INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2012; **69**:57–72
Published online 15 February 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.2535

A third-order velocity correction scheme obtained at the discrete level

H. Owen[1,*,†] and R. Codina[2]

[1]*Barcelona Supercomputing Center (BSC-CNS), Edificio NEXUS I, Campus Nord UPC, Gran Capitán 2-4, 08034 Barcelona, Spain*
[2]*Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain*

SUMMARY

In this work we explore a velocity correction method that introduces the splitting at the discrete level. In order to do so, the algebraic continuity equation is transformed into a discrete pressure Poisson equation and a velocity extrapolation is used. In Badia *et al.* (*IJNMF*, 2008, p. 351), where the method was introduced, the discrete Laplacian that appears in the pressure Poisson equation is approximated by a continuous one using an extrapolation for the pressure. In this work we explore the possibility of actually solving the discrete Laplacian. This introduces significant differences because the pressure extrapolation is avoided and only a velocity extrapolation is needed. Our numerical results indicate that it is the second-order pressure extrapolation which makes third-order methods unstable. Instead, second-order velocity extrapolations do not lead to instabilities. Avoiding the pressure extrapolation allows to obtain stable solutions in problems that become unstable when the Laplacian is approximated. A comparison with a pressure correction scheme is also presented to verify the well-known fact that the use of a second order pressure extrapolation leads to instabilities. Therefore we conclude that it is the combination of a velocity correction scheme with a discrete Laplacian that allows to obtain a stable third-order scheme by avoiding the pressure extrapolation. Copyright © 2011 John Wiley & Sons, Ltd.

Received 12 October 2010; Revised 16 December 2010; Accepted 23 December 2010

KEY WORDS: Navier–Stokes; incompressible flow; stabilized methods; VMS—variational multiscale; finite element methods; stability

1. INTRODUCTION

Pressure segregation methods, also know as fractional step or projection methods, have become one of the most popular schemes for solving the Navier–Stokes equations since their appearance in the late 1960s with the works of Chorin [1] and Temam [2]. The key for such success is that they allow to uncouple the velocity and pressure unknowns, leading not only to smaller, but also better conditioned subproblems. They can be classified into pressure correction methods and velocity correction methods. The former are the most well known and include the Chorin–Temam [1, 2] projection method and the Van Kan method [3]. The latter are more recent [4, 5]. The velocity correction approach we will use has been developed in [6]. Complete reviews on pressure segregation methods can be found in [7, 8]. The name pressure correction arises from the fact that the splitting involves first solving the velocity using an extrapolation for the pressure and then correcting the pressure. If a zero-order extrapolation, $\tilde{P}_0^{n+1} = 0$, is used the original Chorin–Temam [1, 2] scheme is obtained. A first-order extrapolation $\tilde{P}_1^{n+1} = P^n$ leads to the Van Kan scheme [3].

*Correspondence to: H. Owen, Barcelona Supercomputing Center (BSC-CNS), Edificio NEXUS I, Campus Nord UPC, Gran Capitán 2-4, 08034 Barcelona, Spain.
†E-mail: herbert.owen@gmail.com

In the velocity correction method an extrapolation of the velocity is first used to solve for the pressure which is then used to correct the velocity.

The most typical approach is to first uncouple the velocity and the pressure at the space continuous level and then discretize the problem [1, 2, 7]. The approach we will use in this work is to introduce the splitting at the purely algebraic level, once the discretization has been performed. Such approach is advocated in [9–12]. The main difference between both approaches is the way in which the boundary conditions are approximated. When dealing with continuous pressure interpolations, the discrete Laplacian that appears in the discrete approach is usually approximated by a continuous one, reducing the gap between both approaches. For a precise definition for both Laplacians, see Section 3.1.

The velocity correction method we use is based on a discrete pressure Poisson equation (DPPE) and has been introduced in [6]. First the monolithic problem is discretized and then the splitting is introduced at the discrete level. In order to do so the algebraic continuity equation is transformed into a DPPE and a velocity extrapolation is used. A discrete Laplacian for the pressure must then be solved. In [6] the discrete Laplacian that appears in the pressure Poisson equation is approximated by a continuous one using an extrapolation for the pressure. In this work we also explore the possibility of actually solving it.

The splitting of the problem introduces an splitting error that is one order higher than the extrapolation used for the velocity or the pressure. For pressure correction schemes it is well known that extrapolations of order higher than 2 lead to unstable schemes unless very small time steps are used. Therefore only second-order schemes can be obtained. The experience with velocity correction schemes is more limited. In [4] a third-order velocity correction scheme that uses a spectral element space discretization provides stable results. Using finite element discretizations [6] unstable results have been obtained in the third-order case. In [6], where the method we use has been introduced, two alternative approaches have been proposed. The one we use in this work involves only one Poisson solve per time step and therefore introduces a splitting error that is reduced by using a third-order version. The second alternative presented in [6] is called predictor corrector scheme and involves solving for the pressure more than once per time step. Iterating between the pressure and the velocity, the splitting error can be eliminated and the monolithic solution can be recovered. This makes the method much more expensive and such strategy will not be used in this work.

We have verified that when a second-order pressure extrapolation is used to approximate the discrete Laplacian by a continuous one, unstable results are obtained. The instability disappears if the discrete Laplacian is solved instead of approximated. We believe that the improvement can be attributed to the fact that, in this case, no pressure extrapolation is needed and only a second-order velocity extrapolation is used. Results with a pressure correction scheme are also presented. It provides unstable results, even when the discrete Laplacian is used, because a second-order pressure extrapolation must be used to obtain a third-order scheme.

The Split Orthogonal Subgrid Scale Stabilization [13, 14] we use to stabilize the convective term and allow for equal order interpolations for velocity and pressure needs an extrapolation of the pressure gradient projection when it is combined with the velocity correction scheme. In order to avoid any pressure extrapolation we develop an implicit pressure gradient projection.

The straightforward solution of the discrete Laplacian with an iterative method such as CG is computationally more expensive than the solution of the continuous one. We propose a solution based on a preconditioned Richardson iteration using the continuous Laplacian as preconditioner. This is an extension of the enhanced approximation proposed in [6].

In Section 2 we introduce the Navier–Stokes equations, its space and time discretization and the stabilization technique we use. The DPPE, which is the basis for the velocity correction scheme, is presented in Section 3. It uses a lumped mass matrix instead of the consistent mass matrix used in [6]. The velocity correction scheme and its stabilization using a Split Orthogonal Subgrid Scale method with an implicit pressure gradient projection is introduced in Section 4. Section 5 presents the numerical results that show the advantages of using a velocity correction scheme with a discrete Laplacian. The conclusions are summarized in Section 6.

DOI: 10.1002/fld

2. PRELIMINARIES AND PROBLEM STATEMENT

2.1. The continuous Navier–Stokes equations

The Navier–Stokes equations for a fluid moving in the domain Ω bounded by $\Gamma=\partial\Omega$ during the time interval (t_0, t_f) consist in finding a velocity $\mathbf{u}$ and a kinematic pressure p such that

$$\partial_t \mathbf{u}+(\mathbf{u}\cdot\nabla)\mathbf{u}-\nu\Delta\mathbf{u}+\nabla p=\mathbf{f} \quad \text{in } \Omega\times(t_0,t_f), \tag{1}$$

$$\nabla\cdot\mathbf{u}=0 \quad \text{in } \Omega\times(t_0,t_f), \tag{2}$$

where ν is the viscosity and $\mathbf{f}$ the vector of external body forces. Despite examples with Neumann conditions shall be presented, in order to simplify the presentation homogeneous Dirichlet boundary conditions,

$$\mathbf{u}=\mathbf{0} \quad \text{on } \Gamma\times(t_0,t_f), \tag{3}$$

are considered. Initial conditions,

$$\mathbf{u}=\mathbf{u}_0 \quad \text{in } \Omega\times\{t_0\},$$

have to be appended to the problem.

