SIAM J. SCI. COMPUT.
Vol. 37, No. 6, pp. A2656–A2681

HIGH-ORDER TIME STEPPING FOR THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS*

JEAN-LUC GUERMOND† AND PETER MINEV‡

Abstract. This paper introduces a high-order time stepping technique for solving the incompressible Navier–Stokes equations which, unlike coupled techniques, does not require solving a saddle point problem at each time step and, unlike projection methods, does not produce splitting errors and spurious boundary layers. The technique is a generalization of the artificial compressibility method; it is unconditionally stable (for the unsteady Stokes equations), can reach any order in time, and uncouples the velocity and the pressure. The condition number of the linear systems associated with the fully discrete vector-valued problems to be solved at each time step scales like $O(\tau h^{-2})$, where τ is the time step and h is the spatial grid size. No Poisson problem or other second-order elliptic problem has to be solved for the pressure corrections. Unlike projection methods, optimal convergence is observed numerically with Dirichlet and mixed Dirichlet/Neumann boundary conditions.

Key words. Navier–Stokes, incompressibility, artificial compressibility, high-order time approximation, defect correction

AMS subject classifications. 65N12, 65N15, 35Q30

DOI. 10.1137/140975231

1. Introduction.

The problematic and key notation are introduced in this section.

1.1. Objectives of the paper.

The approximation in time of the unsteady Stokes and Navier–Stokes equations is often done by using fractional time stepping techniques à la Chorin–Temam (a.k.a. projection methods). These methods uncouple the velocity and the pressure in a way that avoids solving a saddle point problem at each time step. Denoting by τ the time step, the workload per time step consists of solving one vector-valued problem like $\mathbf{v} - \tau\nu\Delta\mathbf{v} = \mathbf{r}$ (which is associated with the momentum equation) and one scalar Poisson equation $-\Delta\phi = g$ (which is associated with the pressure correction). Projection methods can achieve second-order in time on the velocity in the $\mathbf{L}^2$-norm and $\frac{3}{2}$ order on the pressure in the L^2-norm and the velocity in the $\mathbf{H}^1$-norm; see, e.g., Guermond and Shen [14], Timmermans, Minev, and van de Vosse [36]. Many projection methods have been proposed over the years. Two particular variants having the above convergence properties are the so-called incremental pressure-correction (see, e.g., Kim and Moin [21], Timmermans, Minev, and van de Vosse [36]) and the velocity-correction methods in rotational form (see, e.g., Orszag, Israeli, and Deville [27], Karniadakis, Israeli, and Orszag [20]). A third variant, called either consistent splitting or gauge method, is often used in the

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 30, 2014; accepted for publication (in revised form) August 12, 2015; published electronically November 12, 2015. This material is based upon work supported by the National Science Foundation grants DMS-1015984, DMS-1217262, by the Air Force Office of Scientific Research, USAF, under grant/contract FA99550-12-0358, and a Discovery grant of the National Science and Engineering Research Council of Canada.

http://www.siam.org/journals/sisc/37-6/97523.html

†Department of Mathematics, Texas A&M University 3368 TAMU, College Station, TX 77843 (guermond@math.tamu.edu).

‡Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada T6G 2G1 (minev@ualberta.ca).

literature (see, e.g., Guermond and Shen [13], E and Liu [9]). We refer to Guermond, Minev, and Shen [16] and the references therein for a review on projection methods.

Projection and projection-like fractional time stepping techniques have two limitations: (i) they require solving a Poisson problem at each time step; (ii) they seemingly cannot exceed second-order accuracy in time without losing unconditional stability. Denoting by h the mesh size of the space approximation, the linear system associated with the Poisson equation has a condition number that grows like $\mathcal{O}(h^{-2})$. As a result, solving the Poisson equation may become a bottleneck when the mesh size is very small, in particular, when it is done on massively parallel computers, unless special strategies like that proposed in Guermond and Minev [12] are adopted. In addition to the $\mathcal{O}(h^{-2})$ computational complexity induced by the Laplace operator, projection and projection-like fractional time stepping techniques seem to be facing an order barrier. Actually, to the best of our knowledge, no higher-order extension with provable unconditional stability (for the unsteady Stokes problem) and provable high-order convergence rate has yet been proposed in the literature. For instance, the $\mathcal{O}(\tau^{\frac{3}{2}})$ convergence rate mentioned above on the pressure in the L^2-norm and the velocity in the $\mathbf{H}^1$-norm has never been improved in general domains. Full second order has been established only in channel flows with periodic boundary conditions in the transverse directions (Brown, Cortez, and Minion [7]). This order barrier seems to be deeply rooted in an observation made by Shen [31] where it is argued that no unconditionally stable third-order pressure-correction method can be constructed by using a second-order extrapolation of the pressure.

The objectives of this paper is to introduce a technique that overcomes the two limitations mentioned above. The key idea is to bootstrap a first-order artificial compressibility method. The proposed method is unconditionally stable, avoids solving a saddle point problem at each time step, does not enforce any artificial boundary condition on the pressure, and can achieve high-order accuracy in time. The algorithm requires solving a fixed number of vector-valued PDEs like $\mathbf{v} - \tau\,(\nu\Delta\mathbf{v} + \nabla\nabla{\cdot}\mathbf{v}) = \mathbf{r}$ per time step; the number of problems to be solved depends solely on the order required. The algorithmic complexity is similar to that of solving a sequence of parabolic problems. Assuming $\tau \sim h\mathcal{V}^{-1}$, where $\mathcal{V}$ is a velocity scale, each time step involves solving symmetric positive definite linear systems whose condition numbers scale like $\mathcal{O}((\mathcal{V}h)^{-1})$. Thus, the solution of the unsteady Stokes equations can be reduced to solving a set of parabolic equations only. Finally, the method is observed to be stable for the Navier–Stokes problem under a standard CFL condition $\mathcal{V}\tau \lesssim h$. Existing codes can be easily modified to employ this procedure. Although each of the elementary components of the method are well known (for instance, the benefit of using the $-\nabla\nabla{\cdot}\mathbf{v}$ stabilization has already been recognized in the literature; see, e.g., Angot, Jobelin, and Latché [4], Jobelin et al. [19], Olshanskii et al. [26], and Heister and Rapin [17]), the algorithms proposed below seem to be new to the best of our knowledge.

The paper is organized as follows. We briefly review artificial compressibility methods and recall some theoretical results in section 2. The second-order barrier effect identified by Shen [31] is recalled and investigated in section 3; it is shown in this section that the barrier can be overcome by using subtle regularization effects induced by the time discretization. The new method, based on a bootstrapping technique, is introduced in section 4; the key results of this section are Lemma 4.4 and Theorem 4.5. Some examples of second- and third-order variants of the method are introduced and analyzed in section 5. The third-order schemes (5.3)–(5.5) and (5.19)–(5.21) are new, to the best of our knowledge. Finally, the various methods considered in this paper are

compared in section 6 using manufactured solutions of the unsteady Stokes and the Navier–Stokes equations. The third-order accuracy in time of the schemes (5.3)–(5.5) and (5.19)–(5.21) is illustrated numerically on problems supplemented with Dirichlet and Neumann boundary conditions. Concluding remarks are reported in section 7.

1.2. Model problem and notation. We consider the incompressible Stokes equations written in terms of velocity $\mathbf{u}$ and pressure p on a finite time interval $[0,T]$ and in a domain $\Omega \subset \mathbb{R}^d$ with $d = 2$ or 3:

$$\begin{cases} \partial_t \mathbf{u} + A\mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega \times (0,T], \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega \times [0,T], \\ \mathbf{u}|_\Gamma = 0 & \text{in } (0,T], \\ \mathbf{u}|_{t=0} = \mathbf{u}_0 & \text{in } \Omega, \end{cases} \tag{1.1}$$

where $\mathbf{f}$ is a smooth source term, $\mathbf{u}_0$ is a solenoidal initial velocity field with zero normal trace, and Γ denotes the boundary of Ω. The operator A is assumed to be linear, $\mathbf{H}^1$-coercive, and bounded, i.e., there are two constants $\nu > 0$ and $M < \infty$ such that $\int_\Omega A\mathbf{u}\cdot\mathbf{u}\,\mathrm{d}\mathbf{x} \geq \nu\|\mathbf{u}\|^2_{\mathbf{H}^1}$ and $|\int_\Omega A\mathbf{u}\cdot\mathbf{v}\,\mathrm{d}\mathbf{x}| \leq M\|\mathbf{u}\|_{\mathbf{H}^1}\|\mathbf{v}\|_{\mathbf{H}^1}$ for all $\mathbf{u},\mathbf{v} \in \mathbf{H}^1_0(\Omega)$. Denoting by $(u_1,\dots,u_d)$ the Cartesian components of $\mathbf{u}$, we have defined $\|\mathbf{u}\|^2_{\mathbf{L}^2} := \int_\Omega \sum_{i=1}^d u_i^2\,\mathrm{d}\mathbf{x}$ and $\|\nabla\mathbf{u}\|^2_{\mathbf{L}^2} := \int_\Omega \sum_{i,j=1}^d (\partial_j u_i)^2\,\mathrm{d}\mathbf{x}$, $\|\mathbf{u}\|_{\mathbf{H}^1} := \|\nabla\mathbf{u}\|_{\mathbf{L}^2}$. We will often make use of the Poincaré inequality, $\|\nabla\mathbf{u}\|_{\mathbf{L}^2} \geq c_P\|\mathbf{u}\|_{\mathbf{L}^2}$, without mentioning it. Typical examples for the operator A are as follows: $A\mathbf{u} = -\nu\Delta\mathbf{u}$ for Newtonian fluids, or $A\mathbf{u} = -\nu\nabla\cdot(\nabla^s\mathbf{u}) - \lambda\nabla\nabla\cdot\mathbf{u}$ when one wants to impose boundary conditions that depend on the stresses at the boundary; the tensor $\nabla^s\mathbf{u} := \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^\mathsf{T})$ is the symmetric gradient, or so-called strain rate tensor. The nonlinear term in the momentum equation of the Navier–Stokes equations is not accounted for, since it does not interfere with the incompressibility constraint provided enough smoothness is assumed. The fluid density is assumed to be constant and is hidden in the normalization constants. To account for the fact that the pressure is defined up to a constant, we define $L^2_0(\Omega)$ to be the set composed of the members of $L^2(\Omega)$ with zero average over Ω. We denote by c a generic constant that is independent of the discretization parameter τ and the perturbation parameters ϵ, δ (yet to be defined) but possibly depends on the data, the domain, and the solution. The value of c may change at each occurrence.

2. Artificial compressibility methods. The velocity-pressure coupling in the incompressible Stokes or Navier–Stokes equations makes the numerical solution of these equations somewhat more challenging than that of parabolic equations. In the steady case, this conundrum is usually dealt with iteratively, the simplest iterative method being the Uzawa iteration (see Glowinski [11, chapter IV] for a comprehensive discussion on iterative methods). In the unsteady case, however, it is possible to decouple the velocity and the pressure by a regularization of the incompressibility constraint. Below we briefly review essential features of these methods. The reader is referred to Glowinski [11, section 23] for a more comprehensive discussion. None of what is said in this section is new, but this material is useful to better understand the following sections.

2.1. Penalty method. One of the first artificial compressibility methods introduced in the literature is that proposed by Temam [34]. The idea consists of introducing a penalty parameter $\epsilon > 0$ and replacing the incompressibility constraint $\nabla\cdot\mathbf{u} = 0$ by $\epsilon p + \nabla\cdot\mathbf{u} = 0$. The perturbed system can then be rewritten in the

following form:

$$\begin{cases} \partial_t \mathbf{u}_\epsilon + A\mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f} & \text{in } \Omega \times (0,T], \\ \epsilon p_\epsilon + \nabla\cdot\mathbf{u}_\epsilon = 0 & \text{in } \Omega \times [0,T], \\ \mathbf{u}_\epsilon|_\Gamma = 0 & \text{in } (0,T], \\ \mathbf{u}_\epsilon|_{t=0} = \mathbf{u}_0 & \text{in } \Omega. \end{cases} \tag{2.1}$$

Note that the pressure can be eliminated from the momentum equation as follows: $\partial_t \mathbf{u}_\epsilon + A\mathbf{u}_\epsilon - \frac{1}{\epsilon}\nabla\nabla\cdot\mathbf{u}_\epsilon = \mathbf{f}$. The pressure is recovered as a by-product by setting $p_\epsilon = \epsilon^{-1}\nabla\cdot\mathbf{u}_\epsilon$. The regularized problem can be easily approximated in time by using, for instance, the implicit Euler method. Then, denoting τ as the time step, this technique requires solving $\mathbf{u}^{n+1} + \tau A\mathbf{u}^{n+1} - \frac{\tau}{\epsilon}\nabla\nabla\cdot\mathbf{u}^{n+1} = \mathbf{u}^n + \tau\mathbf{f}^{n+1}$ at each time step $t^{n+1} := t^n + \tau$, where $\mathbf{f}^{n+1} := \mathbf{f}(t^{n+1})$. At least two objections can be raised against this method: (i) the first one is that one must set $\epsilon = \mathcal{O}(\tau)$ to make it first-order accurate in time. For instance, assuming that $\epsilon = \tau$, one has to solve a PDE of the following form $\mathbf{v} - \nabla\nabla\cdot\mathbf{v} + \text{l.o.t.} = \mathbf{g}$ at each time step. The condition number associated with the discrete variant of the symmetric positive definite operator $I - \nabla\nabla\cdot$ scales like $\mathcal{O}(h^{-2})$. This does not compare favorably with the alternative method presented in the next section which is also first-order accurate but whose discrete counterpart yields a linear system with a condition number scaling like $\mathcal{O}(h^{-1})$; (ii) the second objection is that one needs to choose $\epsilon = \tau^l$ to make the accuracy order of the method $\mathcal{O}(\tau^l)$; this then yields linear systems with condition numbers that behave like $\mathcal{O}(\tau^{1-l}h^{-2})$. This type of growth is far too large to be handled efficiently by iterative methods when $l \geq 2$.

2.2. Artificial compressibility methods. An alternative to the penalty technique consists of replacing the constraint $\nabla\cdot\mathbf{u} = 0$ by $\epsilon\partial_t p + \nabla\cdot\mathbf{u} = 0$. This perturbation to incompressibility can be deduced from the compressible mass conservation by taking the low Mach number limit with $\epsilon = \mathcal{M}^{-2}$, where $\mathcal{M}$ is the Mach number.

