Arch Comput Methods Eng
DOI 10.1007/s11831-008-9020-3

ORIGINAL PAPER

Algebraic Pressure Segregation Methods for the Incompressible Navier-Stokes Equations

S. Badia · R. Codina

Received: 12 November 2007 / Accepted: 12 November 2007

Abstract This work is an overview of algebraic pressure segregation methods for the incompressible Navier-Stokes equations. These methods can be understood as an inexact *LU* block factorization of the original system matrix. We have considered a wide set of methods: algebraic pressure correction methods, algebraic velocity correction methods and the Yosida method. Higher order schemes, based on improved factorizations, are also introduced. We have also explained the relationship between these pressure segregation methods and some widely used preconditioners, and we have introduced predictor-corrector methods, one-loop algorithms where nonlinearity and iterations towards the monolithic system are coupled.

1 Introduction

The basic idea behind pressure segregation methods is to split the transient Navier-Stokes equations, a coupled velocity-pressure system, into two or more steps that decouple velocity and pressure computations. The numerical approximation of the uncoupled continuous system has a lower computational cost than the one for the original coupled problem. Because of their computational efficiency, these strategies have become widely popular.

The original approach to pressure segregation methods is based on a splitting over the continuous in space and discrete in time problem. The segregation of the pressure is based on a projection operator, that explains why the original scheme is known as the projection method. This approach requires to introduce appropriate boundary conditions at every step of the algorithm, a point that has been fairly controversial. In Sect. 3 we introduce a motivation to this approach and a state-of-the art review. The reader is referred to [58] for a recent overview of pressure segregation methods motivated at the continuous level (in space).

More recently, these splitting methods have been reinterpreted as inexact *LU* block factorizations. In fact, this approach is not simply formal, and has some advantages over the continuous approach. At the algebraic level, we can skip the discussion about the appropriate boundary conditions to be imposed, because they have already been enforced in the monolithic system.[1] Furthermore, we can easily design stabilized pressure segregation methods. The pros of the algebraic approach are listed in Sect. 4. There, algebraic pressure segregation methods are introduced from two points of view, namely, as approximations based on variable extrapolation and from the inexact *LU* block factorizations mentioned. In both cases one may derive the so called pressure-correction and velocity-correction schemes. The Yosida scheme is presented as a way to improve the accuracy of these factorizations.

The first author's research was supported by the European Community through the Marie Curie contract *NanoSim* (MOIF-CT-2006-039522).

S. Badia (✉) · R. Codina
CIMNE, Universitat Politècnica de Catalunya, Jordi Girona 1-3,
Edifici C1, 08034 Barcelona, Spain
e-mail: sbadia@cimne.upc.edu

R. Codina
e-mail: ramon.codina@upc.edu

Present address:
S. Badia
Applied Mathematics and Applications, MS-1320, Sandia
National Laboratories, Albuquerque, NM 87185-1320, USA

[1] This does not imply that the pressure error layers on the boundaries disappear.

Algebraic pressure-correction schemes are discussed in Sect. 5, whereas more recent algebraic velocity-correction methods are discussed in Sect. 6. The objective of these sections is to present these schemes in conjunction with different time integration algorithms, analyze their formal properties and present some stability results. It is also explained how to incorporate stabilization methods to deal with convection-dominated flows and to avoid any compatibility condition between the velocity and pressure interpolation.

In Sect. 7 we explain the link between algebraic pressure segregation methods and some widely used block preconditioners for the Navier-Stokes equations. We also introduce predictor-corrector methods, one-loop algorithms that couple the nonlinear iterations related to the convective term with the solver iterations towards the monolithic system.

Sections 5, 6 and 7 are extension of the work in [4].

2 The Incompressible Navier-Stokes Equations

Let us start with the basic equations of fluid mechanics for incompressible flows. These equations are obtained from principles of conservation under the basic assumptions assumed for the continuum mechanics theory.

The evolutionary equations for an incompressible Newtonian fluid moving in a domain $\varOmega$ of $\mathbb{R}^d$ ($d = 2$ or 3 being the space dimension) in a time interval $[0, T]$ consists of finding a velocity $\boldsymbol{u}$ and a (kinematic) pressure p such that

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \varDelta \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \varOmega \times (0, T), \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \varOmega \times (0, T), \tag{1b}$$

where $\boldsymbol{f}$ is the force vector and ν the kinematic viscosity.

The set of equations (1a)–(1b) has to be supplemented with appropriate boundary and initial conditions in order to have a well-posed system. The boundary $\varGamma \equiv \partial\varOmega$ is a $(d-1)$-dimensional manifold assumed locally Lipschitz (i.e., smooth enough). It is assumed that $\varGamma$ can be partitioned into two non-overlapping subsets $\varGamma_D$ and $\varGamma_N$, such that $\varGamma_D \cap \varGamma_N = \emptyset$ and $\varGamma_D \cup \varGamma_N = \varGamma$. Then, we have in the simplest case

$$\boldsymbol{u} = \boldsymbol{u}_g \quad \text{on } \varGamma_D \times (0, T),$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} = \boldsymbol{t}_N \quad \text{on } \varGamma_N \times (0, T),$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \quad \text{in } \varOmega \times \{0\},$$

where $\boldsymbol{n}$ is the unit outward normal vector to $\varGamma$ and $\boldsymbol{\sigma} = -p\boldsymbol{I} + \nu(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^t)$ is the stress tensor. For purely Dirichlet type boundary conditions, i.e. $\varGamma_N = \emptyset$, the total flux produced by $\boldsymbol{u}_g$ must be zero in order to have a well posed problem.

For the sake of simplicity we have assumed the homogeneous boundary condition,

$$\boldsymbol{u}_g = \boldsymbol{0} \quad \text{on } \varGamma \times (0, T) \tag{2}$$

in several parts of the following exposition.

When the stationary case is considered, the time derivative disappears, resulting in the steady, incompressible Navier-Stokes (momentum) equation,

$$-\nu \varDelta \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \varOmega. \tag{3}$$

2.1 Some Function Spaces

Let us introduce some notation that will be used hereafter. For a wide exposition of the mathematical concepts mentioned here we refer to any standard functional analysis text (for instance, [75]). More specific aspects of functional analysis when applied to the Navier-Stokes equations can be found in [53] and [96].

Let $\varOmega \subset \mathbb{R}^d$, $d = 2$ or 3, be a bounded domain. We denote by $L^p(\varOmega)$, $1 \le p < \infty$, the space of real functions defined on $\varOmega$ with the p-th power absolutely integrable with respect to the Lebesgue measure. It is a Banach space with the associated norm

$$\|u\|_{L^p(\varOmega)} := \left(\int_\varOmega |u(x)|^p \, \mathrm{d}\varOmega \right)^{1/p}.$$

For $1 < p < \infty$, $L^p(\varOmega)$ is a reflexive space and its dual space is $L^q(\varOmega)$, with q such that $1/p + 1/q = 1$.

The case $p = 2$ is of special interest; $L^2(\varOmega)$ is a Hilbert space endowed with the scalar product

$$(u, v)_\varOmega := \int_\varOmega u(x) v(x) \, \mathrm{d}\varOmega$$

and its induced norm

$$\|u\|_{L^p(\varOmega)} := (u, u)_\varOmega^{1/2}.$$

Furthermore, $L^2(\varOmega)$ is identified with its dual space. In the following, we will likely omit the subscript referred to the domain for which we consider the scalar product when this is the problem domain, that is to say, $\varOmega$. Moreover, we will write the $L^2(\varOmega)$ norm simply as $\|\cdot\|_0$.

The space $L^\infty(\varOmega)$ (for $p = \infty$) consists of essentially bounded functions in $\varOmega$. It is a Banach space equipped with the norm

$$\|u\|_{L^\infty(\varOmega)} := \operatorname*{ess-sup}_{\varOmega} |u(x)|.$$

Its dual space is $L^1(\varOmega)$. Furthermore, $L^\infty(\varOmega) \subset L^p(\varOmega)$ for all $p \in [1, \infty)$.

The Sobolev space $W^{m,p}(\varOmega)$ is the space of functions in $L^p(\varOmega)$ whose weak derivatives of order less than or equal to

m belong to $L^p(\Omega)$, being m an integer and $1 \le p \le \infty$. It is a Banach space endowed with the usual norm, which we denote as $\|\cdot\|_{W^{m,p}(\Omega)}$, or in a more compact manner $\|\cdot\|_{m,p}$. When $p = 2$, the space $W^{m,2}(\Omega) = H^m(\Omega)$ is a Hilbert space endowed with a scalar product and its associated norm $\|\cdot\|_{H^m(\Omega)}$ (also denoted $\|\cdot\|_m$).

The space $H_0^1(\Omega)$ consists of functions of $H^1(\Omega)$ with zero trace on the boundary. As Ω is a bounded domain, the Poincaré inequality holds. Therefore the norm $\|\nabla u\|_0$ is equivalent to $\|u\|_1$ on $H_0^1(\Omega)$. Furthermore, we denote by $H^{-k}(\Omega)$ the dual space of $H_0^k(\Omega)$.

The duality pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$ is denoted by $\langle\cdot,\cdot\rangle_\Omega$. Again, we can omit the domain subscript when Ω is the problem domain.

We shall often consider d-dimensional vector functions with components in one of these spaces. We shall indicate them by boldface letters, for instance $\boldsymbol{H}^m(\Omega) = (H^m(\Omega))^d$. In the next, we will not distinguish between scalar products or norms for scalar or vector-valued functions.

Let us introduce some convenient spaces for the treatment of the incompressibility constraint. The first one is

$$\boldsymbol{H}(\mathrm{div}, \Omega) := \{\boldsymbol{u} \in \boldsymbol{L}^2(\Omega) \mid \nabla\cdot\boldsymbol{u} \in L^2(\Omega)\},$$

which is a Hilbert space with the scalar product $(u, v)_{\mathrm{div}} = (u, v) + (\nabla\cdot u, \nabla\cdot v)$ (scaling coefficients may be introduced in the definition of this norm to make it dimensionally consistent). We also introduce

$$\boldsymbol{H}_0(\mathrm{div}, \Omega) := \{\boldsymbol{u} \in \boldsymbol{H}(\mathrm{div}, \Omega) \mid \boldsymbol{n}\cdot\boldsymbol{u}|_\Gamma = 0\}.$$

We define the spaces whose functions are *weakly divergence-free*,

$$\boldsymbol{J}_0 := \{\boldsymbol{u} \in \boldsymbol{L}^2(\Omega) \mid \nabla\cdot\boldsymbol{u} = 0,\ \boldsymbol{n}\cdot\boldsymbol{u}|_\Gamma = 0\} \tag{4}$$

and,

$$\boldsymbol{J}_1 := \{\boldsymbol{u} \in \boldsymbol{H}_0^1(\Omega) \mid \nabla\cdot\boldsymbol{u} = 0\}.$$

Since $\boldsymbol{J}_0$ is a closed subspace of $\boldsymbol{L}^2(\Omega)$, it can be decomposed as $\boldsymbol{L}^2(\Omega) = \boldsymbol{J}_0 \oplus \boldsymbol{J}_0^\perp$, where

$$\boldsymbol{J}_0^\perp := \{\boldsymbol{u} \in \boldsymbol{L}^2(\Omega) \mid \boldsymbol{u} = \nabla p,\ p \in H^1(\Omega)\}. \tag{5}$$

We define $P_{\boldsymbol{J}_0}$ as the orthogonal $L^2(\Omega)$-projector onto $\boldsymbol{J}_0$. This operator is of main importance, and the basis of the original projection methods (see [25] and [94]). It is obviously continuous in $\boldsymbol{L}^2(\Omega)$ by its definition. In fact, for Ω being an open bounded set of class $\mathcal{C}^2$, the operator $P_{\boldsymbol{J}_0}$ also maps $\boldsymbol{H}^1(\Omega)$ into itself and is continuous with respect to $\|\cdot\|_1$ (see [96]), that is to say, there exists a constant $c > 0$ such that:

$$\|P_{\boldsymbol{J}_0} f\|_1 \le c\|f\|_1, \quad \forall f \in \boldsymbol{H}^1(\Omega). \tag{6}$$

We need to introduce three forms associated to the different terms of the Navier-Stokes equations in their weak form, defined on the appropriate spaces. Let us start with the form related to the viscous term,

$$a(\boldsymbol{u}, \boldsymbol{v}) := \nu(\nabla\boldsymbol{u}, \nabla\boldsymbol{v}), \quad \forall\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{H}^1(\Omega).$$

This is a bilinear continuous form on $\boldsymbol{H}_0^1(\Omega)$ which is coercive with respect to $\|\cdot\|_1$.

The second form to be introduced is required for the pressure gradient and the incompressibility constraint,

$$b(\boldsymbol{v}, q) := -(q, \nabla\cdot\boldsymbol{v}), \quad \forall\boldsymbol{v} \in \boldsymbol{H}_0^1(\Omega),\ \forall q \in L^2(\Omega),$$

which is also continuous with respect to the norms $\|q\|_0$ and $\|\boldsymbol{v}\|_1$.