In order to obtain the weak or variational formulation of the Navier–Stokes equations ((1) and (2)), we introduce the spaces of vector functions $\mathbf{V}_0=\mathbf{H}_0^1(\Omega)$ and $Q=L^2(\Omega)/\mathbb{R}$. As usual, $L^2(\Omega)$ is the space of square-integrable functions, $H^1(\Omega)$ is a subspace of $L^2(\Omega)$ formed by functions whose derivatives also belong to $L^2(\Omega)$, $H_0^1(\Omega)$ is a subspace of $H^1(\Omega)$ whose functions are zero on Γ, and $\mathbf{H}_0^1(\Omega)$ is its vector counterpart in a two- or three-dimensional space. $(\cdot,\cdot)$ indicates the standard L^2 inner product and $\langle f,g\rangle=\int_\Omega fg$ whenever functions f and g are such that the integral is well defined.

For the evolutionary case $\mathbf{V}_t\equiv L^2(t_0,t_f;\mathbf{V}_0)$ and $Q_t\equiv\mathscr{D}'(t_0,t_f;Q)$ are introduced, where $L^p(t_0,t_f;X)$ is the space of time-dependent functions in a normed space X such that $\int_{t_0}^{t_f}\|f\|_X^p\mathrm{d}t<\infty$, $1\leqslant p<\infty$ and Q_t consists of mappings whose Q-norm is a distribution in time. The weak form of problem (1), (2) with the boundary conditions we have just defined is then: Find $\mathbf{u}\in\mathbf{V}_t$, $p\in Q_t$ such that

$$(\partial_t\mathbf{u},\mathbf{v})+\langle\mathbf{u}\cdot\nabla\mathbf{u},\mathbf{v}\rangle+\nu(\nabla\mathbf{u},\nabla\mathbf{v})(p,\nabla\cdot\mathbf{v})=\langle\mathbf{f}\mathbf{v}\rangle,$$

$$(q,\nabla\cdot\mathbf{u})=0,$$

for all $(\mathbf{v},q)\in\mathbf{V}_0\times Q$.

2.2. The discrete Navier–Stokes equations

The problem can then be discretized in time using a uniform partition of the time interval of size δt, and denoting by f^n an approximation to a time-dependent function f at time $t^n=n\delta t$. We use backward difference (BDF) schemes with operator

$$D_k f^{n+1}=\frac{1}{\gamma_k}\left(f^{n+1}-\sum_{i=0}^{k-1}\alpha_k^i f^{n-i}\right), \tag{4}$$

where γ_k and α_k^i are parameters. The operators that lead to time discretization errors of orders 1–3 are

$$D_1 f^{n+1}=\delta f^{n+1}=f^{n+1}-f^n,$$

$$D_2 f^{n+1}=\frac{3}{2}\left(f^{n+1}-\frac{4}{3}f^n+\frac{1}{3}f^{n-1}\right),$$

$$D_3 f^{n+1}=\frac{11}{6}\left(f^{n+1}-\frac{18}{11}f^n+\frac{9}{11}f^{n-1}-\frac{2}{11}f^{n-2}\right).$$

DOI: 10.1002/fld

Given the values of the velocity at the required previous time steps, the time discretized Navier–Stokes equations using a BDF scheme of order k consists in finding $\mathbf{u}^{n+1} \in \mathbf{V}_0$ and $p^{n+1} \in Q$, with $n+1 \geqslant k$, such that

$$\left(\frac{1}{\delta t} D_k \mathbf{u}^{n+1}, \mathbf{v}\right) + \langle \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1}, \mathbf{v} \rangle + \nu(\nabla \mathbf{u}^{n+1}, \nabla \mathbf{v}) - (p^{n+1}, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}^{n+1}, \mathbf{v} \rangle,$$

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

for all $(\mathbf{v}, q) \in \mathbf{V}_0 \times Q$.

The space discretization is built with the finite element method. $\mathbf{V}_{0h} \subset \mathbf{V}_0$ and $Q_h \subset Q$ are the discrete linear subspaces that approximate the respective continuous spaces. The same interpolation will be used for both the velocity and the pressure. $\mathbf{V}_{0h}$ incorporates the Dirichlet conditions for the velocity components and Q_h has one pressure fixed to zero if the normal component of the velocity is prescribed on the whole boundary. The space discretized problem reads: find $\mathbf{u}_h^{n+1} \in \mathbf{V}_{0h}$ and $p^{n+1} \in Q_h$ such that

$$\left(\frac{1}{\delta t} D_k \mathbf{u}_h^{n+1}, \mathbf{v}_h\right) + \langle \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1}, \mathbf{v}_h \rangle + \nu(\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) = \langle \mathbf{f}^{n+1}, \mathbf{v}_h \rangle,$$

$$(q_h, \nabla \cdot \mathbf{u}_h^{n+1}) = 0,$$

for all $(\mathbf{v}_h, q_h) \in \mathbf{V}_{0h} \times Q_h$.

The matrix form of the previous equations is

$$\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1} + \mathsf{K}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} + \mathsf{G}\mathsf{P}^{n+1} = \mathsf{F}^{n+1}, \tag{5}$$

$$\mathsf{D}\mathsf{U}^{n+1} = \mathsf{0}, \tag{6}$$

where U, P are the arrays of the nodal unknowns for $\mathbf{u}$ and p, respectively. If we denote the node indexes with superscripts a, b, the space indexes with subscripts i, j and the standard shape functions of node a by N^a, the components of the arrays involved in the previous equations are

$$\mathsf{M}_{ij}^{ab} = (N^a, N^b)\delta_{ij},$$

$$\mathsf{K}(\mathsf{U}^{n+1})_{ij}^{ab} = \langle N^a, \mathbf{u}_h^{n+1} \cdot \nabla N^b \rangle \delta_{ij} + (\nabla N^a, \nu \nabla N^b)\delta_{ij},$$

$$\mathsf{G}_i^{ab} = (N^a, \partial_i N^b),$$

$$\mathsf{D}_j^{ab} = (N^a, \partial_j N^b),$$

$$\mathsf{F}_i^a = \langle N^a, f_i \rangle,$$

where δ_{ij} is the Kronecker δ.

It is understood that all the arrays are matrices (except F, which is a vector) whose components are obtained by grouping together the left indexes in the previous expressions (a and possibly i) and the right indexes in the previous expressions (b and possibly j). Equation (5) needs to be modified to account for the Dirichlet boundary conditions (matrix G can be replaced by $-\mathsf{D}^{\mathrm{T}}$ when this is done).

2.3. The stabilized problem

The discretized problem presented in the previous subsection needs to be stabilized before it can be solved numerically. Orthogonal Subgrid Scale (OSS) stabilization [13, 14] is used to deal with convection-dominated flows and to circumvent the well-known div-stability restriction for the velocity and pressure finite element spaces [15], allowing in particular for equal interpolation of both unknowns. Two versions of OSS stabilization exist. The one used in this work provides a least-square control of the convective and pressure gradient terms orthogonal to the finite element space