The artificial compressibility regularization can be traced back in the Russian literature to the group of Vladimirova, Kuznetsov, and Yanenko [37] and Yanenko [38, section 8.2]. For instance the following two regularizations can be found in the above references:

(2.2)

$$\begin{cases} \partial_t \mathbf{u}_\epsilon + (\mathbf{u}_\epsilon\cdot\nabla)\mathbf{u}_\epsilon - \Delta\mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f}, \\ \epsilon(\partial_t p_\epsilon + (\mathbf{u}_\epsilon\cdot\nabla)p_\epsilon) + p_\epsilon\nabla\cdot\mathbf{u}_\epsilon = 0, \end{cases} \qquad \begin{cases} \partial_t \mathbf{u}_\epsilon + (\nabla\times\mathbf{u}_\epsilon)\times\mathbf{u}_\epsilon - \Delta\mathbf{u}_\epsilon + \nabla q_\epsilon = \mathbf{f}, \\ \epsilon\partial_t q_\epsilon + \nabla\cdot\mathbf{u}_\epsilon = 0, \end{cases}$$

where $q_\epsilon = p_\epsilon + \frac{1}{2}\mathbf{u}^2$, and these equations are discretized by a direction splitting procedure. At about the same time, the following regularization has also been proposed by Chorin [8], Temam [35] and Ladyzhenskaya [23, section 9, Chap. VI]:

$$\begin{cases} \partial_t \mathbf{u}_\epsilon + (\mathbf{u}_\epsilon\cdot\nabla)\mathbf{u}_\epsilon - \Delta\mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f}, \\ \epsilon\partial_t p_\epsilon + \nabla\cdot\mathbf{u}_\epsilon = 0. \end{cases} \tag{2.3}$$

However, Chorin [8] discretizes the artificial compressibility equation explicitly, and this imposes very severe time step restrictions of the type $\tau \sim h\epsilon^{\frac{1}{2}}$ rendering the method impractical. To overcome this limitation, Temam [35] and Ladyzhenskaya [23] propose various direction splitting schemes for this formulation.

We now turn our attention to the approximation properties of the regularized system (2.3), omitting the nonlinear terms. Let p_0 be the pressure at the initial time, i.e., $\Delta p_0 = \nabla\cdot\mathbf{f}(0)$, $\partial_n p_0 = (\mathbf{f}(0) - A\mathbf{u}_0)\cdot\mathbf{n}|_\Gamma$. Let $\epsilon > 0$ and consider the following

perturbation of (1.1):

$$\begin{cases} \partial_t \mathbf{u}_\epsilon + A\mathbf{u}_\epsilon + \nabla p_\epsilon = \mathbf{f} & \text{in } \Omega \times (0,T], \\ \epsilon \partial_t p_\epsilon + \nabla\cdot \mathbf{u}_\epsilon = 0 & \text{in } \Omega \times [0,T], \\ \mathbf{u}_\epsilon|_\Gamma = 0 & \text{in } (0,T], \\ \mathbf{u}_\epsilon|_{t=0} = \mathbf{u}_0, \quad p_\epsilon|_{t=0} = p_0 & \text{in } \Omega. \end{cases} \tag{2.4}$$

PROPOSITION 2.1. *The PDE system* (2.4) *is stable uniformly with respect to* ϵ. *More precisely,*

$$\|\partial_t \mathbf{u}_\epsilon\|^2_{L^\infty((0,T);\mathbf{L}^2)} + \|\partial_t \mathbf{u}_\epsilon\|^2_{L^2((0,T);\mathbf{H}^1)} + \epsilon^{-1}\|\nabla\cdot\mathbf{u}_\epsilon\|^2_{L^\infty((0,T);\mathbf{L}^2)} \\ \le \|\mathbf{f} - A\mathbf{u}_0 - \nabla p_0\|^2_{\mathbf{L}^2} + \nu^{-1}\|\partial_t \mathbf{f}\|^2_{L^2((0,T);\mathbf{H}^{-1})}. \tag{2.5}$$

Proof. This result is well known, but we reproduce the proof for completeness. Taking one time derivative of the momentum equation gives $\partial_{tt}\mathbf{u}_\epsilon + A\partial_t\mathbf{u}_\epsilon + \nabla\partial_t p_\epsilon = \partial_t \mathbf{f}$, which upon using $\epsilon\partial_t p_\epsilon = -\nabla\cdot\mathbf{u}_\epsilon$, reduces $\partial_{tt}\mathbf{u}_\epsilon + A\partial_t\mathbf{u}_\epsilon - \frac{1}{\epsilon}\nabla\nabla\cdot\mathbf{u}_\epsilon = \partial_t\mathbf{f}$. Testing this equation with $\partial_t\mathbf{u}_\epsilon$ gives

$$\frac{1}{2}\partial_t\|\partial_t\mathbf{u}_\epsilon\|^2_{\mathbf{L}^2} + \nu\|\partial_t\mathbf{u}_\epsilon\|^2_{\mathbf{H}^1} + \frac{1}{2\epsilon}\partial_t\|\nabla\cdot\mathbf{u}_\epsilon\|^2_{\mathbf{L}^2} \le \frac{1}{2\nu}\|\partial_t\mathbf{f}\|^2_{\mathbf{H}^{-1}} + \frac{\nu}{2}\|\partial_t\mathbf{u}_\epsilon\|^2_{\mathbf{H}^1}.$$

Then

$$\|\partial_t \mathbf{u}_\epsilon\|^2_{L^\infty((0,T);\mathbf{L}^2)} + \|\partial_t \mathbf{u}_\epsilon\|^2_{L^2((0,T);\mathbf{H}^1)} + \epsilon^{-1}\|\nabla\cdot\mathbf{u}_\epsilon\|^2_{L^\infty((0,T);\mathbf{L}^2)} \\ \le \|\partial_t\mathbf{u}(0)\|^2_{\mathbf{L}^2} + \nu^{-1}\|\partial_t \mathbf{f}\|^2_{L^2((0,T);\mathbf{H}^{-1})}.$$

Conclude by observing that $\partial_t\mathbf{u}_\epsilon(0) = \mathbf{f}(0) - A\mathbf{u}_0 - \nabla p_0$. ∎

Remark 2.1. L^2-stability of (2.4) can also be established by testing the momentum equation with $\mathbf{u}$, and one then obtains $\|\mathbf{u}_\epsilon\|^2_{L^\infty((0,T);\mathbf{L}^2)} + \nu\|\mathbf{u}_\epsilon\|^2_{L^2((0,T);\mathbf{H}^1)} \le c(\|\mathbf{f}\|^2_{L^2((0,T);\mathbf{H}^{-1})} + \|\mathbf{u}_0\|^2_{\mathbf{H}^1})$. One interest of Proposition 2.1 is to give the straightforward estimate $\|\nabla\cdot\mathbf{u}_\epsilon\|_{L^\infty((0,T);\mathbf{L}^2)} \le c\epsilon^{\frac{1}{2}}$, which shows that the method is at least $\mathcal{O}(\epsilon^{\frac{1}{2}})$ accurate. It is established in Shen [32] that the method is actually $\mathcal{O}(\epsilon)$. The argument is revisited in more detail in section 4.

Remark 2.2. Note that the perturbation introduced in the second equation of (2.4) transforms the Stokes problem into a wave equation $\partial_{tt}\mathbf{u} + A\partial_t\mathbf{u} - \frac{1}{\epsilon}\nabla\nabla\cdot\mathbf{u} = \partial_t\mathbf{f}$, where the wave speed is $\epsilon^{-\frac{1}{2}}$. One can then imagine approximating this problem using explicit methods that are known to be conditionally stable under the CFL condition $\epsilon^{-\frac{1}{2}}\tau h^{-1} \le 1$. Assuming $\epsilon = \tau$ to be formally first-order accurate, this yields the stability condition $\tau \le h^2$, which is unrealistic. This reasoning shows that one needs to be implicit to be at least first-order accurate.

2.3. First-order approximation method. In this section we construct a first-order approximation in time of the artificial compressibility method (2.4). Let τ be the time step. The method is initialized by setting $\mathbf{u}^0 = \mathbf{u}_0$ and $p^0 = p(0)$. Then the new update at time t^{n+1} is computed by solving

$$\begin{cases} \dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\tau} + A\mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1}, \quad \mathbf{u}^{n+1}|_\Gamma = 0, \\ \dfrac{\epsilon}{\tau}(p^{n+1} - p^n) + \nabla\cdot\mathbf{u}^{n+1} = 0. \end{cases} \tag{2.6}$$

Note that the velocity and the pressure are still coupled. This coupling can be untangled by substituting p^{n+1} from the second equation of (2.6) into the first one as follows:

$$\begin{cases} \dfrac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\tau} + A\mathbf{u}^{n+1} - \dfrac{\tau}{\epsilon}\nabla\nabla\cdot\mathbf{u}^{n+1} + \nabla p^n = \mathbf{f}^{n+1}, \quad \mathbf{u}^{n+1}|_\Gamma = 0, \\ p^{n+1} = p^n - \dfrac{\tau}{\epsilon}\nabla\cdot\mathbf{u}^{n+1}. \end{cases} \tag{2.7}$$

If $\epsilon \sim \tau$, the method requires solving at each time step a problem of the type $\mathbf{v} + \tau(A\mathbf{v} - \nabla\nabla\cdot\mathbf{v}) = \mathbf{r}$ whose condition number scales like $O(\tau h^{-2})$. The computational complexity for solving this problem is similar to that for solving a parabolic problem implicitly. Note also that the condition number scales like $\mathcal{O}(\mathcal{V}^{-1}h^{-1})$ in the CFL regime $\mathcal{V}\tau \lesssim h$. The above scheme is analyzed in Shen [32] where it is shown that setting $\epsilon \sim \tau$ yields first-order accuracy. More precisely, we have the following.

PROPOSITION 2.2 (Shen [32, proposition 5.1]). *The following estimate holds under reasonable regularity assumptions of the data and the exact solution to* (1.1)*:*

$$\|\mathbf{u}(t^m) - \mathbf{u}^m\|^2_{\mathbf{L}^2(\Omega)} + \nu\tau\sum_{l=0}^{m}\|\mathbf{u}(t^l) - \mathbf{u}^l\|^2_{\mathbf{H}^1(\Omega)} \le c(\tau^2+\epsilon^2) \quad \forall t^m,\ m \in \{0,\dots,K\}. \tag{2.8}$$

In conclusion, the artificial compressibility method is first-order accurate both in the $\mathbf{L}^2$- and the $\mathbf{H}^1$-norm of the velocity, and its computational cost is similar to that of solving a vector-valued parabolic equation implicitly. Recall that the classical nonincremental projection method is first-order accurate in the $\mathbf{L}^2$-norm of the velocity also, but it is only half-order in the $\mathbf{H}^1$-norm due to spurious boundary effects. Note finally that projection methods require solving a Poisson equation at each time step, which is obviously more expensive than solving a parabolic equation. These observations show that (i) the artificial compressibility method is more efficient than the nonincremental projection method; (ii) the artificial compressibility method may not have been given all the attention it deserves in the literature.

2.4. Comparison with the vector penalty projection method. The artificial compressibility method has been revived lately in a series of papers by Angot, Caltagirone, and Fabrie [3, 2] where the authors introduce the so-called vector penalty projection method. The authors propose to approximate the velocity by using two sequences $\widetilde{\mathbf{u}}^0,\dots,\widetilde{\mathbf{u}}^n$ and $\widehat{\mathbf{u}}^0,\dots,\widehat{\mathbf{u}}^n$ and to approximate the pressure by using a sequence $p^0,\dots,p^n$ (see Angot, Caltagirone, and Fabrie [3, (22)–(23)]. After reasonably initializing the algorithm, the new fields $\widetilde{\mathbf{u}}^{n+1}$, $\widehat{\mathbf{u}}^{n+1}$, and p^{n+1} are obtained by solving

$$\frac{\widetilde{\mathbf{u}}^{n+1}-\widetilde{\mathbf{u}}^n}{\tau} + A\widetilde{\mathbf{u}}^{n+1} - r_0\nabla\nabla\cdot\widetilde{\mathbf{u}}^{n+1} + \nabla p^n = \mathbf{f}^{n+1}, \quad \widetilde{\mathbf{u}}^{n+1}|_\Gamma = 0, \tag{2.9}$$

$$\frac{\widehat{\mathbf{u}}^{n+1}-\widehat{\mathbf{u}}^n}{\tau} + A\widehat{\mathbf{u}}^{n+1} - \frac{1}{\delta}\nabla\nabla\cdot\widehat{\mathbf{u}}^{n+1} = \frac{1}{\delta}\nabla\nabla\cdot\widetilde{\mathbf{u}}^{n+1}, \quad \widehat{\mathbf{u}}^{n+1}|_\Gamma = 0, \tag{2.10}$$

$$\mathbf{u}^{n+1} = \widetilde{\mathbf{u}}^{n+1} + \widehat{\mathbf{u}}^{n+1}, \qquad p^{n+1} = p^n - r_0\nabla\cdot\widetilde{\mathbf{u}}^{n+1} - \frac{1}{\delta}\nabla\cdot\mathbf{u}^{n+1}, \tag{2.11}$$

where $r_0 \ge 0$ is a nonnegative parameter that can be set to zero and $\delta > 0$ is a penalty parameter. The system (2.9)–(2.11) looks a bit odd, but, up to the nonessential parameter r_0, it is simply (2.7) rewritten differently. Actually, setting $r_0 = 0$, summing

(2.9) and (2.10), and using the definition $\mathbf{u}^{n+1} = \widetilde{\mathbf{u}}^{n+1} + \widehat{\mathbf{u}}^{n+1}$ gives

$$\begin{cases} \dfrac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\tau} + A\mathbf{u}^{n+1} - \dfrac{1}{\delta}\nabla\nabla\cdot\mathbf{u}^{n+1} + \nabla p^n = \mathbf{f}^{n+1}, \quad \mathbf{u}^{n+1}|_\Gamma = 0, \\ p^{n+1} = p^n - \dfrac{1}{\delta}\nabla\cdot\mathbf{u}^{n+1}, \end{cases} \tag{2.12}$$

which is exactly (2.7). Hence, the vector penalty projection method is identical to the artificial compressibility method with the choice of parameter $\delta = \frac{\epsilon}{\tau}$. As a byproduct of this observation and upon invoking Proposition 2.2, we conclude that the vector penalty projection method is first-order accurate when $\delta = 1$ (which, in passing, improves by a factor $\tau^{\frac{1}{2}}$ the estimate (i) in Theorem 2.4 in Angot, Caltagirone, and Fabrie [3]).

3. Order barrier. Before moving to the main result of the paper which is stated in section 4, we elaborate on an order barrier identified by Shen [31] and we show that this barrier can be overcome.

