd}-order methods. Again, SBDF-2\* produce undesirable results since the time step is taken too large. Methods with defect correction strategy allows one to choose larger time step while still obtaining satisfactory results.

The 'offset' between PPII and PPIII at three monitoring points can only be explained by the presence of \textbf{grad}-div term in both GM and GM-SRM methods. GM-SRM methods can provide very accurate solutions for flow problems involving Dirichlet boundary condition as for the Taylor’s vortex flow presented above. Boundary conditions alone may not suffice to explain the difference between PPII and PPIII. Therefore, the question concerning which of PPII and PPIII is more accurate requires further investigations.

Table 6 compares the frequency of the periodic solution that we obtain with results found in the literature. The frequency of all our methods stand between 0.4472 and 0.4501, which is in the range published by Pan and Glowinski [39], i.e., 0.4405-0.4505. Meanwhile, the frequency computed by GM and GM-SRM (i.e., 0.4472 and 0.4473, respectively) are very close to 0.4470, the value found by Auteri et al. [2]. The frequency value by Kufferman [26], might be a little underestimated.

5. Conclusions

We have conducted a thorough numerical assessment and benchmark of several 2\textsuperscript{nd}- and 3\textsuperscript{rd}-order semi-implicit methods, e.g., SBDF, DC, GM and GM-SRM methods using two manufactured solutions and two well-known test cases. So far, these methods are proven to be very robust to compute unsteady laminar flows. With a direct solver, SBDF methods are the most efficient method to compute 2D unsteady flows. However, SBDF methods may depend on proper initialization to produce such remarkable results. On the other hand, DC, GM and GM-SRM methods are self-starting and produce the least error in time for a fixed time step since they are based on the same defect correction method. DC methods require the solution of \(k\) similar saddle point problem to obtain \(k\textsuperscript{th}\)-order of accuracy, which turns out to be less efficient at each time step. GM and GM-SRM methods require smaller memory and CPU time for each linear solve. However, more of these linear systems have to be solved at each time step, i.e., one from the momentum
Table 5: The computed average, amplitude, frequency of the \( u \)-velocity of the periodic flow at points near the three corners: bottom left (0.2, 0.3), bottom right (0.8, 0.3) and top right (0.8, 0.7).

<table>
<thead>
<tr>
<th>Method</th>
<th>Time step, ( \tau )</th>
<th>Mean value</th>
<th>Amplitude</th>
<th>Period (s)</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBDF-2*</td>
<td>( 1.4 \times 10^{-3} )</td>
<td>(-1.80109 \times 10^{-1})</td>
<td>3.63427 ( \times 10^{-3})</td>
<td>2.35392</td>
<td>4.24826 ( \times 10^{-1})</td>
</tr>
<tr>
<td>SBDF-2</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>(-1.89676 \times 10^{-1})</td>
<td>6.36170 ( \times 10^{-3})</td>
<td>2.22166</td>
<td>4.50113 ( \times 10^{-1})</td>
</tr>
<tr>
<td>SBDF-3</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>(-1.89676 \times 10^{-1})</td>
<td>6.36155 ( \times 10^{-3})</td>
<td>2.22167</td>
<td>4.50111 ( \times 10^{-1})</td>
</tr>
<tr>
<td>DC-2</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>(-1.89677 \times 10^{-1})</td>
<td>6.36165 ( \times 10^{-3})</td>
<td>2.22167</td>
<td>4.50113 ( \times 10^{-1})</td>
</tr>
<tr>
<td>DC-3</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>(-1.89676 \times 10^{-1})</td>
<td>6.36157 ( \times 10^{-3})</td>
<td>2.22168</td>
<td>4.50111 ( \times 10^{-1})</td>
</tr>
<tr>
<td>GM-2</td>
<td>( 2.5 \times 10^{-3} )</td>
<td>(-1.88397 \times 10^{-1})</td>
<td>6.44025 ( \times 10^{-3})</td>
<td>2.23585</td>
<td>4.47258 ( \times 10^{-1})</td>
</tr>
<tr>
<td>GM-3</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>(-1.88397 \times 10^{-1})</td>
<td>6.43753 ( \times 10^{-3})</td>
<td>2.23607</td>
<td>4.47212 ( \times 10^{-1})</td>
</tr>
<tr>
<td>GM-SRM-2</td>
<td>( 2.5 \times 10^{-3} )</td>
<td>(-1.88406 \times 10^{-1})</td>
<td>6.41144 ( \times 10^{-3})</td>
<td>2.23569</td>
<td>4.47289 ( \times 10^{-1})</td>
</tr>
<tr>
<td>GM-SRM-3</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>(-1.88404 \times 10^{-1})</td>
<td>6.41387 ( \times 10^{-3})</td>
<td>2.23615</td>
<td>4.47198 ( \times 10^{-1})</td>
</tr>
</tbody>
</table>