DOI: 10.1002/fld

separately and can therefore be called split OSS. The matrix version of the split OSS monolithic discrete problem associated with the Navier–Stokes equations ((1) and (2)), discretizing in time using BDF scheme of order k, can be written as follows:

$$\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1}+\mathsf{K}(\mathsf{U}^{n+1})\mathsf{U}^{n+1}+\mathsf{G}\mathsf{P}^{n+1}+\mathsf{S}_u(\tau_1;\mathsf{U}^{n+1})\mathsf{U}^{n+1}-\mathsf{S}_y(\tau_1;\mathsf{U}^{n+1})\mathsf{Y}^{n+1}=\mathsf{F}^{n+1}, \tag{7}$$

$$\mathsf{D}\mathsf{U}^{n+1}+\mathsf{S}_p(\tau_1)\mathsf{P}^{n+1}-\mathsf{S}_z(\tau_1)\mathsf{Z}^{n+1}=\mathsf{0}, \tag{8}$$

$$\mathsf{M}\mathsf{Y}^{n+1}-\mathsf{C}(\mathsf{U}^{n+1})\mathsf{U}^{n+1}=\mathsf{0}, \tag{9}$$

$$\mathsf{M}\mathsf{Z}^{n+1}-\mathsf{G}\mathsf{P}^{n+1}=\mathsf{0}, \tag{10}$$

where Y and Z are the arrays of the nodal unknowns for $\mathbf{y}$ and $\mathbf{z}$, the projections of the convective and pressure gradient terms, respectively. The matrices involved in the stabilization are

$$\mathsf{S}_u(\tau_1;\mathsf{U}^{n+1})_{ij}^{ab}=\langle\tau_1\mathbf{u}_h^{n+1}\cdot\nabla N^a,\mathbf{u}_h^{n+1}\cdot\nabla N^b\rangle\delta_{ij},$$

$$\mathsf{S}_y(\tau_1;\mathsf{U}^{n+1})_{ij}^{ab}=\langle\tau_1\mathbf{u}_h^{n+1}\cdot\nabla N^a,N^b\rangle\delta_{ij},$$

$$\mathsf{S}_p(\tau_1)^{ab}=(\tau_1\nabla N^a,\nabla N^b),$$

$$\mathsf{S}_z(\tau_1)_j^{ab}=(\tau_1\partial_j N^a,N^b),$$

$$\mathsf{C}(\mathsf{U}^{n+1})_{ij}^{ab}=\langle N^a,\mathbf{u}_h^{n+1}\cdot\nabla N^b\rangle\delta_{ij}.$$

The parameter τ_1 is chosen in order to obtain a stable numerical scheme with optimal convergence rates (see [16] and references therein for details). It is computed within each element domain Ω^e as

$$\tau_1=\left[\frac{4\nu}{(h^e)^2}+\frac{2|\mathbf{u}^e|}{h^e}\right]^{-1},$$

where h^e and $|\mathbf{u}^e|$ are a typical length and a velocity norm of element e, respectively. Actually, in our implementation, a lumped mass matrix is used in the projections for Y and Z.

3. THE DPPE

The DPPE is derived from the monolithic problem discretized both in space and time. We shall neglect the stabilization terms to simplify the presentation. First the momentum equation (5) is multiplied by $\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}$, where M_L is the lumped mass matrix and γ_k is a parameter defined in (4). Then the resulting equation is subtracted from the continuity equation (6) to obtain the DPPE. The result is

$$\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}\mathsf{P}^{n+1}=\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}\left(\mathsf{F}^{n+1}-\mathsf{K}\left(\mathsf{U}^{n+1}\right)\mathsf{U}^{n+1}-\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1}\right)+\mathsf{D}\mathsf{U}^{n+1}. \tag{11}$$

The system formed by (5), (11) is equivalent to the original monolithic discretized scheme (5), (6) and the boundary conditions arise naturally from the original scheme. There is no advantage in solving the coupled system that uses the DPPE equation directly. The advantage is that the segregation is now straightforward. The DPPE differs slightly from the one used in [6] that is obtained by multiplying (5) by $\gamma_k\delta t\mathsf{D}\mathsf{M}^{-1}$ instead of $\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}$.

3.1. *Approximation of* $\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}$

The use of the discrete Laplacian $\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}$ is relatively expensive even if a diagonal mass matrix is used. Therefore, when continuous pressure interpolations are used, it is usually approximated as

$$\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}\approx\mathsf{L}\text{ with components }\mathsf{L}^{ab}=-(\nabla N^a,\nabla N^b).$$

DOI: 10.1002/fld

In [6] an *enhanced* approximation is introduced. In this work it is adapted to the case in which a lumped mass matrix is used as follows:

$$\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}\mathsf{P}^{n+1}=\mathsf{L}\mathsf{P}^{n+1}+(\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}-\mathsf{L})\mathsf{P}^{n+1}\approx\mathsf{L}\mathsf{P}^{n+1}+(\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}-\mathsf{L})\widetilde{\mathsf{P}}_p^{n+1}, \tag{12}$$

where $\widetilde{\mathsf{P}}_p^{n+1}$ is an extrapolation of order p of P^{n+1} obtained from previous known values. If a zero-order extrapolation $\widetilde{\mathsf{P}}_p^{n+1}=\mathsf{0}$ is used the basic approximation is recovered. When the *enhanced* approximation is used, the DPPE reads as

$$\gamma_k\delta t\mathsf{L}(\mathsf{P}^{n+1}-\widetilde{\mathsf{P}}_p^{n+1})=\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}\left(\mathsf{F}^{n+1}-\mathsf{K}(\mathsf{U}^{n+1})\mathsf{U}^{n+1}-\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1}-\mathsf{G}\widetilde{\mathsf{P}}_p^{n+1}\right)+\mathsf{D}\mathsf{U}^{n+1}. \tag{13}$$

The *enhanced* approximation can be extended to obtain a preconditioned Richardson iteration to solve for the discrete Laplacian using the continuous Laplacian as preconditioner, leading to the iterative scheme

$$\begin{aligned}&\gamma_k\delta t\mathsf{L}(\mathsf{P}^{n+1,i+1}-\mathsf{P}^{n+1,i})\\&\quad=\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}\left(\mathsf{F}^{n+1}-\mathsf{K}\left(\mathsf{U}^{n+1}\right)\mathsf{U}^{n+1}-\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1}-\mathsf{G}\mathsf{P}^{n+1,i}\right)+\mathsf{D}\mathsf{U}^{n+1},\end{aligned} \tag{14}$$

where the second superscript denotes the iteration counter. $\mathsf{P}^{n+1,i}=\widetilde{\mathsf{P}}_p^{n+1}$ is used for the first iteration, $i=0$. The *enhanced* approximation proposed in [6] is recovered by allowing only one iteration. As we explain in the following section, the velocity U^{n+1} is extrapolated from the values at previous time steps to obtain the velocity correction scheme.

Compared to the scheme presented in [6] the use of a lumped mass matrix instead of the discrete one introduces the following additional term:

$$\mathsf{D}(-\mathsf{M}_L^{-1}\mathsf{M}\mathsf{U}^{n+1}+\mathsf{U}^{n+1}).$$

Our numerical experience indicates that the use of the extrapolation in the second term introduces no difficulties. This seems logical because M_L is a good approximation to M when linear elements are used, as in our examples.

4. THE VELOCITY CORRECTION FRACTIONAL STEP SCHEME

4.1. Galerkin approximation

In order to obtain the velocity correction fractional step scheme we start from the coupled system written with a DPPE (5),(11). The method is called a velocity correction method because it is the velocity that is extrapolated from values at previous time steps instead of the pressure. In the first step the pressure is obtained from the DPPE using an extrapolation of order q of the velocity U^{n+1} (denoted by $\widetilde{\mathsf{U}}_q^{n+1}$). Then, U^{n+1} is obtained from the momentum equation (velocity correction step).

Using a BDF time discretization of order k, the split scheme reads as

$$\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}\mathsf{P}^{n+1}=\gamma_k\delta t\mathsf{D}\mathsf{M}_L^{-1}\left(\mathsf{F}^{n+1}-\mathsf{K}\left(\widetilde{\mathsf{U}}_q^{n+1}\right)\widetilde{\mathsf{U}}_q^{n+1}-\mathsf{M}\frac{D_k}{\delta t}\widetilde{\mathsf{U}}_q^{n+1}\right)+\mathsf{D}\widetilde{\mathsf{U}}_q^{n+1}, \tag{15}$$

$$\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1}+\mathsf{K}(\mathsf{U}^{n+1})\mathsf{U}^{n+1}+\mathsf{G}\mathsf{P}^{n+1}=\mathsf{F}^{n+1}. \tag{16}$$

The extrapolation of the velocity is used not only in the convective and diffusive terms, as happens when a consistent mass matrix is used in the obtention of the DPPE, but also in the term $\mathsf{D}(\mathsf{M}_L^{-1}\mathsf{M}\widetilde{\mathsf{U}}_q^{n+1}-\widetilde{\mathsf{U}}_q^{n+1})$ that only appears when a lumped mass matrix is used.