Finally, the trilinear form associated to the convective term in its standard form is

$$c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \int_\Omega (\boldsymbol{u}\cdot\nabla\boldsymbol{v})\cdot\boldsymbol{w}\,\mathrm{d}\Omega,$$
$$\forall\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{H}^1(\Omega),\ \forall\boldsymbol{w} \in \boldsymbol{H}_0^1(\Omega)$$

which is also continuous and well defined on these spaces. If $\boldsymbol{u} \in \boldsymbol{J}_1$, the form is skew-symmetric in its two last arguments and thus,

$$c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{v}) = 0, \quad \forall\boldsymbol{u} \in \boldsymbol{J}_1,\ \forall\boldsymbol{v} \in \boldsymbol{H}_0^1(\Omega).$$

Its skew-symmetric form, introduced by Temam in [96],

$$\widetilde{c}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \frac{1}{2}(c(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) - c(\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v})),$$
$$\forall\boldsymbol{u} \in \boldsymbol{H}^1(\Omega),\ \boldsymbol{v}, \boldsymbol{w} \in \boldsymbol{H}_0^1(\Omega)$$

is also used in the following. The boundedness properties of $c(\cdot,\cdot,\cdot)$, in particular the continuity, are inherited by this skew-symmetric form. In any case the superscript $\widetilde{}$ will be omitted.

For the treatment of evolutionary problems, we require the following notation. Given $T > 0$, $1 < p < \infty$, and X a Banach space with norm $\|\cdot\|_X$, let $L^p(0, T; X)$ be the space of functions $f : (0, T) \to X$ such that $\|\boldsymbol{u}\|_{L^p(0,T;X)} = (\int_0^T \|\boldsymbol{u}(s)\|_X^p\,\mathrm{d}s)^{1/p} < \infty$. In the case of $p = \infty$, we demand the property $\sup_{0\le s\le T} \|\boldsymbol{u}(s)\|_X < \infty$. These spaces will be often employed.

2.2 Finite Element Approximation

In this section we approximate the previous spaces with finite-dimensional subspaces that can be handled numerically. We start summarizing the definition of a finite element, used for constructing the approximation of the previous functional spaces. The reader is referred to the texts of Ciarlet [28], Strang and Fix [91], Brenner and Scott [14] or

Ern and Guermond [47] for the analysis of the finite element interpolation theory.

Let Θ_h be a finite element partition of the domain Ω in a family of elements $\{K_e\}_{e=1}^{n_{el}}$, that is to say, a sequence of elements K_e for $e = 1, \ldots, n_{el}$, being n_{el} the number of elements. We denote the diameter of the sphere that circumscribes the element K by h_K and the diameter of the sphere inscribed in K by ϱ_K. We also call $h = \max_{K\in\Theta_h}(h_K)$ and $\varrho = \min_{K\in\Theta_h}(\varrho_K)$. We assume that all the element domains $K \in \Theta_h$ are the image of a reference element $\widetilde{K}$ through polynomial mappings F_k, affine for simplicial elements, bilinear for quadrilaterals and trilinear for hexahedra. On $\widetilde{K}$ we define the polynomial spaces $R_k(\widetilde{K})$ where R_k is, for simplicial elements, the set of polynomials in $x_1, \ldots, x_d$ of degree less than or equal to k, called P_k. For quadrilaterals and hexahedra R_k consists of polynomials in $x_1, \ldots, x_d$ of degree less than or equal to k in each variable, called Q_k. The finite element spaces we will use in the following are:

$$
\begin{aligned}
\mathcal{Q}_h = \{ & q_h \in \mathcal{C}^0(\Omega) \cap \boldsymbol{L}^2(\Omega)/\mathbb{R} \mid \\
& q_h|_K = \widetilde{q} \circ F_K^{-1},\ \widetilde{q} \in R_{k_\mathrm{q}}(\widetilde{K}),\ K \in \Theta_h\}, \\
\mathcal{V}_h = \{ & \boldsymbol{v}_h \in (\mathcal{C}^0(\Omega))^d \mid \\
& \boldsymbol{v}_h|_K = \widetilde{\boldsymbol{v}} \circ F_K^{-1},\ \widetilde{\boldsymbol{v}} \in (R_{k_\mathrm{v}}(\widetilde{K}))^d,\ K \in \Theta_h\}, \qquad (7a)\\
\mathcal{V}_{h,0} = \{ & \boldsymbol{v}_h \in \mathcal{V}_h \mid \boldsymbol{v}_h|_\Gamma = \boldsymbol{0}\}, \qquad (7b)
\end{aligned}
$$

which are finite dimensional spaces approximating $L^2(\Omega)$, $\boldsymbol{H}^1(\Omega)$ and $\boldsymbol{H}_0^1(\Omega)$, respectively. In the following, finite element functions will be identified with a subscript h. The space $\mathcal{Q}_h$ will be associated with the pressure (k_q being the degree of the approximation for the pressure) and $\mathcal{V}_{h,0}$ (of degree k_v) with the velocity field. Both spaces are referred to the same partition and both are made up with continuous functions. In some cases, different combinations of $\mathcal{Q}_h$ and $\mathcal{V}_{h,0}$ will be used, for instance accomplishing the inf-sup condition. We will point out these particular cases where used.

2.3 The Variational Formulation of the Navier-Stokes Equations

2.3.1 The Continuous Level

We obtain in this section the variational (or weak) form of the stationary Navier-Stokes equations for incompressible flows. However, all the concepts exposed in this section can be easily extended to the transient case, as well as to other mixed methods. The study of mixed methods was pioneered by Babuška in [3], whose ideas where applied to the specific case of the Stokes equations in [15] by Brezzi. A more recent complete text on this field is the book by Brezzi and Fortin [18].

Let $\mathcal{V} \equiv \boldsymbol{H}^1(\Omega)$, $\mathcal{V}_0 \equiv \boldsymbol{H}_0^1(\Omega)$ and $\mathcal{Q} \equiv \boldsymbol{L}^2(\Omega)/\mathbb{R}$ denote the spaces for velocity and pressure, respectively, with associated norms $\|\boldsymbol{v}\|_\mathcal{V}$ and $\|q\|_\mathcal{Q}$, and $\boldsymbol{f} \in \mathcal{V}'$. Then, the weak form of (3)–(1b) consist of finding $\boldsymbol{u} \in \mathcal{V}$ and $p \in \mathcal{Q}$ such that:

$$
a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) + c(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) = \langle \boldsymbol{f}, \boldsymbol{v}\rangle, \quad \forall \boldsymbol{v} \in \mathcal{V}, \tag{8a}
$$

$$
b(\boldsymbol{u}, q) = 0, \quad \forall q \in \mathcal{Q}, \tag{8b}
$$

where we have used the forms defined above. The study of this problem leads to introduce the operator defined as

$$
B : \mathcal{V}_0 \to \mathcal{Q}' \mid \langle B(\boldsymbol{v}), q\rangle_{\mathcal{Q}'\times\mathcal{Q}} = b(\boldsymbol{v}, q) \quad \forall \boldsymbol{v} \in \mathcal{V}_0,\ \forall q \in \mathcal{Q}
$$

and B^t its adjoint. Then, the condition in order for (8) to be well posed reads as follows: there exists a constant β such that,

$$
\inf_{q\in\mathcal{Q}} \sup_{\boldsymbol{v}\in\mathcal{V}_0} \frac{b(\boldsymbol{v}, q)}{\|\boldsymbol{v}\|_\mathcal{V} \|q\|_{\mathcal{Q}/\ker B^t}} \geq \beta > 0, \tag{9}
$$

in which case existence of the solution can be proved. This condition, referred to as inf-sup condition along this work, is usually called the LBB condition, honoring the works of Ladyzhenskaya [72], Babuška [3] and Brezzi [15].

Ladyzhenskaya proved condition (9) for the particular case of the Stokes problem (for which existence and uniqueness are known) in [72].

For the evolutionary Navier-Stokes equations, let $(\mathcal{V}_0)_t \equiv L^2(0, T; \boldsymbol{H}_0^1(\Omega))$ the space for the velocity and $\mathcal{Q}_t \equiv \mathcal{D}(0, T; \boldsymbol{L}^2(\Omega)/\mathbb{R})$ (the space of functions whose L^2-spatial norm is a distribution in time) the space for the pressure. Then, the weak form of (1a)–(1b) consists of finding $\boldsymbol{u} \in (\mathcal{V}_0)_t$ and $p \in \mathcal{Q}_t$ such that:

$$
\begin{aligned}
&(\partial_t \boldsymbol{u}, \boldsymbol{v}) + a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) + c(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) = \langle \boldsymbol{f}, \boldsymbol{v}\rangle, \\
&\quad \forall \boldsymbol{v} \in (\mathcal{V}_0), \qquad (10a)\\
&b(\boldsymbol{u}, q) = 0, \quad \forall q \in \mathcal{Q}. \qquad (10b)
\end{aligned}
$$

For $2d$ problems, the existence and uniqueness theory is fairly complete. The solution is as regular as allowed by the data and we have continuous dependence of the data in the corresponding function spaces (see [96]). In the three-dimensional case the $2d$ result cannot be extended due to the lack of information concerning the regularity of the weak solution. Only partial results have been proved (see [96]). Regarding *strong* (or classical) solutions, existence and uniqueness have been proved on some time interval depending on the data. It is known since the work of Leray [73] that, provided the data are regular enough, there is locally in time a unique smooth solution.

Unfortunately, uniqueness is not proved for the *weak* solutions for which existence is known. There is a gap between the class of solutions for which existence is proved and the smaller class for which uniqueness is proved. Further, it has been proved that any *weak* solution of (10) belongs to $L^{8/3}(0,T;\boldsymbol{L}^4(\Omega))$ and any *strong* solution of (10) belongs to $L^8(0,T;\boldsymbol{L}^4(\Omega))$. The distance between these two classes of functions seems to be the distance that separate us from the global result of existence and uniqueness.

Summarizing, there are two fundamental problems that still remain open: uniqueness of weak solutions and existence of strong solutions for any time value. The key importance of these questions has motivated their consideration as one of the seven "millennium problems" proposed by the Clay Mathematics Institute [29] with a reward of $1,000,000.

2.3.2 *The Discrete Level*

We now consider the approximation of the weak form (8) using the finite element approximation theory. Let $\mathcal{V}_{h,0}$ and $\mathcal{Q}_h$ be the finite dimensional subspaces approximating $\mathcal{V}_0$ and $\mathcal{Q}$ respectively, constructed as indicated in (7). The discrete version of system (8) consists of finding $\boldsymbol{u}_h \in \mathcal{V}_{h,0}$ and $p_h \in \mathcal{Q}_h$ such that

$$a(\boldsymbol{u}_h,\boldsymbol{v}_h)+b(\boldsymbol{v}_h,p_h)+c(\boldsymbol{u}_h,\boldsymbol{u}_h,\boldsymbol{v}_h)=\langle \boldsymbol{f},\boldsymbol{v}_h\rangle,$$
$$\forall \boldsymbol{v}\in\mathcal{V}_{h,0}, \tag{11a}$$
$$b(\boldsymbol{u}_h,q_h)=0, \quad \forall q\in\mathcal{Q}_h. \tag{11b}$$

We denote by $\mathcal{Q}_h'$ the space of bilinear forms on $\mathcal{Q}_h$, which is a subspace of $\mathcal{Q}'$ (see [18]). Then, we can introduce the operator B_h, the equivalent to B at the discrete level, that is

$$B_h:\mathcal{V}_{h,0}\to\mathcal{Q}_h' \mid \langle B_h(\boldsymbol{v}_h),q_h\rangle_{\mathcal{Q}_h'\times\mathcal{Q}_h}=b(\boldsymbol{v}_h,q_h),$$

where $\boldsymbol{v}_h\in\mathcal{V}_{h,0}$ and $q_h\in\mathcal{Q}_h$. The equivalent version of the inf-sup condition at the discrete level holds if there exists a constant β_d, independent of the mesh size h, such that

$$\inf_{q_h\in\mathcal{Q}_h}\sup_{v_h\in\mathcal{V}_{h,0}}\frac{b(\boldsymbol{v}_h,q_h)}{\|\boldsymbol{v}_h\|_{\mathcal{V}}\|q_h\|_{\mathcal{Q}/\ker B_h^t}}\geq\beta_d>0, \tag{12}$$

in which case existence can be proved. Again, the non-uniqueness of the continuous equation is inherited by the discretized version (11) using the finite element method. The analysis of the approximation of branches of non-singular solutions is due to Brezzi, Rappaz and Raviart [19–21].

Unfortunately, condition (12) does not hold for simple cases, as equal order velocity-pressure interpolation (in particular, with $k_{\mathrm{v}}=k_{\mathrm{q}}$ in (7)). The pairs which are stable, that is, accomplish the discrete inf-sup condition (12), are called *div-stable* in the terminology of [13]. We refer to [16] for a different approach to the inf-sup condition at the fully discrete level.

An alternative to use elements satisfying the inf-sup condition is to use the so called *stabilization techniques* (see [30] for an overview of these methods). The essential idea is to modify the discrete variational form so that pressure stability can be obtained without relying on (12). These type of methods can also be used to avoid the numerical instabilities arising in convection dominated flows, although our main focus here will be their use as a mean to obtain stable pressures. We will introduce them when required.

The space discretization of the evolutionary system (10) consists of finding $\boldsymbol{u}_h\in(\mathcal{V}_{h,0})_t$ and $p_h\in(\mathcal{Q}_h)_t$ such that:

$$(\partial_t\boldsymbol{u}_h,\boldsymbol{v}_h)+a(\boldsymbol{u}_h,\boldsymbol{v}_h)+b(\boldsymbol{v}_h,p_h)+c(\boldsymbol{u}_h,\boldsymbol{u}_h,\boldsymbol{v}_h)$$
$$=\langle\boldsymbol{f},\boldsymbol{v}_h\rangle, \tag{13a}$$
$$b(\boldsymbol{u}_h,q_h)=0, \tag{13b}$$

for all $\boldsymbol{v}_h\in\mathcal{V}_{h,0}$ and $q_h\in\mathcal{Q}_h$, being $(\mathcal{V}_{h,0})_t\equiv L^2(0,T;\mathcal{V}_h)$ and $(\mathcal{Q}_h)_t\equiv\mathcal{D}(0,T;\mathcal{Q}_h)$. Existence and uniqueness of the semi-discrete system (13) are known.