3.1. The PDE argument. One could, in principle, think of augmenting the order of the artificial compressibility method by increasing the order of the time derivative of the pressure in the perturbation of the mass conservation equation. For instance a second-order perturbation can be constructed as follows:

$$\begin{cases} \dfrac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\tau} + A\mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1}, \quad \mathbf{u}^{n+1}|_\Gamma = 0, \\ \dfrac{\epsilon}{\tau^2}(p^{n+1} - 2p^n + p^{n-1}) + \nabla\cdot\mathbf{u}^{n+1} = 0. \end{cases} \tag{3.1}$$

Upon setting $\epsilon = \tau^2$ and after eliminating the pressure from the momentum equation, one is led to solve a problem of the following type $\mathbf{v} + \tau(A\mathbf{v} - \nabla\nabla\cdot\mathbf{v}) = \mathbf{r}$ at each time step. The condition number of the fully discrete counterpart of this problem scales like $O(\tau h^{-2})$, i.e., it is not more expensive than solving a parabolic problem implicitly. Note also that since the perturbation to incompressibility is $\mathcal{O}(\epsilon) = \mathcal{O}(\tau^2)$, the resulting algorithm (if stable) is formally second-order accurate in time. Similar strategies have been proposed in the literature to tentatively increase the accuracy of the incremental pressure-correction methods either in rotational or standard form. Unfortunately, to the best of our knowledge, none of these methods have been proved to be unconditionally stable so far.

To explain why second-order extrapolations of the pressure cannot yield a stable algorithm, Shen [31] proposes considering the following formal limit of (3.1):

$$\begin{cases} \partial_t\mathbf{u} + A\mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega\times(0,T], \\ \epsilon\partial_{tt}p + \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega\times[0,T], \\ \mathbf{u}|_\Gamma = 0 & \text{in } (0,T], \\ \mathbf{u}|_{t=0} = \mathbf{u}_0 \quad p(0) = p_0, \quad \partial_t p(0) = \partial_t p(0) & \text{in } \Omega. \end{cases} \tag{3.2}$$

This perturbation is indeed unstable. More precisely we have the following result.

PROPOSITION 3.1 (Shen [31]). *The PDE system* (3.2) *is unstable.*

Proof. We repeat the argument from Shen [31] for completeness. We simplify the analysis by assuming that $A = -\nu\Delta$ and the boundary conditions are periodic. Taking two time derivatives in the momentum equation, we have $\partial_{ttt}\mathbf{u} - \nu\Delta\partial_{tt}\mathbf{u} + \nabla\partial_{tt}p =$

$\partial_{tt}\mathbf{f}$, which, upon using the perturbation equation $\epsilon\partial_{tt}p + \nabla\cdot\mathbf{u} = 0$, and applying the divergence operator, reduces to

$$\partial_{ttt}\psi - \nu\Delta\partial_{tt}\psi - \frac{1}{\epsilon}\Delta\psi = \partial_{tt}d,$$

where we have set $\psi := \nabla\cdot\mathbf{u}$ and $d := \nabla\cdot\mathbf{f}$. Let $\{(\lambda_n, \varphi_n)\}_{n\in\mathbb{N}}$ be a Hilbertian basis of $\mathbf{L}^2(\Omega)$ composed of eigenpairs of $-\Delta$ equipped with periodic boundary condition. Upon expanding ψ and d with respect to this basis, i.e., $\psi(t,\mathbf{x}) = \sum_{n\in\mathbb{N}} x_n(t)\varphi_n(\mathbf{x})$, $d(t,\mathbf{x}) = \sum_{n\in\mathbb{N}} d_n(t)\varphi_n(\mathbf{x})$, we infer that the following holds for all $n \in \mathbb{N}$:

$$\partial_{ttt}x_n(t) + \nu\lambda_n\partial_{tt}x_n(t) + \frac{\lambda_n}{\epsilon}x_n(t) = \partial_{tt}d_n(t).$$

This is a linear ODE whose characteristic polynomial is $q(x) := \frac{\epsilon}{\lambda_n}x^3 + \epsilon\nu x^2 + 1$. The roots, x_1, x_2, x_3, are such that $x_2x_3 + x_3x_1 + x_1x_2 = 0$ (Viète's formula), which after dividing by $x_1x_2x_3$ implies that $\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} = 0$. This means that there must be a positive root if all the roots are real. If not, there are two conjugate roots $x_\pm = a \pm ib$ and one real root x_1. Then $\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} = \frac{1}{x_1} + \frac{2a}{a^2+b^2} = 0$, which implies that either a or x_1 is positive. In conclusion one of the roots of q has a positive real part, thereby proving that $x_n(t)$ grows exponentially. ∎

3.2. Overcoming the order barrier. Although the analysis of the limit PDE (3.2) is informative, it does not tell the whole story. Actually we show in this section that (3.1) can be made stable provided the artificial compressibility coefficient ϵ is chosen small enough, say $\epsilon = \mathcal{O}(\tau^3)$.

PROPOSITION 3.2. *Assume periodic boundary conditions and $A = -\nu\Delta$. Let λ_1 be the smallest eigenvalue of A. The semidiscrete system* (3.1) *is stable for all* $\epsilon \in [0, \lambda_1\tau^3]$.

Proof. Since the boundary conditions are periodic and $A = -\nu\Delta$, we can apply the mode analysis directly to the semidiscrete system (3.1). Upon setting $d^m := \nabla\cdot\mathbf{f}(t^m)$ and $\psi^m = \nabla\cdot\mathbf{u}^m$, $m = 0, 1, \ldots$, we obtain

$$\delta_3\psi^{n+1} - \nu\tau\Delta\delta_2\psi^{n+1} - \tau^3\epsilon^{-1}\Delta\psi^{n+1} = \delta_2 d^{n+1},$$

where $\delta_{l+1}\psi^{n+1} := \delta_l\psi^{n+1} - \delta_l\psi^n$, $l \in \mathbb{N}$, with the convention $\delta_0\psi^{n+1} := \psi^{n+1}$. Let $\{(\lambda_i, \varphi_i)\}_{i\in\mathbb{N}}$ be a Hilbertian basis of $\mathbf{L}^2(\Omega)$ composed of all the eigenpairs of the operator $-\Delta$ equipped with periodic boundary condition. Upon expanding ψ^{n+1} and d^{n+1} with respect to this basis, i.e., $\psi^{n+1}(\mathbf{x}) = \sum_{i\in\mathbb{N}} x_i^{n+1}\varphi_i(\mathbf{x})$, $d^{n+1}(\mathbf{x}) = \sum_{i\in\mathbb{N}} d_i^{n+1}\varphi_i(\mathbf{x})$, we infer that the following holds for all $i \in \mathbb{N}$:

$$(1 + \nu\tau\lambda_i + \lambda_i\tau^3\epsilon^{-1})x_i^{n+1} - (3 + 2\nu\tau\lambda_i)x_i^n + (3 + \nu\tau\lambda_i)x_i^{n-1} - x_i^{n-2} = \delta_2 d_i^{n+1}.$$

This recurrence relation depends on two parameters $\alpha_i := \nu\tau\lambda_i$ and $\beta_i := \lambda_i\tau^3\epsilon^{-1}$. It is stable provided the roots of the associated characteristic polynomial

$$h(x) := (1 + \alpha_i + \beta_i)x^3 - (3 + 2\alpha_i)x^2 + (3 + \alpha_i)x - 1$$

are all less than unity in absolute value. Observe that $h(x) = x(\alpha_i(1-x)^2 + \beta_i x^2) + (x-1)^3$ and h has at least one real root, say x_1. Note that x_1 cannot be negative, otherwise $h(x_1) \le (x_1 - 1)^3 \le -1 < 0$; note also that x_1 cannot be larger than 1, otherwise $h(x_1) \ge (x_1 - 1)^3 > 0$. In conclusion $x_1 \in (0, 1)$ and the proof ends here if

the three roots of h are real. Otherwise, there are two other complex conjugate roots, say x_2 and $\overline{x_2}$. Viète's formula implies that

$$x_1 x_2 \overline{x_2} = x_1 |x_2|^2 = \frac{1}{1+\alpha_i+\beta_i}.$$

We will be able to conclude that $|x_2| \le 1$ once we establish that $\frac{1}{(1+\alpha_i+\beta_i)x_1} \le 1$. Note that the assumption $\epsilon \le \lambda_1 \tau^3$ implies that $\alpha_i + \beta_i := \nu\tau\lambda_i + \lambda_i \tau^3 \epsilon^{-1} \ge \lambda_1 \tau^3 \epsilon^{-1} \ge 1$. Hence, if $x_1 \ge \frac{1}{2}$, then $\frac{1}{x_1} \le 2 \le 1 + \alpha_i + \beta_i$, thereby proving that $\frac{1}{(1+\alpha_i+\beta_i)x_1} \le 1$. Assume now that $x_1 \le \frac{1}{2}$, then using that $h(x_1) = 0$, i.e.,

$$x_1(\alpha_i(1-x_1)^2 + \beta_i x_1^2) = (1-x_1)^3,$$

and $\frac{x_1^2}{(1-x_1)^2} \le 1$, since $x_1 \le \frac{1}{2}$, we infer that $x_1(\alpha_i + \beta_i) \ge 1 - x_1$, or $x_1 \ge \frac{1}{1+\alpha_i+\beta_i}$. This concludes the proof. ∎

The above proposition shows that although the singular perturbation of the Stokes problem formulated in (3.2) is unstable, its discrete form can be stabilized; the key stabilization mechanism comes from the time approximation. In other words, the order barrier identified by Shen [31] can be overcome provided $\epsilon \le c\tau^3$. In particular the following algorithm is stable:

$$\begin{cases} \dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\tau} + A\mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1}, \quad \mathbf{u}^{n+1}|_\Gamma = 0, \\ \gamma\tau(p^{n+1} - 2p^n + p^{n-1}) + \nabla\cdot\mathbf{u}^{n+1} = 0, \end{cases} \tag{3.3}$$

provided $\gamma > 0$ is chosen smaller than λ_1. This algorithm uncouples the velocity and the pressure and is formally third-order accurate on the divergence. The numerical cost of the method is prohibitive though, since one needs to solve one vector-valued elliptic problem like $\mathbf{v} - \nabla\nabla\cdot\mathbf{v} + \tau A\mathbf{v} = \mathbf{r}$ per time step, and the condition number of the discrete counterpart of this problem behaves like $\mathcal{O}(h^{-2})$.

Whether the above observations can be useful to make higher-order projection method is unclear. We abandon projection methods in the rest of the paper and from now on focus only on artificial compressibility methods.

4. Bootstrapping technique. We have seen in the previous sections that the artificial compressibility technique with $\epsilon = \tau$ is a very efficient first-order approximation technique with a computational cost similar to that of solving a parabolic equation. We show in this section that it is possible to bootstrap the method to increase its accuracy without sacrificing the algorithmic complexity.

4.1. Heuristic argument. Let $(\mathbf{u}, p)$ be the solution to (1.1). Assume for the time being that r is some approximation of the exact pressure p and let us assume that both p and r are smooth functions of time. Let ϵ be a positive number and consider the following problem:

$$\begin{cases} \partial_t \mathbf{w} + A\mathbf{w} + \nabla s = \mathbf{f}, \quad \mathbf{w}|_\Gamma = 0, \qquad \mathbf{w}|_{t=0} = \mathbf{u}_0, \\ \epsilon\partial_t(s - r) + \nabla\cdot\mathbf{w} = 0, \quad s|_{t=0} = p_0. \end{cases} \tag{4.1}$$

Let us denote $\mathbf{e} := \mathbf{u} - \mathbf{w}$ and $\delta := p - s$, where recall that the pair $(\mathbf{u}, p)$ solves (1.1). Then,

$$\begin{cases} \partial_t \mathbf{e} + A\mathbf{e} + \nabla\delta = 0, \qquad \mathbf{e}|_\Gamma = 0, \qquad \mathbf{e}|_{t=0} = 0, \\ \epsilon\partial_t\delta + \nabla\cdot\mathbf{e} = \epsilon\partial_t(p - r), \quad \delta|_{t=0} = 0. \end{cases} \tag{4.2}$$

We observe that if r is an $\mathcal{O}(\epsilon^l)$ approximation of p, then $\epsilon\partial_t(p-r) = \mathcal{O}(\epsilon^{l+1})$ and if (4.2) is stable with respect to perturbations in the mass equation, then one should get $\mathbf{e} = \mathcal{O}(\epsilon^{l+1})$ and $\delta = \mathcal{O}(\epsilon^{l+1})$, i.e., the accuracy on the pair $(\mathbf{w}, s)$ is increased by one order in ϵ. Provided it can be established that (4.2) is stable with respect to perturbations in the mass equation, this observation opens a door to a new family of high-order approximation methods based on bootstrapping.

4.2. Stability analysis. We start with a standard lemma.

LEMMA 4.1. *There is $c > 0$ such that for all $s \in L^2_0(\Omega)$ there is $\mathbf{w}(s) \in \mathbf{H}^1_0(\Omega)$ such that $\nabla\cdot(\mathbf{w}(s)) = s$ and $\|\mathbf{w}(s)\|_{\mathbf{H}^1(\Omega)} \le c\|s\|_{L^2_0(\Omega)}$*

Proof. This is a consequence of the divergence operator $\nabla\cdot : \mathbf{H}^1_0(\Omega) \longrightarrow L^2_0(\Omega)$ being bounded and surjective; see, e.g., Girault and Raviart [10, pp. 18–26]. ∎

We now revisit the stability analysis of (2.4). More precisely, let $\mathbf{k} \in L^2((0,+\infty); \mathbf{L}^2(\Omega))$ and $g \in H^1((0,+\infty); L^2_0(\Omega))$ be some data, and let $(\mathbf{v}, q)$ be the solution to

$$\begin{cases} \partial_t\mathbf{v} + A\mathbf{v} + \nabla q = \mathbf{k}, & \mathbf{v}|_\Gamma = 0, \quad \mathbf{v}|_{t=0} = \mathbf{v}_0, \\ \epsilon\partial_t q + \nabla\cdot\mathbf{v} = g, & q|_{t=0} = q_0. \end{cases} \tag{4.3}$$

LEMMA 4.2. *There exists constants c and c' such that the solution to (4.3) satisfies the following a priori estimate:*

$$\begin{aligned} \|\mathbf{v}\|^2_{L^\infty((0,T);\mathbf{L}^2)} + \epsilon\|q\|^2_{L^\infty((0,T);L^2)} + \nu\|\mathbf{v}\|^2_{L^2((0,T);\mathbf{H}^1)} \\ \le c\,(\|\mathbf{v}_0\|^2_{\mathbf{L}^2} + \epsilon\|q_0\|^2_{L^2}) + c'\,(\|\mathbf{k}\|^2_{L^2((0,T);\mathbf{H}^{-1})} + \|g\|^2_{H^1((0,T);L^2)}). \end{aligned} \tag{4.4}$$

