SIAM J. SCI. COMPUT.
Vol. 41, No. 2, pp. A770–A788

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

JEAN-LUC GUERMOND† AND PETER MINEV‡

Abstract. In this paper we develop a high-order time stepping technique for the incompressible Navier–Stokes equations. The method is based on an artificial compressibility perturbation made high order by using a Taylor series technique. The method is suitable for time step control. It is unconditionally stable in the case of the unsteady Stokes equations and conditionally stable for the full Navier–Stokes equations. The numerical results presented in the paper suggest that the stability condition in the second case is of CFL type; i.e., the time step should be of the order of the ratio of the meshsize and the magnitude of the velocity. In principle, the technique can be developed to any order in time. We illustrate the idea by giving the third-order version of the methodology. We numerically illustrate the third-order convergence rate of the method on a manufactured solution. The scheme converges with time steps randomly chosen at each time level as the size of the average time step decreases. We also demonstrate the efficiency of a simple time step control on a realistic incompressible flow in 2D.

Key words. differential-algebraic equations, incompressible Navier–Stokes equations, artificial compressibility, high-order time approximation, Taylor series schemes, adaptive time stepping

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

DOI. 10.1137/18M1209301

1. Introduction.

In the literature dedicated to approximations of differential-algebraic equations (DAEs) there are two major approaches used to avoid the simultaneous solution of the discretized ODE system and the discrete algebraic constraint. The first one consists of enforcing the constraint by projecting a consistently produced approximation of the solution of the differential equations onto the manifold defined by the constraint (see, for example, Hairer and Wanner [12, sect. VII.4]). This class of methods is known in the computational fluid dynamics literature as projection methods, the most famous instance being the first-order, nonincremental technique known as the Chorin–Temam method, which appeared in the late 1960s. Unfortunately, all subsequent attempts to increase the order of the method beyond second order in time have been unsuccessful so far, due to the fact that the projection operator incorporates some boundary conditions on the pressure that must be extrapolated in time. All attempts for higher-than-second-order extrapolations known so far compromise the stability of the method (see Guermond, Minev, and Shen [11] for an extended discussion on this issue). The second approach consists of regularizing the constraint by adding a properly chosen perturbation (see, for example, Baumgarte [3]). In the com-

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 23, 2018; accepted for publication (in revised form) January 10, 2019; published electronically March 14, 2019.

http://www.siam.org/journals/sisc/41-2/M120930.html

Funding: This work was supported by the National Science Foundation under grants DMS-1619892 and DMS-1620058, by the Air Force Office of Scientific Research, USAF, under grant FA9550-15-1-0257, by the Army Research Office under grant W911NF-15-1-0517, and by a Discovery grant of the National Science and Engineering Research Council of Canada. The work of the second author was supported by the American Chemical Society Petroleum Research Fund under grant 55484-ND9.

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

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

putational fluid dynamics literature this regularization technique has been proposed in two variants. The incompressibility constraint can be regularized as

$$\epsilon p + \nabla\cdot\mathbf{u} = 0, \tag{1.1}$$

yielding the so-called penalty method (see, for example, Bercovier and Engelman [4]), or, as proposed in Vladimirova, Kuznetsov, and Yanenko [25], Yanenko [27], Temam [24], Ladyzhenskaya [14], Shen [22], it can be regularized as follows:

$$\epsilon \partial_t p + \nabla\cdot\mathbf{u} = 0. \tag{1.2}$$

In (1.1), ϵ is a small positive number that scales like a time over the square of a length, and in (1.2), ϵ is a small positive number that scales like the inverse of the square of a velocity (say the "sound speed"). The main advantage of the perturbation (1.2) over (1.1) is that the linear system for $\mathbf{u}$ that is obtained after substituting p in the discretization of the momentum equation can have a condition number that is significantly smaller than when using (1.1). For example, using $\epsilon \sim \tau/\lambda$, where τ is the time step and λ is a normalizing parameter having the appropriate units (the square of a reference length scale for (1.1) and the square of a velocity times a reference time scale for (1.2)), one obtains in both cases a method that is $\mathcal{O}(\tau)$ accurate. But the algebraic system for $\mathbf{u}$ resulting from the space discretization of the penalty equation (1.1) gives a condition number of order $\mathcal{O}(h^{-2})$, where h is the meshsize, whereas the linear system resulting from the space discretization of the artificial compressibility equation (1.2) gives a condition number that is of order $\mathcal{O}(\tau h^{-2})$, which is the typical condition number that one obtains when approximating the heat equation (see Guermond and Minev [8] for more details). These considerations demonstrate also that any attempt to increase the order of the perturbation in (1.1) or (1.2) by choosing, for example, $\epsilon \sim \tau^k/\lambda$, with $k > 1$, would give a condition number that is of order $\mathcal{O}(\tau^{1-k}h^{-2})$, which would make the linear system difficult to solve. In [8] we proposed a technique to construct time discretizations of any order in time that is based on the perturbation (1.2) and requires only the solution of optimally conditioned linear systems. It employs the defect-correction approach to reduce both the order of the perturbation and the time-discretization error. As proven in [8], each substep in this algorithm leads to a decrease in the overall error by one order in τ. A procedure for reducing the perturbation, also based on (1.2), has been proposed for linear and nonlinear DAEs by Asher and Lin [1, 2], and a particular adaptation of this idea to the incompressible Navier–Stokes equations has been proposed by Lin [16]. But, unlike our approach, it uses an iterative penalty method, essentially an implicit version of the augmented Lagrangian iteration, with the overall accuracy not exceeding first order in time. Moreover, the parameter ϵ that is used in [16] to obtain stability yields linear systems with condition numbers of order $\mathcal{O}(h^{-2})$ or higher.

The method proposed in [8] has been further extended in [9] so as to avoid the appearance of the grad-div operator in the linear system resulting from the discretization of the momentum equations. In [8, 9], we demonstrated the accuracy of the so-constructed schemes using a third-order variant. To our knowledge, this is the first noniterative scheme (i.e., not based on Uzawa or augmented Lagrangian iterations) that can achieve higher-than-second-order accuracy in time on the velocity in the H^1-norm and on pressure in the L^2-norm. An additional advantage of this scheme is that it does not involve an elliptic problem for the pressure that has a condition number scaling like $O(h^{-2})$; the method requires instead solving a set of parabolic problems for the velocity with condition numbers scaling like $O(\tau h^{-2})$, which is significantly better than $O(h^{-2})$. This may turn out to be important for the parallel implementation of these schemes.

Unfortunately, all the schemes proposed in [8, 9] with a convergence rate in time higher-than-first are not suitable for adaptive time stepping and time step control since they involve more than two time levels. The objective of the present paper is to present an alternative version of these schemes that is high order in time (second order and higher) and involves only two time levels, which makes it suitable for time step control. The new idea consists of using some variations of the so-called Taylor method for time discretization modified so as to avoid the explicit differentiation of the nonlinear terms. To the best of our knowledge, the proposed approach is new and allows one to construct schemes of any order in time using only two time levels.

The paper is organized as follows. In section 2 we introduce the problem and establish the notation used in the paper. We also recall the third-order algorithm proposed in [8] for completeness. The main contribution of the paper is in section 3, where we introduce the novel high-order two-level technique mentioned above; see (3.9)–(3.11) or (3.12)–(3.14). We discuss in section 4 some linear algebra details that facilitate the implementation of the method. The performance of the proposed method is illustrated in section 5 using the marker-and-cell (MAC) technique on uniform Cartesian grids and mixed $\mathbb{P}_2/\mathbb{P}_1$ finite elements on nonuniform triangular grids. We confirm therein that the proposed method is indeed third-order accurate in time and performs well with adaptive time stepping. Concluding remarks are reported in section 6.

2. Preliminaries.

In this section we introduce the model problem along with the notation that is used in the paper. We also recall the third-order algorithm proposed in [8] since it is our starting point.

2.1. The model problem.

The incompressible Navier–Stokes equations can be recast in the following abstract operator form:

$$\partial_t \mathbf{u} + \mathbf{N}(\mathbf{u}) + \mathbf{A}\mathbf{u} - \mathbf{B}^{\mathsf{T}} p = \mathbf{f}(t), \qquad \mathbf{B}\mathbf{u} = 0, \qquad \mathbf{u}_{|t=0} = \mathbf{v}_0, \tag{2.1}$$

where $\mathbf{A} : \mathbf{V} \to \mathbf{V}'$ and $\mathbf{B} : \mathbf{V} \to M$ are linear operators, and $\mathbf{N}(\mathbf{u})$ is a nonlinear operator. Here $\mathbf{V} \hookrightarrow \mathbf{H} \equiv \mathbf{H}' \hookrightarrow \mathbf{V}'$ is a Gelfand triple and $\mathbf{M} \equiv \mathbf{M}'$ is a Hilbert space. $\mathbf{f}$ is some source term and $\mathbf{v}_0$ is the initial data. The boundary conditions are encoded in the spaces $\mathbf{V}$ and M and in the definitions of the operators $\mathbf{A}$ and $\mathbf{B}$. Also, $\mathbf{B}^{\mathsf{T}}$ denotes the adjoint of $\mathbf{B}$. We assume that the above problem makes sense and there is a unique solution, say $\mathbf{u} \in L^2((0,\infty);\mathbf{V})$, $\mathbf{N}(\mathbf{u}) \in L^2((0,\infty);\mathbf{V}')$, $\partial_t \mathbf{u} \in L^2((0,\infty);\mathbf{V}')$, and $p \in L^2((0,\infty);M)$. It is not the purpose of this paper to discuss the validity of these assumptions. One can think of (2.1) either as a time-dependent partial differential equation (PDE), in which case $\mathbf{V}$, $\mathbf{H}$, and M are functional spaces (Sobolev or otherwise), or as a nonlinear system of ODEs (possibly resulting from the application of the method of lines to a time-dependent PDE), in which case $\mathbf{V}$, $\mathbf{H}$, and M are finite-dimensional vectors spaces. Although the above setting is quite general and may model many physical situations, we henceforth refer to (2.1) as the Navier–Stokes system, and we refer to $\mathbf{u}$ and p as the velocity and the pressure, respectively.

The problem we investigate in the present paper consists of approximating (2.1) in time so that the velocity and the pressure are uncoupled and the resulting scheme is high-order accurate with respect to τ and produces reasonable condition numbers when approximated in space.