2.4 Time Discretization

Let us introduce some notation that we will use throughout the work. Consider a uniform partition of the time interval of size δt, and let us denote by f^n the approximation of a time dependent function f at time level $t^n=n\delta t$. For a parameter $\theta\in[0,1]$, we will denote

$$f^{n+\theta}=\theta f^{n+1}+(1-\theta)f^n,$$
$$\delta f^{n+1}\equiv\delta^{(1)}f^{n+1}=f^{n+1}-f^n,$$
$$\delta^{(i+1)}f^{n+1}=\delta^{(i)}f^{n+1}-\delta^{(i)}f^n,\quad i=1,2,3,\ldots,$$
$$\delta_t f^{n+1}=\frac{\delta f^{n+1}}{\delta t}.$$

The discrete operators $\delta^{(i+1)}$ are centered. We will use the backward difference operators

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

where, for $k=1,2,3$ the coefficients γ_k and α_k^i correspond to

$$\delta_1 f^{n+1}=\delta f^{n+1}=f^{n+1}-f^n,$$
$$\delta_2 f^{n+1}=\frac{3}{2}\left(f^{n+1}-\frac{4}{3}f^n+\frac{1}{3}f^{n-1}\right),$$
$$\delta_3 f^{n+1}=\frac{11}{6}\left(f^{n+1}-\frac{18}{11}f^n+\frac{9}{11}f^{n-1}-\frac{2}{11}f^{n-2}\right).$$

We will also need the backward extrapolation operators

$$\widetilde{f}_i^{n+1}=f^{n+1}-\delta^{(i)}f^{n+1}=f^{n+1}+\mathcal{O}(\delta t^i), \tag{14}$$

which for $i = 1, 2, 3$ are

$$\tilde{f}_1^{n+1} = f^n,$$
$$\tilde{f}_2^{n+1} = 2f^n - f^{n-1},$$
$$\tilde{f}_3^{n+1} = 3f^n - 3f^{n-1} + f^{n-2}.$$

For the time integration of problem (13) we consider two sorts of finite difference approximations. The first is the generalized trapezoidal rule, which consists of solving the following problem: from known $\boldsymbol{u}^n$, find $\boldsymbol{u}^{n+1} \in \mathcal{V}_0$ and $p^{n+1} \in \mathcal{Q}$ such that

$$\left(\frac{\delta_1 \boldsymbol{u}^{n+1}}{\delta t}, \boldsymbol{v}\right) + c(\boldsymbol{u}^{n+\theta}, \boldsymbol{u}^{n+\theta}, \boldsymbol{v}) + a(\boldsymbol{u}^{n+\theta}, \boldsymbol{v}) + b(p^{n+\theta}, \boldsymbol{v}) = \langle \bar{\boldsymbol{f}}^{n+\theta}, \boldsymbol{v}\rangle, \tag{15a}$$

$$b(q, \boldsymbol{u}^{n+\theta}) = 0, \tag{15b}$$

for all $(\boldsymbol{v}, q) \in \mathcal{V}_0 \times \mathcal{Q}$. The force term $\bar{\boldsymbol{f}}^{n+\theta}$ in (15a) and below has to be understood as the time average of the force in the interval $[t^n, t^{n+1}]$, even though we use a superscript $n + \theta$ to characterize it. The pressure value computed here has been identified as the pressure evaluated at $t^{n+\theta}$, although this is irrelevant for the velocity approximation. The values of interest of θ are $\theta = 1/2$, corresponding to the second order Crank-Nicolson scheme, and $\theta = 1$, which corresponds to the backward Euler method.

Backward differencing (BDF) time integration schemes will also be considered for the methods proposed hereafter. The first order one (BDF1) coincides with the backward Euler method. BDF1 and the second order scheme BDF2 are $\mathcal{A}$-stable. Higher order methods do not keep this interesting property anymore, a limitation known as the *second Dahlquist barrier*. Nevertheless, BDF3 holds a less demanding $\mathcal{A}(\alpha)$-stability, with $\alpha = 86°$, that makes this method appropriate. See [69] for a complete exposition of BDF methods and their stability properties.

For the second order scheme BDF2, $\boldsymbol{u}^1$ can be computed using the backward Euler method, whereas for $n \geq 1$ the unknowns $\boldsymbol{u}^{n+1} \in \mathcal{V}_0$ and $p^{n+1} \in \mathcal{Q}$ are found by solving the problem

$$\left(\frac{\delta_2 \boldsymbol{u}^{n+1}}{\delta t}, \boldsymbol{v}\right) + c(\boldsymbol{u}^{n+1}, \boldsymbol{u}^{n+1}, \boldsymbol{v}) + a(\boldsymbol{u}^{n+1}, \boldsymbol{v}) + b(p^{n+1}, \boldsymbol{v}) = \langle \bar{\boldsymbol{f}}^{n+1}, \boldsymbol{v}\rangle, \tag{16a}$$

$$b(q, \boldsymbol{u}^{n+1}) = 0, \tag{16b}$$

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

The BDF3, which requires also $\boldsymbol{u}^2$ for being initialized, has been analyzed in [8] for the time discretization of the incompressible Navier-Stokes problem.

For a k-th order BDF method, the unknowns $\boldsymbol{u}^{n+1} \in \mathcal{V}_0$ and $p^{n+1} \in \mathcal{Q}$, with $n + 1 \geq k$, are found by solving a problem analogous to (16), obtained by replacing $\delta_2 \boldsymbol{u}^{n+1}$ by $\delta_k \boldsymbol{u}^{n+1}$.

2.5 The Algebraic System

We assume that $\boldsymbol{u}_h^n$ and p_h^n are constructed using the standard finite element interpolation from the nodal values. We introduce the Lagrange basis $\{\boldsymbol{\phi}_{i,j}\}_{i\in\mathcal{N}_u, j=1,\dots,d} \oplus \{\boldsymbol{\phi}_{i,j}\}_{i\in\mathcal{N}_d, j=1,\dots,d}$ and $\{\pi_i\}_{i\in\mathcal{N}_p}$ associated to $\mathcal{V}_h$ and $\mathcal{Q}_h$, respectively. Here, $\mathcal{N}_u$ and $\mathcal{N}_d$ denote the set of free and fixed velocity nodes and $\boldsymbol{\phi}_{i,j}$ is the vector that contains the standard shape function ϕ_i at position j and the rest of components are zero. The set of pressure nodes are denoted by $\mathcal{N}_p$. We aim at writing the system resulting from the fully discretized Navier-Stokes problem. We start by writing the finite element approximation of the unknowns:

$$\boldsymbol{u}_h^{n+1}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}_u} \phi_i(\boldsymbol{x})(\mathbf{U}^{n+1})_i + \sum_{j\in\mathcal{N}_d} \phi_j(\boldsymbol{x})(\mathbf{U}_d^{n+1})_j,$$
$$p_h^{n+1}(\boldsymbol{x}) = \sum_{k\in\mathcal{N}_p} \pi_k(\boldsymbol{x})(\mathbf{P}^{n+1})_k,$$

where $\mathbf{U}^{n+1}$ and $\mathbf{P}^{n+1}$ are the arrays of nodal values for the velocity and the pressure, respectively. $\mathbf{U}_d^{n+1}$ is the array of fixed nodal values for the velocity, which is a datum. The symbol $(\cdot)_i$ denotes the sub-array with the values of node i.

In order to write the fully discretized coupled problem for a given time value t^{n+1}, we need to define a set of matrices. Superscripts a and b will refer to the nodes and subscripts i and j to the spatial dimensions, going from 1 to d. We define:

$$K_{ij}^{ab}(\mathbf{U}) := a(\boldsymbol{\phi}_{a,i}, \boldsymbol{\phi}_{b,j}) + c(\boldsymbol{u}_h, \boldsymbol{\phi}_{a,i}, \boldsymbol{\phi}_{b,j}), \quad a, b \in \mathcal{N}_u,$$
$$M_{ij}^{ab} := \left(\boldsymbol{\phi}_{a,i}, \boldsymbol{\phi}_{b,j}\right), \quad a, b \in \mathcal{N}_u,$$
$$G_i^{ab} := b(\pi_b, \boldsymbol{\phi}_{a,i}), \quad a \in \mathcal{N}_u,\ b \in \mathcal{N}_p.$$

From these terms we define matrices K, M and G whose components are obtained by grouping together spatial and nodal indexes. We also define matrix $D := -G^t$.

Let us start considering the trapezoidal rule. In this case, $\mathbf{U}^{n+1}$ and $\mathbf{P}^{n+1}$ are solution of the nonlinear algebraic system

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+\theta})\mathbf{U}^{n+\theta} + G\mathbf{P}^{n+\theta} = \mathbf{f}^{n+\theta}, \tag{17a}$$

$$D\mathbf{U}^{n+\theta} = \mathbf{g}^{n+\theta}, \tag{17b}$$

where the arrays $\mathbf{f}^{n+\theta}$ and $\mathbf{g}^{n+\theta}$ stand for the arrays that incorporate body forces and Dirichlet boundary conditions.

For homogeneous boundary conditions $\mathbf{g}^{n+\theta} = \mathbf{0}$. If the skew-symmetric form of the convective term is used, the convective contribution to matrix $K(\mathbf{U}^{n+\theta})\mathbf{U}^{n+\theta}$ is skew-symmetric. Let us introduce the matrices

$$C(\cdot) := \frac{1}{\delta t}M + K(\cdot),$$

and

$$C_\theta(\cdot) := \frac{1}{\theta\delta t}M + K(\cdot).$$

At a given time value t^{n+1}, (17) can be written in matrix form as:

$$A\mathbf{X}^{n+\theta} = \mathbf{b}^{n+\theta}, \tag{18}$$

where

$$A = \begin{bmatrix} C_\theta(\mathbf{U}^{n+\theta}) & G \\ D & 0 \end{bmatrix}, \qquad \mathbf{X}^{n+\theta} = \begin{bmatrix} \mathbf{U}^{n+\theta} \\ \mathbf{P}^{n+\theta} \end{bmatrix},$$
$$\mathbf{b}^{n+\theta} = \begin{bmatrix} \mathbf{f}^{n+\theta} - \delta t^{-1}M\mathbf{U}^n \\ \mathbf{g}^{n+\theta} \end{bmatrix}. \tag{19}$$

The force term $\mathbf{b}^{n+\theta}$ accounts now for body forces, boundary conditions and time integration terms.

When considering the semi-implicit treatment of the convective term, we will refer to the case is which K, and therefore C, are evaluated with $\mathbf{U}^n$. The scheme in this case will be at most first order accurate, regardless of the value of θ.

For the BDF2 time integration scheme the resulting algebraic system is analogous to (17), simply replacing $\delta_t\mathbf{U}^{n+1}$ by $\delta t^{-1}\delta_2\mathbf{U}^{n+1}$ and evaluating the rest of the terms at $n+1$ instead of $n+\theta$.

3 The Continuous Approach to Pressure Segregation Methods

Pressure segregation methods were originally motivated for the continuous problem in space and discrete in time. This section is a state-of-the-art review of the continuous approach to these methods. For simplicity, we shall assume homogeneous Dirichlet conditions (2) in the following exposition.

3.1 Some Pressure Segregation Algorithms

The pressure segregation methods were originated independently by Chorin (see [24–27]) and Temam (see [92–95]). The original methods are known as *projection methods* because they are based on the orthogonal decomposition $\boldsymbol{L}^2(\Omega) = \boldsymbol{J}_0 \oplus \boldsymbol{J}_0^\perp$ (see (5) and (6)) which derives from a theorem due to Ladyzhenskaya (see [72]). This theorem is based on the classical Helmholtz decomposition of a vector field into the sum of a solenoidal field and a gradient of a scalar function and a more general theorem proved by De Rham (see [42]).

Hence, in the projection method an intermediate velocity obtained from the momentum equation without the pressure term is decomposed into a solenoidal field, the velocity, and the gradient of a scalar field, the pressure. We present this method in its semi-discrete version in time. Moreover, for the sake of conciseness we will restrict the bulk of this subsection to the backward Euler scheme and homogeneous Dirichlet conditions, commenting later about higher order methods.

This basic procedure involves two basic steps. The first step consists of finding an intermediate velocity $\tilde{\boldsymbol{u}}^{n+1}$ such that:

$$\frac{\tilde{\boldsymbol{u}}^{n+1} - \boldsymbol{u}^n}{\delta t} - \nu\Delta\tilde{\boldsymbol{u}}^{n+1} + \tilde{\boldsymbol{u}}^{n+1}\cdot\nabla\tilde{\boldsymbol{u}}^{n+1} = \boldsymbol{f}^{n+1}, \tag{20a}$$

$$\tilde{\boldsymbol{u}}^{n+1}|_\Gamma = \mathbf{0}, \tag{20b}$$

that does not satisfy the incompressibility condition. In (20a) the convective term is treated implicitly. Alternatively, semi-implicit and explicit approaches can be used. Note that the full Dirichlet boundary conditions are imposed on $\tilde{\boldsymbol{u}}^{n+1}$. The weak form of this equation is required for the numerical approximation. Its weak formulation consists of finding a $\tilde{\boldsymbol{u}}^{n+1} \in \mathcal{V}_0$ such that

$$\left(\frac{\tilde{\boldsymbol{u}}^{n+1} - \boldsymbol{u}^n}{\delta t}, \boldsymbol{v}\right) + a(\tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{v}) + c(\tilde{\boldsymbol{u}}^{n+1}, \tilde{\boldsymbol{u}}^{n+1}, \boldsymbol{v})$$
$$= \langle \boldsymbol{f}^{n+1}, \boldsymbol{v}\rangle, \tag{21}$$

for all $\boldsymbol{v} \in \mathcal{V}_0$. Temam in [96] proposed the use of the skew-symmetric form of the convective term (see (21)). This form is usually assumed in order to obtain stability bounds for the solution of implicit projection methods.