If in addition $\mathbf{k} \in H^1((0,+\infty); \mathbf{L}^2(\Omega))$ and $g \in H^2((0,+\infty); L^2_0(\Omega))$, and the initial data $(\mathbf{v}_0, q_0)$ are smooth enough, say $\partial_t\mathbf{v}(0) := \mathbf{k}(0) - A\mathbf{v}_0 - \nabla q_0 \in \mathbf{L}^2(\Omega)$, then

$$\|q\|^2_{L^2((0,T);L^2)} \le c\,(\|\partial_t\mathbf{v}_0\|^2_{\mathbf{L}^2} + \epsilon^{-1}\|g(0)\|^2_{L^2} + \|\mathbf{k}\|^2_{H^1((0,T);\mathbf{H}^{-1})} + \|g\|^2_{H^2((0,T);L^2)}). \tag{4.5}$$

Proof. Testing the momentum equation with $\mathbf{v}$ and the mass conservation equation with q and adding the two results gives

$$\frac{1}{2}\partial_t\|\mathbf{v}\|^2_{\mathbf{L}^2} + \frac{\epsilon}{2}\partial_t\|q\|^2_{\mathbf{L}^2} + \frac{\nu}{2}\|\mathbf{v}\|^2_{\mathbf{H}^1} \le \frac{2}{\nu}\|\mathbf{k}\|^2_{\mathbf{H}^{-1}} + \int_\Omega qg\,\mathrm{d}\mathbf{x}.$$

Let $\mathbf{w}(g)$ be as defined in Lemma 4.1, then

$$\begin{aligned} \int_\Omega qg\,\mathrm{d}\mathbf{x} &= \int_\Omega q\nabla\cdot(\mathbf{w}(g))\,\mathrm{d}\mathbf{x} = \int_\Omega(-\mathbf{k} + \partial_t\mathbf{v} + A\mathbf{v})\cdot\mathbf{w}(g)\,\mathrm{d}\mathbf{x} \\ &\le \|\mathbf{k}\|_{\mathbf{H}^{-1}}\|\mathbf{w}(g)\|_{\mathbf{H}^1} + M\|\mathbf{v}\|_{\mathbf{H}^1}\|\mathbf{w}(g)\|_{\mathbf{H}^1} - \int_\Omega \partial_t\mathbf{v}\cdot\mathbf{w}(g)\,\mathrm{d}\mathbf{x}. \end{aligned}$$

We integrate the above inequality over time and use the property $\partial_t\mathbf{w}(g) = \mathbf{w}(\partial_t g)$ together with the stability estimate from Lemma 4.1 to obtain

$$\begin{aligned} &-\int_0^T\int_\Omega \partial_t\mathbf{v}\cdot\mathbf{w}(g)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t \\ &= \int_0^T\int_\Omega \mathbf{v}\cdot\mathbf{w}(\partial_t g)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t - \int_\Omega \mathbf{v}(T)\cdot\mathbf{w}(g(T))\,\mathrm{d}\mathbf{x} + \int_\Omega \mathbf{v}(0)\cdot\mathbf{w}(g(0))\,\mathrm{d}\mathbf{x} \\ &\le \|\mathbf{v}\|_{L^2((0,T);\mathbf{L}^2)}\|\mathbf{w}(\partial_t g)\|_{L^2((0,T);\mathbf{L}^2)} + \|\mathbf{v}(T)\|_{\mathbf{L}^2}\|\mathbf{w}(g(T))\|_{\mathbf{L}^2} \\ &\quad + \|\mathbf{v}(0)\|_{\mathbf{L}^2}\|\mathbf{w}(g(0))\|_{\mathbf{L}^2} \\ &\le c(\|\mathbf{v}\|_{L^2((0,T);\mathbf{H}^1)}\|\partial_t g\|_{L^2((0,T);L^2)} + \|\mathbf{v}(T)\|_{\mathbf{L}^2}\|g(T)\|_{L^2} + \|\mathbf{v}_0\|_{\mathbf{L}^2}\|g_0\|_{L^2}), \end{aligned}$$

which in turn gives

$$\begin{aligned}&\int_0^T\int_\Omega qg\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\\&\quad\le c(\|\mathbf{k}\|_{L^2((0,T);\mathbf{H}^{-1})}+\|\mathbf{v}\|_{L^2((0,T);\mathbf{H}^1)})\|g\|_{L^2((0,T);L^2)}\\&\qquad+c'(\|\mathbf{v}\|_{L^2((0,T);\mathbf{H}^1)}\|\partial_t g\|_{L^2((0,T);L^2)}+\|\mathbf{v}(T)\|_{\mathbf{L}^2}\|g(T)\|_{L^2}+\|\mathbf{v}_0\|_{\mathbf{L}^2}\|g_0\|_{L^2}).\end{aligned}$$

In conclusion,

$$\begin{aligned}&\frac12\|\mathbf{v}(T)|_{\mathbf{L}^2}^2+\frac{\epsilon}{2}\|q(T)\|_{\mathbf{L}^2}^2+\frac{\nu}{2}\|\mathbf{v}\|_{L^2((0,T);\mathbf{H}^1)}^2\\&\quad\le c\|\mathbf{k}\|_{L^2((0,T);\mathbf{H}^{-1})}^2+\frac14\|\mathbf{v}(T)\|_{\mathbf{L}^2}^2+\frac{\nu}{4}\|\mathbf{v}\|_{L^2((0,T);\mathbf{H}^1)}^2\frac12\|\mathbf{v}_0\|_{\mathbf{L}^2}^2+\frac{\epsilon}{2}\|q_0\|_{L^2}^2\\&\qquad+c'(\|g\|_{L^2((0,T);L^2)}^2+\|\partial_t g\|_{L^2((0,T);L^2)}^2+\|g\|_{L^\infty((0,T);L^2)}^2+\|\mathbf{v}_0\|_{\mathbf{L}^2}\|g_0\|_{L^2}).\end{aligned}$$

The estimate (4.4) follows readily.

Upon considering the time derivative of the system (4.3) and using the estimate (4.4) for the time derivatives, we infer that

$$\|\partial_t\mathbf{v}\|_{L^2((0,T);\mathbf{L}^2)}^2\le c\,(\|\partial_t\mathbf{v}_0\|_{\mathbf{L}^2}^2+\epsilon\|\partial_t q_0\|_{L^2}^2)+c'\,(\|\mathbf{k}\|_{H^1((0,T);\mathbf{H}^{-1})}^2+\|g\|_{H^2((0,T);L^2)}^2).$$

Since $\nabla\cdot(\mathbf{v}(0))=\nabla\cdot\mathbf{v}_0=0$, we infer that $\epsilon\partial_t q(0)=g(0)$, which in turn implies $\epsilon\|\partial_t q(0)\|_{L^2}^2=\epsilon^{-1}\|g(0)\|_{L^2}$. This bound together with the estimate on $\|\mathbf{v}\|_{L^2((0,T);\mathbf{H}^1)}$ provides an estimate on $\|\nabla q\|_{L^2((0,T);\mathbf{H}^{-1})}$ from which (4.5) follows; see, e.g., Girault and Raviart [10, pp. 18–26]. ∎

We now apply the above result to the perturbed system (4.1).

THEOREM 4.3. *Let $r\in H^2((0,T);L_0^2(\Omega))$, let $(\mathbf{u},p)$ be the solution to* (1.1)*, and let $(\mathbf{w},s)$ be the solution to* (4.1)*. There is a constant $c>0$ so that*

$$\begin{aligned}&\|\mathbf{u}-\mathbf{w}\|_{L^\infty((0,T);\mathbf{L}^2)}^2+\epsilon\|p-s\|_{L^\infty((0,T);L^2)}^2+\nu\|\mathbf{u}-\mathbf{w}\|_{L^2((0,T);\mathbf{H}^1)}^2\\&\qquad\le c\,\epsilon^2\|p-r\|_{H^2((0,T);L^2)}^2,\qquad(4.6)\\&\|p-s\|_{L^2((0,T);L^2)}^2\\&\qquad\le c\,(\epsilon\|\partial_t(p-r)(0)\|_{L^2}^2+\epsilon^2\|p-r\|_{H^3((0,T);L^2)}^2).\qquad(4.7)\end{aligned}$$

Proof. Let us denote $\mathbf{e}:=\mathbf{u}-\mathbf{w}$ and $\delta:=p-s$. Then,

$$\begin{cases}\partial_t\mathbf{e}+A\mathbf{e}+\nabla\delta=0, & \mathbf{e}|_\Gamma=0, \qquad \mathbf{e}|_{t=0}=0,\\ \epsilon\partial_t\delta+\nabla\cdot\mathbf{e}=\epsilon\partial_t(p-r), & \delta|_{t=0}=0.\end{cases}$$

Apply Lemma 4.2 and observe that $\mathbf{v}_0=0$, $q_0=0$, $\mathbf{k}=0$, and $g=\epsilon\partial_t(p-r)$. Note also that $\mathbf{v}_0=0$, $q_0=0$, $\mathbf{k}=0$ imply $\partial_t\mathbf{v}(0)=0$. ∎

This result confirms the heuristic argument above. More precisely, if r is a scalar field such that $\|\partial_t(p-r)\|_{H^3((0,T);L^2)}=\mathcal{O}(\epsilon^l)$ then $\mathbf{u}-\mathbf{w}=\mathcal{O}(\epsilon^{l+1})$ in the $L^\infty((0,T);\mathbf{L}^2)$- and $L^2((0,T);\mathbf{H}^1)$-norm, and $p-s=\mathcal{O}(\epsilon^{l+\frac12})$ in the $L^\infty((0,T);L^2)$-norm. If in addition $\|\partial_t(p-r)(0)\|_{L^2}=\mathcal{O}(\epsilon^{\frac12})$ then $\|p-s\|_{L^2((0,T);L^2)}=\mathcal{O}(\epsilon^{l+1})$, i.e., the accuracy on the pair $(\mathbf{w},s)$ is increased by one order in ϵ.

Remark 4.1. The additional assumption $\|\partial_t(p-r)(0)\|_{L^2}=\mathcal{O}(\epsilon^{\frac12})$ can be avoided by considering time-dependent norms weighted by $\min(t,1)$ as in Shen [32, Lemma 3.2].

4.3. High-order artificial compressibility. The above argument leads us to consider the following family of approximation methods. Let ℓ be a positive integer, set $s_0 = 0$, and consider the following velocity-pressure pairs $(\mathbf{w}_1, s_1), \dots, (\mathbf{w}_\ell, s_\ell)$ solving

$$\begin{cases} \partial_t \mathbf{w}_i + A\mathbf{w}_i + \nabla s_i = \mathbf{f}, & \mathbf{v}|_\Gamma = 0, \qquad \mathbf{w}_i|_{t=0} = \mathbf{u}_0, \\ \epsilon \partial_t (s_i - s_{i-1}) + \nabla\cdot \mathbf{w}_i = 0, & s_i|_{t=0} = p_0, \end{cases} \tag{4.8}$$

for all $1 \le i \le \ell$. The above PDE systems are easy to solve since each of them corresponds to the artificial compressibility method with nonzero right-hand side in the mass conservation equation.

To avoid issues with compatibility conditions at $t = 0$, let us assume that $\mathbf{u}_0 = 0$, and $\mathbf{f}(0) = 0, \dots, \partial_t^m \mathbf{f}(0) = 0$, for some $m \in \mathbb{N}$. Then it can be shown that $\partial_t \mathbf{u}(0) = 0, \dots, \partial_t^{m+1}\mathbf{u}(0) = 0, p(0) = 0, \partial_t p(0) = 0, \dots, \partial_t^m p(0) = 0$. Recalling that $s_0 = 0$, then, upon setting $\delta_0 = p - s_0$, the above assumption implies that $\delta_0(0) = 0$, $\partial_t \delta_0(0) = 0, \dots, \partial_t^m \delta_0(0) = 0$.

LEMMA 4.4. *Let $\delta_0 := p$ and for $i \in \{1, \dots, \ell\}$ let $\mathbf{e}_i := \mathbf{u} - \mathbf{w}_i$, $\delta_i := p - s_i$. Assume that there is some $l \in \mathbb{N}$ such that $\partial_t \delta_{i-1}(0) = 0, \dots, \partial_t^{l+1}\delta_{i-1}(0) = 0$, then $\partial_t e_i(0) = 0, \dots, \partial_t^{l+1} e_i(0) = 0$, $\partial_t \delta_i(0) = 0, \dots, \partial_t^{l+1}\delta_i(0) = 0$, and there is $c > 0$ so that*

$$\begin{aligned} &\|\partial_t^l \mathbf{e}_i\|^2_{L^\infty((0,T);\mathbf{L}^2)} + \epsilon \|\partial_t^l \delta_i\|^2_{L^\infty((0,T);L^2)} + \nu \|\partial_t^l \mathbf{e}_i\|^2_{L^2((0,T);\mathbf{H}^1)} \\ &\qquad \le c\,\epsilon^2 \|\delta_{i-1}\|^2_{H^{l+2}((0,T);L^2)}, \end{aligned} \tag{4.9}$$

$$\begin{aligned} &\|\delta_i\|_{H^l((0,T);L^2)} \\ &\qquad \le c\epsilon \|\delta_{i-1}\|_{H^{l+3}((0,T);L^2)}. \end{aligned} \tag{4.10}$$

Proof. Let us observe first that

$$\begin{cases} \partial_t \mathbf{e}_i + A\mathbf{e}_i + \nabla \delta_i = 0, & \mathbf{e}_i|_\Gamma = 0, \qquad \mathbf{e}_i|_{t=0} = 0, \\ \epsilon \partial_t \delta_i + \nabla\cdot \mathbf{e}_i = \epsilon \partial_t \delta_{i-1}, & \delta_i|_{t=0} = 0. \end{cases}$$

We prove the fist statement by induction, i.e., $\mathbf{e}_i(0) = 0, \dots, \partial_t^k \mathbf{e}_i(0) = 0$, $\delta_i(0) = 0, \dots, \partial_t^k \delta_i(0) = 0$ for all $k = 0, \dots, l+1$. The statement is true for $k = 0$ by hypothesis. Assume that the induction hypothesis holds for some k, $0 \le k \le l$. Then the momentum equation implies that $\partial_t^{k+1}\mathbf{e}_i(0) = -A\partial_t^k \mathbf{e}_i(0) - \nabla \partial_t^k \delta_i(0) = 0$. Moreover, the equation $\epsilon \partial_t^{k+1}\delta_i + \nabla\cdot \partial_t^k \mathbf{e} = \epsilon \partial_t^{k+1}\delta_{i-1}$ and the identities $\partial_t^k \mathbf{e}(0) = 0$, $\partial_t^{k+1}\delta_{i-1}(0) = 0$ imply that $\partial_t^{k+1}\delta_i(0) = 0$, which proves that the induction hypothesis holds for $k+1$. The second part of the statement is proved by applying Lemma 4.2. Taking l time derivatives in the error equation gives

$$\begin{cases} \partial_t(\partial_t^l \mathbf{e}_i) + A(\partial_t^l \mathbf{e}_i) + \nabla(\partial_t^l \delta_i) = 0, & (\partial_t^l \mathbf{e}_i)|_\Gamma = 0, \\ \epsilon \partial_t(\partial_t^l \delta_i) + \nabla\cdot(\partial_t^l \mathbf{e}_i) = \epsilon \partial_t(\partial_t^l \delta_{i-1}). \end{cases}$$

Observing that we have proved above that $\partial_t^l \mathbf{e}_i(0) = 0$ and $\partial_t^l \delta_i(0) = 0$, Lemma 4.2 implies that

$$\|\partial_t^l \mathbf{e}_i\|^2_{L^\infty((0,T);\mathbf{L}^2)} + \epsilon \|\partial_t^l \delta_i\|^2_{L^\infty((0,T);L^2)} + \nu \|\partial_t^l \mathbf{e}_i\|^2_{L^2((0,T);\mathbf{H}^1)} \le c\,\epsilon^2 \|\partial_t^l \delta_{i-1}\|^2_{H^2((0,T);L^2)}.$$

Moreover we have also $\partial_t^{l+1}\mathbf{e}_i(0) = 0$ and $\partial_t^{l+1}\delta_{i-1}(0) = 0$, which by applying

Lemma 4.2 again imply that

$$\|\partial_t^l \delta_i\|_{L^2((0,T);L^2)} \leq c\,\epsilon \|\delta_{i-1}\|_{H^{l+3}((0,T);L^2)}.$$

This completes the proof. ∎

We now formulate the main result of this section.