2.2. The multistep artificial compressibility method.

We now recall the third-order method proposed in [8]. Let τ be some time step. Let $t^n \geq 0$ be the current time, and let $t^{n+1} = t^n + \tau$. We denote by $\mathbf{f}^{n+1}$, $\mathbf{f}^n$, ..., the quantities $\mathbf{f}(t^{n+1})$, $\mathbf{f}(t^n)$, ..., if $\mathbf{f}$ is continuous in time, or some approximations thereof otherwise. Let λ be a normalizing parameter independent of τ and having the unit of the square of a velocity times a reference time scale. It is shown in [8] that the sequences $(\mathbf{u}_0^n, p_0^n)_{n\geq 0}$, $(\mathbf{u}_0^n + \tau\mathbf{u}_1^n, p_0^n + \tau p_1^n)_{n\geq 1}$, $(\mathbf{u}_0^n + \tau\mathbf{u}_1^n + \tau^2\mathbf{u}_2^n, p_0^n + \tau p_1^n + \tau^2 p_2^n)_{n\geq 2}$ are, respectively, first-order, second-order, and third-order approximations in time of the solution to (2.1), with

$$\begin{cases} \dfrac{\mathbf{u}_0^{n+1} - \mathbf{u}_0^n}{\tau} + \mathbf{A}\mathbf{u}_0^{n+1} + \lambda\mathbf{B}^\mathsf{T}\mathbf{B}\mathbf{u}_0^{n+1} - \mathbf{B}^\mathsf{T} p_0^n = \mathbf{f}^{n+1} - \mathbf{N}(\mathbf{u}_0^n), \\ p_0^{n+1} = p_0^n - \lambda\mathbf{B}\mathbf{u}_0^{n+1}, \quad d\mathbf{u}_0^{n+1} = \dfrac{\mathbf{u}_0^{n+1} - \mathbf{u}_0^n}{\tau}, \quad dp_0^{n+1} = \dfrac{p_0^{n+1} - p_0^n}{\tau}, \end{cases} \tag{2.2}$$

$$\begin{cases} \mathbf{nl}_1^n = \mathbf{N}(\mathbf{u}_0^n + \tau\mathbf{u}_1^{n-1}),\ d^2\mathbf{u}_0^{n+1} = \dfrac{d\mathbf{u}_0^{n+1} - d\mathbf{u}_0^n}{\tau}, \\ \dfrac{\mathbf{u}_1^n - \mathbf{u}_1^{n-1}}{\tau} + \mathbf{A}\mathbf{u}_1^n + \lambda\mathbf{B}^\mathsf{T}\mathbf{B}\mathbf{u}_1^n - \mathbf{B}^\mathsf{T}(p_1^{n-1} + dp_0^n) \\ \qquad\qquad\qquad\qquad = -\dfrac{1}{2}d^2\mathbf{u}_0^{n+1} - \dfrac{\mathbf{nl}_1^n - \mathbf{nl}_0^n}{\tau}, \\ p_1^n = p_1^{n-1} + dp_0^n - \lambda\mathbf{B}\mathbf{u}_1^n, \quad d\mathbf{u}_1^n = \dfrac{\mathbf{u}_1^n - \mathbf{u}_1^{n-1}}{\tau}, \quad dp_1^n = \dfrac{p_1^n - p_1^{n-1}}{\tau}, \end{cases} \tag{2.3}$$

$$\begin{cases} \mathbf{nl}_2^{n-1} = \mathbf{N}(\mathbf{u}_0^{n-1} + \tau\mathbf{u}_1^{n-1} + \tau^2\mathbf{u}_2^{n-2}), \\ d^2\mathbf{u}_1^n = \dfrac{d\mathbf{u}_1^n - d\mathbf{u}_1^{n-1}}{\tau},\ d^3\mathbf{u}_0^{n+1} = \dfrac{d^2\mathbf{u}_0^{n+1} - d^2\mathbf{u}_0^n}{\tau}, \\ \dfrac{\mathbf{u}_2^{n-1} - \mathbf{u}_2^{n-2}}{\tau} + \mathbf{A}\mathbf{u}_2^{n-1} + \lambda\mathbf{B}^\mathsf{T}\mathbf{B}\mathbf{u}_2^{n-1} - \lambda\mathbf{B}^\mathsf{T}(p_2^{n-2} + dp_1^{n-1}), \\ \qquad\qquad\qquad = -\dfrac{1}{2}d^2\mathbf{u}_1^n + \dfrac{1}{6}d^3\mathbf{u}_0^{n+1} - \dfrac{\mathbf{nl}_2^{n-1} - \mathbf{nl}_1^{n-1}}{\tau^2}, \\ p_2^{n-1} = p_2^{n-2} + dp_1^{n-1} - \lambda\mathbf{B}\mathbf{u}_2^{n-1}, \\ \mathbf{u}^{n-1} = \mathbf{u}_0^{n-1} + \tau\mathbf{u}_1^{n-1} + \tau^2\mathbf{u}_2^{n-1}, \quad p^{n-1} = p_0^{n-1} + \tau p_1^{n-1} + \tau^2 p_2^{n-1}. \end{cases} \tag{2.4}$$

The basic stability and accuracy of this scheme have been established in Lemma 5.2 in [8]; more precisely, the method (2.2)–(2.4) is third-order accurate in time and is unconditionally stable if $\mathbf{N}(\mathbf{u}) \equiv 0$. The numerical results reported in Table 6.2 in [8] also suggest that the method is stable for reasonably large Courant numbers.

The above method achieves high-order accuracy in time by relying on finite difference approximations of the time derivatives in the Taylor expansions of the solution around the previous time level. Therefore, the algorithm is intrinsically a multistep method, which makes it unsuitable for time step adaption. In fact, all the efforts we have made since we published [8] to develop a robust time step control algorithm for this method have failed. The main objective of this paper is to overcome this difficulty.

3. The Taylor series technique.

In this section, we revisit the above algorithm (2.2)–(2.4), but instead of approximating the various time derivatives by using finite differences (which makes the algorithm a multistep method), we are going to solve separate evolutionary equations for each of the necessary derivatives.

3.1. Taylor series method. In this section we use the Taylor series method to develop a third-order two-step algorithm for the approximation in time of (2.1). The method can be extended to any order in time.

Methods for approximating the solutions of ODEs, or DAEs, based on truncating the Taylor series expansion of the solution are not widely used in the literature, as compared to, for example, the Runge–Kutta (RK) methods. Important contributions to the development of methods based on Taylor series can be found in Corliss and Chang [6], Pryce [21], Nedialkov and Pryce [18], Nedialkov and Pryce [19], and Tan, Nedialkov, and Pryce [23]. The idea of the Taylor series method for ODEs is to first formulate a recursive system of ODEs satisfied by the solution of the original ODE and its derivatives to a given order, then approximate this system with a method of appropriate accuracy, and finally recover high-order accuracy by using Taylor series expansions.

Let us apply the Taylor series strategy to the system (2.1) up to the third order. How to extend the method to any order will be clear at the end of this section, and we leave further extensions to fourth and higher orders to the interested reader. Denoting by $\mathbf{u}_l$, p_l, $\mathbf{f}_l$ the lth-order partial derivative with respect to t of $\mathbf{u}$, p, and $\mathbf{f}$, i.e., $\mathbf{u}_l := \partial_t^l \mathbf{u}$, $p_l := \partial_t^l p$, $\mathbf{f}_l := \partial_t^l \mathbf{f}$, for $l \in \{0,1,2\}$, we have

$$\partial_t \mathbf{u}_2 + \partial_{tt}\mathbf{N}(\mathbf{u}_0) + \mathbf{A}\mathbf{u}_2 - \mathbf{B}^{\mathsf{T}} p_2 = \mathbf{f}_2(t), \quad \mathbf{B}\mathbf{u}_2 = 0, \tag{3.1}$$

$$\partial_t \mathbf{u}_1 + \partial_{t}\mathbf{N}(\mathbf{u}_0) + \mathbf{A}\mathbf{u}_1 - \mathbf{B}^{\mathsf{T}} p_1 = \mathbf{f}_1(t), \quad \mathbf{B}\mathbf{u}_1 = 0, \tag{3.2}$$

$$\partial_t \mathbf{u}_0 + \mathbf{N}(\mathbf{u}_0) + \mathbf{A}\mathbf{u}_0 - \mathbf{B}^{\mathsf{T}} p_0 = \mathbf{f}_0(t), \quad \mathbf{B}\mathbf{u}_0 = 0. \tag{3.3}$$

Furthermore, if we discretize the equation for the second derivative (3.1) by a first-order scheme, i.e., approximating $\partial_t \mathbf{u}_2$ by a first divided difference, then the result can be used to approximate the derivative $\partial_t \mathbf{u}_1$ by the following second-order accurate Taylor expansion:

$$\frac{\partial \mathbf{u}_1^{n+1}}{\partial t} = \frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\tau} + \frac{\tau}{2}\frac{\partial \mathbf{u}_2^{n+1}}{\partial t} + O(\tau^2) = \frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\tau} + \frac{\tau}{2}\frac{\mathbf{u}_2^{n+1} - \mathbf{u}_2^n}{\tau} + O(\tau^2). \tag{3.4}$$

This approximation can in turn be used to approximate the first derivative of $\mathbf{u}_0$ using the third-order Taylor expansion:

$$\begin{aligned}\frac{\partial \mathbf{u}_0^{n+1}}{\partial t} &= \frac{\mathbf{u}_0^{n+1} - \mathbf{u}_0^n}{\tau} + \frac{\tau}{2}\frac{\partial \mathbf{u}_1^{n+1}}{\partial t} - \frac{\tau^2}{6}\frac{\partial \mathbf{u}_2^{n+1}}{\partial t} + O(\tau^3)\\ &= \frac{\mathbf{u}_0^{n+1} - \mathbf{u}_0^n}{\tau} + \frac{\tau}{2}\left(\frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\tau} + \frac{\tau}{2}\frac{\mathbf{u}_2^{n+1} - \mathbf{u}_2^n}{\tau}\right) - \frac{\tau^2}{6}\frac{\mathbf{u}_2^{n+1} - \mathbf{u}_2^n}{\tau} + O(\tau^3)\\ &= \frac{\mathbf{u}_0^{n+1} - \mathbf{u}_0^n}{\tau} + \frac{\tau}{2}\frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\tau} + \frac{\tau^2}{12}\frac{\mathbf{u}_2^{n+1} - \mathbf{u}_2^n}{\tau} + O(\tau^3).\end{aligned} \tag{3.5}$$

The derivatives of the nonlinear terms in (3.1)–(3.2) can be computed directly by evaluating exactly the Jacobian and Hessian of $\mathbf{N}$. Alternatively, these derivatives can be approximated by using finite differences of the corresponding order, and, since this approach is more universal, we adopt it here. We also choose to discretize the nonlinear terms explicitly in order to avoid solving nonlinear algebraic problems. Notice, however, that using an implicit discretization of the nonlinear term is perfectly admissible if desired, and we leave these possible extensions to the interested reader. The resulting third-order Taylor series scheme for the incompressible Navier–Stokes equations can then be written as follows:

$$\begin{aligned}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} &+ \frac{\mathbf{N}(\mathbf{u}_0^n+2\tau\mathbf{u}_1^n+2\tau^2\mathbf{u}_2^n)-2\mathbf{N}(\mathbf{u}_0^n+\tau\mathbf{u}_1^n+\frac{\tau^2}{2}\mathbf{u}_2^n)+\mathbf{N}(\mathbf{u}_0^n)}{\tau^2}\\ &+\mathbf{A}\mathbf{u}_2^{n+1}-\mathbf{B}^{\mathsf{T}}p_2^{n+1}=\mathbf{f}_2^{n+1}, \qquad \mathbf{B}\mathbf{u}_2^{n+1}=0,\end{aligned}\tag{3.6}$$

$$\begin{aligned}\frac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau}+\frac{\tau}{2}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} &+ \frac{\mathbf{N}(\mathbf{u}_0^n+2\tau\mathbf{u}_1^n+2\tau^2\mathbf{u}_2^n)-\mathbf{N}(\mathbf{u}_0^n)}{2\tau}\\ &+\mathbf{A}\mathbf{u}_1^{n+1}-\mathbf{B}^{\mathsf{T}}p_1^{n+1}=\mathbf{f}_1^{n+1}, \qquad \mathbf{B}\mathbf{u}_1^{n+1}=0,\end{aligned}\tag{3.7}$$

$$\begin{aligned}\frac{\mathbf{u}_0^{n+1}-\mathbf{u}_0^n}{\tau}+\frac{\tau}{2}\frac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau}+\frac{\tau^2}{12}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} &+\mathbf{N}\left(\mathbf{u}_0^n+\tau\mathbf{u}_1^n+\frac{\tau^2}{2}\mathbf{u}_2^n\right)\\ &+\mathbf{A}\mathbf{u}_0^{n+1}-\mathbf{B}^{\mathsf{T}}p_0^{n+1}=\mathbf{f}_0^{n+1}, \qquad \mathbf{B}\mathbf{u}_0^{n+1}=0.\end{aligned}\tag{3.8}$$

The above algorithm has to be properly initialized since the original problem provides no initial conditions on the time derivatives of the solution. We propose an algorithm for the initialization of (3.6)–(3.8) in section 3.3.