The second step of this method consists of finding an end-of-step velocity $\boldsymbol{u}^{n+1} \in \boldsymbol{J}_0$ and a pressure $p^{n+1} \in \mathcal{Q}$ such that

$$\frac{\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}^{n+1}}{\delta t} + \nabla p^{n+1} = \mathbf{0}, \tag{22a}$$

$$\nabla\cdot\boldsymbol{u}^{n+1} = 0, \tag{22b}$$

$$\boldsymbol{n}\cdot\boldsymbol{u}^{n+1}|_\Gamma = 0, \tag{22c}$$

which is equivalent to find the projection of $\tilde{\boldsymbol{u}}^{n+1}$ onto the space $\boldsymbol{J}_0$ (see (4)),

$$\boldsymbol{u}^{n+1} = P_{\boldsymbol{J}_0}(\tilde{\boldsymbol{u}}^{n+1}).$$

This is the reason why these methods are popularly known as *projection methods*. We point out that at this step only the normal component of the velocity is prescribed, and therefore the end-of-step velocity $\boldsymbol{u}^{n+1}$ does not satisfy the proper boundary conditions. Hence, we infer easily from (20b) and (22c) that the pressure is satisfying on the Dirichlet boundary an *artificial Neumann condition*

$$\boldsymbol{n} \cdot \nabla p^{n+1}|_{\Gamma} = 0. \tag{23}$$

This is one of the most controversial points of the classical projection method. There has been much speculation about whether the pressure p^{n+1} is a good approximation to the exact pressure $p(t^{n+1})$ (see [97]). It is conjectured by Rannacher in [82] and by Gresho in [55] that the non-physical boundary condition lives only in a narrow boundary layer of width $\mathcal{O}(\sqrt{\nu \delta t})$.

In 1996 Timmermans, Minev and Van de Vosse proposed in [98] a modified version of the projection method that leads to improved pressure approximations. Once solved the classical projection method, the modification consists of a correction of the pressure p^{n+1} as follows:

$$\hat{p}^{n+1} = p^{n+1} - \nu \nabla \cdot \tilde{\boldsymbol{u}}^{n+1}.$$

It is evident that this update does not cause a severe additional numerical effort. Furthermore, it has been observed in [63] that with this method, unlike (23), the following pressure boundary condition is imposed

$$\boldsymbol{n} \cdot \nabla p^{n+1}|_{\Gamma} = \boldsymbol{n} \cdot (\boldsymbol{f}(t^{n+1}) - \nu \nabla \times \nabla \times \boldsymbol{u}^{n+1})|_{\Gamma},$$

which is consistent for the Stokes problem. Due to the fact that the operator $\nabla \times \nabla \times$ plays a key role, the pressure-correction methods using this improvement are called *rotational* pressure-correction methods, as proposed in [63]. A similar *enhanced* boundary condition for the pressure was previously proposed in [76] and [70].

The main interest of the projection method is the possible uncoupling of the pressure from the velocity in its numerical approximation. It is achieved by taking the divergence of (22a), obtaining a Pressure Poisson Equation (PPE from here onwards) for the pressure,

$$\Delta p^{n+1} = \frac{1}{\delta t} \nabla \cdot \tilde{\boldsymbol{u}}^{n+1}, \tag{24a}$$

$$\boldsymbol{n} \cdot \nabla p^{n+1}|_{\Gamma} = 0. \tag{24b}$$

Once the pressure p^{n+1} is calculated, the end-of-step velocity is recovered from (22a).

Another interesting issue of this method is its inherent stability. It can be seen by numerical experimentation that equal order velocity and pressure space approximations lead to stabilized systems, so that the discrete inf-sup condition (12) can be violated. However, the stability is decreasing with the time step size. This effect has been hidden by the convergence analysis, where it is required a time step size *small enough*. A complete study of the machinery that makes the fully discretized projection method stable was developed in [32] and extended in [35].

The improvement of the error estimates for the pressure taking the *rotational form* has been studied in [78] and [61].

The long term behavior, dissipativity and unconditional stability of the projection scheme was analyzed in [90] and [1] by Simo and Armero. They also showed that this method exhibits an absorbing set, as does the continuum flow generated by the Navier-Stokes equations.

A modified scheme was proposed by Shen in [85]. It consists of introducing in the momentum equation an approximation of the pressure, more specifically, the term ∇p^n. This method is usually called *incremental projection method*. The original scheme proposed by Shen was composed by a first step where the intermediate velocity $\tilde{\boldsymbol{u}}^{n+1}$ is obtained,

$$\frac{1}{\delta t}(\tilde{\boldsymbol{u}}^{n+1} - \tilde{\boldsymbol{u}}^{n}) - \nu \Delta \tilde{\boldsymbol{u}}^{n+1} + \boldsymbol{u}^{n} \cdot \nabla \tilde{\boldsymbol{u}}^{n+1} + \nabla p^{n} = \boldsymbol{f}^{n+1}, \tag{25a}$$

$$\tilde{\boldsymbol{u}}^{n+1}|_{\Gamma} = \boldsymbol{0}, \tag{25b}$$

and a second step for the end-of-step velocity $\boldsymbol{u}^{n+1}$ and pressure p^{n+1},

$$\frac{1}{\delta t}(\boldsymbol{u}^{n+1} - \tilde{\boldsymbol{u}}^{n+1}) + \nabla(p^{n+1} - p^{n}) = \boldsymbol{0}, \tag{26a}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0, \tag{26b}$$

$$\boldsymbol{n} \cdot \boldsymbol{u}^{n+1}|_{\Gamma} = 0. \tag{26c}$$

Alternatively, fully implicit or fully explicit convective terms can be used. As before, the pressure can be obtained from a Poisson equation. However, the variable obtained in this case is a *correction* of the pressure, δp^{n+1}, instead of the total pressure p^{n+1},

$$\Delta(p^{n+1} - p^{n}) = \frac{1}{\delta t} \nabla \cdot \tilde{\boldsymbol{u}}^{n+1},$$

$$\boldsymbol{n} \cdot \nabla(p^{n+1} - p^{n})|_{\Gamma} = 0.$$

This scheme is also called *pressure-correction* method. This is the nomenclature we choose in this work. We note that the classical projection method can also be considered as a pressure correction method. The only difference is that in the classical method a zero order approximation of the pressure is used in the momentum equation whereas in (25a) a first order approximation is chosen.

So far, the approach adopted is first the discretization in time, then the splitting. We have a semi-discrete problem which can be approximated after being discretized in space. Hence, the controversy arises on which boundary conditions must be imposed in order to have well-posed problems at the continuous level.

At the fully discrete level, the PPE introduces a perturbation term in the continuity equation which consists of a difference between two different Laplacian approximations. As it is showed in [32], this term is the origin of the inherent stability of the discretized form of the projection method. The characterization of the *stabilizer* motivated the Orthogonal Subgrid Scale (OSS) stabilization methods proposed in [33].

The pressure-correction methods showed above are first order schemes in the time step size. Several alternatives have been suggested to achieve second order methods. The first one was proposed by Kim and Moin in [71]. Therein they introduce a method where the convective term is evaluated by an explicit second order Adams-Bashfort method and the viscous term by a second order implicit Crank-Nicholson method. The main novelty of this scheme are the boundary condition imposed on the intermediate velocity.

Van Kan proposed in [100] a second order accurate pressure correction method. This method is equivalent to Shen's method but replacing the backward Euler time integration scheme by the second order Crank-Nicholson scheme. It is proved in [100] that the velocity and pressure solution of this method differ from the solution of the monolithic problem by a term of order $\mathcal{O}(\delta t^2)$.

In 1989 Bell Colella and Glaz introduced in [9] the first predictor-corrector method. They proposed a second order iterative scheme which converges to the Crank-Nicholson monolithic system. For every iteration, first the intermediate velocity is obtained, treating the convection term explicitly and the diffusion term implicitly. Then, the end-of-step velocity for the current iteration and pressure are calculated in a coupled way. The convective term is differentiated using a second order Godunov procedure.

Second order accuracy in time can be obtained also using a BDF2 time integration method instead of backward Euler in Shen's scheme (25)–(26).

In different papers second order pressure-correction schemes have been treated analytically. E and Liu studied in [43] Kim and Moin's scheme. Shen analyzed the Van Kan method in [85] and [89], obtaining optimal error estimates assuming a sufficiently accurate initial data for the velocity field and pressure field and some regularity assumptions. In [78] optimal error estimates are recovered with weaker regularity requirements.

The situation is much less studied for third order methods. Apart from using a third order time integration scheme (for example BDF3), the pressure has to be extrapolated to second order in the momentum equation, that is to say, using notation (14), the pressure gradient in the momentum equation should be approximated by $\nabla \widetilde{p}_2^{n+1}$.

The first author who proposed a third order pressure-correction method was Gresho in [55]. However, numerical experiments showed that this method is unstable in time. In [87] Shen tried to explain why the solutions obtained with third order pressure-correction methods cannot be bounded uniformly in time. In fact, he obtained that *all the pressure-correction methods of order higher than two are unstable*. However, according to the authors' analysis, the explanation of Shen is not appropriate because some terms are neglected. In fact Shen pointed out that the *equivalent* continuous system he analyzed was obtained *by intuition*. The reason why higher order pressure-correction methods are unstable remains an open question that has not been fully explained yet.

Very recently, Guermond and Shen have introduced in [62] a new kind of pressure segregation methods, coined velocity-correction methods because it is the velocity instead of the pressure the extrapolated unknown. We will also discuss these methods in Sect. 6.

3.2 Numerical Analysis

The projection method has been studied analytically in order to prove convergence and obtain error estimates. For periodic boundary conditions and a centered finite difference space approximation, Chorin proved in [26] that the convergence was of first order in the time step size δt, provided $\delta t = \mathcal{O}(h^2)$, h being the diameter of the space discretization. A proof of convergence of the semi-discrete (space continuous) projection method to a continuous solution $\boldsymbol{u}(t)$ was given by Temam (see [94]) in a general setting.

The pioneering work of Shen [86] analyzed the semi-implicit version of the semi-discrete projection method, taking the convective term as $\boldsymbol{u}^n \cdot \nabla \tilde{\boldsymbol{u}}^{n+1}$. He obtained the first error estimates for the velocity, but in a weak norm, $L^2(0,T;\boldsymbol{L}^2(\Omega))$. The proof of the error estimates in this work is plagued by a mistake when using dual norms which was originally observed by Guermond in [57]. In [88] these mistakes were corrected. Furthermore, for obtaining the pressure estimates, $\partial_t p \in L^2(0,T;H^1(\Omega)/\mathbb{R})$ has to be assumed. This requirement is not appropriate, because of the lack of compatibility with the given data at time $t \to 0$ (see [64]).

Rannacher proposed an alternative version of the projection method in [82]. In this new format the end-of-step velocity disappears, and the problem can be understood as a penalized method with a lag in the evaluation of the pressure in the momentum equation. The method reads

$$\frac{1}{\delta t}(\tilde{\boldsymbol{u}}^{n+1} - \tilde{\boldsymbol{u}}^n) - \nu \Delta \tilde{\boldsymbol{u}}^{n+1} + \tilde{\boldsymbol{u}}^n \cdot \nabla \tilde{\boldsymbol{u}}^{n+1} + \nabla p^n = \boldsymbol{f}^{n+1},$$

$$\nabla \cdot \tilde{\boldsymbol{u}}^{n+1} + \delta t \Delta p^{n+1} = 0.$$

Based on this approach, Prohl obtained sharp error estimates for the semi-discrete in time projection method (see [78]). For instance, first order error estimates for the velocity were obtained with a stronger norm, $L^\infty(0,T;\boldsymbol{L}^2(\Omega))$.

The improvement of the error estimates for the pressure taking the *rotational form* of the Navier-Stokes equations has been studied in [78] and [61].

Shen obtained improved error estimates for method (25) in [85]. Both the intermediate $\tilde{\boldsymbol{u}}^{n+1}$ and end-of-step $\boldsymbol{u}^{n+1}$ velocities are first order approximations to the continuous velocities in $L^\infty(0,T;\boldsymbol{L}^2(\Omega))$.

Error estimates for a fully discrete pressure-correction method were introduced in [59, 60]. The method analyzed therein is the incremental scheme proposed by Shen. Furthermore, the pressure is not calculated from a Poisson equation, but from a Stokes problem. This analysis requires velocity and pressure interpolations satisfying the inf-sup condition for obtaining error estimates of first order in the time step size and of optimal order in the mesh size.

In [5] the authors have obtained convergence results for the fully discrete projection method in two different situations. The first analysis applies to div-stable elements that satisfy a discrete inf-sup condition. The second part of this work analyzes the fully discrete projection method using a Poisson equation for the pressure. But in this case the velocity-pressure interpolations do not need to satisfy the inf-sup condition. Convergence results are obtained only relying on the inherent stability of the algorithm. In both cases error estimates are obtained comparing these schemes with an auxiliary problem semi-discretized in time for which optimal convergence results are known. The analysis of the auxiliary problem follows the analysis in [78], due to the similarity of the auxiliary problem with the total projection method studied therein. The analysis in [5] is the first convergence analysis for the fully discrete first order projection method in the original version proposed by Chorin and Temam.