THEOREM 4.5. *Assume that* $\mathbf{u}_0 = 0$, *and* $\mathbf{f}(0) = 0, \ldots, \partial_t^{3\ell}\mathbf{f}(0) = 0$, *and assume that* $p \in H^{3\ell}((0,T);L^2(\Omega))$, *then there is* c, *uniform with respect to* ϵ, *so that the following error estimates hold for all* $i \in \{0, \ldots, \ell\}$:

$$\|\mathbf{u} - \mathbf{w}_i\|_{L^2((0,T);\mathbf{L}^2)} + \nu^{\frac{1}{2}} \|\mathbf{u} - \mathbf{w}_i\|_{L^2((0,T);\mathbf{H}^1)} \leq c\epsilon^i \|p\|_{H^{3i-1}((0,T);L^2)}, \tag{4.11}$$

$$\|p - s_i\|_{L^2((0,T);L^2)} \leq c\epsilon^i \|p\|_{H^{3i}((0,T);L^2)}. \tag{4.12}$$

Proof. Apply Lemma 4.4 repeatedly. ∎

Remark 4.2. The assumption $\mathbf{f}(0) = 0, \ldots, \partial_t^{3\ell}\mathbf{f}(0) = 0$, may not seem to be realistic, but it could be weakened by using weighted norms in time as in Shen [32, Lemma 3.2].

5. Examples of high-order artificial compressibility methods. We show in this section some examples of high-order artificial compressibility methods based on the bootstrapping technique introduced in section 4. In this section we set $\epsilon = \frac{\tau}{\chi}$, where χ is any user-defined positive number of order unity, i.e., $\chi \sim 1$. In all the numerical results presented in the next section, we use the value $\chi = 1$.

5.1. Backward differentiation- (BDF-)based generalization. We introduce in this section discrete versions of the bootstrapping technique based on multistep methods.

5.1.1. Second-order BDF. Using $\epsilon = \frac{\tau}{\chi}$, a second-order method can be constructed as follows by using BDF1 for the sequence $(\mathbf{w}_1, s_1)$ and BDF2 for the sequence $(\mathbf{w}_2, s_2)$:

$$\begin{cases} \dfrac{\mathbf{w}_1^{n+1} - \mathbf{w}_1^n}{\tau} + A\mathbf{w}_1^{n+1} + \nabla s_1^{n+1} = \mathbf{f}^{n+1} & \text{for } n \geq 0, \\ s_1^{n+1} - s_1^n + \chi \nabla{\cdot}\mathbf{w}_1^{n+1} = 0, \end{cases} \tag{5.1}$$

$$\begin{cases} \dfrac{3\mathbf{w}_2^{n+1} - 4\mathbf{w}_2^n + \mathbf{w}_2^{n-1}}{2\tau} + A\mathbf{w}_2^{n+1} + \nabla s_2^{n+1} = \mathbf{f}^{n+1} & \text{for } n \geq 1, \\ (s_2^{n+1} - s_2^n) - (s_1^{n+1} - s_1^n) + \chi \nabla{\cdot}\mathbf{w}_2^{n+1} = 0 \end{cases} \tag{5.2}$$

with $\mathbf{w}_1^0 = \mathbf{u}_0$, $s_1^0 = p_0$, and the BDF2 algorithm could be initialized by setting $\mathbf{w}_2^0 = \mathbf{u}_0$, $\mathbf{w}_2^1 = \mathbf{w}_1^1$, $s_2^1 = s_1^1$. After eliminating s_1^{n+1} and s_2^{n+1} in the momentum equation, the algorithm can be rewritten into the following form which uncouples the velocity and the pressure:

$$\begin{cases} \dfrac{\mathbf{w}_1^{n+1} - \mathbf{w}_1^n}{\tau} + A\mathbf{w}_1^{n+1} - \chi\nabla\nabla{\cdot}\mathbf{w}_1^{n+1} + \nabla s_1^n = \mathbf{f}^{n+1}, & n \geq 0, \\ s_1^{n+1} = s_1^n - \chi\nabla{\cdot}\mathbf{w}_1^{n+1}, \quad \delta s_1^{n+1} = s_1^{n+1} - s_1^n, \end{cases} \tag{5.3}$$

$$\begin{cases} \dfrac{3\mathbf{w}_2^{n+1} - 4\mathbf{w}_2^n + \mathbf{w}_2^{n-1}}{2\tau} + A\mathbf{w}_2^{n+1} - \chi\nabla\nabla{\cdot}\mathbf{w}_2^{n+1} + \nabla(s_2^n + \delta s_1^{n+1}) = \mathbf{f}^{n+1}, & n \geq 1, \\ s_2^{n+1} = s_2^n + \delta s_1^{n+1} - \chi\nabla{\cdot}\mathbf{w}_2^{n+1}, \quad \delta s_2^{n+1} = s_2^{n+1} - s_2^n. \end{cases} \tag{5.4}$$

Remark 5.1. Note the sequential structure of the algorithm: at each time step, one solves for $\mathbf{w}_1^{n+1}$, then computes s_1^{n+1}, then solves for $\mathbf{w}_2^{n+1}$, then finally computes s_2^{n+1}.

Remark 5.2. The BDF2 stepping for the sequence $(\mathbf{w}_2, s_2)$ can be replaced by the Crank–Nicolson time stepping without any difficulty.

5.1.2. Third-order BDF. A third-order method can be constructed as follows: compute the sequences $(\mathbf{w}_1, s_1)$ and $(\mathbf{w}_2, s_2)$ as in (5.3) and (5.4), respectively, then compute the sequence $(\mathbf{w}_3, s_3)$ by solving the following problem for $n \geq 2$:

$$\begin{cases} \dfrac{11\mathbf{w}_3^{n+1} - 18\mathbf{w}_3^n + 9\mathbf{w}_3^{n-1} - 2\mathbf{w}_3^{n-2}}{6\tau} + A\mathbf{w}_3^{n+1} - \chi\nabla\nabla\cdot\mathbf{w}_3^{n+1} \\ \qquad +\nabla(s_3^n + \delta s_2^{n+1}) = \mathbf{f}^{n+1}, \\ s_3^{n+1} = s_3^n + \delta s_2^{n+1} - \chi\nabla\cdot\mathbf{w}_3^{n+1}. \end{cases} \tag{5.5}$$

The algorithm can be initialized by setting $\mathbf{w}_3^0 = \mathbf{u}_0$, $\mathbf{w}_3^1 = \mathbf{w}_1^1$, $\mathbf{w}_3^2 = \mathbf{w}_2^2$, $s_3^2 = s_2^2$.

Remark 5.3. Instead of (5.3)–(5.5) it is possible to derive another third-order scheme by using a third-order BDF time stepping in (3.3). However, this scheme would require solving at each time step a discrete elliptic vector-valued problem with a condition number scaling like $O(h^{-2})$ as opposed to solving discrete parabolic problems resulting from (5.3)–(5.5) and having condition numbers scaling like $O(\tau h^{-2})$.

Remark 5.4. The schemes (5.3)–(5.4) and (5.3)–(5.5) can be extended to the Navier–Stokes equations by approximating the advection term with the appropriate order in each substep, i.e., first order in (5.3), second order in (5.4), and third order in (5.5). For an explicit treatment of the nonlinear term one could replace $\mathbf{f}^{n+1}$ by $\mathbf{f}^{n+1} - \mathbf{w}_1^n\cdot\nabla\mathbf{w}_1^n$ in (5.3), by $\mathbf{f}^{n+1} - 2\mathbf{w}_2^n\cdot\nabla\mathbf{w}_2^n - \mathbf{w}_2^{n-1}\cdot\nabla\mathbf{w}_2^{n-1}$ in (5.4), and by $\mathbf{f}^{n+1} - 3\mathbf{w}_3^n\cdot\nabla\mathbf{w}_3^n + 3\mathbf{w}_3^{n-1}\cdot\nabla\mathbf{w}_3^{n-1} - \mathbf{w}_3^{n-1}\cdot\nabla\mathbf{w}_3^{n-1}$ in (5.5). This will make the resulting schemes conditionally stable under a CFL-type condition. Note that the scheme (5.3)–(5.4) can be made unconditionally stable if the nonlinear terms are treated semi-implicitly. Unconditional stability cannot be achieved for (5.3)–(5.5) since BDF3 is not A-stable.

5.2. Defect correction generalization. As an alternative to the multistep BDF techniques described in the previous section, we show now that the bootstrapping methodology can be implemented within the framework of the so-called defect (or deferred) correction methods (see, for example, Stetter [33], Kress and Gustafsson [22]). This methodology has been successfully applied to the pressure-correction projection method in Minion [25], assuming periodic boundary conditions, and to the consistent splitting method in Jia and Liu [18]. The key difference with what we propose now is that the pressure is computed by solving a Poisson equation in the above references.

5.2.1. ODE theory. Although the principle of defect corrections methods is well known, we recall it here for the sake of completeness (see, e.g., Kress and Gustafsson [22]). Let us first illustrate the principle of the method on the ODE

$$\partial_t u(t) = f(t), \quad u(0) = u_0, \tag{5.6}$$

where we assume that f is a smooth function of t. The objective is to construct an approximation of u at the time step t^n by adding successive corrections. For instance

a defect correction technique of order k constructs an approximation of u at time t^n as follows: $u(t^n) = u_0^n + \tau u_1^n + \tau^2 u_2^n + \cdots + \tau^k u_k^n + \mathcal{O}(\tau^{k+1})$, where the corrections $u_0^n, \ldots, u_k^n$ are computed one after the other. There are various ways to construct the successive corrections, but for the sake of simplicity (and robustness) we are going to restrict ourselves to the Euler time stepping, i.e., we are going to construct $u_0^n, \ldots, u_k^n$ by using an implicit step for the stiff part of the ODE and an explicit step for the remaining part. For instance, assuming that we restrict ourselves to a third-order method, i.e., $k = 2$, we observe that

$$\frac{u(t^{n+1}) - u(t^n)}{\tau} = f(t^{n+1}) - \frac{\tau}{2}\partial_{tt}u(t^{n+1}) + \frac{\tau^2}{6}\partial_{ttt}u(t^{n+1}) + \mathcal{O}(\tau^3). \tag{5.7}$$

Then upon inserting the expansion $u_0^n + \tau u_1^n + \tau^2 u_2^n + \mathcal{O}(\tau^3)$ into this identity and regrouping terms having the same asymptotic order we obtain the following sequence of formal problems:

$$\frac{u_0^{n+1} - u_0^n}{\tau} = f(t^{n+1}), \qquad \frac{u_1^{n+1} - u_1^n}{\tau} = -\frac{1}{2}\partial_{tt}u_0^{n+1},$$

$$\frac{u_2^{n+1} - u_2^n}{\tau} = -\frac{1}{2}\partial_{tt}u_1^{n+1} + \frac{1}{6}\partial_{ttt}u_0^{n+1}.$$

The rest of the algorithm consists of approximating the higher-order derivatives. Again, there are many ways to proceed, but the simplest one consists of using divided differences. To make the algorithm depend on two stages only, we are going to authorize a time delay between each correction, say u_0 is approximated at time t^{n+1} (for $n \geq 0$), u_1 is approximated at time t^n (for $n \geq 1$), and u_2 is approximated at time t^{n-1} (for $n \geq 2$). The algorithm is initialized by setting $u_0^0 = u_0$, $u_1^0 = 0$, $u_2^0 = 0$, and it advances in time as follows:

(5.8)

$$\text{for } n \geq 0, \qquad \begin{cases} \dfrac{u_0^{n+1} - u_0^n}{\tau} = f^{n+1}, \\ du_0^{n+1} = (u_0^{n+1} - u_0^n)/\tau, \end{cases}$$

(5.9)

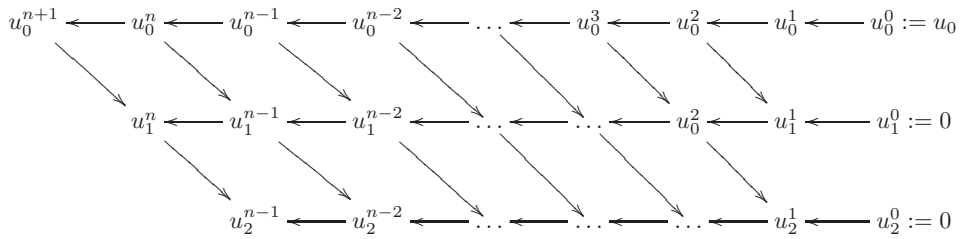
$$\text{for } n \geq 1, \qquad \begin{cases} d^2u_0^{n+1} = (du_0^{n+1} - du_0^n)/\tau, \\ \dfrac{u_1^n - u_1^{n-1}}{\tau} = -\dfrac{1}{2}d^2u_0^{n+1}, \\ du_1^n = (u_1^n - u_1^{n-1})/\tau, \end{cases}$$