3.2. Taylor series method with bootstrapping.

Now we attack the real issue under investigation in this paper, which consists of approximating properly the constraints $\mathbf{B}\mathbf{u}_0^{n+1}=0$, $\mathbf{B}\mathbf{u}_1^{n+1}=0$, $\mathbf{B}\mathbf{u}_2^{n+1}=0$. This will be done by using the bootstrapping technique introduced in Guermond and Minev [8].

We first replace $\mathbf{B}\mathbf{u}_2^{n+1}=0$ by $\epsilon\partial_t p_2^{n+1}+\mathbf{B}\mathbf{u}_2^{n+1}=0$, which is an $\mathcal{O}(\epsilon)$ perturbation of the exact constraint. Then, setting $\epsilon:=\tau/\lambda$ and using first-order differences to approximate $\partial_t p_2^{n+1}$, we replace the constraint $\mathbf{B}\mathbf{u}_2^{n+1}=0$ by $p_2^{n+1}=p_2^n-\lambda\mathbf{B}\mathbf{u}_2^{n+1}$. Notice that under appropriate smoothness assumptions, we have $\mathbf{B}\mathbf{u}_2^{n+1}=\mathcal{O}(\tau)$. Hence, the sequence $(\mathbf{u}_2^n,p_2^n)_{n\geq 0}$ is a first-order approximation in τ of $(\partial_{tt}\mathbf{u},\partial_{tt}p)$. This statement is proved in Theorem 4.3 in Guermond and Minev [8] without the nonlinear term (and with $r\equiv 0$ therein); it is also proved in Theorem 3.1 in Shen [22]. In conclusion, we can replace (3.6) by

$$\begin{aligned}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} &+ \frac{\mathbf{N}(\mathbf{u}_0^n+2\tau\mathbf{u}_1^n+2\tau^2\mathbf{u}_2^n)-2\mathbf{N}(\mathbf{u}_0^n+\tau\mathbf{u}_1^n+\frac{\tau^2}{2}\mathbf{u}_2^n)+\mathbf{N}(\mathbf{u}_0^n)}{\tau^2}\\ &+\mathbf{A}\mathbf{u}_2^{n+1}-\mathbf{B}^{\mathsf{T}}p_2^{n+1}=\mathbf{f}_2^{n+1}, \qquad p_2^{n+1}=p_2^n-\lambda\mathbf{B}\mathbf{u}_2^{n+1}.\end{aligned}\tag{3.9}$$

We now use the bootstrapping technique to replace the constraint $\mathbf{B}\mathbf{u}_1^{n+1}=0$ in (3.7) by $\epsilon(\partial_t p_1^{n+1}-p_2^{n+1})+\mathbf{B}\mathbf{u}_1^{n+1}=0$. Notice that now we have $\mathbf{B}\mathbf{u}_1^{n+1}=\mathcal{O}(\epsilon\tau)$ by virtue of Theorem 4.3 in Guermond and Minev [8] (with $\partial_t r\equiv p_2$), since we have established above that the sequence $(p_2^n)_{n\geq}$ is an $\mathcal{O}(\tau)$ approximation of $\partial_t p_1=\partial_{tt}p$. Replacing $\partial_t p_1^{n+1}$ by the first-order approximation $\frac{p_1^{n+1}-p_1^n}{\tau}$ and ϵ by τ/λ gives $p_1^{n+1}=p_1^n+\tau p_2^{n+1}-\lambda\mathbf{B}\mathbf{u}_1^{n+1}$, and now we have $\mathbf{B}\mathbf{u}_1^{n+1}=\mathcal{O}(\tau^2)$ because $\frac{p_1^{n+1}-p_1^n}{\tau}=\partial_t p_1^{n+1}+\mathcal{O}(\tau)$. Hence, the sequence $(\mathbf{u}_1^n,p_1^n)_{n\geq 0}$ is a second-order approximation in τ of $(\partial_t\mathbf{u},\partial_t p)$. In conclusion, we can replace (3.7) by

$$\begin{aligned}\frac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau}+\frac{\tau}{2}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} &+ \frac{\mathbf{N}(\mathbf{u}_0^n+2\tau\mathbf{u}_1^n+2\tau^2\mathbf{u}_2^n)-\mathbf{N}(\mathbf{u}_0^n)}{2\tau}\\ &+\mathbf{A}\mathbf{u}_1^{n+1}-\mathbf{B}^{\mathsf{T}}p_1^{n+1}=\mathbf{f}_1^{n+1}, \qquad p_1^{n+1}=p_1^n+\tau p_2^{n+1}-\lambda\mathbf{B}\mathbf{u}_1^{n+1}.\end{aligned}\tag{3.10}$$

Finally we replace $\mathbf{B}\mathbf{u}_0^{n+1} = 0$ in (3.8) by $\epsilon(\partial_t p_0^{n+1} - p_1^{n+1}) + \mathbf{B}\mathbf{u}_0^{n+1} = 0$. Since the sequence $(p_1^n)_{n\geq 0}$ is an $\mathcal{O}(\tau^2)$ approximation of $\partial_t p_0$, by virtue of Theorem 4.3 in [8] (with $\partial_t r \equiv p_1$), we infer that $\mathbf{B}\mathbf{u}_0^{n+1} = \mathcal{O}(\epsilon\tau^2)$. Now, using the Taylor expansion $p_0^n = p_0^{n+1} - \tau\partial p_0^{n+1} + \frac{1}{2}\tau^2\partial_{tt}p_0^{n+1} + \mathcal{O}(\tau^2)$, we replace $\partial_t p_0^{n+1}$ by the second-order approximation $\frac{p_0^{n+1}-p_0^n}{\tau} + \frac{1}{2}\tau p_2^{n+1}$. Then, using $\epsilon := \tau/\lambda$, the constraint $\mathbf{B}\mathbf{u}_0^{n+1} = 0$ is replaced by $p_0^{n+1} = p_0^n + \tau p_1^{n+1} - \frac{1}{2}\tau^2 p_2^{n+1} - \lambda\mathbf{B}\mathbf{u}_0^{n+1}$. Now we have $\mathbf{B}\mathbf{u}_0^{n+1} = \mathcal{O}(\tau^3)$ because $\frac{p_0^{n+1}-p_0^n}{\tau} + \frac{1}{2}\tau p_2^{n+1} = \partial_t p_0^{n+1} + \mathcal{O}(\tau^2)$. Hence, the sequence $(\mathbf{u}_1^n, p_1^n)_{n\geq 0}$ is a third-order approximation in τ of $(\mathbf{u}, p)$. Hence we replace (3.8) by

$$\begin{aligned}\frac{\mathbf{u}_0^{n+1}-\mathbf{u}_0^n}{\tau} + \frac{\tau}{2}\frac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau} + \frac{\tau^2}{12}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} + \mathbf{N}\left(\mathbf{u}_0^n + \tau\mathbf{u}_1^n + \frac{\tau^2}{2}\mathbf{u}_2^n\right)&\\ + \mathbf{A}\mathbf{u}_0^{n+1} - \mathbf{B}^{\mathsf{T}}p_0^{n+1} = \mathbf{f}_0^{n+1}, \qquad p_0^{n+1} = p_0^n + \tau p_1^{n+1} - \frac{\tau^2}{2}p_2^{n+1} - \lambda\mathbf{B}\mathbf{u}_0^{n+1}&.\end{aligned} \tag{3.11}$$

Notice that the velocity and the pressure are indeed uncoupled in (3.9)–(3.11). Indeed, using that $p_2^{n+1} = p_2^n - \lambda\mathbf{B}\mathbf{u}_2^{n+1}$, $p_1^{n+1} = p_1^n + \tau p_2^{n+1} - \lambda\mathbf{B}\mathbf{u}_1^{n+1}$, and $p_0^{n+1} = p_0^n + \tau p_1^{n+1} - \frac{\tau^2}{2}p_2^{n+1} - \lambda\mathbf{B}\mathbf{u}_0^{n+1}$, one observes that the momentum equations in (3.9)–(3.11) can be solved as follows:

$$\begin{aligned}\frac{\mathbf{u}_2^{n+1}}{\tau} + \mathbf{A}\mathbf{u}_2^{n+1} + \lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}_2^{n+1} = \frac{\mathbf{u}_2^n}{\tau} + \mathbf{f}_2^{n+1} + \mathbf{B}^{\mathsf{T}}p_2^n&\\ - \frac{\mathbf{N}(\mathbf{u}_0^n + 2\tau\mathbf{u}_1^n + 2\tau^2\mathbf{u}_2^n) - 2\mathbf{N}(\mathbf{u}_0^n + \tau\mathbf{u}_1^n + \frac{\tau^2}{2}\mathbf{u}_2^n) + \mathbf{N}(\mathbf{u}_0^n)}{\tau^2}&,\end{aligned} \tag{3.12}$$

$$\begin{aligned}\frac{\mathbf{u}_1^{n+1}}{\tau} + \mathbf{A}\mathbf{u}_1^{n+1} + \lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}_1^{n+1} = \frac{\mathbf{u}_1^n}{\tau} + \mathbf{f}_1^{n+1} + \mathbf{B}^{\mathsf{T}}(p_1^n + \tau p_2^{n+1})&\\ - \frac{\tau}{2}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} - \frac{\mathbf{N}(\mathbf{u}_0^n + 2\tau\mathbf{u}_1^n + 2\tau^2\mathbf{u}_2^n) - \mathbf{N}(\mathbf{u}_0^n)}{2\tau}&,\end{aligned} \tag{3.13}$$

$$\begin{aligned}\frac{\mathbf{u}_0^{n+1}}{\tau} + \mathbf{A}\mathbf{u}_0^{n+1} + \lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}_0^{n+1} = \frac{\mathbf{u}_0^n}{\tau} + \mathbf{f}_0^{n+1} + \mathbf{B}^{\mathsf{T}}\left(p_0^n + \tau p_1^{n+1} - \frac{\tau^2}{2}p_2^{n+1}\right)&\\ - \frac{\tau}{2}\frac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau} - \frac{\tau^2}{12}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau} - \mathbf{N}\left(\mathbf{u}_0^n + \tau\mathbf{u}_1^n + \frac{\tau^2}{2}\mathbf{u}_2^n\right)&.\end{aligned} \tag{3.14}$$

The pressure is updated by setting $p_2^{n+1} = p_2^n - \lambda\mathbf{B}\mathbf{u}_2^{n+1}$, $p_1^{n+1} = p_1^n + \tau p_2^{n+1} - \lambda\mathbf{B}\mathbf{u}_1^{n+1}$, and $p_0^{n+1} = p_0^n + \tau p_1^{n+1} - \frac{\tau^2}{2}p_2^{n+1} - \lambda\mathbf{B}\mathbf{u}_0^{n+1}$.