4 The Discrete Approach

4.1 General Comments

A different approach to the projection method in a discrete setting (in space and time) was initially proposed by Perot in [77] in the framework of finite volumes. He suggested a pressure correction method for the solution of the discrete linearized system, obtained from the Navier-Stokes equations, based on an *incomplete LU block factorization*. In [81] this idea was generalized to different methods. The convergence analysis of a first order algebraic pressure-correction method can be found in [5]. The main differences with respect to the continuous approach are:

- Dirichlet boundary conditions: At the algebraic level we can skip the discussion about the correct boundary conditions to be imposed. In fact, at the discrete level, the end-of-step velocity satisfies the good boundary conditions. Even more important, no boundary conditions have to be imposed for the pressure. Even though the artificial boundary conditions are not explicitly imposed, the algebraic approach leads to the same wrong pressure boundary conditions, although in a weak sense. This last point has been initially misunderstood; some works claimed that the discrete approach was not affected by this problem (see [79]). Numerical experiments comparing the pressure error layers in both cases have been carried out in [6].
- Neumann boundary conditions: The discrete approach is clearly superior to the continuous approach for problems with open boundaries. At the discrete level, Neumann boundary conditions are taken into account in the force term as for the monolithic system. However, for the continuous approach, when using the pressure Poisson equation, Dirichlet pressure conditions for the pressure must be enforced on the open boundaries, introducing an extra perturbation to the problem.
- Discrete Laplacian: The discrete approach motivates a discrete pressure Poisson equation that differs from the one obtained at the continuous level. A discrete Laplacian can also be motivated at the continuous level if, instead of using the pressure Poisson equation, a Darcy problem that couples pressure and end-of-step velocity is used. The use of the discrete Laplacian is more accurate than the continuous Laplacian, because no perturbation is introduced in the mass conservation equation (see [32, 35]).
- Relationship with preconditioners: At the algebraic level, pressure segregation methods motivate effective preconditioners for the monolithic Navier-Stokes system. In fact, these methods can be understood as preconditioners for the pressure Schur complement (see [99]).
- More accurate splitting procedures: At the discrete level, more accurate pressure segregation algorithms have been designed by using enhanced inexact *LU* block factorizations. Third and fourth order splitting methods that are claimed to be unconditionally stable can be found in [50–52]. Also the Yosida method has been designed at the algebraic level by modifying the end-of-step computation (see [80, 81]). The continuous counterpart of these methods has not been explored.
- Stabilization techniques: As it has been mentioned earlier, a possible way to avoid the need for satisfying the inf-sup condition (12) is to use stabilization techniques, as those described later on. At the discrete level, it is easy to infer how to deal with the stabilization terms in the splitting process, particularly those terms related to the pressure stabilization. Unfortunately, things are not so clear at the continuous level (in space), where the stabilization terms do not have any sense.

Before proceeding to the analysis and description of particular algebraic pressure-correction and velocity-correction schemes, we will describe next two ideas that may be used to motivate the algebraic point of view to pressure segregation methods. To present these ideas in the simplest setting, we will consider the backward Euler scheme for the time integration and a semi-implicit treatment of the nonlinearity, leaving for the next sections the analysis of other schemes. Likewise, we will consider homogeneous Dirichlet conditions ($\mathbf{g}=\mathbf{0}$) and constant-in-time forcing terms.

4.2 A First Algebraic Point of View: Extrapolation

Let us consider the algebraic system (18) with $\theta=1$. At time step $n+1$ we may write it as

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1}-\mathbf{U}^n)+K\mathbf{U}^{n+1}+G\mathbf{P}^{n+1}=\mathbf{f}, \tag{27a}$$

$$\begin{aligned}&DM^{-1}K\mathbf{U}^{n+1}+DM^{-1}G\mathbf{P}^{n+1}\\&\quad=DM^{-1}\mathbf{f}+\frac{1}{\delta t}D\mathbf{U}^n.\end{aligned} \tag{27b}$$

The reason for writing the discrete version of the PPE (27b) is to be able to obtain an equation for the pressure if the velocity is known. Note also that we have kept the term $D\mathbf{U}^n$, since after the approximations to be made it will not be exactly zero.

The idea now is *to extrapolate either the pressure in* (27a) *or the velocity in* (27b), *solving for the remaining variable and then correcting both variables.* Let us describe the two options.

4.2.1 Pressure Extrapolation

In this case, *the pressure is extrapolated in* (27a). This allows to compute an approximate velocity which can be used to compute the pressure form (27b). The velocity is corrected after the pressure is computed. Using a pressure extrapolation of order q and the notation of (14), the three steps are

$$M\frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1}-\mathbf{U}^n)+K\widetilde{\mathbf{U}}^{n+1}=\mathbf{f}-G\widetilde{\mathbf{P}}_q^{n+1}, \tag{28a}$$

$$DM^{-1}G\mathbf{P}^{n+1}=-DM^{-1}K\widetilde{\mathbf{U}}^{n+1}+DM^{-1}\mathbf{f}+\frac{1}{\delta t}D\mathbf{U}^n, \tag{28b}$$

$$\begin{aligned}&M\frac{1}{\delta t}(\mathbf{U}^{n+1}-\widetilde{\mathbf{U}}^{n+1})+\alpha K(\mathbf{U}^{n+1}-\widetilde{\mathbf{U}}^{n+1})\\&\quad+G(\mathbf{P}^{n+1}-\widetilde{\mathbf{P}}_q^{n+1})=\mathbf{0}.\end{aligned} \tag{28c}$$

Note that $\widetilde{\mathbf{U}}^{n+1}$ is just an intermediate variable computed with the extrapolated pressure $\widetilde{\mathbf{P}}_q^{n+1}$. The parameter $\alpha\in[0,1]$ in (28c) accounts for a possible correction of the error of $K\widetilde{\mathbf{U}}^{n+1}$ compared to $K\mathbf{U}^{n+1}$, which is eliminated if $\alpha=1$, as can be seen if (28a) and (28c) are added up (this approach was used in [11] only for the viscous part of $K\mathbf{U}^{n+1}$). The classical choice corresponds to $\alpha=0$, whereas we shall see later on that $\alpha=1$ is related to the so called Yosida scheme. Let us finally mention that the right-hand-side of (28b) is not written in the usual way, which is $\delta t^{-1}D\widetilde{\mathbf{U}}^{n+1}+DM^{-1}G\widetilde{\mathbf{P}}_q^{n+1}$. This term is easier to compute and arises naturally in simpler motivations of pressure-corrections schemes, as described in the next section. However, our purpose here is to highlight the connection with velocity-correction methods introduced next.

4.2.2 Velocity Extrapolation

In this case, *the velocity is extrapolated in* (27b). This allows us to compute and intermediate pressure with which the velocity can be computed. The pressure could be finally corrected, leading to the scheme:

$$DM^{-1}G\widetilde{\mathbf{P}}^{n+1}=-DM^{-1}K\widetilde{\mathbf{U}}_q^{n+1}+DM^{-1}\mathbf{f}+\frac{1}{\delta t}D\mathbf{U}^n, \tag{29a}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1}-\mathbf{U}^n)+K\mathbf{U}^{n+1}=\mathbf{f}-G\widetilde{\mathbf{P}}^{n+1}, \tag{29b}$$

$$DM^{-1}G(\mathbf{P}^{n+1}-\widetilde{\mathbf{P}}^{n+1})=-DM^{-1}K(\mathbf{U}^{n+1}-\widetilde{\mathbf{U}}_q^{n+1}). \tag{29c}$$

The parallelism between system (29) and (28) is obvious. Note also that the velocity does not depend on $\mathbf{P}^{n+1}$ and only $\widetilde{\mathbf{P}}^{n+1}$ is needed to go to the next time step. Therefore, (29c) is in fact *not needed*, and therefore not solved in practice.

4.3 A Second Algebraic Point of View: Inexact Factorization

We will see in the following that it is possible to motivate both pressure-correction and velocity-correction schemes from a completely different point of view. Our starting point now will be (17) and its compact form (18) instead of (27).

4.3.1 Pressure-Correction Schemes as Inexact Factorization

Let us consider the *exact LU* factorization of the system matrix of (19)

$$A=\begin{bmatrix}C & 0\\ D & -DC^{-1}G\end{bmatrix}\begin{bmatrix}I & C^{-1}G\\ 0 & I\end{bmatrix}, \tag{30}$$

where I is the identity matrix associated to free velocity nodes. System (18) (with $\theta=1$)

$$A\mathbf{X}^{n+1}=LU\mathbf{X}^{n+1}=\mathbf{b}^{n+1},$$

can be solved in two steps as

$$L\widetilde{\mathbf{X}}^{n+1} = \mathbf{b}^{n+1},$$
$$U\mathbf{X}^{n+1} = \widetilde{\mathbf{X}}^{n+1},$$

where $\widetilde{\mathbf{X}}^{n+1}$ is an intermediate variable. In velocity-pressure components, the first step can be written as

$$C\widetilde{\mathbf{U}}^{n+1} = \mathbf{f} - \frac{1}{\delta t} M\mathbf{U}^n,$$
$$D\widetilde{\mathbf{U}}^{n+1} - DC^{-1}G\widetilde{\mathbf{P}}^{n+1} = \mathbf{0},$$

which is followed by the second step

$$\mathbf{U}^{n+1} + C^{-1}G\mathbf{P}^{n+1} = \widetilde{\mathbf{U}}^{n+1},$$
$$\mathbf{P}^{n+1} = \widetilde{\mathbf{P}}^{n+1}.$$

The computational cost of this scheme can be substantially reduced by using a *cheap* approximation to C^{-1}. One possibility is to use the first order approximation (in time) of the inverse matrix (see [77, 81])

$$C^{-1} = \left(M\frac{1}{\delta t} + K \right)^{-1} \simeq \delta t M^{-1}. \tag{31}$$

Then, by using this approximation in (30) we get the following *inexact LU* block factorization:

$$A_{PC} = \begin{bmatrix} C & 0 \\ D & -\delta t DM^{-1}G \end{bmatrix} \begin{bmatrix} I & \delta t M^{-1}G \\ 0 & I \end{bmatrix}. \tag{32}$$

It is easily checked that this system of equations is equivalent to (28) with $\alpha = 0$ and $q = 0$ (order of pressure extrapolation), in its linearized version. In compact form, this method reads as

$$A_{PC}\mathbf{X}^{n+1} = \mathbf{b}^{n+1}.$$

The incremental form (equivalent to (28) with $q = 1$) can be written as:

$$A_{PC} \begin{bmatrix} \mathbf{U}^{n+1} \\ \delta\mathbf{P}^{n+1} \end{bmatrix} = \mathbf{b}^{n+1} - \begin{bmatrix} G\mathbf{P}^n \\ 0 \end{bmatrix}.$$

Even though the approximation described can be implemented as such, it is sometimes convenient to make a further approximation. Observe that $DM^{-1}G$ represents an approximation to the Laplacian operator. In order to avoid dealing with this matrix, we can approximate

$$DM^{-1}G \approx L, \quad \text{with components } L^{ab} = -(\nabla\pi_a, \nabla\pi_b),$$
$$a, b \in \mathcal{N}_p. \tag{33}$$

Matrix L is the standard approximation to the Laplacian operator. Clearly, this approximation is only possible when continuous pressure interpolations are employed. In [56], the use of approximation (33) is referred to as "approximate projection". It is easily seen that approximation (33) keeps the order of accuracy of the pressure-correction scheme.

Using (33), the approximated LU block factorization reads as

$$A_{PC} = \begin{bmatrix} C & 0 \\ D & -\delta t L \end{bmatrix} \begin{bmatrix} I & \delta t M^{-1}G \\ 0 & I \end{bmatrix}.$$

This interpretation of the algebraic pressure segregation algorithms have motivated new splitting methods based on the idea of getting improved inexact LU block factorizations. Some of these methods are introduced in Sects. 4.3.3 and 4.3.4.

Observe that the error in the discrete operator made by replacing the exact matrix (30) by the approximate matrix (32) is given by

$$E_{PC} = A - A_{PC} = \begin{bmatrix} 0 & -\delta t KM^{-1}G \\ 0 & 0 \end{bmatrix}. \tag{34}$$

4.3.2 Velocity-Correction Schemes as Inexact Factorization

We can also motivate the algebraic velocity-correction methods introduced above as an inexact factorization method. Let us consider a slightly different *exact LU* block factorization of A:

$$A = \begin{bmatrix} I & 0 \\ DC^{-1} & -DC^{-1}G \end{bmatrix} \begin{bmatrix} C & G \\ 0 & I \end{bmatrix}, \tag{35}$$

which can be considered "non-canonical", in the sense that matrix U has non-unit diagonal. Now, we can apply again approximation (31), obtaining the following matrix:

$$A_{VC} = \begin{bmatrix} I & 0 \\ \delta t DM^{-1} & -\delta t DM^{-1}G \end{bmatrix} \begin{bmatrix} C & G \\ 0 & I \end{bmatrix}. \tag{36}$$

Similarly to the pressure-correction scheme, it is easily checked that scheme (29a)–(29b) with $q = 0$ (order of velocity extrapolation) corresponds precisely to solving $A_{VC}\mathbf{X}^{n+1} = \mathbf{b}^{n+1}$.