DOI: 10.1002/fld

Using approximation (12) for the discrete Laplacian suggested in [6], we can obtain the following system:

$$\gamma_k \delta t \mathsf{L}(\mathsf{P}^{n+1} - \widetilde{\mathsf{P}}_p^{n+1}) = \gamma_k \delta t \mathsf{D}\mathsf{M}_L^{-1}\left(\mathsf{F}^{n+1} - \mathsf{K}\left(\widetilde{\mathsf{U}}_q^{n+1}\right)\widetilde{\mathsf{U}}_q^{n+1} - \mathsf{M}\frac{D_k}{\delta t}\widetilde{\mathsf{U}}_q^{n+1} - \mathsf{G}\widetilde{\mathsf{P}}_p^{n+1}\right) + \mathsf{D}\widetilde{\mathsf{U}}_q^{n+1},$$

$$\frac{1}{\delta t}\mathsf{M}\left(\mathsf{U}^{n+1} - \mathsf{U}^n\right) + \mathsf{K}\left(\mathsf{U}^{n+1}\right)\mathsf{U}^{n+1} + \mathsf{G}\mathsf{P}^{n+1} = \mathsf{F}^{n+1}.$$

A first-order method in time can be obtained taking $k=1$ and $q=p=0$. The second-order scheme can be obtained with $k=2$ and $q=p=1$, and the third order version with $k=3$ and $q=p=2$.

The first remarkable fact about the case when an approximation for the discrete Laplacian based on the continuous one is used is that despite using a velocity correction method which should be characterized by the fact that in going from one step to the next only an extrapolation for the velocity is used, actually extrapolations for both the velocity and the pressure are used. The need for the pressure extrapolation comes from the use of an approximation of $\mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}$. The reason for needing the pressure extrapolation is therefore different from the reason for needing the velocity extrapolation (that is the true spirit of the velocity correction method) but, in any case, if the approximation of the discrete Laplacian is used, both are needed. Instead, in the pressure correction method [9] an extrapolation of the pressure is needed no matter whether the discrete Laplacian or an approximation to it is used. Therefore, we can say that in order to obtain a 'pure' velocity correction scheme the discrete Laplacian must be used. Extrapolations of order higher than 1 should help to reduce the splitting error but can cause instabilities. In [6] a third-order method that used a BDF3 time discretization and second-order extrapolations for both the velocity and the pressure with the enhanced approximation for the discrete Laplacian was tested. Numerical experimentation showed that it was unstable as happens for third-order pressure correction methods.

In this work we have solved the discrete Laplacian, directly or using a Richardson iteration, to obtain a 'pure' third-order velocity correction method that only uses a second-order velocity extrapolation. This has allowed us to obtain a third-order method that has shown to be stable in numerical examples we present in Section 5. When the same cases are run with the third-order method with an approximation of the discrete Laplacian used in [6], they are unstable (also if a pressure correction method is used). Moreover we have tested the velocity correction BDF3 scheme with an approximation to the discrete Laplacian with a first-order pressure extrapolation and second-order velocity extrapolation. In that case, the third-order accuracy is lost but stability is recovered. Therefore we deduce that the instability observed in [6] was caused by the use of second-order pressure extrapolations and different conclusions could have been drawn if a 'pure' velocity correction scheme had been used. Instabilities for third-order velocity correction schemes are also observed in [7] where the fractional step scheme is obtained at the continuous level precluding the possibility of using a discrete Laplacian.

Second-order velocity extrapolations may work better than second-order pressure extrapolations because the velocity satisfies an evolutionary equation; instead the pressure adapts itself instantaneously to satisfy the incompressibility constraint. From the convergence analysis of different pressure segregation methods it is known that the error estimates for the velocity are sharper than for the pressure [6, 8]. Even in the monolithic case, where the order of the velocity error depends only on the time integration scheme used, pressure errors of order equal or higher than 2 cannot always be obtained for time integration schemes of order 2 or higher [17].

The third-order accurate scheme used in the numerical examples is obtained by combining a BDF3 time discretization and a second-order velocity extrapolation ($q=2$). It reads as

$$\frac{6}{11}\delta t \mathsf{D}\mathsf{M}_L^{-1}\mathsf{G}\mathsf{P}^{n+1} = \frac{6}{11}\delta t \mathsf{D}\mathsf{M}_L^{-1}\left[\mathsf{F}^{n+1} - \mathsf{K}\left(\widetilde{\mathsf{U}}_2^{n+1}\right)\widetilde{\mathsf{U}}_2^{n+1} - \frac{11}{6\delta t}\mathsf{M}\left(\widetilde{\mathsf{U}}_2^{n+1} - \frac{18}{11}\mathsf{U}^n + \frac{9}{11}\mathsf{U}^{n-1} - \frac{2}{11}\mathsf{U}^{n-2}\right)\right] + \mathsf{D}\widetilde{\mathsf{U}}_2^{n+1}, \quad (17)$$

DOI: 10.1002/fld

$$\frac{11}{6\delta t}\mathbf{M}\mathbf{U}^{n+1}+\mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1}+\mathbf{G}\mathbf{P}^{n+1}=\mathbf{F}^{n+1}+\frac{18}{6\delta t}\mathbf{M}\mathbf{U}^{n}-\frac{9}{6\delta t}\mathbf{M}\mathbf{U}^{n-1}+\frac{2}{6\delta t}\mathbf{M}\mathbf{U}^{n-2}, \tag{18}$$

where $\widetilde{\mathbf{U}}_2^{n+1}=2\mathbf{U}^n-\mathbf{U}^{n-1}$. Although we do not have an analytical proof but only numerical evidence, this is one of the few [4, 18] third-order schemes in which the velocity and the pressure are segregated of which we are aware. Contrary to schemes in which the pressure is extrapolated, we have not observed any numerical instability for the time step sizes tested. The only time step limitation to be expected is that coming from the conditional stability of the BDF3 scheme [19]. If Equation (17) is solved using the iterative scheme (14) good convergence behavior has been found (see Section 5).

4.2. Stabilized velocity correction scheme

In order to obtain the stabilized version of the velocity correction scheme one starts from the matrix form of the stabilized momentum and continuity equations (7), (8) and proceeds as in the non-stabilized case to obtain the corresponding DPPE. The equation that replaces (11) is

$$[\gamma_k\delta t\mathbf{D}\mathbf{M}_L^{-1}\mathbf{G}-\mathbf{S}_p(\tau_1)]\mathbf{P}^{n+1}=\gamma_k\delta t\mathbf{D}\mathbf{M}_L^{-1}[\mathbf{F}^{n+1}-\mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1}-\mathbf{M}\frac{D_k}{\delta t}\mathbf{U}^{n+1}$$
$$-\mathbf{S}_u(\tau_1;\mathbf{U}^{n+1})\mathbf{U}^{n+1}+\mathbf{S}_y(\tau_1;\mathbf{U}^{n+1})\mathbf{Y}^{n+1}]+\mathbf{D}\mathbf{U}^{n+1}-\mathbf{S}_z(\tau_1)\,\mathbf{Z}^{n+1}.$$

In order to obtain the velocity correction scheme an extrapolation of the velocity is used in the terms already present in the non-stabilized case and also in the terms introduced by the stabilization that involve the velocity. For the projections $\mathbf{Y}^{n+1}$ and $\mathbf{Z}^{n+1}$ used in the OSS stabilization, extrapolations $\widetilde{\mathbf{Y}}^{n+1}$ and $\widetilde{\mathbf{Z}}^{n+1}$ are also introduced. In [6] first-order extrapolations are used for both $\widetilde{\mathbf{Y}}^{n+1}$ and $\widetilde{\mathbf{Z}}^{n+1}$. In this work we use the same order of extrapolation as used in the velocity for $\mathbf{Y}^{n+1}$ and therefore call it $\widetilde{\mathbf{Y}}_q^{n+1}$ and the same order of extrapolation as used in the pressure for $\mathbf{Z}^{n+1}$, which we call $\widetilde{\mathbf{Z}}_p^{n+1}$.