The two key arguments to establish stability and convergence of the algorithm (3.9)–(3.11) when $\mathbf{N}(\mathbf{u}) \equiv 0$ are Theorem 4.3 in Guermond and Minev [8] and Proposition 5.1 in Shen [22]. That $(\mathbf{u}_2^n, p_2^n)_{n\geq 0}$ is a first-order approximation in τ of $(\partial_{tt}\mathbf{u}, \partial_{tt}p)$ is a direct consequence of [22, Prop. 5.1]. After noticing that $(\mathbf{u}_1^n, p_1^n)_{n\geq 0}$ depends linearly on the sequence $(\mathbf{u}_2^n, p_2^n)_{n\geq 0}$, one proves that $(\mathbf{u}_1^n, p_1^n)_{n\geq 0}$ is a second-order approximation in τ of $(\partial_t\mathbf{u}, \partial_t p)$ by invoking [8, Thm. 4.3] combined with [22, Prop. 5.1]. Finally, one proves that $(\mathbf{u}_0^n, p_0^n)_{n\geq 0}$ is a third-order approximation in τ of $(\mathbf{u}, p)$ by invoking [8, Thm. 4.3] combined with [22, Prop. 5.1], and using that $(\mathbf{u}_0^n, p_0^n)_{n\geq 0}$ depends linearly on the sequences $(\mathbf{u}_1^n, p_1^n)_{n\geq 0}$ and $(\mathbf{u}_2^n, p_2^n)_{n\geq 0}$. In conclusion, the algorithm (3.9)–(3.11) is unconditionally stable if $\mathbf{N}(\mathbf{u}) \equiv 0$ and third-order accurate. The numerical results presented below suggest that the method is stable with reasonably large Courant numbers in case of the Navier–Stokes equations.

3.3. Initialization. As mentioned above, the initial values of the derivatives of $\mathbf{u}$ and p are unknown, so the method (3.9)–(3.11) has to be started appropriately by using a lower-order scheme. Note that in order to maintain the overall accuracy of the scheme we need initial values for $(\mathbf{u}_2, p_2)$, $(\mathbf{u}_1, p_1)$, and $(\mathbf{u}_0, p_0)$ that are first, second, and third-order accurate, respectively.

We propose here one possibility to generate the full set of data required at the time $t = \tau$ by combining the Richardson extrapolation technique with the first-order artificial compressibility scheme over the time interval $(0, 2\tau]$. Since the original problem (2.1) does not contain an initial condition for p at the time $t = 0$, this quantity has to be generated by using, for example, the pressure initialization proposed in Guermond, Minev, and Shen [11] (see Hypothesis 3.1 and the subsequent comments therein). Given the initial velocity $\mathbf{v}_0$ and the pressure at $t = 0$, say q_0, let us denote by $(\mathbf{u}^1_{\tau/3}, p^1_{\tau/3})$, $(\mathbf{u}^1_{\tau/2}, p^1_{\tau/2})$, and $(\mathbf{u}^1_{\tau}, p^1_{\tau})$ the solutions obtained at time $t = \tau$ by the first-order artificial compressibility scheme using, respectively, the time steps $\tau/3$, $\tau/2$, and τ. Similarly, let $(\mathbf{u}^2_{\tau/3}, p^2_{\tau/3})$, $(\mathbf{u}^2_{\tau/2}, p^2_{\tau/2})$, and $(\mathbf{u}^2_{\tau}, p^2_{\tau})$ be the solutions obtained at time $t = 2\tau$. Next we generate third-order approximations at τ and 2τ of $(\mathbf{u}, p)$, say $(\mathbf{u}^1_R, p^1_R)$ and $(\mathbf{u}^2_R, p^2_R)$, respectively, using the extrapolation formula $\phi^k_R = \frac{9}{2}\phi^k_{\tau/3} - 4\phi^k_{\tau/2} + \frac{1}{2}\phi^k_{\tau}$, with $\phi = \mathbf{u}$ or $\phi = p$, and $k = 1, 2$. These approximations can be used in turn to generate a first-order approximation to the second derivative of $\mathbf{u}$ and p at $t = \tau$ by means of a second difference i.e., $\partial_{tt}\phi(\tau) = (\phi^2_R - 2\phi^1_R + \phi(0))/\tau^2 + O(\tau)$. Similarly, the first derivative at τ can be approximated by $\partial_t\phi(\tau) = (\phi^2_R - \phi(0))/(2\tau) + O(\tau^2)$, and the value at $t = \tau$ is simply given by $\phi(\tau) = \phi^1_R + O(\tau^3)$. Of course there is no initialization problem if $\mathbf{v}_0 = 0$, $\mathbf{f}(0) = \mathbf{0}$, $\partial_t\mathbf{f}(0) = \mathbf{0}$, and $\partial_{tt}\mathbf{f}(0) = \mathbf{0}$.

Remark 3.1. It is well known that the inconsistency of the initial data with the perturbed incompressibility constraint of the artificial compressibility method leads to the appearance of spurious acoustic modes in the pressure that decrease the accuracy of the approximation close to $t = 0$ (see Ohwada and Asinari [20], DeCaria, Layton, and McLaughlin [7] for a detailed discussion on this issue). These acoustic modes can be filtered using various correction techniques, and we again refer the reader to Ohwada and Asinari [20] and DeCaria, Layton, and McLaughlin [7] for details on the available options. In our opinion, the easiest and computationally least intrusive among them is to use the so-called Robert–Asselin filter (see Williams [26], Hurl et al. [13], Layton, Li, and Trenchea [15], and DeCaria, Layton, and McLaughlin [7]). Essentially, this technique corrects a posteriori the results for the velocity and pressure at each time level using second time differences of these quantities. Notice, though, that none of these techniques is applied in the numerical tests reported at the end of the paper in section 5. The results from our numerical tests are not filtered.

Remark 3.2. As usual for perturbation techniques, the method (3.9)–(3.11) yields an end-of-step velocity that satisfies the incompressibility constraint up to an additional truncation error in time. This might be viewed as a defect by some readers since it is claimed in the literature that a major advantage of projection methods is that they yield an end-of-step velocity field that satisfies the divergence constraint up to machine precision. However, the end-of-step velocity field contains an additional truncation error that is rarely mentioned in the literature: it does not solve the discrete momentum balance equations and does not satisfy the proper boundary condition. In addition, it satisfies "exactly" only the *discrete* incompressibility constraint (i.e., this constraint already contains a spatial truncation error), and the resulting approximation to the velocity *is not pointwise* divergence-free. For a detailed discussion on this issue we refer the reader to the review on projection methods in Guermond, Minev, and Shen [11].

4. Further splitting.

We show in this section that the algorithm (3.9)–(3.11) can be further simplified, and we give implementation details for mixed finite elements.

4.1. Abstract splitting.

Let $\mathbf{C}^{\triangle} : \mathbf{V} \rightarrow \mathbf{V}'$ be another bounded linear operator such that $\frac{1}{\tau}\mathbf{I} + \mathbf{A} + \lambda\mathbf{B}^{\mathsf{T}}\mathbf{B} - \mathbf{C}^{\triangle}$ is a bijective operator for any $\tau > 0$, where $\mathbf{I}$ is the identity operator. Then it has been shown in Guermond and Minev [9] that the defect-correction technique can be further simplified by replacing $\mathbf{A}\mathbf{u}_2^{n+1}$ by $\mathbf{A}\mathbf{u}_2^{n+1} - \mathbf{C}^{\triangle}(\mathbf{u}_2^{n+1} - \mathbf{u}_2^n)$ in (3.9), $\mathbf{A}\mathbf{u}_1^{n+1}$ by $\mathbf{A}\mathbf{u}_1^{n+1} - \mathbf{C}^{\triangle}(\mathbf{u}_1^{n+1} - \mathbf{u}_1^n - \tau\mathbf{u}_2^{n+1})$ in (3.10), and $\mathbf{A}\mathbf{u}_0^{n+1}$ by $\mathbf{C}^{\triangle}(\mathbf{u}_0^{n+1} - \mathbf{u}_0^n - \tau\mathbf{u}_1^{n+1} + \frac{\tau^2}{2}\mathbf{u}_2^{n+1})$ in (3.11). In summary the system (3.12)–(3.14) can be replaced without loss of accuracy by

$$\left(\frac{1}{\tau}\mathbf{I}+\mathbf{A}+\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}-\mathbf{C}^{\triangle}\right)\mathbf{u}_2^{n+1}=\frac{\mathbf{u}_2^n}{\tau}+\mathbf{f}_2^{n+1}+\mathbf{C}^{\triangle}\mathbf{u}_2^n+\mathbf{B}^{\mathsf{T}}p_2^n \\ -\frac{\mathbf{N}(\mathbf{u}_0^n+2\tau\mathbf{u}_1^n+2\tau^2\mathbf{u}_2^n)-2\mathbf{N}(\mathbf{u}_0^n+\tau\mathbf{u}_1^n+\frac{\tau^2}{2}\mathbf{u}_2^n)+\mathbf{N}(\mathbf{u}_0^n)}{\tau^2}, \tag{4.1}$$

$$\left(\frac{1}{\tau}\mathbf{I}+\mathbf{A}+\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}-\mathbf{C}^{\triangle}\right)\mathbf{u}_1^{n+1}=\frac{\mathbf{u}_1^n}{\tau}+\mathbf{f}_1^{n+1}+\mathbf{C}^{\triangle}(\mathbf{u}_1^n-\tau\mathbf{u}_2^{n+1}) \\ +\mathbf{B}^{\mathsf{T}}(p_1^n+\tau p_2^{n+1})-\frac{\tau}{2}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau}-\frac{\mathbf{N}(\mathbf{u}_0^n+2\tau\mathbf{u}_1^n+2\tau^2\mathbf{u}_2^n)-\mathbf{N}(\mathbf{u}_0^n)}{2\tau}, \tag{4.2}$$

$$\left(\frac{1}{\tau}\mathbf{I}+\mathbf{A}+\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}-\mathbf{C}^{\triangle}\right)\mathbf{u}_0^{n+1}=\frac{\mathbf{u}_0^n}{\tau}+\mathbf{f}_0^{n+1}+\mathbf{C}^{\triangle}\left(\mathbf{u}_0^n-\tau\mathbf{u}_1^{n+1}+\frac{\tau^2}{2}\mathbf{u}_2^{n+1}\right) \\ +\mathbf{B}^{\mathsf{T}}\left(p_0^n+\tau p_1^{n+1}-\frac{\tau^2}{2}p_2^{n+1}\right)-\frac{\tau}{2}\frac{\mathbf{u}_1^{n+1}-\mathbf{u}_1^n}{\tau}-\frac{\tau^2}{12}\frac{\mathbf{u}_2^{n+1}-\mathbf{u}_2^n}{\tau}-\mathbf{N}\left(\mathbf{u}_0^n+\tau\mathbf{u}_1^n+\frac{\tau^2}{2}\mathbf{u}_2^n\right). \tag{4.3}$$

The heuristic reason for these modifications to be reasonable and give the same properties as the original algorithm is that $\mathbf{C}^{\triangle}(\mathbf{u}_2^{n+1}-\mathbf{u}_2^n)$ introduces an $\mathcal{O}(\tau)$ perturbation on $\mathbf{u}_2$, which is not a problem since the accuracy that is needed on $\mathbf{u}_2$ to achieve the $\mathcal{O}(\tau^3)$ accuracy on $\mathbf{u}_0$ is only $\mathcal{O}(\tau)$. Then $\mathbf{C}^{\triangle}(\mathbf{u}_1^{n+1} - \mathbf{u}_1^n - \tau\mathbf{u}_2^{n+1})$ introduces an $\mathcal{O}(\tau^2)$ perturbation on $\mathbf{u}_1$, which again is not a problem since the accuracy that is needed on $\mathbf{u}_1$ is only $\mathcal{O}(\tau^2)$. Finally, $\mathbf{C}^{\triangle}(\mathbf{u}_0^{n+1} - \mathbf{u}_0^n - \tau\mathbf{u}_1^{n+1} + \frac{\tau^2}{2}\mathbf{u}_2^{n+1})$ introduces a $\mathcal{O}(\tau^3)$ perturbation on $\mathbf{u}_0$, which again is optimal since we expect the accuracy on $\mathbf{u}_0$ to be $\mathcal{O}(\tau^3)$. We show in the next section that one can use the additional operator $\mathbf{C}^{\triangle}$ to simplify the solution process for the momentum equations in (3.9)–(3.11).