(5.10)

$$\text{for } n \geq 2, \qquad \begin{cases} d^2u_1^n = (du_1^n - du_1^{n-1})/\tau, \qquad d^3u_0^{n+1} = (d^2u_0^{n+1} - d^2u_0^n)/\tau, \\ \dfrac{u_2^{n-1} - u_2^{n-2}}{\tau} = -\dfrac{1}{2}d^2u_1^n + \dfrac{1}{6}d^3u_0^{n+1}, \\ u^{n-1} = u_0^{n-1} + \tau u_1^{n-1} + \tau^2 u_2^{n-1}. \end{cases}$$

Note that u_0^{n+1} must be computed before u_1^n, which itself must be computed before u_2^{n-1}, i.e., the flow of dependences of the sequences $(u_0^n)_{n\geq 0}$, $(u_1^n)_{n\geq 0}$, $(u_2^n)_{n\geq 0}$ is as

follows:



LEMMA 5.1. *The algorithm* (5.8)–(5.9)–(5.10) *is third-order consistent and has the same stability properties as the Euler time stepping.*

Proof. The stability is evident. To prove the consistency statement we add the three equations (5.8) plus τ(5.9), replacing n by $(n+1)$, plus τ^2(5.10), replacing n by $n+2$, and we obtain

$$\frac{u_0^{n+1}-u_0^n}{\tau}+\tau\frac{u_1^{n+1}-u_1^n}{\tau}+\tau^2\frac{u_2^{n+1}-u_2^n}{\tau} = f^{n+1}-\tau\frac{1}{2}d^2u_0^{n+2}-\tau^2\frac{1}{2}d^2u_1^{n+2}+\tau^2\frac{1}{6}d^3u_0^{n+3}$$

which, upon setting setting $w^n := u_0^n + \tau u_1^n + \tau^2 u_2^n$, can also be rewritten

$$\frac{w^{n+1}-w^n}{\tau} = f^{n+1}-\tau\frac{1}{2}d^2w^{n+2}+\tau^3\frac{1}{2}d^2u_2^{n+2}+\tau^2\frac{1}{6}d^3u_0^{n+3}.$$

Finally, replacing $d^2u_0^{n+2}$ by $d^2w^{n+2}-\tau d^2u_1^{n+2}-\tau d^2u_2^{n+2}$, we infer that

$$\frac{w^{n+1}-w^n}{\tau} = f^{n+1}-\tau\frac{1}{2}d^2w^{n+2}+\tau^2\frac{1}{6}d^3w^{n+3}+\tau^3\frac{1}{2}d^2u_2^{n+2}-\tau^3\frac{1}{6}d^3u_1^{n+3}-\tau^4\frac{1}{6}d^3u_2^{n+3}.$$

This identity is then recast into the following form

$$\frac{w^{n+1}-w^n}{\tau} = f^{n+1}-\frac{\tau}{2}d^2w^{n+2}+\frac{\tau^2}{6}d^3w^{n+2}+R^{n+1}, \tag{5.11}$$

where $R^{n+1} := \frac{\tau^3}{2}d^2u_2^{n+2}-\frac{\tau^3}{6}d^3u_1^{n+3}-\frac{\tau^4}{6}d^3u_2^{n+3}$. Let $T>0$ be some fixed time and let $N=[T/\tau]$. Note that $N\geq 2$ provided τ is small enough. Assuming enough regularity on f with respect to t, say $f\in\mathcal{C}^3([0,T];\mathbb{R})$, one can show that there is a uniform constant $c(T)$ so that $\max_{3\leq n\leq N}(\tau|d^3u_2^n|,|d^2u_2^n|,|d^3u_1^n|)\leq c(T)\,\|f\|_{\mathcal{C}^3}$. As a result $R^{n+1}=\mathcal{O}(\tau^3)$. Upon observing that $\partial_{tt}(u(t^{n+1}))=d^2u(t^{n+2})+\mathcal{O}(\tau^2)$, $\partial_{ttt}(u(t^{n+3}))=d^3u(t^{n+1})+\mathcal{O}(\tau)$, provided the time step is constant, we infer that

$$\frac{u^{n+1}-u^n}{\tau} = f^{n+1}-\frac{\tau}{2}d^2u^{n+2}+\frac{\tau^2}{6}d^3u^{n+2}+\mathcal{O}(\tau^3),$$

which, when compared to (5.11), proves that $(w^n)_{1\leq n\leq N}$ is a third-order consistent approximation of $(u(t^n))_{1\leq n\leq N}$. ∎

Convergence can be proved using standard arguments; we skip the details of the full convergence analysis for the sake of brevity.

Remark 5.5 (time delay). Note that the delay in time in (5.8)–(5.9)–(5.10) can be avoided provided the algorithm is initialized properly (the details are left to the reader). In this case, (5.9) must be replaced by $\frac{u_1^{n+1}-u_1^n}{\tau}=-\frac{1}{2}d^2u_0^{n+1}$, and (5.10) must be replaced by $\frac{u_2^{n+1}-u_2^n}{\tau}=-\frac{1}{2}d^2u_1^{n+1}-\frac{1}{3}d^3u_0^{n+1}$.

5.2.2. PDE extension. The above technique naturally extends to nonlinear PDEs. Consider for instance the nonlinear PDE

$$\partial_t u + Au = f + B(u), \qquad u(0) = u_0, \tag{5.12}$$

where $A : D(T) \subset L \longrightarrow L$ is a densely defined closed unbounded linear operator on a Banach space L and B is a nonlinear operator well defined on $D(T)$. Let us assume that A is maximal and monotone, i.e., owing to the Hille–Yosida theorem, A generates a contraction semigroup. Without getting into more technical details, let us assume that the presence of B still makes the above problem well posed in some reasonable sense. The algorithm (5.8)–(5.9)–(5.10) can be generalized to this case as follows:

(5.13)
$$\text{for } n \geq 0, \qquad \begin{cases} \mathrm{nl}_0^{n+1} = Bu_0^n, \\ \dfrac{u_0^{n+1} - u_0^n}{\tau} + Au_0^{n+1} = f^{n+1} - \mathrm{nl}_0^{n+1}, \\ du_0^{n+1} = (u_0^{n+1} - u_0^n)/\tau, \end{cases}$$

(5.14)
$$\text{for } n \geq 1, \qquad \begin{cases} d^2u_0^{n+1} = (du_0^{n+1} - du_0^n)/\tau, \qquad \mathrm{nl}_1^n = B(u_0^n + \tau u_1^{n-1}), \\ \dfrac{u_1^n - u_1^{n-1}}{\tau} + Au_1^n = -\dfrac{1}{2}d^2u_0^{n+1} - \dfrac{\mathrm{nl}_1^n - \mathrm{nl}_0^n}{\tau}, \\ du_1^n = (u_1^n - u_1^{n-1})/\tau, \end{cases}$$

(5.15)
$$\text{for } n \geq 2, \qquad \begin{cases} d^2u_1^n = (du_1^n - du_1^{n-1})/\tau, \qquad d^3u_0^{n+1} = (d^2u_0^{n+1} - d^2u_0^n)/\tau, \\ \mathrm{nl}_2^{n-1} = B(u_0^{n-1} + \tau u_1^{n-1} + \tau^2 u_2^{n-2}), \\ \dfrac{u_2^{n-1} - u_2^{n-2}}{\tau} + Au_2^{n-1} = -\dfrac{1}{2}d^2u_1^n + \dfrac{1}{6}d^3u_0^{n+1} - \dfrac{\mathrm{nl}_2^{n-1} - \mathrm{nl}_1^{n-1}}{\tau^2}, \\ u^{n-1} = u_0^{n-1} + \tau u_1^{n-1} + \tau^2 u_2^{n-1}. \end{cases}$$

LEMMA 5.2. *The algorithm* (5.13)–(5.14)–(5.15) *is third-order accurate, and it is unconditionally stable if* $B = 0$.

5.2.3. Navier–Stokes extension. We finish this section by showing how to implement the above algorithm in the context of the bootstrapping artificial compressibility technique developed in section 4.

In order to understand the principle of the method, let us forget for the time being about the consistency error on the time derivatives of the velocity and let us focus instead on the handling of the bootstrapping on the pressure. For instance denoting by $\mathbf{u}_0$, $\mathbf{u}_1$ the first two corrections on the velocity and p_0, p_1 the first two corrections on the pressure, let us set $\mathbf{w}_1 := \mathbf{u}_0$, $s_1 = p_0$, $\mathbf{w}_2 = \mathbf{u}_0 + \tau\mathbf{u}_1$, $s_2 = p_0 + \tau p_1$. Then, still restricting ourselves to the Stokes system for the time being, we should have

$$\begin{cases} \dfrac{\mathbf{w}_1^{n+1} - \mathbf{w}_1^n}{\tau} + A\mathbf{w}_1^{n+1} + \nabla s_1^{n+1} = \mathbf{f}^{n+1}, \\ s_1^{n+1} - s_1^n + \chi\nabla\cdot\mathbf{w}_1^{n+1} = 0, \end{cases} \tag{5.16}$$

$$\begin{cases} \dfrac{\mathbf{w}_2^{n+1} - \mathbf{w}_2^n}{\tau} + A\mathbf{w}_2^{n+1} + \nabla s_2^{n+1} = \mathbf{f}^{n+1}, \\ (s_2^{n+1} - s_2^n) - (s_1^{n+1} - s_1^n) + \chi\nabla\cdot\mathbf{w}_2^{n+1} = 0, \end{cases} \tag{5.17}$$

which, after subtracting (5.16) from (5.17) and dividing the result by τ, gives the

following expression for the system solved by the second correction $(\mathbf{u}_1, p_1)$:

$$\begin{cases} \dfrac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau} + A\mathbf{u}_1^{n+1} + \nabla p_1^{n+1} = 0, \\ (p_1^{n+1}-p_1^n) - \tau^{-1}(p_0^{n+1}-p_0^n) + \chi\nabla\cdot\mathbf{u}_1^{n+1} = 0. \end{cases} \tag{5.18}$$

The same structure holds at every correction level; that is to say, the mass conservation equation at level k must be written $(p_k^{n+1}-p_k^n) - \tau^{-1}(p_{k-1}^{n+1}-p_{k-1}^n) + \chi\nabla\cdot\mathbf{u}_k^{n+1} = 0$. Once this is understood, the rest of the algorithm proceeds as in (5.13)–(5.14)–(5.15).

We are now ready to apply the above algorithm to the Navier–Stokes system. Upon setting $B\mathbf{u} := \mathbf{u}\cdot\nabla\mathbf{u}$, the third-order defect correction variant of the bootstrapping method can be written as follows. Initialize the algorithm by setting $\mathbf{u}_0^0 = \mathbf{u}(0)$ and $p_0^0 = p(0)$ (recall that the pressure at the initial time can be computed by solving $\Delta p_0 = \nabla\cdot\mathbf{f}(0)$, $\partial_n p_0|_\Gamma = (\mathbf{f}(0) - A\mathbf{u}_0)\cdot\mathbf{n}|_\Gamma$. Then update the velocity and pressure as follows:

(5.19)

$$\text{for } n \ge 0, \quad \begin{cases} \mathbf{nl}_0^{n+1} = B\mathbf{u}_0^n, \\ \dfrac{\mathbf{u}_0^{n+1}-\mathbf{u}_0^n}{\tau} + A\mathbf{u}_0^{n+1} - \chi\nabla\nabla\cdot\mathbf{u}_0^{n+1} + \nabla p_0^n = \mathbf{f}^{n+1} - \mathbf{nl}_0^{n+1}, \\ p_0^{n+1} = p_0^n - \chi\nabla\cdot\mathbf{u}_0^{n+1}, \\ d\mathbf{u}_0^{n+1} = (\mathbf{u}_0^{n+1}-\mathbf{u}_0^n)/\tau, \quad dp_0^{n+1} = (p_0^{n+1}-p_0^n)/\tau, \end{cases}$$

(5.20)

$$\text{for } n \ge 1, \quad \begin{cases} d^2\mathbf{u}_0^{n+1} = (d\mathbf{u}_0^{n+1} - d\mathbf{u}_0^n)/\tau, \\ \mathbf{nl}_1^n = B(\mathbf{u}_0^n + \tau\mathbf{u}_1^{n-1}), \\ \dfrac{\mathbf{u}_1^n-\mathbf{u}_1^{n-1}}{\tau} + A\mathbf{u}_1^n - \chi\nabla\nabla\cdot\mathbf{u}_1^n + \nabla(p_1^{n-1} + dp_0^n) \\ \qquad\qquad = -\dfrac{1}{2}d^2\mathbf{u}_0^{n+1} - \dfrac{\mathbf{nl}_1^n - \mathbf{nl}_0^n}{\tau}, \\ p_1^n = p_1^{n-1} + dp_0^n - \chi\nabla\cdot\mathbf{u}_1^n, \\ d\mathbf{u}_1^n = (\mathbf{u}_1^n - \mathbf{u}_1^{n-1})/\tau, \quad dp_1^n = (p_1^n - p_1^{n-1})/\tau, \end{cases}$$

(5.21)

$$\text{for } n \ge 2, \quad \begin{cases} d^2\mathbf{u}_1^n = (d\mathbf{u}_1^n - d\mathbf{u}_1^{n-1})/\tau, \qquad d^3\mathbf{u}_0^{n+1} = (d^2\mathbf{u}_0^{n+1} - d^2\mathbf{u}_0^n)/\tau, \\ \mathbf{nl}_2^{n-1} = B(\mathbf{u}_0^{n-1} + \tau\mathbf{u}_1^{n-1} + \tau^2\mathbf{u}_2^{n-2}), \\ \dfrac{\mathbf{u}_2^{n-1}-\mathbf{u}_2^{n-2}}{\tau} + A\mathbf{u}_2^{n-1} - \chi\nabla\nabla\cdot\mathbf{u}_2^{n-1} + \nabla(p_2^{n-2} + dp_1^{n-1}) \\ \qquad\qquad = -\dfrac{1}{2}d^2\mathbf{u}_1^n + \dfrac{1}{6}d^3\mathbf{u}_0^{n+1} - \dfrac{\mathbf{nl}_2^{n-1} - \mathbf{nl}_1^{n-1}}{\tau^2}, \\ p_2^{n-1} = p_2^{n-2} + dp_1^{n-1} - \chi\nabla\cdot\mathbf{u}_2^{n-1}, \\ \mathbf{u}^{n-1} = \mathbf{u}_0^{n-1} + \tau\mathbf{u}_1^{n-1} + \tau^2\mathbf{u}_2^{n-1}, \quad p^{n-1} = p_0^{n-1} + \tau p_1^{n-1} + \tau^2 p_2^{n-1}. \end{cases}$$

PROPOSITION 5.3. *Let* $(\mathbf{u}, p)$ *be the solution to* (1.1). *In the absence of the nonlinear term* $B\mathbf{u}$, *the algorithm* (5.19)–(5.20)–(5.21) *is unconditionally stable with any* $\chi \sim 1$.