The stabilized version of the velocity correction scheme that replaces (15)–(16) then reads as

$$[\gamma_k\delta t\mathbf{D}\mathbf{M}_L^{-1}\mathbf{G}-\mathbf{S}_p(\tau_1)]\mathbf{P}^{n+1}=\gamma_k\delta t\mathbf{D}\mathbf{M}_L^{-1}[\mathbf{F}^{n+1}-\mathbf{K}(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1}-\mathbf{M}\frac{D_k}{\delta t}\widetilde{\mathbf{U}}_q^{n+1}-\mathbf{S}_u(\tau_1;\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1}$$
$$+\mathbf{S}_y(\tau_1;\widetilde{\mathbf{U}}_q^{n+1})\,\widetilde{\mathbf{Y}}_q^{n+1}]+\mathbf{D}\widetilde{\mathbf{U}}_q^{n+1}-\mathbf{S}_z(\tau_1)\widetilde{\mathbf{Z}}_p^{n+1}, \tag{19}$$

$$\mathbf{M}\frac{D_k}{\delta t}\mathbf{U}^{n+1}+\mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1}+\mathbf{G}\mathbf{P}^{n+1}+\mathbf{S}_u(\tau_1;\mathbf{U}^{n+1})\mathbf{U}^{n+1}-\mathbf{S}_y(\tau_1;\mathbf{U}^{n+1})\mathbf{Y}^{n+1}=\mathbf{F}^{n+1}, \tag{20}$$

$$\mathbf{M}\mathbf{Y}^{n+1}-\mathbf{C}(\mathbf{U}^{n+1})\mathbf{U}^{n+1}=\mathbf{0}, \tag{21}$$

$$\mathbf{M}\mathbf{Z}^{n+1}-\mathbf{G}\mathbf{P}^{n+1}=\mathbf{0}. \tag{22}$$

Equations (20)–(21) need to be solved iteratively. A natural initial guess to start the iterative procedure is $\mathbf{U}^{n+1,0}=\widetilde{\mathbf{U}}_q^{n+1}$ and $\mathbf{Y}^{n+1,0}=\widetilde{\mathbf{Y}}_q^{n+1}$. In the numerical examples $\mathbf{M}$ has been approximated by $\mathbf{M}_L$ in the OSS stabilization projection steps onto the finite element space.

4.3. Implicit pressure gradient projection

Since our final objective is to obtain a 'pure' velocity correction scheme in the sense that it does not require any pressure extrapolation, it seems logical to also avoid the extrapolation of the pressure gradient projection. This can be achieved quite naturally when a discrete Laplacian is used. The projection can be written as

$$\mathbf{Z}^{n+1}=\mathbf{M}^{-1}\mathbf{G}\mathbf{P}^{n+1}.$$

DOI: 10.1002/fld

Now the term corresponding to Z^{n+1} in the DPPE can be sent to the left-hand side, to obtain the following scheme:

$$[\gamma_k \delta t \mathsf{D}\mathsf{M}_L^{-1}\mathsf{G} - \mathsf{S}_p(\tau_1) + \mathsf{S}_z(\tau_1)\mathsf{M}^{-1}\mathsf{G}]\mathsf{P}^{n+1} = \gamma_k \delta t \mathsf{D}\mathsf{M}_L^{-1}\left[\mathsf{F}^{n+1} - \mathsf{K}(\widetilde{\mathsf{U}}_q^{n+1})\widetilde{\mathsf{U}}_q^{n+1} - \mathsf{M}\frac{D_k}{\delta t}\widetilde{\mathsf{U}}_q^{n+1} - \mathsf{S}_u(\tau_1; \widetilde{\mathsf{U}}_q^{n+1})\widetilde{\mathsf{U}}_q^{n+1} + \mathsf{S}_y(\tau_1; \widetilde{\mathsf{U}}_q^{n+1})\widetilde{\mathsf{Y}}_q^{n+1}\right] + \mathsf{D}\widetilde{\mathsf{U}}_q^{n+1}.$$

$$\mathsf{M}\frac{D_k}{\delta t}\mathsf{U}^{n+1} + \mathsf{K}(\widetilde{\mathsf{U}}_q^{n+1})\mathsf{U}^{n+1} + \mathsf{G}\mathsf{P}^{n+1} + \mathsf{S}_u(\tau_1; \widetilde{\mathsf{U}}_q^{n+1})\mathsf{U}^{n+1} - \mathsf{S}_y(\tau_1; \widetilde{\mathsf{U}}_q^{n+1})\widetilde{\mathsf{Y}}_q^{n+1} = \mathsf{F}^{n+1},$$

$$\mathsf{M}\mathsf{Y}^{n+1} - \mathsf{C}(\mathsf{U}^{n+1})\mathsf{U}^{n+1} = 0.$$

5. NUMERICAL EXAMPLES

In this section we present two numerical examples where we compare the velocity correction scheme described in this paper against the results obtained with a monolithic solver. Comparisons with the more widely used pressure correction scheme [9] are also presented. The first objective is to test the order of the extrapolations that can be used for the velocity and the pressure to obtain a stable scheme. This determines the order of the splitting error. The first example shows that the velocity correction method combined with a discrete Laplacian is the only method that remains stable using second-order extrapolations. This leads to smaller splitting errors than the other schemes.

The second-order extrapolation leads to a third-order splitting error that combined with a third-order time discretization results in a third-order scheme. In the second example a convergence test is performed to show that the velocity correction scheme with discrete Laplacian allows to obtain third-order accuracy in the $L^2(\Omega)$ norm of the velocity at a certain time.

In order to simplify the presentation of the examples the following nomenclature is used. The pressure correction and velocity correction schemes are denoted 'PC' and 'VC', respectively. The use of a discrete Laplacian (with Lumped mass matrix) is denoted by 'LD' and its approximation by the continuous Laplacian is denoted by 'LC'.

5.1. Flow behind a cylinder

The first example is the flow behind a cylinder at Reynolds number $Re = 190$. It is perhaps the most typical example of oscillating flow. This example is essentially 2-D but it has been run with a 3-D mesh. The mesh, provided by Professor Rainald Lohner, has a special placement of points in the vicinity of the cylinder [20]. It is formed by 108 147 tetrahedral linear elements and 30 000 nodes. In Figure 1 the surface mesh is shown. The computational domain is $\Omega = [0, 19] \times [0, 8] \times [0, 0.2]/D$, with the cylinder D of diameter 1 centered at (4, 4). The velocity at $x = 0$ is prescribed to (1, 0, 0). At $y = 0$, $y = 8$, $z = 0$, and $z = 0.2$ the normal component of the velocity is set to zero and the tangential components are left free. At the outflow ($x = 19$) zero traction is prescribed. The time step is $\delta t = 0.05$ and the total time is $t_f = 100.0$. The time step we have used is 80.37 times bigger than the critical time step for an explicit Forward Euler scheme calculated as the minimum over all elements of

$$dtc_{\text{elem}} = \left[\frac{4\nu}{(h^e)^2} + \frac{2|\mathbf{u}^e|}{h^e}\right]^{-1}.$$

The value is within the range of 10 and 100 typically used in implicit flow calculations. Despite the BDF3 scheme being conditionally stable [19] we have observed that it introduces no instability in the monolithic case nor when it is combined with the velocity correction scheme for our examples.