4.2. Splitting of the grad-div operator.

Let us illustrate the technique introduced in section 4.1 in the context of the Navier–Stokes equations in a bounded domain $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$. Let us set $\mathbf{A}\mathbf{u} := -\frac{1}{Re}\Delta\mathbf{u}$, $\mathbf{B}\mathbf{u} := \nabla\cdot\mathbf{u}$, and $\mathbf{B}^{\mathsf{T}}p := -\nabla p$, where all these operators are defined with the appropriate boundary conditions. Then, once the pressure is eliminated from (3.9)–(3.11), one has to solve the linear systems (3.12)–(3.14), each involving the operator $\frac{1}{\tau}\mathbf{I} + \mathbf{A} + \lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}$. Notice that the (negative of the) grad-div operator $\mathbf{B}^{\mathsf{T}}\mathbf{B}$ couples all the Cartesian components of the velocity, which is not the case (a priori) of the operator $\frac{1}{\tau}\mathbf{I} + \mathbf{A}$. Hence, we have uncoupled the velocity and the pressure at the cost of coupling the Cartesian components of the velocity. This coupling can be removed by a block-triangular splitting of the operator

$\mathbf{B}^{\mathsf{T}}\mathbf{B}$. Let ∂_i be the partial derivative with respect to the ith Cartesian coordinate. Let $(u_1,\ldots,u_d)$ be the Cartesian components of the field $\mathbf{u}:\Omega\to\mathbb{R}^d$, and let us set

$$(\mathbf{C}^{\triangle}\mathbf{u})_i := -\lambda\partial_i\Bigg(\sum_{j=i+1}^{d}\partial_j u_j\Bigg) \qquad \forall i\in\{1,\ldots,d\}. \tag{4.4}$$

Then $\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}-\mathbf{C}^{\triangle}$ is a lower triangular operator in the sense that

$$(\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}-\mathbf{C}^{\triangle}\mathbf{u})_i = -\lambda\partial_i\Bigg(\sum_{j=1}^{i}\partial_j u_j\Bigg) \qquad \forall i\in\{1,\ldots,d\}. \tag{4.5}$$

That is to say, $(\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}-\mathbf{C}^{\triangle}\mathbf{u})_1=-\lambda\partial_{11}u_1$, $(\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}-\mathbf{C}^{\triangle}\mathbf{u})_2=-\lambda(\partial_{21}u_1+\partial_{22}u_2)$, and $(\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}\mathbf{u}-\mathbf{C}^{\triangle}\mathbf{u})_3=-\lambda(\partial_{31}u_1+\partial_{32}u_2+\partial_{33}u_3)$.

For instance, the solution of the linear system $(\frac{1}{\tau}\mathbf{I}+\mathbf{A}+\lambda\mathbf{B}^{\mathsf{T}}\mathbf{B}-\mathbf{C}^{\triangle})\mathbf{u}=\mathbf{f}$ in two dimensions with homogeneous Dirichlet conditions and a generic right-hand side $\mathbf{f}$ can be obtained as follows. First we solve the scalar equation

$$\frac{1}{\tau}u_1-\left(\frac{1}{Re}+\lambda\right)\partial_{11}u_1-\frac{1}{Re}\partial_{22}u_1=f_1, \qquad u_{1|\partial\Omega}=0.$$

Then we solve the second scalar equation:

$$\frac{1}{\tau}u_2-\left(\frac{1}{Re}+\lambda\right)\partial_{22}u_2-\frac{1}{Re}\partial_{11}u_2=f_2+\lambda\partial_{21}u_1, \qquad u_{2|\partial\Omega}=0.$$

This splitting facilitates the solution of the system of momentum equations and allows us to easily modify existing codes for the Navier–Stokes equations. Moreover, our numerical experience (see Guermond and Minev [9] and section 5 below) shows that the accuracy of the splitting scheme (4.1)–(4.3) is practically the same as the fully coupled scheme (3.9)–(3.11). We finally refer the reader to Linke and Rebholz [17], where different splittings with similar properties are proposed.

4.3. Finite elements implementation. When using mixed finite elements satisfying the inf-sup condition to solve the Navier–Stokes equations, one must be careful when eliminating the pressure in order to avoid possible locking at high Reynolds numbers or having to invert the pressure mass matrix. More precisely, let us denote by $\mathcal{M}$ the mass matrix for the velocity. Let $\mathcal{N}$ be the mass matrix for the pressure, or its lumped version, or any diagonal matrix with entries equal to the volume of the support of the pressure shape functions. Let $\mathcal{A}$ be the stiffness matrix associated with the operator $\mathbf{A}$. Similarly, we denote by $\mathcal{B}$ the matrix associated with the divergence operator $\mathbf{B}$. Then $\mathcal{B}^{\mathsf{T}}$ is the matrix associated with the negative of the gradient operator.

At every time step, the system (3.9)–(3.11) requires solving linear equations of the form $\frac{1}{\tau}\mathbf{u}+\mathbf{A}\mathbf{u}-\lambda\mathbf{B}^{\mathsf{T}}p=\mathbf{f}$, $p=q-\lambda\mathbf{B}\mathbf{u}$, where $\mathbf{f}$ and q are given. Then the matrix form of the first equation is $(\frac{1}{\tau}\mathcal{M}+\mathcal{A})U-\lambda\mathcal{B}^{\mathsf{T}}P=F$, and the matrix form of the second equation is $\mathcal{N}P=\mathcal{N}Q-\lambda\mathcal{B}U$. Notice here that the exact matrix version of $p=q-\lambda\mathbf{B}\mathbf{u}$ requires that $\mathcal{N}$ be the pressure mass matrix. But this constraint can be relaxed since, without loss of accuracy, instead of approximating $\epsilon\partial_t p+\nabla\cdot\mathbf{u}=0$, we could also approximate the perturbation $\epsilon\partial_t L(p)+\nabla\cdot\mathbf{u}=0$, where $L:M\to M$ is any perturbation of the identity operator. Hence, as said above, instead of using the mass matrix for the pressure, one does not lose the properties of the scheme by using either the lumped mass matrix or any diagonal matrix with entries equal to the volume of the support of the pressure shape functions. In conclusion, one eliminates the pressure in the velocity equation by using $p = q - \lambda\mathcal{N}^{-1}\mathcal{B}U$, and one obtains $(\frac{1}{\tau}\mathcal{M} + \mathcal{A} + \lambda\mathcal{B}^{\mathsf{T}}\mathcal{N}^{-1}\mathcal{B})U = F + \lambda\mathcal{B}^{\mathsf{T}}Q$. We insist again that $\mathcal{N}$ need not be the consistent pressure mass matrix; actually, we recommend using either the lumped mass matrix or any diagonal matrix appropriately scaled.

We finish this section by mentioning that, on special grids, it is possible to avoid the pressure mass matrix altogether by simply approximating the operator $-\lambda\nabla\nabla\cdot$ with the bilinear form $c(\mathbf{u},\mathbf{v}) := \int_\Omega \nabla\cdot\mathbf{u}\nabla\cdot\mathbf{v}\,\mathrm{d}x$ as shown in Case et al. [5].

5. Numerical illustrations.

In this section, we illustrate the performance of the method proposed in the paper with the Navier–Stokes equations: Using the notation of (2.1), from now on we use $\mathbf{Au} := -\frac{1}{Re}\Delta\mathbf{u}$, $\mathbf{Bu} := \nabla\cdot\mathbf{u}$, $-\mathbf{B}^{\mathsf{T}}p := \nabla p$, and $\mathbf{N}(\mathbf{u}) := \mathbf{u}\cdot\nabla\mathbf{u}$. For homogeneous Dirichlet boundary conditions, we have $\mathbf{V} := \mathbf{H}^1_0(\Omega)$, $M := L^2(\Omega)/\mathbb{R}$, where $\Omega \subset \mathbb{R}^d$ is the fluid domain. First we confirm that the convergence rate in time is indeed third order by using a manufactured solution. Then we show that the method performs as expected when the time step is chosen either randomly or adaptively. In all the computation reported below we take $\lambda = 1$.

5.1. Manufactured solutions.

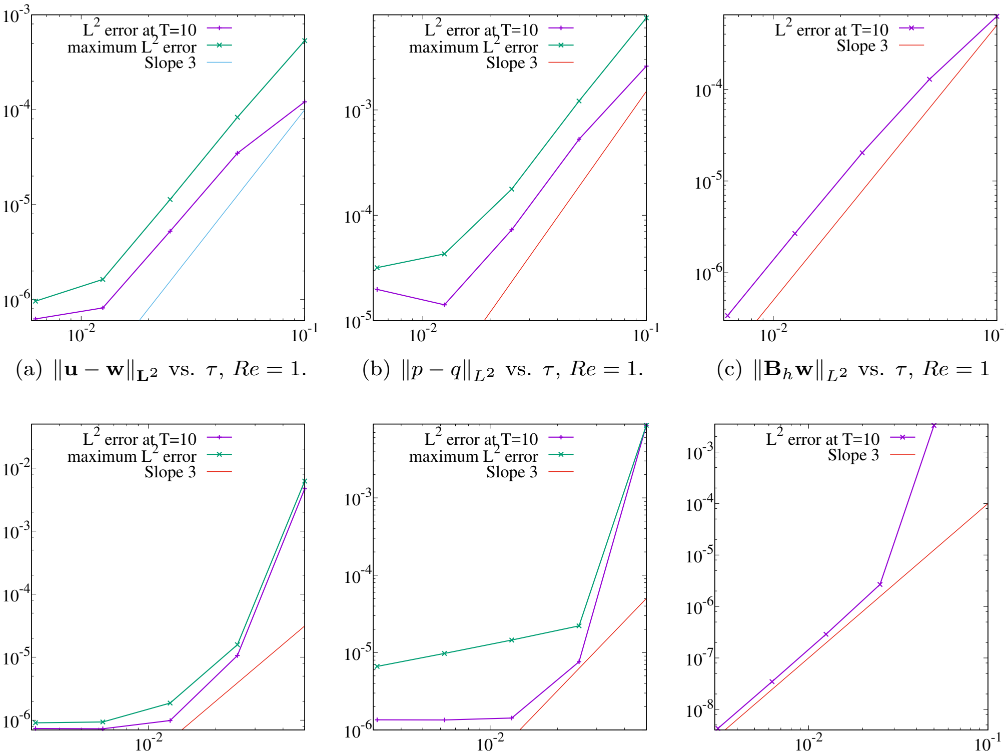
Using $\mathbf{x} = (x, y)^{\mathsf{T}}$ in $\mathbb{R}^2$, we consider here the Navier–Stokes equations with the source term chosen so that the solution is given by

$$\begin{aligned} &\mathbf{u}(\mathbf{x}, t) = (\sin(x)\sin(y+t), \cos(x)\cos(y+t))^{\mathsf{T}}, \\ &p(\mathbf{x}, t) = \cos(x)\sin(y+t) - \frac{1}{|\Omega|}\int_\Omega \cos(x)\sin(y+t)d\Omega. \end{aligned} \tag{5.1}$$

The problem is solved in $\Omega = (0, 1)\times(0, 1)$ over a time interval $(0, T]$. We enforce the Dirichlet conditions on the velocity. The initial condition for the velocity and the pressure and their first and second partial derivatives in time are taken equal to those of the exact solution. The purpose of this test is to estimate the convergence rate in time of the proposed scheme. We do two series of tests. In the first series, the spatial discretization is done by using the second-order MAC technique on a uniform Cartesian grid, and we use the fully uncoupled scheme (4.1)–(4.3) with $\mathbf{C}^{\triangle}$ defined in (4.4). In the second series, the space approximation is done with mixed $\mathbb{P}_2/\mathbb{P}_1$ finite elements on unstructured triangular meshes, and we use (3.12)–(3.14). In both cases we denote by $(\mathbf{w}, q)$ the approximate solution.