It can be seen that this method introduces a different perturbation matrix than the pressure-correction scheme. From (35) and (36) it follows that

$$E_{VC} = \begin{bmatrix} 0 & 0 \\ -\delta t DM^{-1}K & 0 \end{bmatrix}. \tag{37}$$

Therefore, this method is *momentum preserving* but perturbs the mass conservation equation, in contraposition to the classical pressure-correction algorithm, which is mass preserving and only perturbs the momentum conservation equation. Again, the Laplacian approximation (33) can be used.

In the case of homogeneous Dirichlet conditions that we are considering, $D = -G^t$. From (34) and (37) it is seen that if matrix K is symmetric (Stokes problem or explicit treatment of the nonlinear term) then $E_{PC} = -E^t_{VC}$. In both cases this error matrix is of order $\mathcal{O}(\delta t)$, corresponding to the order zero extrapolation in either pressures or velocities, with the approach adopted in the previous subsection.

4.3.3 The Yosida Method

An alternative pressure segregation algorithm motivated from the exact LU block factorization (30) has been introduced in [81] and analyzed in [80]. This method has been coined the *Yosida* scheme, because it can be interpreted as a Yosida regularization of the Laplace operator. The difference with the previous pressure-correction algorithms relies on the computation of the end-of-step velocity. The idea is to use approximation (31) only in the pressure Schur complement matrix, computing exactly the U-block. It leads to the following inexact LU block factorization instead of (32):

$$A_Y = \begin{bmatrix} C & 0 \\ D & -\delta t D M^{-1} G \end{bmatrix} \begin{bmatrix} I & C^{-1} G \\ 0 & I \end{bmatrix}.$$

The perturbation introduced by the Yosida method is

$$E_Y = A - A_Y = \begin{bmatrix} 0 & 0 \\ 0 & D(\delta t M^{-1} - C^{-1}) G \end{bmatrix}.$$

Clearly, as the velocity-correction scheme this method is momentum preserving. Moreover, since $C^{-1} = \delta t M^{-1} + \mathcal{O}(\delta t^2)$, it is seen that perturbation E_Y is formally $\mathcal{O}(\delta t^2)$ (in any norm). Therefore, the splitting error due to the Yosida scheme is expected to be of second order.

This method can also be written as a three-step algorithm:

$$M \frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + K \widetilde{\mathbf{U}}^{n+1} = \mathbf{f}, \tag{38a}$$

$$\delta t D M^{-1} G \mathbf{P}^{n+1} = D \widetilde{\mathbf{U}}^{n+1}, \tag{38b}$$

$$M \frac{1}{\delta t}(\mathbf{U}^{n+1} - \widetilde{\mathbf{U}}^{n+1}) + K \mathbf{U}^{n+1} - K \widetilde{\mathbf{U}}^{n+1} + G \mathbf{P}^{n+1} = \mathbf{0}, \tag{38c}$$

which corresponds to (28) with $q = 0$ and $\alpha = 1$.

Remark 1 In [80, 81] the nonlinearity has always been considered in a explicit way. A reasonable implicit treatment of the nonlinearity would be to consider $K(\mathbf{U}^n)\widetilde{\mathbf{U}}^{n+1}$ in (38a) (semi-explicit treatment) and $K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - K(\mathbf{U}^n)\widetilde{\mathbf{U}}^{n+1}$ in (38c) (implicit treatment), with similar extensions to other values of θ of the trapezoidal rule.

Remark 2 The incremental form of the Yosida algorithm is obtained taking $\alpha = 1$ and $q = 1$ in (28). This method can be written as

$$A_Y \begin{bmatrix} \mathbf{U}^{n+1} \\ \delta \mathbf{P}^{n+1} \end{bmatrix} = \mathbf{b}^{n+1} - \begin{bmatrix} G\mathbf{P}^n \\ 0 \end{bmatrix}.$$

This algorithm perturbs the monolithic problem by adding the term

$$D(\delta t M^{-1} - C^{-1}) G \delta \mathbf{P}^{n+1},$$

to the mass conservation equation, which is formally of order $\mathcal{O}(\delta t^3)$. Therefore, the incremental Yosida algorithm combined with the BDF3 time integration scheme leads to an overall third order method. As far as we know, such a method has never been proposed before.

4.3.4 Higher Order Methods

Third and fourth order algebraic pressure segregation algorithms have been designed and analyzed in [50–52, 84]. These methods are again based on incomplete LU block factorization. However, in these methods a different approximate factorization is used:

$$A_{HO} = \begin{bmatrix} C & 0 \\ D & -\delta t D M^{-1} G \end{bmatrix} \begin{bmatrix} I & H^{-1} G R \\ 0 & Q \end{bmatrix}.$$

Matrices Q and R are chosen in order to minimize the perturbation error. Matrix H can take two different values:

- $H = \delta t^{-1} M$ for pressure-correction algorithms (with $q = 0$).
- $H = C$ for the Yosida scheme.

This method is equivalent either to the previous pressure-correction method or to the Yosida method supplemented with an extra correction of the pressure, dictated by Q, and a modification of the end-of-step equation, dictated by R. Thus, we have to distinguish between intermediate and end-of-step pressures. This method involves four steps:

$$M \frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + K \widetilde{\mathbf{U}}^{n+1} = \mathbf{f},$$

$$\delta t D M^{-1} G \widetilde{\mathbf{P}}^{n+1} = D \widetilde{\mathbf{U}}^{n+1},$$

$$Q \mathbf{P}^{n+1} = \widetilde{\mathbf{P}}^{n+1},$$

$$H(\mathbf{U}^{n+1} - \widetilde{\mathbf{U}}^{n+1}) + G R \mathbf{P}^{n+1} = \mathbf{0}.$$

The key ingredient is the expression for matrix Q. Two alternative approaches have been considered:

- Third order methods have been considered in [84], namely a *Chorin-Temam pressure-correction* type scheme and the so called *Yosida-3* scheme, depending on the expression for H. Both methods use the same expression for Q:

  $$Q = \delta t T^{-1} D M^{-1} G$$

with $T=\delta t^2DM^{-1}CM^{-1}G$. These two methods differ in the expression for matrix R. The Chorin-Temam type pressure-correction method takes $R=Q$, whereas Yosida-3 uses $R=I$.
- A fourth order method, *Yosida-4*, has been designed in [51] and analyzed in [50]. We refer to these works for the expression of Q that leads to a fourth order method.

Remark 3 This approach leads to high order pressure segregation algorithms which are stable because the enhanced accuracy is not attained using high order extrapolations for the pressure. The only way to improve the order of accuracy when the splitting is carried out at the continuous level turns out to be higher order pressure predictions. However, using this approach, third and higher order schemes (in time) become unstable.

Remark 4 Unfortunately, these schemes are not useful for real (large) problems, because matrices Q are dense. Krylov methods should be used but an appropriate preconditioner for Q is not available and finding one seems to be a hard task.

5 Algebraic Pressure-Correction Algorithms

In this section we analyze pressure-correction schemes introduced in Sect. 4. We start re-introducing these schemes in the simple setting of either zero or first order pressure extrapolations, leading to first or second splitting errors, respectively, as we will explain in the following. As time integration schemes we will consider both the trapezoidal rule and BDF schemes.

5.1 Formulation of the Algorithms

Let us start considering schemes based on the trapezoidal rule. The fully discrete nonlinear problem (17) is exactly equivalent to

$$M\frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1}-\mathbf{U}^n)+K(\mathbf{U}^{n+\theta})\mathbf{U}^{n+\theta}+\gamma G\mathbf{P}^n=\mathbf{f}^{n+\theta}, \tag{39a}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1}-\widetilde{\mathbf{U}}^{n+1})+G(\mathbf{P}^{n+1}-\gamma\mathbf{P}^n)=\mathbf{0}, \tag{39b}$$

$$D\mathbf{U}^{n+1}=\mathbf{0}, \tag{39c}$$

where $\widetilde{\mathbf{U}}^{n+1}$ is an auxiliary variable and γ is a numerical parameter, whose values of interest are 0 and 1. At this point we can make the essential approximation

$$K(\mathbf{U}^{n+\theta})\mathbf{U}^{n+\theta}\approx K(\widetilde{\mathbf{U}}^{n+\theta})\widetilde{\mathbf{U}}^{n+\theta}, \tag{40}$$

where $\widetilde{\mathbf{U}}^{n+\theta}:=\theta\widetilde{\mathbf{U}}^{n+1}+(1-\theta)\mathbf{U}^n$. Expressing $\mathbf{U}^{n+1}$ in terms of $\widetilde{\mathbf{U}}^{n+1}$ using (39b) and inserting the result in (39c), the set of equations to be solved is

$$M\frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1}-\mathbf{U}^n)+K(\widetilde{\mathbf{U}}^{n+\theta})\widetilde{\mathbf{U}}^{n+\theta}+\gamma G\mathbf{P}^n=\mathbf{f}^{n+\theta}, \tag{41a}$$

$$\delta t\, DM^{-1}G(\mathbf{P}^{n+1}-\gamma\mathbf{P}^n)=D\widetilde{\mathbf{U}}^{n+1}, \tag{41b}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1}-\widetilde{\mathbf{U}}^{n+1})+G(\mathbf{P}^{n+1}-\gamma\mathbf{P}^n)=\mathbf{0}, \tag{41c}$$

which have been ordered according to the sequence of solution, for $\widetilde{\mathbf{U}}^{n+1}$, $\mathbf{P}^{n+1}$ and $\mathbf{U}^{n+1}$. This uncoupling of variables has been made possible by virtue of (40).

We will consider three possibilities depending on the choice of θ and γ. Formally, it is easy to see that the perturbation term introduced by approximation (40) is of order $\mathcal{O}(\delta t)$ when $\gamma=0$, and of order $\mathcal{O}(\delta t^2)$ when $\gamma=1$ (incremental form). Observe from (39b) that $\mathcal{O}(\|\mathbf{U}^{n+1}-\widetilde{\mathbf{U}}^{n+1}\|)$ $=\delta t\mathcal{O}(\|\mathbf{P}^{n+1}-\gamma\mathbf{P}^n\|)$ in any norm $\|\cdot\|$. Thus, we will refer to the case $\gamma=0$ as the case with splitting error of order 1, called SE1 in the following, and the case $\gamma=1$ as the case with splitting error of order 2, called SE2. The three possibilities mentioned are:

- $\theta=1$, $\gamma=0$. Method BDF1-SE1.
- $\theta=1$, $\gamma=1$. Method BDF1-SE2.
- $\theta=1/2$, $\gamma=1$. Method CN-SE2.

Method BDF1-SE2 will obviously be first order, and thus the second order splitting error unnecessary. However, this method is interesting, because reduces the artificial dissipation introduced by the first order splitting error.

So far, we have considered the trapezoidal rule for the time integration. If, instead, we use BDF2 with a second order splitting error, the final algebraic system will be

$$M\frac{1}{2\delta t}(3\widetilde{\mathbf{U}}^{n+1}-4\mathbf{U}^n+\mathbf{U}^{n-1})+K(\widetilde{\mathbf{U}}^{n+1})\widetilde{\mathbf{U}}^{n+1}+G\mathbf{P}^n$$
$$=\mathbf{f}^{n+1},$$

$$\frac{2}{3}\delta t\, DM^{-1}G(\delta\mathbf{P}^{n+1})=D\widetilde{\mathbf{U}}^{n+1},$$

$$M\frac{1}{2\delta t}(3\mathbf{U}^{n+1}-3\widetilde{\mathbf{U}}^{n+1})+G(\delta\mathbf{P}^{n+1})=\mathbf{0}.$$

We will call this method BDF2-SE2.

As mentioned in Sect. 4, it is often convenient from the numerical point of view to make use of approximation (33). Using this, together with (40), the problem to be solved, instead of (41), is:

$$M\frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1}-\mathbf{U}^n)+K(\widetilde{\mathbf{U}}^{n+\theta})\widetilde{\mathbf{U}}^{n+\theta}+\gamma G\mathbf{P}^n=\mathbf{f}^{n+\theta}, \tag{42a}$$

$$\delta t L(\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) = D\widetilde{\mathbf{U}}^{n+1}, \tag{42b}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \widetilde{\mathbf{U}}^{n+1}) + G(\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) = \mathbf{0}. \tag{42c}$$

Remark 5 The system matrix $DM^{-1}G$ can be used without any approximation when an iterative solver that only involves matrix-vector products is used (for instance, GMRES). Moreover, the computational cost associated to M^{-1} can be reduced using a diagonal mass matrix, that can be obtained by using a closed rule for the space integration (for example, in the case of Lagrangian elements) or some lumping technique (see [49, 65]). In this case, the discrete Laplacian is a sparse matrix that can be easily built. In any case, this discrete Laplacian also introduces artificial boundary conditions for the pressure although not explicitly imposed.

5.2 Equivalent Monolithic Formulation

At this point we will compare the pressure-correction method with the monolithic system as it was done in [32]. The objective is to extend the expression of the error term given by (34) to the nonlinear case and considering the possibility to use approximation (33).