DOI: 10.1002/fld

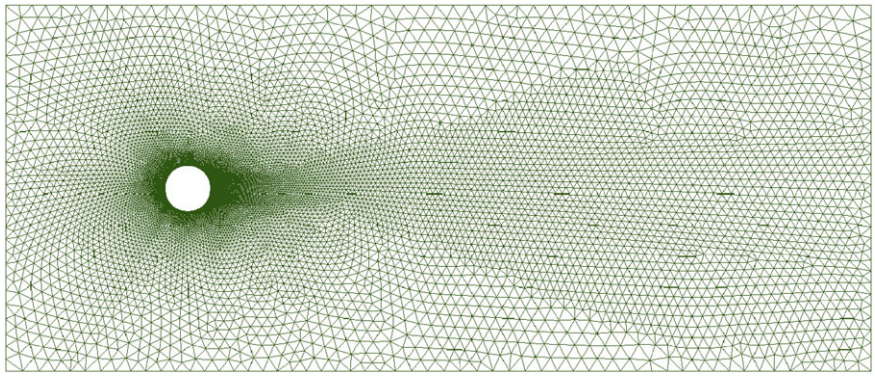

Figure 1. Mesh used for the flow behind a cylinder.

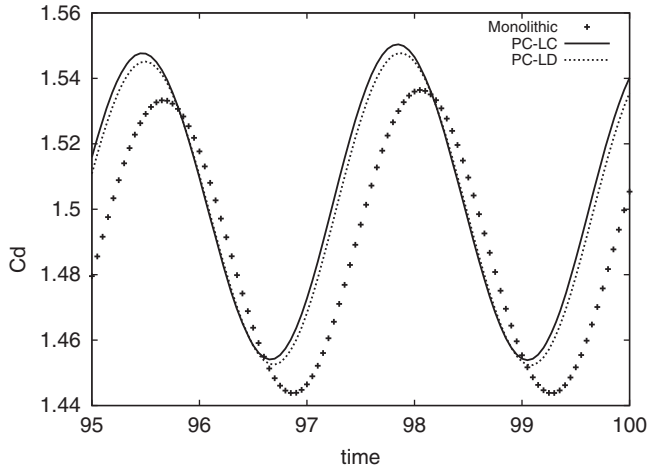

Figure 2. Drag coefficient for the flow behind a cylinder using pressure correction schemes.

First a BDF2 time discretization is used to compare the different methods against the monolithic solution. Numerical experimentation shows that the use of a second-order velocity extrapolation generates no stability problems. Instead a second-order extrapolation for the pressure leads in all cases (VC and PC) to unstable results. Therefore a first-order pressure extrapolation and a second-order velocity extrapolation was used.

The two most representative values for the flow around a cylinder are the Drag and Lift coefficients. We have observed that the Drag coefficient is much more sensitive to the scheme used and therefore it shall be used for the comparison. In Figure 2 the results obtained with a pressure correction using both the discrete Laplacian and its approximation by a continuous one are compared against the monolithic solution. The error compared with the monolithic solution is due to the splitting. The results with the velocity correction scheme are shown in Figure 3.

The velocity correction scheme combined with a discrete Laplacian provides the best results. This can be attributed to the fact that, except for the stabilization of the pressure, it does not depend on pressure extrapolations. Therefore, neglecting the stabilization, a third-order splitting error is obtained thanks to the use of a second-order velocity extrapolation. The use of the pressure extrapolation for the stabilization can be avoided by the use of an implicit projection as has been discussed in Section 4.3.

The second best results are provided by the VC scheme with the discrete Laplacian approximated by a continuous one (LC). In this scheme not only a second-order extrapolation is used for the velocity but also a first-order extrapolation is needed in the approximation of the Discrete Laplacian

DOI: 10.1002/fld

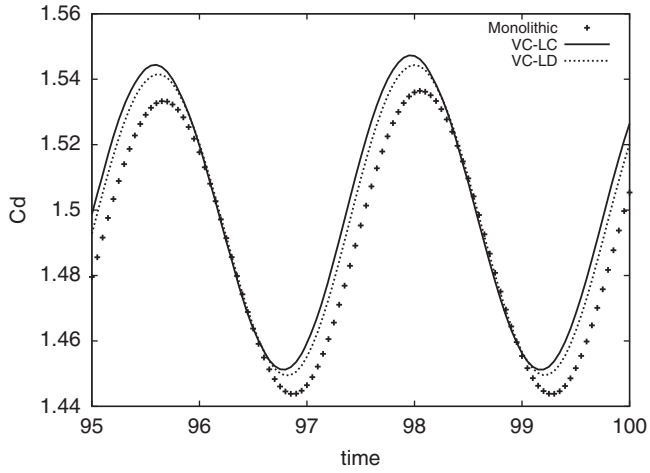

Figure 3. Drag coefficient for the flow behind a cylinder using velocity correction schemes.

Table I. Total cpu time for the cylinder.

	PC	VC
L cont	66 979 s	69 642 s
Ldisc	80 274 s	83 020 s

by a continuous one. If a second-order extrapolation for the pressure is used in the approximation of the Laplacian we have verified that, as already noted in [6], the scheme becomes unstable.

Except for the stabilization, where the velocity extrapolation is needed, the pressure correction schemes use only the pressure extrapolation that is of first order. Therefore it seems logical that they provide the worse results. The use of a discrete Laplacian provides only a small improvement in this case. We have verified that if a first-order extrapolation is used for the velocity the advantage of the velocity correction method over the pressure correction scheme disappears.

We would like to remark that the errors we have observed in the previous figures are not the most important results we have presented. They just confirm what one could expect from the extrapolations being used. The key point is that we have confirmed that the use of second-order extrapolations for the pressure leads to unstable schemes as is well known for pressure correction schemes [21] and has been observed in [6] for the velocity correction case. In [6] first-order extrapolations for both the velocity and the pressure provided stable results and second-order extrapolations for both unknowns led to instability. In this work we test the intermediate case of a first-order extrapolation for the pressure and a second-order extrapolation for the velocity that proves to be stable. Such extrapolation is not of great interest because it does not allow to obtain a third-order splitting error but it shows where the instability comes from. The key point is that if instead of approximating the discrete Laplacian by a continuous one, as done in [6], we actually solve the discrete Laplacian, no pressure extrapolation is needed and a stable scheme with a third-order splitting error can be obtained. Actually in the previous example for the velocity correction scheme with discrete Laplacian the pressure extrapolation has been used for the pressure stabilization but this can be avoided as we shall show in the next example.

Table I shows the CPU time for 2000 time steps for the previous four cases. The velocity correction scheme is slightly more expensive than the pressure correction scheme with both Laplacians. On the other hand the use of the discrete Laplacian results in an increase in CPU time of approximately 15%. The use of the velocity correction scheme with a discrete Laplacian is the most expensive option but for problems that require time accurate solutions, the possibility of obtaining third-order accuracy in time, as we shall show in the next examples, compensates for the additional cost.