Table 6: Comparison between our work with numerical values from the literature for frequencies of the lid-driven cavity flow at \( Re = 8500 \)

<table>
<thead>
<tr>
<th>Source of results</th>
<th>Time-stepping schemes</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>SBDF, DC, GM, GM-SRM</td>
<td>0.4501, 0.4501, 0.4472, 0.4473</td>
</tr>
<tr>
<td>Auerer et. al. (2002) [2]</td>
<td>2\textsuperscript{nd}, order projection</td>
<td>0.4470</td>
</tr>
<tr>
<td>Kupferman (2001) [26]</td>
<td>2\textsuperscript{nd}, order CN-midpoint rule</td>
<td>0.4900</td>
</tr>
<tr>
<td>Pan &amp; Glowinski (2000) [40]</td>
<td>2\textsuperscript{nd}, order projection</td>
<td>0.4405 – 0.4505</td>
</tr>
</tbody>
</table>

43
and one from the continuity equation, and this \( k \) times to reach \( k^{\text{th}} \)-order of accuracy. GM and GM-SRM methods have good potential to compute unsteady flows involving millions of unknowns (e.g., 3D flows) since the linear systems with symmetric positive definite matrices can be easily handled by any efficient iterative solvers such as the conjugate gradient method.

In terms of stability, GM and GM-SRM methods are more stable than SBDF and DC methods due to the presence of a \( \text{grad-div} \) stabilization term. The combination of \( \text{grad-div} \) term and defect correction strategy allows us to choose larger time step with GM and GM-SRM methods while still producing good results. The \( \text{grad-div} \) stabilization term in GM and GM-SRM helps to enforce incompressibility condition with \( \mathbb{P}_2-\mathbb{P}_1 \) elements, thus reducing even more the error in space. This property has been demonstrated by GM-SRM methods with a nearly ‘optimal’ choice of the stabilization parameters when computing the manufactured solution II. This was attempted with GM methods with much smaller \( \lambda \) (e.g. \( \lambda < 3.3 \)) but it resulted in more oscillations at start-up before they finally damp at larger \( t \) to produce similar accuracy to that with GM-SRM methods (result not shown here). We mention that the optimal choice of the stabilization parameters, \( \lambda \) in GM methods and \( \alpha_1, \alpha_2 \) in GM-SRM methods is difficult to obtain to produce the least error in space on a fixed mesh. The analysis of these stabilization terms in terms of the errors induced on velocity and pressure has not been done yet for unsteady flows. The optimal choice of these stabilization parameters \( \lambda \) and \( \alpha \) for Navier–Stokes equations is still an open problem.

In the last two test cases, we observed that the use of higher-order semi-implicit methods with reasonably small time step is crucial for highly nonlinear flows (large Reynolds number). Besides reproducing results from the literature, e.g., the lift and drag coefficients along the cylinder and pulsating frequency, Strouhal number in both flows, we managed to capture additional vortices in the lid-driven cavity at \( Re = 8500 \) with all our methods. To the best of our knowledge, this finding has not been reported in the recent literature. This indicates that higher-order time-stepping methods with smaller time step are indispensable to obtain very accurate solutions for inertia-dominant flows.

6. Acknowledgement

We would like to express our gratitute for funding given by the NSERC Discovery Grant program to conduct this research. The first author, Kak
Choon, Loy is a recipient of Ph.D scholarship granted by Universiti Malaysia Terengganu and The Ministry of Education Malaysia, for this we deeply acknowledge. Thanks to the resources of High Performance Computing Virtual Laboratory (HPCVL) at Queen’s University and the FreeFEM++ development team headed by Professor Frédéric Hecht at the Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie. Last but not least, we indebted to Peter Minev and Jean-Luc Guermond for their willingness to share their submitted manuscript, long before their work is finally published online in November 12, 2015 [21].

Bibliography

References