The algorithm (5.19)–(5.20)–(5.21) is formally third-order accurate provided the solution is smooth enough and enough compatibility conditions are satisfied at $t = 0$. For instance, the proof of Theorem 4.5 suggests that sufficient compatibility conditions could be $\mathbf{u}_0 = 0$, $\mathbf{f}(0) = 0, \ldots, \partial_t^9\mathbf{f}(0) = 0$. A full proof of third-order convergence

would require reproving the discrete versions of Lemma 4.2 and Theorem 4.3 where the partial derivatives with respect to time are replaced by backward Euler time stepping, i.e., this would actually reprove Proposition 2.2 plus an estimate on the pressure. The conclusion would follow by reproving the discrete version of Lemma 4.4 and applying this lemma repeatedly (two times here) as in the proof of Theorem 4.5

Remark 5.6 (higher order). We have restricted ourselves to third-order accuracy, but the above methodology generalizes to arbitrary order.

Remark 5.7 (time delay). Note that the time delay in (5.19)–(5.20)–(5.21) can be avoided by proceeding as in Remark 5.5, provided the algorithm is initialized appropriately.

6. Numerical illustrations. The second-order method (5.3)–(5.4) and the two third-order methods (5.3)–(5.5) and (5.19)–(5.21) are tested numerically in this section. First we test the performance of the methods with Dirichlet boundary conditions, then we test (5.19)–(5.21) with Neumann boundary conditions. In all the tests we take $\chi = 1$.

6.1. Spatial discretization and solution of the linear system. The algorithms (5.3)–(5.4), (5.3)–(5.5), and (5.19)–(5.21) are tested with two codes: one using mixed $\mathbb{P}_2/\mathbb{P}_1$ finite elements and the other using the MAC approximation on a Cartesian grid.

The $\nabla\nabla\cdot$ operator in the finite element case is discretized using the bilinear form $\int_\Omega \nabla\cdot\mathbf{u}\nabla\cdot\mathbf{v}\,\mathrm{d}\mathbf{x}$. In the case of the Cartesian MAC discretization, the $\nabla\nabla\cdot$ operator is discretized using the classical form $(\nabla\nabla\cdot\mathbf{u})_i = \sum_{j=1}^d \partial_{ij}u_j, i = 1,\dots,d$, where the derivatives are approximated by standard central differences. This discretization can be generalized to unstructured finite volume grids by using the following identity on each finite volume V: $\int_V \nabla\nabla\cdot\mathbf{w}\,\mathrm{d}\mathbf{x} = \int_{\partial V}(\nabla\cdot\mathbf{w})\mathbf{n}\,\mathrm{d}s$, where $\mathbf{n}$ is the outward normal to the surface of V.

The linear systems resulting from the full discretization of any of the schemes discussed above couple all the components of the velocity. The structure of these systems is very similar to that resulting from the discretization of linear elasticity problems; therefore, one may adopt a solution technique that has been developed in this context (see, e.g., Blaheta [6]). Many other alternative techniques are available from the literature like, for instance, those from Benzi, Olshanskii, and Wang [5] and Heister and Rapin [17]. In this paper the linear systems resulting from the finite element discretization are solved with a method similar to that presented in [17]. Assume for instance that the matrix of the linear system has the following block structure in two dimensions:

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_{xx} & \mathcal{A}_{xy} \\ \mathcal{A}_{yx} & \mathcal{A}_{yy} \end{bmatrix}. \tag{6.1}$$

Let $\widetilde{\mathcal{A}}_{xx}$ and $\widetilde{\mathcal{A}}_{yy}$ be incomplete LU factorizations of $\mathcal{A}_{xx}$ and $\mathcal{A}_{yy}$, then we solve the linear system $\mathcal{A}z = b$ by using GMRES preconditioned by the upper triangular block structure

$$\mathcal{P} = \begin{bmatrix} \widetilde{\mathcal{A}}_{xx} & \mathcal{A}_{xy} \\ 0 & \widetilde{\mathcal{A}}_{yy} \end{bmatrix}. \tag{6.2}$$

The incomplete LU factorization contains 3 to 4 times the number of nonzero entries of $\mathcal{A}$. This techniques fully uncouples the Cartesian components of the velocity. It is

TABLE 1
$\mathbb{P}_2/\mathbb{P}_1$ *finite elements, method* (5.19)–(5.21), *CFL* $= 4.6$, *Re* $\in \{1, 100\}$.

	# Vel. dofs	93,130		370,938		1,480,938		1,991,930	
	Meth.	Dir.	Ite.	Dir.	Ite.	Dir.	Ite.	Dir.	Ite.
Re $= 1$	cpu/τ (sec.)	0.8	1.5	4.3	10.3	15.8	37.0	20.9	61.3
	# it		10		13		15		16
Re $= 100$	cpu/τ (sec.)	0.9	3.9	4.2	15.8	16.5	61.5	24.4	95.3
	# it		31		27		31		29

used for both the finite element approximation and the MAC approximation. As a reference we also use the sparse direct solver PARDISO; see, e.g., Schenk, Bollhöfer, and Römer [29].

To illustrate the method (5.19)–(5.21) we show in Table 1 tests done on four triangular Delaunay meshes composed of 93,130, 370,938, 1,480,938, and 1,991,930 velocity degrees of freedom (dofs). All the computations are done on a single processor. The source term is defined such that the exact solution is (6.6). The time step in these tests has been adjusted so that the CFL is equal to 4.6. Two Reynolds numbers are tested: Re $= 1$ and Re $= 100$. The results obtained with the direct method are reported in the column "Dir." and those obtained with the iterative method are listed in the column "Ite." We report in each case the elapsed time per time step in seconds, and for the iterative method we show the total number of calls to the preconditioner for the three linear systems in (5.19)–(5.21), (on average the number of calls to the preconditioner per linear system is the number reported in the table divided by three). The GMRES iterations are stopped when the relative ℓ^2-norm of the residual is less than 10^{-9}. We observe first that the sparse direct solver is really fast and well optimized. Second, the number of calls to the preconditioner grows slowly with the number of dofs. The growth is compatible with the fact that the incomplete LU factorization is known not to be optimal and should be replaced by an algebraic multigrid solver. Modulo this technicality, which is far outside the scope of the present paper, the behavior of the iterative method is satisfactory.

Note that the finite element implementation of the method described above is simplistic and runs into the risk of locking at high Reynolds number unless the velocity space, say $\mathbf{X}_h$, and the pressure space, say M_h, are such that

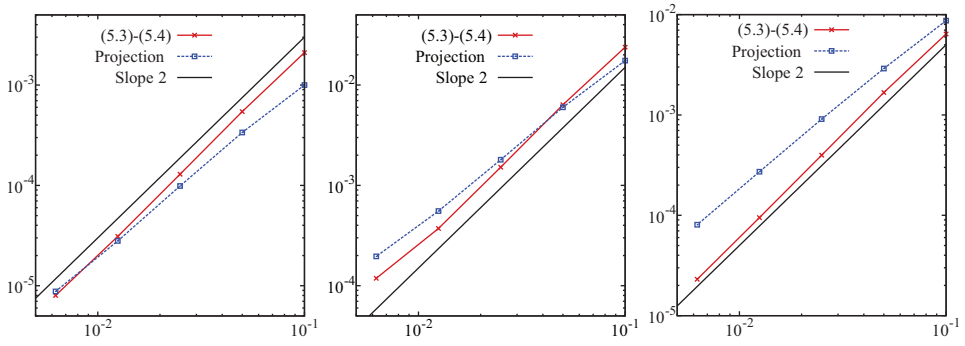
$$\{\mathbf{v}_h \in \mathbf{X}_h;\ \int_\Omega q_h \nabla\cdot\mathbf{v}_h \,\mathrm{d}\mathbf{x} = 0\} \subset \{\mathbf{v} \in \mathbf{H}^1_0(\Omega);\ \nabla\cdot\mathbf{v} = 0\}.$$

Such elements are described in Scott and Vogelius [30], Zhang [39]. But, in general, a locking free implementation must involve the inverse of the pressure mass matrix. More precisely, let $\mathcal{M}$ be the velocity mass matrix, $\mathcal{N}$ be the pressure mass matrix, $\mathcal{A}$ be the matrix associated with the operator A, $\mathcal{B}$ be the matrix associated with minus the divergence operator, and $\mathcal{B}^\mathsf{T}$ the matrix associated with the gradient operator. At every time step and at every level of the bootstrapping technique, the linear system has the following form:

$$(\mathcal{M} + \tau\mathcal{A} + \tau\chi\mathcal{B}^\mathsf{T}\mathcal{N}^{-1}\mathcal{B})z = b. \tag{6.3}$$

To simplify the argument, let us assume the following block structure in two space dimensions:

$$\mathcal{A} = \begin{bmatrix} \mathcal{A}_{xx} & \mathcal{A}_{xy} \\ \mathcal{A}_{yx} & \mathcal{A}_{yy} \end{bmatrix}, \qquad \mathcal{M} = \begin{bmatrix} \mathcal{M}_{xx} & 0 \\ 0 & \mathcal{M}_{yy} \end{bmatrix}, \qquad \mathcal{B} = \begin{bmatrix} \mathcal{B}_x & \mathcal{B}_y \end{bmatrix}. \tag{6.4}$$



(a) $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$ versus τ (b) $\|p-q\|_{L^2}$ versus τ (c) $\|\nabla\cdot\mathbf{w}\|_{L^2}$ versus τ

FIG. 1. *Two-dimensional (2D) unsteady Stokes equations on* 100×100 *MAC grid using second-order schemes. Log-log plot of the* L^2*-norm of errors at* $T=10$ *versus* τ*: velocity* (a)*; pressure* (b)*; divergence of velocity* (c)*. Second-order slope (solid line), BDF2 incremental rotational projection scheme (dashed line with* □ *symbols), bootstrapping scheme* (5.3)–(5.4) *(solid line with* × *symbols).*

Then the linear system has the following structure:

$$\begin{bmatrix} \mathcal{M}_{xx}+\tau\mathcal{A}_{xx}+\tau\chi\mathcal{B}_x^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B}_x & \tau\mathcal{A}_{xy}+\tau\chi\mathcal{B}_x^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B}_y \\ \tau\mathcal{A}_{yx}+\tau\chi\mathcal{B}_y^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B}_x & \mathcal{M}_{yy}+\tau\mathcal{A}_{yy}+\tau\chi\mathcal{B}_y^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B}_y \end{bmatrix} \begin{bmatrix} z_x \\ z_y \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}. \tag{6.5}$$

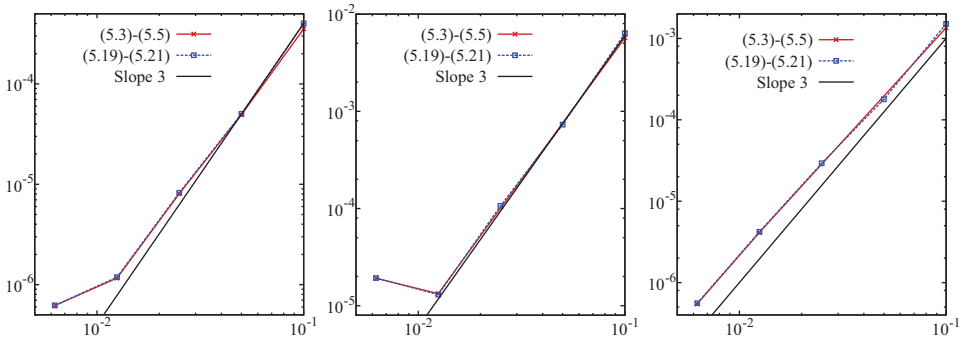
Instead of using (6.2), the linear system can be preconditioned by using the upper triangular part of the above matrix, where the blocks $\mathcal{B}_x^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B}_x$ and $\mathcal{B}_y^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B}_y$ are replaced by $\mathcal{B}_x^{\mathsf{T}}\mathcal{N}_L^{-1}\mathcal{B}_x$ and $\mathcal{B}_y^{\mathsf{T}}\mathcal{N}_L^{-1}\mathcal{B}_y$, where $\mathcal{N}_L$ is either the lumped pressure mass matrix or an appropriately weighted diagonal matrix. A complete description of suitable preconditioners for the above problem is out of the scope of the paper. We refer to Benzi, Olshanskii, and Wang [5] where other suitable preconditioners are described.

6.2. Dirichlet boundary conditions. Let us consider the following manufactured solution

$$\mathbf{u}=(\sin x\sin(y+t),\cos x\cos(y+t)),\quad p=\cos x\sin(y+t) \tag{6.6}$$

in the domain $\Omega=(0,1)\times(0,1)$. The errors in the L^2-norm are evaluated at $T=10$. For the MAC approximation the errors are measured in the discrete L^2-norm, which for a discrete field ψ with point values ψ_{ij} is defined by $\|\psi\|_{\ell^2}^2=h^2\sum_{i,j}|\psi_{ij}|^2$, where h is the mesh size. Since on the MAC stencil the two velocity components are located on different faces of the cells, we approximate both components at the centroids of the cells by averaging. For the finite element code the errors are estimated by using high-order Gaussian quadratures.

In the first series of tests we use only the MAC approximation on a grid composed of 100×100 cells. We solve the Stokes problem (1.1) with $A\mathbf{u}=-\Delta\mathbf{u}$, i.e., the viscosity is equal to 1; the source term is adjusted so that the exact solution is given by (6.6); there is no nonlinear term in this test. We compare the second-order bootstrapping method (5.3)–(5.4) with the second-order incremental projection scheme in rotational form; see Guermond, Minev, and Shen [16]. The results are shown in Figure 1. The scheme (5.3)–(5.4) achieves second-order on the velocity, the pressure, and the divergence of the velocity. The error levels on the velocity in the $\mathbf{L}^2$-norm



(a) $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$ versus τ (b) $\|p-q\|_{L^2}$ versus τ (c) $\|\nabla\cdot\mathbf{w}\|_{L^2}$ versus τ

FIG. 2. *2D unsteady Stokes equations on* 200×200 *MAC grid using third-order schemes* (5.3)–(5.5) *(solid line with* × *symbols) and* (5.19)–(5.21) *(dashed line with* □ *symbols). Log-log plot of the* L^2*-norm of errors at* $T = 10$ *versus time step* τ*: velocity* (a); *pressure* (b); *velocity divergence* (c).