DOI: 10.1002/fld

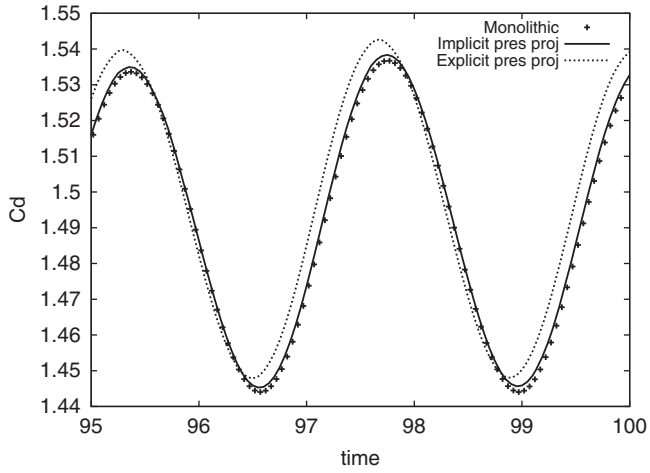

Figure 4. Drag coefficient using velocity correction scheme with discrete Laplacian and BDF3 time discretization.

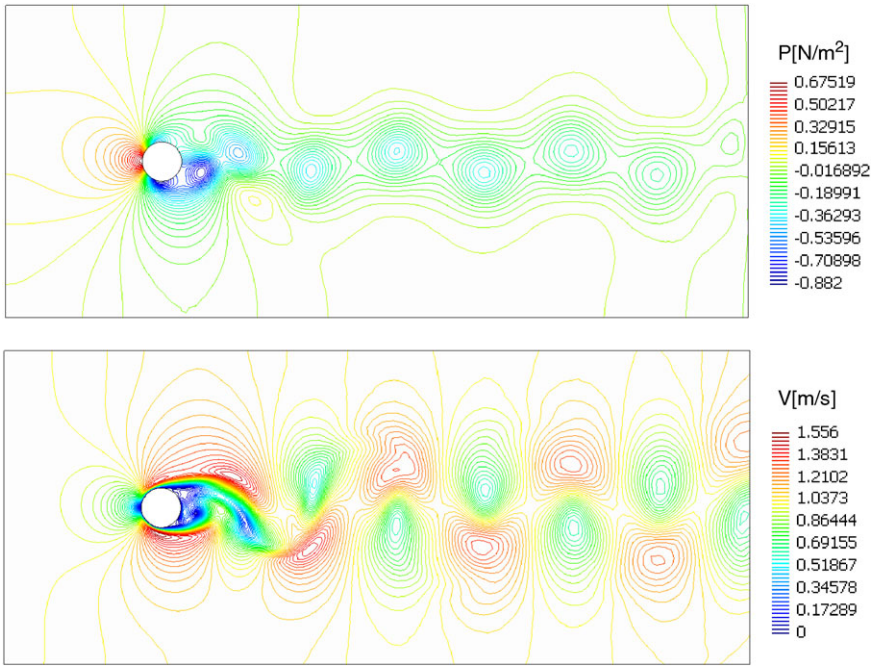

Figure 5. Pressure (top) and velocity (bottom) at $t = 100\,\mathrm{s}$.

Now we have seen that the velocity correction scheme with a discrete Laplacian can provide a third-order splitting error, we combine it with a third-order time discretization scheme such as BDF3. In order to avoid the need for any pressure extrapolation, we use the implicit pressure gradient projection as described in Section 4.3. According to our knowledge this is the first time this projection has been treated implicitly for the OSS stabilization. In Figure 4 the drag coefficient is compared to the results obtained with the monolithic scheme. The case with explicit pressure gradient projection is also included in the comparison. It can be seen that the use of a pure velocity correction scheme, one without any pressure extrapolation, leads to very small splitting errors.

The velocity and pressure contours obtained with the velocity correction BDF3 scheme with discrete Laplacian and implicit pressure gradient projection are shown in Figure 5.

DOI: 10.1002/fld

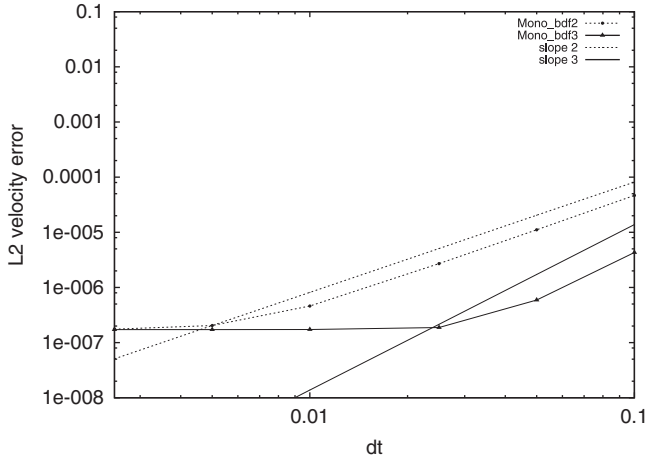

Figure 6. Convergence test with monolithic solver.

5.2. *Convergence test*

The second example is used to test the time convergence rate numerically. It has been borrowed from [5].

The Stokes problem is solved on the unit square, $]0,1[^2$. The force term is set so that the exact solution is

$$p(x,y,t)=\cos(\pi x)\sin(\pi y)\sin(t),$$
$$u(x,y,t)=\pi\sin(2\pi y)\sin^2(\pi x)\sin(t),$$
$$v(x,y,t)=-\pi\sin(2\pi x)\sin^2(\pi y)\sin(t).$$

The domain is discretized using $Q2/Q2$ finite elements of size $h=1/200$. Boundary and initial conditions are forced to satisfy the previous equations. The time step size we use varies from 0.0025 to 0.1 s and the results at $t=1.0$ s are presented.

In Figure 6 we present the convergence results for the $L^2(\Omega)$ norm of the velocity error at the final time using a monolithic formulation. These results can be used as a reference against which the results obtained with the fractional step results can be compared because they have no splitting error. With the BDF2 scheme the second-order slope can easily be observed. For the third-order scheme the spatial discretization error soon becomes dominant and the third-order slope can only be seen for the two bigger time steps.

In Figure 7 we present the results with the velocity correction fractional step scheme. Both BDF2 and BDF3 results show third-order convergence. In the BDF2 case, this can be explained by the fact that the error due to the temporal discretization is small compared with the splitting error which is third-order accurate.

In Figure 8 we present the convergence results for the $L^2(\Omega)$ norm of the pressure error with the velocity correction fractional step scheme using BDF2 and BDF3 time discretizations. The convergence rate is slightly higher than 2.

Finally, in Figure 9 we present some results with the VC BDF3 scheme and continuous Laplacian. The third order is lost and only second-order accuracy is obtained due to the error introduced by the approximation of the discrete Laplacian by the continuous one. Remember we are using a first-order extrapolation for the pressure because we have seen that a second-order extrapolation leads to an unstable scheme. We have also included in the comparison the results obtained with the discrete Laplacian solved by a Richardson iteration. Actually only three Richardson iterations have been allowed per time step. It is interesting to note how two extra Richardson iterations allow to recover results that are nearly third-order accurate and very close to that obtained when the discrete Laplacian is solved with a conjugate gradient method.

DOI: 10.1002/fld

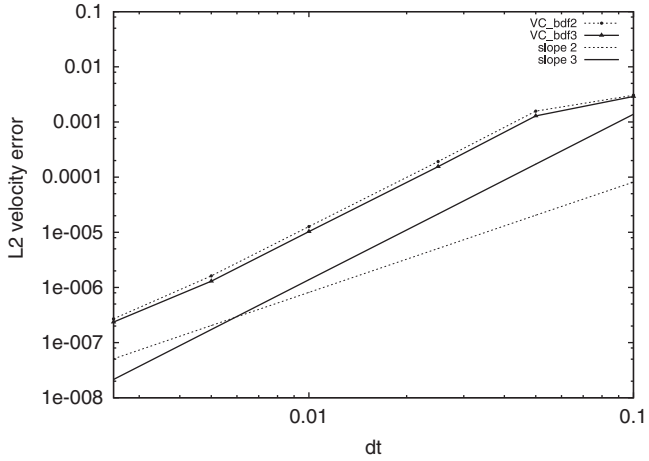

Figure 7. Convergence test with velocity correction scheme.