We start with the MAC approximation and use a fixed grid composed of 200×200 cells. The simulations are run until $T = 10$ with various time steps. The time step is kept constant for each simulation. We show in Figure 1 the $\mathbf{L}^2$-norm of the error on the velocity, and the L^2-norm of the error on the pressure and the (discrete) divergence of the velocity at $T = 10$. All theses errors are estimated at $T = 10$. We also show the maximum in time of the error on the velocity in the $\mathbf{L}^2$-norm, and the maximum in time of the error on the pressure in the L^2-norm. The computations reported in the top panels of Figure 1 are done at $Re = 1$, and those reported in the bottom panels are done at $Re = 100$. At $Re = 1$, the order $\mathcal{O}(\tau^3)$ is clearly visible. Notice that the error saturates at small time steps, i.e., when $\tau^3 \lesssim h^3$ or $\tau^3 \lesssim h^2$, depending on the norm considered, because in this regime the spatial error is dominant. We also observe third-order accuracy at $Re = 100$, but the saturation occurs earlier than at $Re = 1$. Notice that the error on the discrete divergence does not saturate, i.e., the L^2-norm of discrete divergence is $\mathcal{O}(\tau^3)$ independently of the meshsize.

We now report the convergence tests done with mixed finite elements. We show



(a) $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$ vs. τ, $Re = 1$. (b) $\|p-q\|_{L^2}$ vs. τ, $Re = 1$. (c) $\|\mathbf{B}_h\mathbf{w}\|_{L^2}$ vs. τ, $Re = 1$

(d) $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$ vs. τ, $Re = 10^2$. (e) $\|p-q\|_{L^2}$ vs. τ, $Re = 10^2$. (f) $\|\mathbf{B}_h\mathbf{w}\|_{L^2}$ vs. τ, $Re = 10^2$.

FIG. 1. 200×200 *MAC grid, at* $Re = 1$ *(top) and* $Re = 10^2$ *(bottom).*

TABLE 1
Mixed $\mathbb{P}_2/\mathbb{P}_1$ *finite element approximation,* $Re = 1$.

τ	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$	Rate	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{H}^1}$	Rate	$\|p-q\|_{L^2}$	Rate
1/20	1.42E-04	–	8.91E-04	–	3.95E-03	–
1/40	1.45E-05	3.29	5.59E-05	3.99	3.45E-04	3.52
1/80	3.88E-06	1.91	1.49E-05	1.90	9.12E-05	1.92
1/160	6.13E-07	2.66	2.94E-06	2.34	1.47E-05	2.63
1/320	8.42E-08	2.86	1.64E-06	0.85	2.41E-06	2.61
1/640	1.11E-08	2.92	1.60E-06	0.03	1.22E-06	0.98

in Tables 1 and 2 the error in the $\mathbf{L}^2$-norm on the velocity, the error in the $\mathbf{H}^1$-seminorm on the velocity, and the error in the $\mathbf{L}^2$-norm on the pressure. All the errors are computed at $T = 10$ and are relative, i.e., the errors are normalized by the corresponding norm or seminorm of the solution at $T = 10$. The computations are done on a nonuniform mesh composed of 118837 $\mathbb{P}_2$ grid points and 59098 triangles. The tests reported in Table 1 have been done with $Re = 1$, and those reported in Table 2 have been done with $Re = 10^2$. We observe the third-order convergence in τ in the $\mathbf{L}^2$-norm for the velocity in both cases. We also observe $\mathcal{O}(\tau^3)$ convergence on the $\mathbf{H}^1$-seminorm of the error of the velocity and on the L^2-norm of the error on the pressure for large τ, but just as with the MAC approximation there is saturation at small time steps due to the spatial error.

TABLE 2
Mixed $\mathbb{P}_2/\mathbb{P}_1$ finite element approximation, $Re = 10^2$.

τ	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$	Rate	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{H}^1}$	Rate	$\|p-q\|_{L^2}$	Rate
1/20	5.71E-05	–	1.17E-03	–	1.21E-04	–
1/40	4.41E-06	3.70	5.49E-05	4.42	1.51E-05	3.00
1/80	5.50E-07	3.00	7.32E-06	2.91	2.29E-06	2.72
1/160	6.86E-08	3.00	2.08E-06	1.82	1.22E-06	0.92
1/320	8.90E-09	2.95	1.88E-06	0.15	1.17E-06	0.05
1/640	2.67E-09	1.74	1.88E-06	0.00	1.17E-06	0.00

TABLE 3
Mixed $\mathbb{P}_2/\mathbb{P}_1$ finite element approximation, $Re = 10^2$. The meshsize and the time step are refined simultaneously, $\tau \sim \frac{1}{2}h$.

h	τ	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$	Rate	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{H}^1}$	Rate	$\|p-q\|_{L^2}$	Rate
0.1	1/20	2.13E-04	–	4.55E-03	–	6.95E-03	–
0.05	1/40	2.52E-05	3.66	1.04E-03	2.53	1.74E-03	2.37
0.025	1/80	2.31E-06	3.61	1.92E-04	2.55	2.56E-04	2.89
0.0125	1/160	2.38E-07	3.35	3.96E-05	2.33	1.11E-04	1.23
0.00625	1/320	2.50E-08	3.27	8.48E-06	2.24	2.87E-05	1.97
0.003125	1/640	2.67E-09	3.24	1.88E-06	2.18	1.17E-06	4.63

We show in Table 3 the relative errors at $Re = 100$ using meshes and time steps that are refined simultaneously, as is routinely done in practical applications. The number of $\mathbb{P}_2$ grid points and the size of the time step used for each of these meshes are as follows: $(161, 1/20)$, $(517, 1/40)$, $(1945, 1/80)$, $(7545, 1/160)$, $(29857, 1/320)$, $(118837, 1/640)$. The meshes are not hierarchically nested. We observe that the convergence rate on the velocity in the $\mathbf{L}^2$-norm is 3, and the convergence rate on the other two quantities is 2, as expected. The maximum order of convergence with respect to space on the velocity in the $\mathbf{H}^1$-seminorm and on the pressure in the L^2-norm cannot exceed 2 with mixed $\mathbb{P}_2/\mathbb{P}_1$ finite elements. This test and the previous ones confirm that the proposed scheme delivers the error estimates $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2} \leq c_1\tau^3 + c_2h^3$, $|\mathbf{u}-\mathbf{w}|_{\mathbf{H}^1} \leq c_3\tau^3 + c_4h^2$, and $\|p-q\|_{L^2} \leq c_5\tau^3 + c_6h^2$ for any $T > 0$.

5.2. Robustness with respect to open boundary conditions. We now consider a test to illustrate the robustness of the proposed scheme with respect to open boundary conditions and incompatible initial data. We compute the flow around a wing equipped with a flap as shown in the top left panel in Figure 2.

The distance between the leading edge of the wing and the trailing edge of the flap is 1.0558. The computational domain is the parallelogram defined by the following four corners: $\mathbf{x}_0 = (-1, -0.7)$, $\mathbf{x}_1 = (-1, 1.3)$, $\mathbf{x}_2 = (-4, 0.8)$, $\mathbf{x}_3 = (4, 1.2)$. We enforce the inflow boundary condition $\mathbf{u} = (1, 0)^\mathsf{T}$ at $\{x = -1, y \in (-0.7, 1.3)$. We enforce the no-slip boundary condition on the wing and the flap for $t > 0$. The initial velocity at $t = 0$ is $\mathbf{v}_0 = (1, 0)$. Notice that there is incompatibility between the initial velocity and the boundary condition on the wing and the flap. Notice also that the boundary connecting $\mathbf{x}_0$ to $\mathbf{x}_3$ is in principle an inflow boundary condition since it is slightly inclined, but to make sure that the scheme is robust with respect to open boundary conditions, we nevertheless enforce the natural boundary condition $-\frac{1}{Re}\mathbf{n}\cdot\nabla\mathbf{u} + p\mathbf{n} = 0$ on this segment. We also enforce the natural boundary $-\frac{1}{Re}\mathbf{n}\cdot\nabla\mathbf{u} + p\mathbf{n} = 0$ on the segments $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_2, \mathbf{x}_3)$. The chosen Reynolds number is 10^4. The simulation is run until $T = 5$. The size of the time step is kept constant. The mesh is Delaunay and composed of 57720 triangles. This makes 116483 $\mathbb{P}_2$ nodes for the velocity and