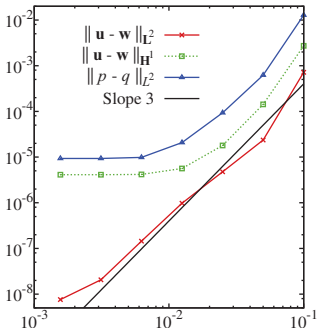


FIG. 3. *2D unsteady Navier–Stokes equations with* $\mathbb{P}_2/\mathbb{P}_1$ *finite elements on unstructured grid (*23,082 *elements,* 46,565 $\mathbb{P}_2$ *nodes) using third-order scheme* (5.19)–(5.21). *Log-log plot of the norm of errors at* $T = 10$ *versus time step* τ*:* $\mathbf{L}^2$*-norm of velocity (solid line with* × *symbols);* $\mathbf{H}^1$*-norm of velocity (dashed line with* □ *symbols);* L^2*-norm of pressure (solid line with* △ *symbols);.*

are comparable for both the projection method and the algorithm (5.3)–(5.4), but the accuracy of the projection method on the pressure in the $\mathbf{L}^2$-norm and the velocity in the $\mathbf{H}^1$-norm is suboptimal; this is the $\mathcal{O}(\tau^{\frac{3}{2}})$-order barrier of projection methods mentioned in the introduction. Note that the bootstrapping algorithm (5.3)–(5.4) does not suffer from this order barrier.

In a second series of tests we evaluate the third-order schemes (5.3)–(5.5) and (5.19)–(5.21). We solve the same Stokes problem as above, but this time the MAC approximation is done on a grid composed of 200×200 cells. The results obtained with the algorithms (5.3)–(5.5) and (5.19)–(5.21) are shown in Figure 2. The L^2-norm of the errors on the velocity, the pressure, and the divergence converge like $\mathcal{O}(\tau^3)$. The results obtained with the linear version of the scheme (5.19)–(5.21) are undistinguishable from those obtained with (5.3)–(5.5). However, these two schemes have very different stability limits when solving the full Navier–Stokes equations as shown in the next section.

In a third series of tests we solve the Navier–Stokes system; we use $A\mathbf{u} = -\Delta\mathbf{u}$ again, but this time the nonlinear term $\mathbf{u}\cdot\nabla\mathbf{u}$ is accounted for. The finite element mesh is composed of unstructured triangles (23,082 elements, 46,565 $\mathbb{P}_2$ nodes). Only the third-order algorithm (5.19)–(5.21) is tested. The results are shown in Figure 3.

TABLE 2
Limit of stability on the ratio τ/h for the third-order schemes (5.3)–(5.5) and (5.19)–(5.21), and various Reynolds numbers and grid sizes.

Approx	Scheme	Re=100, h=0.01	Re=500, h=0.005
MAC	(5.3)–(5.5)	1.25	0.76
MAC	(5.19)–(5.21)	5	1.5
FE ($\mathbb{P}_2/\mathbb{P}_1$)	(5.19)–(5.21)	12.5	3.5

The $\mathbf{L}^2$-norm of the error on the velocity behaves like $\mathcal{O}(\tau^3)$. The $\mathbf{H}^1$-norm of the error on the velocity and the L^2-norm of the error on the pressure both converge like $\mathcal{O}(\tau^3)$ for $\tau \geq 4\times 10^{-2}$; the convergence rate saturates for small time steps due to the spatial approximation error.

6.3. Stability in the presence of explicit advection terms. To illustrate the behavior of the algorithms (5.3)–(5.5) and (5.19)–(5.21) when solving the Navier–Stokes equations, we use the same manufactured solution as above but this time the nonlinear advection term is included in the algorithms. The tests are done with the MAC approximation and the mixed $\mathbb{P}_2/\mathbb{P}_1$ finite element codes. We show in Table 2 the limiting value of the ratio τ/h that guarantees stability for the two schemes at various Reynolds numbers and grid sizes. The key conclusion of these tests is that the scheme (5.19)–(5.21) has superior stability properties to (5.3)–(5.5).

6.4. Neumann boundary conditions. We finish this series of tests by evaluating the performance of the bootstrapping technique (5.19)–(5.21) on a problem involving mixed Dirichlet and Neumann boundary conditions. It is well established that classical projection methods suffer from accuracy loss for this type of boundary conditions; see, e.g., Guermond, Minev, and Shen [15, 16]. Many techniques have been proposed to fix projection methods in order to recover quasi-optimal convergence, but no third-order method with provable stability and consistency has yet been proposed in the literature (besides the technique that consists of solving the coupled system); see, e.g., Poux et al. [28], Angot and Cheaytou [1], Linke et al. [24]. Angot and Cheaytou [1] demonstrated that the penalty projection method of Angot, Caltagirone, and Fabrie [3, 2] yields second-order accuracy in time on both the velocity and the pressure, thereby suggesting that the $\nabla\nabla\cdot$ operator has a beneficial effect on open boundary conditions. We demonstrate here that the third-order version of the method proposed in the present paper, i.e., (5.19)–(5.21), gives third-order in time on all the quantities.

We evaluate (5.19)–(5.21) using the 2D manufactured solution from section 6.2 in the domain $\Omega = (0,1)\times(0,1)$ with the Neumann boundary conditions $-\partial_x u_y|_{\Gamma_N} = 0$, $(p - \partial_x u_x)|_{\Gamma_N} = 0$, where $\Gamma_N = \{(x,y) \in \Gamma,\ x = 0\}$. Dirichlet boundary conditions are imposed on $\Gamma_D := \Gamma\backslash\Gamma_N$. We use $A\mathbf{u} = -\Delta\mathbf{u}$ and the nonlinear term $\mathbf{u}\cdot\nabla\mathbf{u}$ is accounted for. This problem is solved up to $T = 2$ using $\mathbb{P}_2/\mathbb{P}_1$ finite elements. The unstructured mesh is composed of 23,082 triangles and 46,565 $\mathbb{P}_2$ nodes. The results for the third-order algorithm (5.19)–(5.21) are shown in Figure 4. All the errors behave like $\mathcal{O}(\tau^3)$. The saturation of the convergence rate in the $\mathbf{H}^1$-norm of the error on the velocity and the L^2-norm of the error on the pressure for small time steps is due to the spatial approximation error.

7. Conclusions. This paper has introduced a generalization of the artificial compressibility method for approximating the time-dependent incompressible Navier–Stokes equations. In principle, any order in time can be reached, say $\mathcal{O}(\tau^k)$, provided

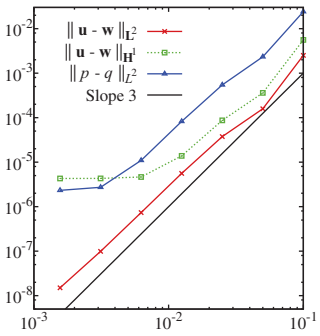


FIG. 4. *Mixed Dirichlet/Neumann boundary condition. 2D unsteady Navier–Stokes equations with $\mathbb{P}_2/\mathbb{P}_1$ finite elements on unstructured grid (*23082 *elements,* 46565 $\mathbb{P}_2$ *nodes) using third-order scheme* (5.19)–(5.21). *Log-log plot of the norm of errors at $T = 10$ versus time step τ: $\mathbf{L}^2$-norm of velocity (solid line with* × *symbols); $\mathbf{H}^1$-norm of velocity (dashed line with* □ *symbols); L^2-norm of pressure (solid line with* △ *symbols);.*

(at most) k vector-valued parabolic problems are solved at each time step. The condition number of the linear system associated with each fully discrete vector-valued problem scales like $O(\tau h^{-2})$, with τ being the time step and h being the spatial grid size. This approach has several advantages in comparison to traditional projection schemes widely used for solving the unsteady Navier–Stokes equations. First, it allows for the construction of schemes of any order in time for both the velocity and pressure. Second, in addition to requiring the solution of a vector-valued parabolic problem for the velocity at each time step, every projection scheme requires the solution of a scalar elliptic problem for the pressure which has a condition number scaling like $O(h^{-2})$. This makes these algorithms potentially expensive on massively parallel computers when the number of dofs is extremely large. Third, the artificial compressibility schemes of second or higher order based on a defect correction approach presented in this paper have nonlinear stability properties that are superior to that of second-order projection schemes. Finally, the proposed schemes have the potential to work with variable time stepping and time step control.

REFERENCES

[1] P. ANGOT AND R. CHEAYTOU, *Vector penalty-projection method for incompressible fluid flows with open boundary conditions*, in ALGORITMY 2012, 19th Conference on Scientific Computing, Vysoké Tatry–Podbanské, Slovakia, A. Handlovičová, Z. Minarechová, and D. Ševčovič, eds., Slovak University of Technology in Bratislava, Bratislava, Slovakia, 2012, pp. 219–229.

[2] P. ANGOT, J. CALTAGIRONE, AND P. FABRIE, *A fast vector penalty-projection method for incompressible non-homogeneous or multiphase Navier-Stokes problems*, Appl. Math. Lett., 25 (2012), pp. 1681–1688.

[3] P. ANGOT, J. CALTAGIRONE, AND P. FABRIE, *A new fast method to compute saddle-points in constrained optimization and applications*, Appl. Math. Lett., 25 (2012), pp. 245–251.

[4] P. ANGOT, M. JOBELIN, AND J.-C. LATCHÉ, *Error analysis of the penalty-projection method for the time dependent Stokes equations*, Int. J. Finite Vol., 6 (2009).

[5] M. BENZI, M. A. OLSHANSKII, AND Z. WANG, *Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations*, Internat. J. Numer. Methods Fluids, 66 (2011), pp. 486–508.

[6] R. BLAHETA, *Displacement decomposition-incomplete factorization preconditioning techniques for linear elasticity problems*, Numer. Linear Algebra Appl., 1 (1994), pp. 107–128.

[7] D. BROWN, R. CORTEZ, AND M. MINION, *Accurate projection methods for the incompressible Navier-Stokes equations*, J. Comput. Phys., 168 (2001), pp. 464–499.

[8] A. Chorin, *A numerical method for solving incompressible viscous flow problems*, J. Comput. Phys., 2 (1967), pp. 12–26.
[9] W. E and J.-G. Liu, *Gauge method for viscous incompressible flows*, Commun. Math. Sci., 1 (2003), pp. 317–332.
[10] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier-Stokes Equations. Theory and Algorithms*, Springer Ser. Computat. Math., Springer-Verlag, Berlin, 1986.
[11] R. Glowinski, *Finite Element Methods for Incompressible Viscous Flow*, Handb. Numer. Anal. 9, Elsevier, Amsterdam, 2003.
[12] J.-L. Guermond and P. D. Minev, *A new class of massively parallel direction splitting for the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 2083–2093.
[13] J.-L. Guermond and J. Shen, *A new class of truly consistent splitting schemes for incompressible flows*, J. Comput. Phys., 192 (2003), pp. 262–276.
[14] J.-L. Guermond and J. Shen, *On the error estimates for the rotational pressure-correction projection methods*, Math. Comp., 73 (2004), pp. 1719–1737.
[15] J. L. Guermond, P. Minev, and J. Shen, *Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions*, SIAM J. Numer Anal., 43 (2005), pp. 239–258.
[16] J.-L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 6011–6045.
[17] T. Heister and G. Rapin, *Efficient augmented Lagrangian-type preconditioning for the Oseen problem using grad-div stabilization*, Internat. J. Numer. Methods Fluids, 71 (2013), pp. 118–134.
[18] J. Jia and J. Liu, *Stable and spectrally accurate schemes for the Navier–Stokes equations*, SIAM J. Sci. Comput., 33 (2011), pp. 2421–2439.
[19] M. Jobelin, C. Lapuerta, J.-C. Latché, P. Angot, and B. Piar, *A finite element penalty-projection method for incompressible flows*, J. Comput. Phys., 217 (2006), pp. 502–518.
[20] G. Karniadakis, M. Israeli, and S. Orszag, *High-order splitting methods for the incompressible Navier-Stokes equations*, J. Comput. Phys., 97 (1991), pp. 414–443.
[21] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations*, J. Comput. Phys., 59 (1985), pp. 308–323.
[22] W. Kress and B. Gustafsson, *Deferred correction methods for initial boundary value problems*, J. Sci. Comput., 17 (2002), pp. 241–251.
[23] O. A. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, 2nd ed. rev. extended, Nauka, Moscow, 1970 (in Russian).
[24] A. Linke, M. Neilan, L. Rebholz, and N. Wilson, *Improving Efficiency of Coupled Schemes for Navier-Stokes Equations by a Connection to Grad-Div Stabilized Projection Methods*, Technical report 1835, Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS), Leibniz-Institut im Forschungsverbund Berlin, 2013.
[25] M. L. Minion, *Semi-implicit projection methods for incompressible flow based on spectral deferred corrections*, Appl. Numer. Math., 48 (2004), pp. 369–387.
[26] M. Olshanskii, G. Lube, T. Heister, and J. Löwe, *Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 3975–3988.
[27] S. A. Orszag, M. Israeli, and M. Deville, *Boundary conditions for incompressible flows*, J. Sci. Comput., 1 (1986), pp. 75–111.
[28] A. Poux, S. Glockner, E. Ahusborde, and M. Azaïez, *Open boundary conditions for the velocity-correction scheme of the Navier-Stokes equations*, Comput. & Fluids, 70 (2012), pp. 29–43.
[29] O. Schenk, M. Bollhöfer, and R. A. Römer, *On large-scale diagonalization techniques for the Anderson model of localization*, SIAM Rev., 50 (2008), pp. 91–112.
[30] L. R. Scott and M. Vogelius, *Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials*, RAIRO Model. Math. Anal. Numer., 19 (1985), pp. 111–143.
[31] J. Shen, *A remark on the projection-3 method*, Internat. J. Numer. Methods Fluids, 16 (1993), pp. 249–253.
[32] J. Shen, *On error estimates of the penalty method for unsteady Navier–Stokes equations*, SIAM J. Numer. Anal., 32 (1995), pp. 386–403.
[33] H. J. Stetter, *The defect correction principle and discretization methods*, Numer. Math., 29 (1978), pp. 425–443.
[34] R. Temam, *Une méthode d'approximation de la solution des équations de Navier-Stokes*, Bull. Soc. Math. France, 96 (1968), pp. 115–152.

[35] R. Temam, *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires* (I), Arch. Ration. Mech. Anal., 32 (1969), pp. 135–153.

[36] L. Timmermans, P. Minev, and F. van de Vosse, *An approximate projection scheme for incompressible flow using spectral elements*, Internat. J. Numer. Methods Fluids, 22 (1996), pp. 673–688.

[37] N. Vladimirova, B. Kuznetsov, and N. Yanenko, *Numerical calculation of the symmetrical flow of viscous incompressible liquid around a plate*, in Some Problems in Computational and Applied Mathematics, Nauka, Novosibirsk, 1966 (in Russian).

[38] N. N. Yanenko, *The Method of Fractional Steps. The Solution of Problems of Mathematical Physics in Several Variables*, Springer-Verlag, New York, 1971.

[39] S. Zhang, *A new family of stable mixed finite elements for the* 3*D Stokes equations*, Math. Comp., 74 (2005), pp. 543–554.