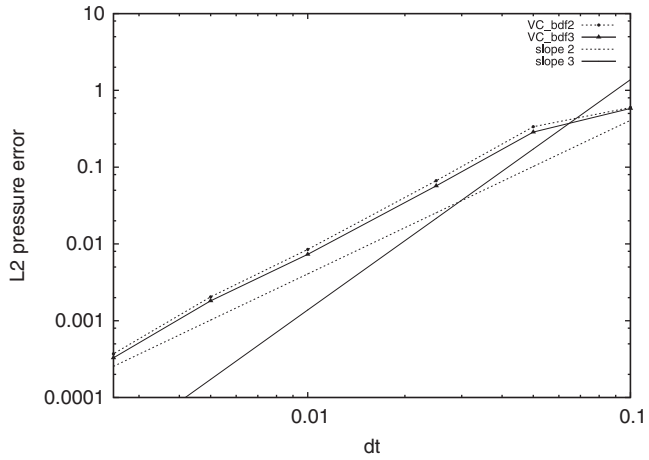

Figure 8. Pressure convergence with the velocity correction scheme.

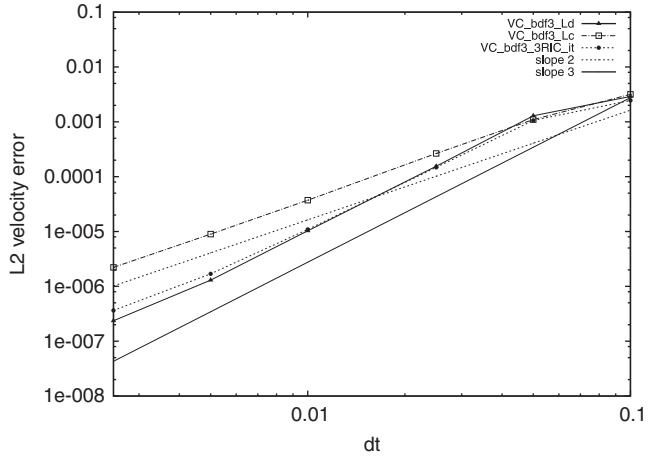

Figure 9. Convergence test with BDF3 VC scheme and different Laplacians.

DOI: 10.1002/fld

6. CONCLUSIONS

In this work we have explored numerically the velocity correction scheme proposed in [6] and introduced several modifications that have allowed us to obtain stable third-order results, something that had previously not been possible. The first key point is that we have found that is the second-order pressure extrapolation that makes the original scheme unstable. In [6] only equal order (first or second) extrapolations for the velocity and the pressure had been tested. The use of second-order extrapolations had proved unstable but it had not been clarified whether it was the second-order extrapolation for the pressure, the velocity, or both that caused the instability. An intermediate case with first-order pressure extrapolation and second-order velocity extrapolation is tested in this work to show that it provides stable results. Despite such does not having much real interest because the third-order splitting error is lost, it shows that it is the second-order pressure extrapolation that makes the scheme unstable and that the second-order velocity extrapolation does not harm stability.

In order to obtain a third-order splitting error a 'pure' velocity correction scheme is proposed. That is, the original scheme is modified so that no pressure extrapolation is needed and only a second-order velocity extrapolation is used. Two key modifications are introduced to avoid the need of a pressure extrapolation. The use of the pressure extrapolation in the approximation of the discrete Laplacian by a continuous one is avoided by solving for the discrete Laplacian directly. The second modification is related to the split OSS stabilization technique used. In order to avoid needing an extrapolation for the pressure gradient projection an implicit pressure gradient projection is implemented. Finally the scheme is combined with a BDF3 time discretization to obtain a third-order temporal error.

For the solution of the discrete Laplacian two options have been implemented. The first one is the straightforward application of a conjugate gradient iterative procedure. The second option is to use a preconditioned Richardson iteration with the continuous Laplacian as preconditioner. More elaborate options such as a flexible conjugate gradient solver for the discrete Laplacian preconditioned by a conjugate gradient solver for the continuous Laplacian can also be proposed.

In the convergence test borrowed from [5] the expected convergence slope for the $L^2(\Omega)$ velocity error is verified. For the velocity correction case it is observed that both BDF2 and BDF3 schemes show third-order accuracy. For the second-order time discretization this can be explained by the fact that the splitting error which is third-order accurate is dominant. When the continuous Laplacian approximation is used the third-order accuracy obtained with the velocity correction scheme is lost. This is caused by the use of a first-order pressure extrapolation. It has also been shown that the use of a preconditioned Richardson iteration with only three iterations allows to recover results that are nearly third-order accurate and very close to that obtained with the discrete Laplacian.

From the previous examples we can conclude that the superior behavior of velocity correction methods can be attributed to the fact that second-order velocity extrapolations lead to stable schemes. In contrast, second-order pressure extrapolations lead to unstable schemes. This not only happens in the pressure correction case but also in the velocity correction with continuous Laplacian approximation. Therefore we can say that the use of a discrete Laplacian is more significant in the velocity correction scheme because it provides a 'pure' VC scheme where no pressure extrapolation is needed.

ACKNOWLEDGEMENTS

H. Owen acknowledges the support received from the Departament d'Universitats, Recerca i Societat de la Informació of the Generalitat de Catalunya (Catalan Government) and the European Social Fund through a doctoral grant.

REFERENCES

1. Chorin AJ. A numerical method for solving incompressible viscous problems. *Journal of Computational Physics* 1967; **2**:12–26.
2. Temam R. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionaires (I). *Archives for Rational Mechanics and Analysis* 1969; **32**:135–153.

3. van Kan J. A second-order accurate pressure correction scheme for viscous incompressible flow. *SIAM Journal on Scientific Computing* 1986; **7**:870–891.
4. Karniadakis GE, Israeli M, Orzag SE. High order splitting methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics* 1991; **59**:414–443.
5. Guermond JL, Shen J. Velocity-correction projection method for incompressible flows. *SIAM Journal on Numerical Analysis* 2003; **41**:112–134.
6. Badia S, Codina R. Pressure segregation methods based on a discrete pressure Poisson equation. An algebraic approach. *International Journal for Numerical Methods in Fluids* 2008; **56**:351–382.
7. Guermond JL, Minev P, Shen J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:6011–6045.
8. Badia S, Codina R. Algebaric pressure segregation methods for the incompressible Navier–Stokes equations. *Archives of Computational Methods in Engineering* 2008; **15**:343–369.
9. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112–140.
10. Perot JB. An analysis of the fractional step method. *Journal of Computational Physics* 1993; **108**:51–58.
11. Quarteroni A, Saleri F, Veneziani A. Factorization methods for the numerical approximation of Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2000; **188**:505–526.
12. Turek S. A comparative study of time-stepping techniques for the incompressible Navier–Stokes equations: from fully implicit nonlinear schemes to semi-implicit projection methods. *International Journal for Numerical Methods in Fluids* 1996; **22**:987–1011.
13. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
14. Codina R. Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales. *Applied Numerical Mathematics* 2008; **58**:264–283.
15. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: Berlin, 1991.
16. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
17. Heywood JG, Rannacher R. Finite element approximation of the nonstationary Navier–Stokes problem. IV: Error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis* 1990; **27**:353–384.
18. Nikitin N. Third-order-accurate semi-implicit Runge–Kutta scheme for incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2006; **51**:221–233.
19. Hundsdorfer W, Verwer JG. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer: Berlin, 2003.
20. R. Löhner, Yang C, Cebral J, Camelli F, Soto O, Waltz J. Improving the speed and accuracy of projection-type incompressible flow solvers. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3087–3109.
21. Shen J. A remark on the projection-3 method. *International Journal for Numerical Methods in Fluids* 1993; **16**:249–253.

DOI: 10.1002/fld