Being $\widetilde{\mathbf{P}}_p^{n+1}$ an extrapolation of $\mathbf{P}^{n+1}$ of order p obtained from previous known values, we can write an *approximated* pressure-correction method as follows:

$$M\frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + K(\widetilde{\mathbf{U}}^{n+1})\widetilde{\mathbf{U}}^{n+1} + G\widetilde{\mathbf{P}}_p^{n+1} = \mathbf{f}^{n+1}, \tag{43a}$$

$$\delta t DM^{-1}G(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1}) = D\widetilde{\mathbf{U}}^{n+1}, \tag{43b}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \widetilde{\mathbf{U}}^{n+1}) + G(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1}) = \mathbf{0}. \tag{43c}$$

We assume BDF1 for the sake of simplicity. Let us introduce the integer β, that will take the value 1 when approximation (33) is used and 0 otherwise. We could rewrite (43) as an equivalent stabilized monolithic formulation (see [32]) in order to identify the perturbation terms introduced by the splitting, getting

$$\begin{aligned} &M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} \\ &\quad + E(\mathbf{U}^{n+1}) + G\mathbf{P}^{n+1} = \mathbf{f}^{n+1}, \\ &D\mathbf{U}^{n+\theta} - \beta\delta t(DM^{-1}G - L)(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1}) = \mathbf{0}. \end{aligned}$$

The term $E(\mathbf{U}^{n+1})$ is given by

$$\begin{aligned} &E(\mathbf{U}^{n+1}) := K(\mathbf{S}^{n+1})\mathbf{U}^{n+1} + K(\mathbf{U}^{n+1})\mathbf{S}^{n+1}, \\ &\mathbf{S}^{n+1} := \delta t M^{-1}G(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1}). \end{aligned}$$

Remark 6 The perturbation of the momentum equation $E(\mathbf{U}^{n+1})$ is formally of order $\mathcal{O}(\delta t^{p+1})$ where p is the approximation of the $\widetilde{\mathbf{P}}_p^{n+1}$ extrapolation. This result generalizes the expression of the perturbation term obtained in (34).

Remark 7 The continuity equation is only perturbed when using the standard Laplacian L. In this case, the perturbation term is $\delta t B(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1})$, with $B = DM^{-1}G - L$, which is formally of order $\mathcal{O}(\delta t^{p+1})$. It was noted in [32] that this term is the reason of the inherent stability of pressure segregation methods. As mentioned earlier, this stabilizer has motivated the use of Orthogonal Subgrid Scale (OSS) stabilization (see e.g. [31, 33, 36, 37]). Let us stress that L is the kind of Laplacian we obtain when using the continuous approach with the pressure Poisson equation. Unfortunately, this stabilization is reduced with the time step size and too weak when using the incremental version. In the next section we list the stability properties of these methods, proved in [32, 35].

Remark 8 This method is unconditionally stable for $p \leq 1$. Higher order methods seem to be conditionally stable.

5.3 Stability of Pressure-Correction Methods

The goal of this section is to present stability estimates for the pressure-correction methods introduced previously. Let us first introduce some additional notation. If $\mathbf{Y}$, $\mathbf{Z}$ are arrays, $\{\mathbf{Y}^n\}_{n=0,1,\dots,N}$ is a sequence of arrays of $N+1$ terms and A a symmetric positive semi-definite matrix, we define

$$(\mathbf{Y}, \mathbf{Z})_A := \mathbf{Y} \cdot A\mathbf{Z},$$

$$\|\mathbf{Y}\|_A := (\mathbf{Y} \cdot A\mathbf{Y})^{1/2},$$

$$\|\mathbf{Y}\|_{-A} := \sup_{\mathbf{Z} \neq \mathbf{0}} \frac{\mathbf{Y} \cdot \mathbf{Z}}{\|\mathbf{Z}\|_A}$$

(here A is assumed to be positive definite),

$$\{\mathbf{Y}^n\} \in \ell^\infty(A) \quad \Longleftrightarrow \quad \|\mathbf{Y}^n\|_A \leq c < \infty, \quad \forall n = 0, 1, \dots, N,$$

$$\{\mathbf{Y}^n\} \in \ell^p(A) \quad \Longleftrightarrow \quad \sum_{n=0}^{N} \delta t \|\mathbf{Y}^n\|_A^p \leq c < \infty, \quad 1 \leq p < \infty.$$

Here and in the following, c denotes a positive constant, not necessarily the same at different appearances.

A remark is needed when $A = K$. This matrix is not symmetric, but it has the contribution from the convective term, which is skew-symmetric, and the contribution from the viscous term, K_{visc}, which is symmetric and positive definite. We will simply write $\mathbf{U} \cdot K(\mathbf{U})\mathbf{U} = \mathbf{U} \cdot K_{\text{visc}}\mathbf{U} \equiv \|\mathbf{U}\|_K^2$.

We will make use also of $L_+ := -L$, which is the positive semi-definite matrix corresponding to the discretization of $-\Delta$.

These definitions will allow us to express our stability results in a compact manner. The basic assumption in all the cases will be that

$$\sum_{n=0}^{N} \delta t \|\mathbf{f}^n\|_{-K}^2 \leq c < \infty, \tag{44}$$

which is the matrix version of the classical condition required for the problem to be well posed. Apart from this, *no other regularity assumptions will be required*. Thus, the following estimates hold for the minimum velocity-pressure regularity.

In what follows, we state the stability estimates for L. In case of using the discrete Laplacian $DM^{-1}G$, the stability estimates for the velocity are identical but *there is not any control over the pressure*.

For method BDF1-SE1, the following stability results have been obtained in [32]:

Stability of BDF1-SE1:
$\{\mathbf{U}^n\} \in \ell^\infty(M)$, $\{\widetilde{\mathbf{U}}^n\} \in \ell^\infty(M) \cap \ell^2(K)$,
$\{\sqrt{\delta t}\, \mathbf{P}^n\} \in \ell^2(L_+)$

The stability estimate for the pressure shows that the pressure gradient *multiplied by* δt is ℓ^2-bounded. When δt is of order $\mathcal{O}(h^2)$ this is optimal [17, 48, 82]. For the velocity, the stability estimates are optimal.

Method BDF1-SE1 is first order because of the order of both the time integration and the splitting error. However, if we consider method BDF1-SE2, with a second order splitting error, we obtain the same estimates for the velocity but much weaker estimates for the pressure. The result is:

Stability of BDF1-SE2:
$\{\mathbf{U}^n\} \in \ell^\infty(M)$, $\{\widetilde{\mathbf{U}}^n\} \in \ell^\infty(M) \cap \ell^2(K)$,
$\{\delta t \mathbf{P}^n\} \in \ell^\infty(L_+)$

The stability estimate for the pressure is now multiplied by δt instead of $\sqrt{\delta t}$ as in the previous case, which makes it weaker (even though the temporal norm is stronger). The way to improve it is by making use of the inf-sup condition, if it holds for the velocity-pressure interpolation employed, or by using stabilization techniques.

For the CN-SE2 the stability estimate was also obtained in [32]. The result is the following:

Stability of CN-SE2:
$\{\mathbf{U}^n\} \in \ell^\infty(M)$, $\{\widetilde{\mathbf{U}}^n\} \in \ell^\infty(M)$,
$\{\widetilde{\mathbf{U}}^{n+1/2}\} \in \ell^2(K)$, $\{\delta t \mathbf{P}^n\} \in \ell^\infty(L_+)$,
$\{\sqrt{\delta t}\, \delta\mathbf{P}^n\} \in \ell^2(L_+)$

The same remarks as those made concerning the stability of method BDF1-SE2 apply now. We therefore conclude that the pressure stability depends on how the splitting is done rather than on the time integration scheme. This is also corroborated by the stability estimate for method BDF2-SE2, obtained in [35]:

Stability of BDF2-SE2:
$\{\mathbf{U}^n\} \in \ell^\infty(M)$, $\{\widetilde{\mathbf{U}}^n\} \in \ell^2(K)$,
$\{\delta t \mathbf{P}^n\} \in \ell^\infty(L_+)$, $\{\sqrt{\delta t}\, \delta\mathbf{P}^n\} \in \ell^2(L_+)$

Remark 9 The inherent pressure stability of these methods is very weak when L is used in place of $DM^{-1}G$ and nonexistent when approximation (33) is not adopted. As commented above, we must rely on div-stable elements (see [18]) or stabilization techniques. This last approach is the one considered in the next section.

5.4 Stabilized Pressure-Correction Methods

5.4.1 *Stabilized Formulation for Incompressible Flows*

In this section we consider the stabilization of the previous methods using the Orthogonal Subgrid Scale (OSS) finite element method described in detail in [33].

Let us introduce the OSS formulation for the backward Euler discretization of problem (13). The idea is to add a least-squares form of the component of the convective and pressure gradient terms orthogonal to the velocity finite element space without boundary conditions. Let P_h be the L^2 projection onto this space. The term to be added to the discrete variational form of the problem is

$$\begin{aligned} &(P_h^\perp(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{v}_h^{n+1} + \nabla q_h^{n+1}), \\ &\quad \tau P_h^\perp(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1})) \\ &= (\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{v}_h^{n+1} + \nabla q_h^{n+1}, \\ &\quad \tau P_h^\perp(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1})), \end{aligned} \tag{45}$$

where τ is the stabilization parameter on which the formulation depends, that we compute as

$$\tau = \left(c_1 \frac{\nu}{h^2} + c_2 \frac{|\boldsymbol{u}_h^{n+1}|}{h} \right)^{-1}, \tag{46}$$

where c_1 and c_2 are algorithmic constants. Both in (45) and in (46) the advection velocity $\boldsymbol{u}_h^{n+1}$ is assumed to be known. It can be taken as a value computed in a previous iteration within an iterative loop or as an extrapolation from velocity values at previous time steps. Note that in (46) the Euclidean norm of this velocity appears, and therefore τ needs to be computed at each integration point.

The discrete Navier-Stokes system stabilized with OSS stabilization consists of finding $\boldsymbol{u}_h^{n+1} \in \mathcal{V}_{h,0}$ and $p_h^{n+1} \in \mathcal{Q}_h$ such that

$$\begin{aligned}
&(\delta_t \boldsymbol{u}_h^{n+1} + \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1}, \boldsymbol{v}_h) + \nu(\nabla \boldsymbol{u}_h^{n+1}, \nabla \boldsymbol{v}_h) \\
&\quad - (p_h^{n+1}, \nabla \cdot \boldsymbol{v}_h) \\
&\quad + (\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{v}_h^{n+1}, \tau P_h^{\perp}(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1})) \\
&= \langle \boldsymbol{f}^{n+1}, \boldsymbol{v}_h \rangle, \qquad (47a)
\end{aligned}$$

$$\begin{aligned}
&(q_h, \nabla \cdot \boldsymbol{u}_h^{n+1}) + (\nabla q_h^{n+1}, \tau P_h^{\perp}(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1})) \\
&= 0, \qquad (47b)
\end{aligned}$$

for all $[\boldsymbol{v}_h, q_h] \in \mathcal{V}_{h,0} \times \mathcal{Q}_h$. Our objective in the remaining of this section is *to extend the previous pressure-correction methods to include the OSS stabilization* we have just described.

5.4.2 Matrix Version of the Stabilized Monolithic System

Prior to writing the matrix version of the stabilized pressure-correction scheme, let us consider the matrix version of the monolithic velocity-pressure calculation.

We start writing the orthogonal projection of the convective and pressure gradient terms as

$$\begin{aligned}
&P_h^{\perp}(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1}) \\
&= \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1} - \boldsymbol{y}_h^{n+1},
\end{aligned}$$

where $\boldsymbol{y}_h^{n+1}$ is the projection of these terms onto the finite element space, that is,

$$\begin{aligned}
&(\boldsymbol{y}_h^{n+1}, \boldsymbol{v}_h) = (\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} + \nabla p_h^{n+1}, \boldsymbol{v}_h), \\
&\forall \boldsymbol{v}_h \in \mathcal{V}_h. \qquad (48)
\end{aligned}$$

From these expressions it is easily checked that the discrete variational problem (47) together with the projection equation (48) lead to the nonlinear algebraic system

$$\begin{aligned}
&M \frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} \\
&\quad + S_{uu}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + S_{up}(\mathbf{U}^{n+1})\mathbf{P}^{n+1} \\
&\quad - S_{uy}(\mathbf{U}^{n+1})\mathbf{Y}^{n+1} = \mathbf{f}^{n+1}, \qquad (49a)
\end{aligned}$$

$$D\mathbf{U}^{n+1} + S_{pp}\mathbf{P}^{n+1} + S_{pu}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - S_{py}\mathbf{Y}^{n+1} = \mathbf{0}, \qquad (49b)$$

$$M\mathbf{Y}^{n+1} - V(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - G\mathbf{P}^{n+1} = \mathbf{0}, \qquad (49c)$$

where $\mathbf{Y}$ is an array with the unknown nodal values of $\boldsymbol{y}$. In the notation used above, we have introduced the new stabilization matrices:

$$S_{uu}(\mathbf{U}^{n+1})_{ij}^{ab} = (\tau \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{\phi}_{a,i}, \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{\phi}_{b,j}),$$

$$S_{up}(\mathbf{U}^{n+1})_i^{ab} = (\tau \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{\phi}_{a,i}, \nabla \pi_b),$$

$$S_{uy}(\mathbf{U}^{n+1})_{ij}^{ab} = (\tau \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{\phi}_{a,i}, \boldsymbol{\phi}_{b,j}),$$

$$S_{pp}{}^{ab} = (\tau \nabla \pi_a, \nabla \pi_b),$$

$$S_{pu}(\mathbf{U}^{n+1})_j^{ab} = (\tau \nabla \pi_a, \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{\phi}_{b,j}),$$

$$S_{py}{}_j^{ab} = (\tau \nabla \pi_a, \boldsymbol{\phi}_{b,j}),$$

$$V(\mathbf{U}^{n+1})_{ij}^{ab} = (\boldsymbol{\phi}_{a,i}, \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{\phi}_{b,j}),$$

where the notation is the same as in Sect. 2.