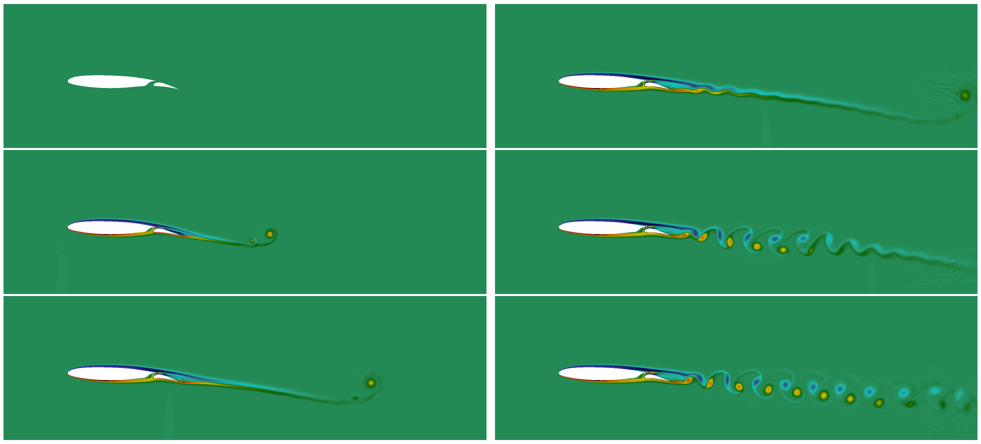


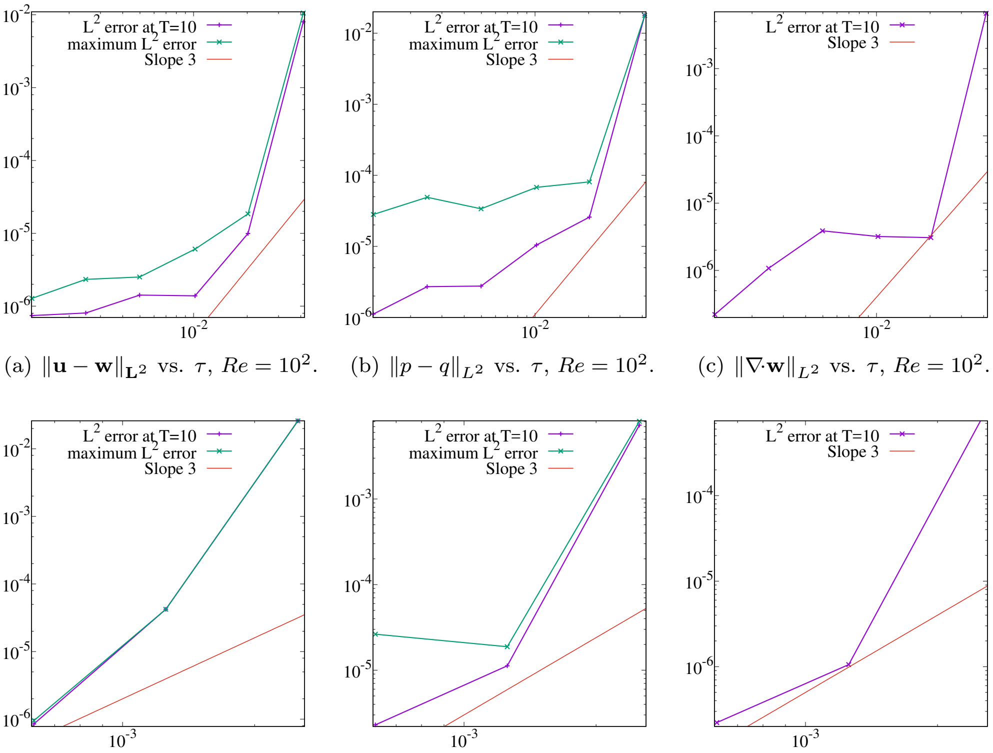
FIG. 2. *Wing with flap. $Re = 10^4$; $t = 0$ (top left); $t = 1$ (middle left); $t = 2$ (bottom left); $t = 3$ (top right); $t = 4$ (middle right); $t = 5$ (bottom right).*

29381 $\mathbb{P}_1$ nodes for the pressure. We show the vorticity in Figure 2 at the times $t \in \{0, 1, 2, 3, 4, 5\}$. We observe that the flow develops correctly in time. An unstable wake forms since the angle of attack is 0° and the flap is not properly trimmed (on purpose) for this angle of attack. The bottom, left, and top edges of the figures are cropped, but the right edge corresponds to the actual outflow boundary. We observe that the vortices cross this boundary without creating spurious artifacts.

5.3. Random time stepping. Next, we solve the Navier–Stokes equations using again the manufactured solution (5.1) and the scheme (4.1)–(4.3); however, now the time step is chosen randomly using a random number generator. The strategy we use is as follows: We run K simulations, with the kth simulation done with the average time step $2^{-k}\tau_0$, $k \in \{1, \dots, K\}$. For each simulation the time step τ is redefined at each time level by the expression $\tau = 2r2^{-k}\tau_0$, where each time r is a random number in $(0, 1)$. One series of simulations is done at $Re = 10^2$ and another one is done at $Re = 10^3$. This test is meant to evaluate the robustness of the algorithm. It is a very severe test since the size of τ may vary by several orders of magnitude from one time level to the other, because we allow the random number r to be arbitrarily close to 0.

The first series of tests is done with the MAC scheme on a 200×200 uniform Cartesian grid. We use $\tau_0 = 0.04$ and $K = 6$ for $Re = 10^2$, and we use $\tau_0 = 0.0025$ and $K = 3$ for $Re = 10^3$. The errors on the velocity, the pressure, and the (discrete) divergence of the velocity are measured in the L^2-norm at $T = 10$. We also compute the maximum in time of the L^2-norm of the errors on the velocity and the pressure. The results are shown in Figure 3. The results at $Re = 10^2$ are shown in the top panels, and the results at $Re = 10^3$ are shown in the bottom panels. The ideal third-order convergence rate is obviously hampered by the random time stepping, but we observe that the scheme is clearly robust and converges.

In the second series of tests we use mixed $\mathbb{P}_2/\mathbb{P}_1$ triangular continuous finite elements. We use $\tau_0 = 0.05$, $K = 6$, and $T = 10$ for $Re = 1$ and $Re = 100$. The tests are done on meshes that are refined by a factor 2^k for all $k \in \{1, \dots. K\}$. The meshes are nonuniform and are not hierarchically nested. The number of $\mathbb{P}_2$ grid points and average time step are as follows: $(161, 1/20)$, $(517, 1/40)$, $(1945, 1/80)$, $(7545, 1/160)$,



(a) $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$ vs. τ, $Re = 10^2$. (b) $\|p-q\|_{L^2}$ vs. τ, $Re = 10^2$. (c) $\|\nabla{\cdot}\mathbf{w}\|_{L^2}$ vs. τ, $Re = 10^2$.

(d) $\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$ vs. τ, $Re = 10^3$. (e) $\|p-q\|_{L^2}$ vs. τ, $Re = 10^3$. (f) $\|\nabla{\cdot}\mathbf{w}\|_{L^2}$ vs. τ, $Re = 10^3$.

Fig. 3. *Random time step; 200×200 MAC grid; Re* $= 10^2$ *(top); Re* $= 10^3$ *(bottom). The average value of the random time step is shown on the horizontal axes.*

Table 4

Mixed $\mathbb{P}_2/\mathbb{P}_1$ *finite elements, Re* $= 1$ *with random time stepping. The meshsize and the average time step are refined simultaneously, average*$(\tau) \sim \frac{1}{2}h$.

h	avg(τ)	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$	Rate	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{H}^1}$	Rate	$\|p-q\|_{L^2}$	Rate
0.1	1/20	1.33E-04	–	1.74E-03	–	7.51E-03	–
0.05	1/40	1.20E-05	4.13	4.27E-04	2.41	1.61E-03	2.64
0.025	1/80	1.44E-06	3.20	1.04E-04	2.13	2.65E-04	2.72
0.0125	1/160	1.89E-07	3.00	2.58E-05	2.05	1.03E-04	1.39
0.00625	1/320	4.91E-07	-1.39	6.66E-06	1.97	3.12E-05	1.74
0.003125	1/640	3.61E-08	3.78	1.61E-06	2.06	2.58E-06	3.61

$(29857, 1/320)$, $(118837, 1/640)$. The results are reported in Tables 4 and 5.

Similarly to the MAC scheme, we observe that the method converges, although the theoretical convergence rate is not exactly obtained due to the randomness of the time stepping.

5.4. Time step control. The main advantage of the Taylor series third-order defect-correction scheme proposed in the present paper, as compared to the multistep defect-correction in [9], is that it involves only two time levels, which makes it suitable for time step control. The purpose of the present section is to propose one time step control algorithm that is robust in the context of the present scheme. All our

TABLE 5

Mixed $\mathbb{P}_2/\mathbb{P}_1$ finite elements, $Re = 10^2$ with random time stepping. The meshsize and the average time step are refined simultaneously such that average$(\tau) \sim \frac{1}{2}h$.

h	τ	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{L}^2}$	Rate	$\|\mathbf{u}-\mathbf{w}\|_{\mathbf{H}^1}$	Rate	$\|p-q\|_{L^2}$	Rate
0.1	1/20	1.55E-03	–	2.07E-02	–	5.72E-03	–
0.05	1/40	6.56E-05	5.42	1.22E-03	4.86	1.55E-03	2.23
0.025	1/80	7.81E-06	3.21	2.02E-04	2.71	3.77E-04	2.14
0.0125	1/160	5.43E-06	0.54	5.63E-05	1.88	7.55E-05	2.37
0.00625	1/320	1.04E-06	2.40	1.09E-05	2.39	8.74E-05	-0.21
0.003125	1/640	4.35E-06	-2.07	3.39E-05	-1.65	9.88E-05	-0.18

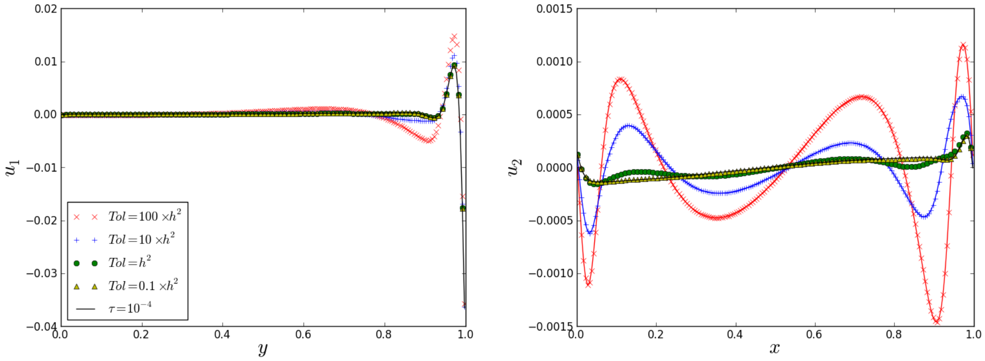
attempts to implement such an algorithm with the scheme proposed in [9] have been unsuccessful. Our quest for a robust scheme amenable to time step control is actually what led us to develop the method explained in the present paper.

It is not our purpose here to develop an elaborate strategy of time step adaption for the Navier–Stokes equations; we just want to show that the proposed scheme (3.9)–(3.11) performs as expected with a very simple time step control. The adaption strategy we consider is based on the assumption that the local error in the vicinity of the current time level is $\mathcal{O}(\tau^2\partial_{tt}\mathbf{u})$. For example, let e be the error on the velocity in the L^2-norm, and let us assume that $e = \mathcal{O}(\tau^2)\|\partial_{tt}\mathbf{u}^n\|_{L^2(\Omega)}$. Since $\mathbf{u}_2^n$ is an approximation of $\partial_{tt}\mathbf{u}(t^n)$, then $e = \mathcal{O}(\tau^2)\|\mathbf{u}_2^n\|_{L^2(\Omega)}$. Let us make it our objective that e should be bounded by $\text{Tol}\times\|\mathbf{u}_2^n\|_{L^2(\Omega)}$, where Tol is the relative error we would like to reach. Then we propose to control the error by computing the new time step as $\tau^{n+1} = s_1^n\tau^n$, where the scaling factor s_1^n is given by $s_1^n := \sqrt{\text{Tol}\,\|\mathbf{u}_0^n\|_{L^2(\Omega)}/\|\mathbf{u}_2^n\|_{L^2(\Omega)}}/\tau^n$. In addition, when the algorithm is combined with space approximation, we can also control the stability of the method by requiring that the local Courant number, $C := \tau^{n+1}\max_{\mathbf{x}\in\Omega}\|\mathbf{u}_0^n(\mathbf{x})\|_{\ell^2}/h(\mathbf{x})$, not be too large, say $C \in [\frac{1}{2}, 1]$, $h(\mathbf{x})$ being the local grid size and $\|\mathbf{u}_0^n(\mathbf{x})\|_{\ell^2}$ the Euclidean norm of $\mathbf{u}_0^n(\mathbf{x})$. Then, letting $\tau^{n+1} = s_2^n\tau^n$, the stability constraint induced by the Courant number is $s_2^n := C\min_{\mathbf{x}\in\Omega} h(\mathbf{x})/(\|\mathbf{u}_0^n(\mathbf{x})\|_{\ell^2}\tau^n)$. In conclusion, the next time step is defined by setting $\tau^{n+1} = \tau^n\min(s_1^n, s_2^n, s_{\max})$, where $s_{\max}$ is a fixed upper bound on the growth rate of the size of the time step. If it happens that $\min(s_1^n s_2^n)$ is too small, say $\min(s_1^n s_2^n) \leq s_{\min}$, where $s_{\min}$ is a fixed lower bound on the decrease rate of the time step, then the computation should be redone. We have found that the bounds $s_{\min} = 0.75$, $s_{\max} = 1.25$ give satisfactory results, but other choices are legitimate. The method to compute the new time step τ^{n+1} is described in Algorithm 1.

Algorithm 1 Time step control algorithm.

```
$s = \min(s_1^n, s_2^n, s_{\max})$
$\tau^{n+1} = s\tau^n$
if $s < s_{\min}$ then
    Time step is repeated with $\tau^{n+1}$
end if
return Time step $\tau^{n+1}$ and flag whether to repeat time step or not
```

5.5. Transient driven cavity. In this section we consider the classical lid-driven cavity problem, but in order to introduce essential unsteadiness in the solution, we make the horizontal velocity of the lid equal to $\sin(2\pi t)$. We solve the problem until $T = 1$, i.e., over one full period of the lid oscillation. The computations are done at $Re = 10^3$ with the MAC stencil. The problem is first solved with very high accuracy using the parallel method discussed in Guermond and Minev [10], on a uniform grid composed of $2\,000{\times}2\,000$ MAC cells and a time step $\tau = 10^{-4}$. This solution is further referred to as the reference solution. Then the algorithm (4.1)–(4.3) is used to solve the problem on a MAC Cartesian grid composed of $200{\times}200$ cells. The results of five different simulations are presented: (i) The first simulation is done with a constant time step $\tau = 10^{-4}$. The other four utilize the time step control described in section 5.4, where the control tolerance is defined as follows: (ii) $\mathrm{Tol} = 100{\times}h^2$, (iii) $\mathrm{Tol} = 10h^2$, (iv) $\mathrm{Tol} = h^2$, and (v) $\mathrm{Tol} = 0.1{\times}h^2$. In all cases we take $C = 1$, $s_{\min} = 0.5$, $s_{\max} = 2$. The differences between the reference solution and the solutions in cases (i)–(v) at $T = 1$ are reported in Figure 4. We show in the left panel the differences in the horizontal component of the velocity along the segment $x = \frac{1}{2}$, $y \in [0, 1]$. In the right panel we show the difference in the vertical component of the velocity along the segment $x \in [0, 1]$, $y = \frac{1}{2}$. The computation requires 145 time steps with $\mathrm{Tol} = 100{\times}h^2$, 318 time steps with $\mathrm{Tol} = 10{\times}h^2$, 1023 time step with $\mathrm{Tol} = h^2$, and 3258 time steps with $\mathrm{Tol} = 0.1{\times}h^2$. The solutions with the tolerances $\mathrm{Tol} = h^2$ and $\mathrm{Tol} = 0.1{\times}h^2$ and with the constant time step $\tau = 10^{-4}$ are almost identical; the overall error in these cases is dominated by the spatial error.



(a) Differences in horizontal velocity.

(b) Differences in vertical velocity.

FIG. 4. *Deviations from the reference solution in the horizontal* (a) *and vertical* (b) *components of the velocity along the vertical and horizontal center lines, respectively, at* $T = 1$.

We show in Figure 5 the time step size versus the time level for cases (ii)–(v). The cusp in the top graph (labeled $\mathrm{Tol} = 100h^2$) is due to the stability condition $C < 1$; i.e., at this time moment, the error control yields a time step that violates the stability condition, and this leads to its sudden decrease.

6. Conclusions.

This paper proposes a two-level high-order algorithm for the incompressible Navier–Stokes equations combining the artificial compressibility regularization (1.2) with a Taylor series scheme for the resulting ODE system. (The algorithm can actually be applied to any PDE or system of DAEs that can be put in the abstract form (2.1).) The main advantage of this setting as compared to the artificial compressibility method of Guermond and Minev [8, 9], which is of similar high order, is that the method is amenable to robust time step control. The robustness of the method with variable time stepping has been verified computationally using random time steps with decreasing averages. The results show that the method converges despite drastic variations of the time step from one time level to the other. A simple

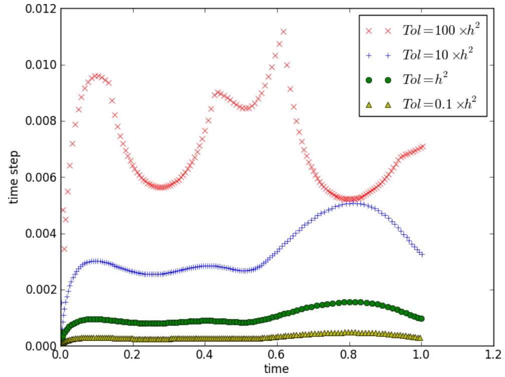


FIG. 5. *Plot of the time step versus the time level.*

strategy for the time step adaption has also been proposed. The performance of the proposed time step control has been verified on the lid-driven cavity with an oscillating lid velocity. The results demonstrate that the control of the second time derivative and the local Courant number yield a robust and efficient algorithm. Finally, the method discussed in this paper is third order in time, but there is no theoretical obstacle to extend it to any order by following the steps described in sections 3.1–3.2.

REFERENCES

[1] U. Asher and P. Lin, *Sequential regularization methods for higher index DAEs with constraint singularities: The linear index-2 case*, SIAM J. Numer. Anal., 33 (1996), pp. 1921–1940, https://doi.org/10.1137/S0036142993253254.

[2] U. Asher and P. Lin, *Sequential regularization methods for nonlinear higher-index DAEs*, SIAM J. Sci. Comput, 18 (1996), pp. 160–181, https://doi.org/10.1137/S1064827595287778.

[3] J. Baumgarte, *Stabilization of constraints and integrals of motion in dynamical systems*, Comput. Methods Appl. Mech. Engrg., 1 (1972), pp. 1–16, https://doi.org/10.1016/0045-7825(72)90018-7.

[4] M. Bercovier and M. Engelman, *A finite element for the numerical solution of viscous incompressible flows*, J. Comput. Phys., 30 (1979), pp. 181–201, https://doi.org/10.1016/0021-9991(79)90098-6.

[5] M. A. Case, V. J. Ervin, A. Linke, and L. G. Rebholz, *A connection between Scott–Vogelius and grad-div stabilized Taylor–Hood FE approximations of the Navier–Stokes equations*, SIAM J. Numer. Anal., 49 (2011), pp. 1461–1481, https://doi.org/10.1137/100794250.

[6] G. Corliss and Y. Chang, *Solving ordinary differential equations using Taylor series*, ACM Trans. Math. Software, 8 (1982), pp. 114–144, https://doi.org/10.1145/355993.355995.

[7] V. DeCaria, W. Layton, and M. McLaughlin, *A conservative, second order, unconditionally stable artificial compression method*, Comput. Methods Appl. Mech. Engrg., 325 (2017), pp. 733–747, https://doi.org/10.1016/j.cma.2017.07.033.

[8] J.-L. Guermond and P. Minev, *High-order time stepping for the incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 37 (2015), pp. A2656–A2681, https://doi.org/10.1137/140975231.

[9] J.-L. Guermond and P. Minev, *High-order time stepping for the incompressible Navier-Stokes equations with minimal computational complexity*, J. Comput. Appl. Math., 310 (2017), pp. 92–103, https://doi.org/10.1016/j.cam.2016.04.033.

[10] J.-L. Guermond and P. D. Minev, *Start-up flow in a three-dimensional lid-driven cavity by means of a massively parallel direction splitting algorithm*, Internat. J. Numer. Methods Fluids, 68 (2012), pp. 856–871, https://doi.org/10.1002/fld.2583.

[11] 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, https://doi.org/10.1016/j.cma.2005.10.010.

[12] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II: *Stiff and Differential-Algebraic Systems*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, Heidelberg, 1991, https://doi.org/10.1007/978-3-642-05221-7.

[13] N. Hurl, W. Layton, Y. Li, and C. Trenchea, *Stability analysis of the Crank–Nicolson–Leapfrog method with the Robert–Asselin–Williams time filter*, BIT, 54 (2014), pp. 1009–1021, https://doi.org/10.1007/s10543-014-0493-1.

[14] O. A. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, Second Russian Edition, revised and extended, Nauka, Moscow, 1970 (in Russian).

[15] W. Layton, Y. Li, and C. Trenchea, *Recent developments in IMEX methods with time filters for systems of evolution equations*, J. Comput. Appl. Math., 299 (2016), pp. 50–67, https://doi.org/10.1016/j.cam.2015.09.038.

[16] P. Lin, *A sequential regularization method for time-dependent incompressible Navier–Stokes equations*, SIAM J. Numer. Anal., 34 (1997), pp. 1051–1071, https://doi.org/10.1137/S0036142994270521.

[17] A. Linke and L. G. Rebholz, *On a reduced sparsity stabilization of grad-div type for incompressible flow problems*, Comput. Methods Appl. Mech. Engrg., 261/262 (2013), pp. 142–153, https://doi.org/10.1016/j.cma.2013.04.005.

[18] N. S. Nedialkov and J. Pryce, *Solving differential-algebraic equations by Taylor series* (I): *Computing the Taylor coefficients*, BIT, 45 (2005), pp. 561–591, https://doi.org/10.1007/s10543-005-0019-y.

[19] N. S. Nedialkov and J. Pryce, *Solving differential-algebraic equations by Taylor series* (II): *Computing the system Jacobean*, BIT, 47 (2007), pp. 121–135, https://doi.org/10.1007/s10543-006-0106-8.

[20] T. Ohwada and P. Asinari, *Artificial compressibility method revisited: Asymptotic numerical method for incompressible Navier–Stokes equations*, J. Comput. Phys., 229 (2010), pp. 1698–1723, https://doi.org/10.1016/j.jcp.2009.11.003.

[21] J. Pryce, *Solving high-index DAEs by Taylor series*, Numer. Algorithms, 19 (1998), pp. 195–211, https://doi.org/10.1023/A:1019150322187.

[22] J. Shen, *On error estimates of the penalty method for unsteady Navier–Stokes equations*, SIAM J. Numer. Anal., 32 (1995), pp. 386–403, https://doi.org/10.1137/0732016.

[23] G. Tan, N. S. Nedialkov, and J. Pryce, *Conversion methods for improving structural analysis of differential-algebraic equation systems*, BIT, 57 (2017), pp. 845–865, https://doi.org/10.1007/s10543-017-0655-z.

[24] 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, https://doi.org/10.1007/BF00247678.

[25] 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).

[26] P. Williams, *The RAW filter: An improvement to the Robert-Asselin filter in semi-implicit integrations*, Mon. Weather Rev., 139 (2010), pp. 1996–2007, https://doi.org/10.1175/2010MWR3601.1.

[27] N. N. Yanenko, *The Method of Fractional Steps. The Solution of Problems of Mathematical Physics in Several Variables*, Springer-Verlag, New York, 1971, https://doi.org/10.1007/978-3-642-65108-3.