An alternative version of the orthogonal projection terms has been used in [41]. The projection term is split into two least squares parts, after neglecting crossed terms. In this case, there are two projection arrays to be introduced, one term for the projection of the pressure gradient and the other associated to the convective term:

$$P_h^{\perp}(\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1}) = \boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1} - \boldsymbol{y}_h^{n+1},$$

$$P_h^{\perp}(\nabla p_h^{n+1}) = \nabla p_h^{n+1} - \boldsymbol{z}_h^{n+1},$$

where $\boldsymbol{y}_h^{n+1}$ and $\boldsymbol{z}_h^{n+1}$ are the solution of

$$(\boldsymbol{y}_h^{n+1}, \boldsymbol{v}_h) = (\boldsymbol{u}_h^{n+1} \cdot \nabla \boldsymbol{u}_h^{n+1}, \boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \mathcal{V}_h,$$

$$(\boldsymbol{z}_h^{n+1}, \boldsymbol{v}_h) = (\nabla p_h^{n+1}, \boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \mathcal{V}_h.$$

This approximation slightly simplifies the final stabilized system, which now is

$$\begin{aligned}
&M \frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} \\
&\quad + S_{uu}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - S_{uy}(\mathbf{U}^{n+1})\mathbf{Y}^{n+1} = \mathbf{f}^{n+1}, \qquad (50a)
\end{aligned}$$

$$D\mathbf{U}^{n+1} + S_{pp}\mathbf{P}^{n+1} - S_{py}\mathbf{Z}^{n+1} = \mathbf{0}, \qquad (50b)$$

$$M\mathbf{Y}^{n+1} - V(\mathbf{U}^{n+1})\mathbf{U}^{n+1} = \mathbf{0}, \qquad (50c)$$

$$M\mathbf{Z}^{n+1} - G\mathbf{P}^{n+1} = \mathbf{0}. \qquad (50d)$$

The projections $\mathbf{Y}^{n+1}$ and $\mathbf{Z}^{n+1}$ in (50a) and (50b) are usually replaced by the values from the previous time steps. It has been proved in [10, 38] that the explicit treatment of the projections gives slightly stronger pressure stability. Stabilized pressure-correction and predictor-corrector algorithms from system (49) have been obtained in [41]. The *enhanced* stability properties of these methods has been fully discussed in [32].

Remark 10 OSS is a symmetric stabilization technique. However, this is not the case for some of the most popular stabilization methods, like SUPG, GLS or ASGS (see, e.g., [22, 66–68]), applied to transient problems. These methods lead to non-symmetric formulations even for the Stokes problem.

5.4.3 Matrix Version of the Stabilized Pressure Segregation Algorithms

The stabilized pressure-correction algorithm is obtained either from (49) or from (50) using the same approximations as for the non-stabilized scheme. Introducing the matrices

$$\widetilde{K} \equiv \widetilde{K}(\mathbf{U}^{n+1}) = K(\mathbf{U}^{n+1}) + S_{uu}(\mathbf{U}^{n+1}),$$

$$\widetilde{G} \equiv \widetilde{G}(\mathbf{U}^{n+1}) = G + S_{up}(\mathbf{U}^{n+1}),$$

$$\widetilde{D} \equiv \widetilde{D}(\mathbf{U}^{n+1}) = D + S_{pu}(\mathbf{U}^{n+1}),$$

the first option leads to the following algorithm

$$M\frac{1}{\delta t}(\widetilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \widetilde{K}\widetilde{\mathbf{U}}^{n+1} + \gamma\widetilde{G}\mathbf{P}^n - S_{uy}(\mathbf{U}^{n+1})\mathbf{Y}^{n+\rho} = \mathbf{f}^{n+1}, \tag{51a}$$

$$\widetilde{D}\widetilde{\mathbf{U}}^{n+1} - \delta t\widetilde{D}M^{-1}\widetilde{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^n) + S_{pp}\mathbf{P}^{n+1} - S_{py}\mathbf{Y}^{n+\rho} = \mathbf{0}, \tag{51b}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \widetilde{\mathbf{U}}^{n+1}) + \widetilde{G}(\mathbf{P}^{n+1} - \gamma\mathbf{P}^n) = \mathbf{0}, \tag{51c}$$

$$M\mathbf{Y}^{n+1} - V(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - G\mathbf{P}^{n+1} = \mathbf{0}. \tag{51d}$$

With $\gamma = 0$ we have the non-incremental version whereas $\gamma = 1$ is used for the incremental version. When $\rho = 0$ the projection is treated explicitly and for $\rho = 1$ implicitly. Again, the discrete Laplacian, that now involves stabilization terms, can be approximated by the standard Laplacian L.

It has been proved in [32, 34] that the stabilized versions of all the methods analyzed in the previous section have the extra pressure stability

$$\{\sqrt{\tau}\mathbf{P}^n\} \in \ell^1(L_+)$$

independently of the Laplacian used. This pressure stability does not depend on the time step size and does not decrease when this value goes to zero.

Again, we can motivate the stabilized pressure segregation method as an inexact LU block factorization. Let $\widetilde{C} = \delta t^{-1}M + \widetilde{K}$. The stabilized system matrix for the monolithic system (49) can be written as:

$$\begin{bmatrix} \widetilde{C} & \widetilde{G} \\ \widetilde{D} & S_{pp} \end{bmatrix}\begin{bmatrix} \mathbf{U}^{n+1} \\ \mathbf{P}^{n+1} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{b}}_u^{n+1} \\ \widetilde{\mathbf{b}}_p^{n+1} \end{bmatrix}.$$

The rest of stabilization terms, related to the projections, are included in $\widetilde{\mathbf{b}}_u^{n+1}$ and $\widetilde{\mathbf{b}}_p^{n+1}$. The exact factorization of the matrix of this system, $\widetilde{A}$, is

$$\widetilde{A} = \begin{bmatrix} \widetilde{C} & 0 \\ \widetilde{D} & -\widetilde{D}\widetilde{C}^{-1}\widetilde{G} + S_{pp} \end{bmatrix}\begin{bmatrix} I & \widetilde{C}^{-1}\widetilde{G} \\ 0 & I \end{bmatrix}.$$

Under the assumption of Remark 11 below, we may now approximate

$$\widetilde{C}^{-1} \simeq \delta t M^{-1}, \tag{52}$$

from where we get the approximate factorization of matrix $\widetilde{A}$

$$\widetilde{A}_{PC} = \begin{bmatrix} \widetilde{C} & 0 \\ \widetilde{D} & -\delta t\widetilde{D}M^{-1}\widetilde{G} + S_{pp} \end{bmatrix}\begin{bmatrix} I & \delta t M^{-1}\widetilde{G} \\ 0 & I \end{bmatrix},$$

which corresponds precisely to the factorization implied by the split system (51).

Remark 11 For orthogonal quasi-static subscales, there is not any stabilization term related to the time derivative. That is to say, there is not any term of order $\tau\delta t^{-1}$. A term of this kind would be a problem if the following condition is not satisfied

$$\tau \leq c\delta t.$$

It would imply that the zero order approximation of $\widetilde{C}^{-1}$ (52) should involve extra terms, complicating the method. In fact, this would be the case when using quasi-static subscales without orthogonal projection, like for the classical SUPG and GLS methods (see [40]).

Remark 12 Let us note that, for τ constant, $S_{pp} = \tau L$.

6 Algebraic Velocity-Correction Algorithms

In Sect. 4 we have already introduced algebraic velocity-correction methods, both using extrapolation or inexact factorization as driving motivation. These arguments are parallel to those used for algebraic pressure-correction methods. However, for the latter we had a continuous counterpart presented in Sect. 3 that we lack so far in the case of velocity-correction methods. We start this section introducing a continuous pressure Poisson equation which, as we shall see, is not a good approach. Then, we will get a discrete pressure Poisson equation that will motivate velocity-correction algorithms. We then move to the re-statement of these algorithms in a setting more general than in Sect. 4 and discuss their properties, including their stability and the way to use stabilized formulations in conjunction with them.

6.1 The Continuous Problem

An alternative form of the Navier-Stokes equations where the continuity equation is replaced by a new equation that replaces the mass conservation can also be considered. A pressure Poisson equation (PPE) that implies mass conservation can be obtained by taking the divergence of the momentum equation (1a) and invoking (1b), leading to the system

$$\partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f}, \tag{53a}$$

$$\Delta p = \nabla\cdot(\boldsymbol{f} + \nu\Delta\boldsymbol{u} - \boldsymbol{u}\cdot\nabla\boldsymbol{u}). \tag{53b}$$

The pressure boundary condition is obtained by imposing that the normal component of the pressure gradient is equal to the normal component of the term within parenthesis in the right hand side of (53b). The term $\nabla\cdot(\nu\Delta\boldsymbol{u})$ complicates the finite element approximation of the PPE equation. We can extract this term from the equation by commuting operators, leading to a *simplified* PPE:

$$\Delta p = \nabla\cdot(\boldsymbol{f} - \boldsymbol{u}\cdot\nabla\boldsymbol{u}). \tag{54}$$

Unfortunately, this equation is ill-posed. It can be shown that (53a)–(54) admits extra *spurious* solutions that do not satisfy the original Navier-Stokes equations. These *spurious* solutions satisfy

$$\partial_t(\nabla\cdot\boldsymbol{u}) - \nu\Delta(\nabla\cdot\boldsymbol{u}) = 0,$$

which does not necessarily imply $\nabla\cdot\boldsymbol{u} = 0$ for $t \in (0,T)$. Thus, the false equivalence between the original continuity equation (1b) and the simplified PPE obtained at the continuous level makes (53a)–(54) an inappropriate alternative to the original Navier-Stokes equations. The problems arising when using the simplified PPE are analyzed in [54].

If we want to obtain the weak form of (53) with homogeneous Dirichlet boundary conditions, we need some more regularity in the solution. Let us consider the functional spaces $(\widetilde{\mathcal{V}}_0)_t \equiv L^2(0,T;\widetilde{\mathcal{V}}_0)$ and $\widetilde{\mathcal{Q}}_t \equiv \mathcal{D}(0,T;\widetilde{\mathcal{Q}})$, with $\widetilde{\mathcal{V}}_0 \equiv \boldsymbol{H}^1_0(\Omega)\cap\boldsymbol{H}^2(\Omega)$ and $\widetilde{\mathcal{Q}} \equiv H^1(\Omega)/\mathbb{R}$. Now we assume that the force vector belongs to $\boldsymbol{L}^2(\Omega)$. The weak form consists of finding $[\boldsymbol{u},p] \in (\widetilde{\mathcal{V}}_0)_t \times (\widetilde{\mathcal{Q}})_t$ such that:

$$(\partial_t\boldsymbol{u},\boldsymbol{v}) + \nu(\nabla\boldsymbol{u},\nabla\boldsymbol{v}) + (\boldsymbol{u}\cdot\nabla\boldsymbol{u},\boldsymbol{v}) + (\nabla p,\boldsymbol{v}) = (\boldsymbol{f},\boldsymbol{v}), \tag{55a}$$

$$(\nabla p,\nabla q) = (\boldsymbol{f} - \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nu\Delta\boldsymbol{u},\nabla q), \tag{55b}$$

for all $[\boldsymbol{v},q] \in \widetilde{\mathcal{V}}_0 \times \widetilde{\mathcal{Q}}$. We could recover the regularity requirements of the previous formulation by taking the pressure test function in $H^2(\Omega)$. However, either this choice or (55b) imply the need to construct finite element approximations to $H^2(\Omega)$, which is an involved task (at least for conforming interpolations, as we consider throughout). Thus, the finite element discretization of the pressure Poisson equation (55b) will not be studied. We could try to circumvent the regularity assumptions neglecting the diffusive term in (55b). Unfortunately, the simplified Poisson equation is ill-posed, as commented above.

6.2 Formulation of the Algorithms

At the continuous level the PPE is not an appropriate equation to be discretized as it has been discussed above. The alternative is to use *directly the discrete PPE* (DPPE) given by (27b) in the case of the backward Euler time discretization. This starting point was proposed in [7].

Obviously, for the derivation of scheme (27) *no extra regularity conditions are required for the pressure*, a main difference in comparison to the continuous PPE (53b). Scheme (27), which has been obtained after algebraic manipulation, *is equivalent to the original monolithic discretized scheme* (17). Furthermore, boundary conditions arise naturally from the original problem.

In Sect. 4, and in the context of pressure-correction schemes, we considered the possibility to use approximation (33) to avoid the need to deal with matrix $DM^{-1}G$. Before formulating the velocity-correction schemes we are interested in, let us comment on a possible improvement of this approximation, which consists in

$$\begin{aligned} DM^{-1}G\mathbf{P}^{n+1} &= L\mathbf{P}^{n+1} + (DM^{-1}G - L)\mathbf{P}^{n+1} \\ &\approx L\mathbf{P}^{n+1} + (DM^{-1}G - L)\widetilde{\mathbf{P}}^{n+1}_p, \end{aligned} \tag{56}$$

where $\widetilde{\mathbf{P}}^{n+1}_p$ is an extrapolation of order p to compute $\widetilde{\mathbf{P}}^{n+1}$. This new approximation couples space and time errors, yielding better accuracy.

Using (56) in the DPPE (27b) we get

$$\begin{aligned} &\delta t L(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}^{n+1}_p) \\ &\quad = \delta t DM^{-1}(\mathbf{f}^{n+1} - K\mathbf{U}^{n+1} - G\widetilde{\mathbf{P}}^{n+1}_p) + D(\mathbf{U}^n), \end{aligned}$$

which is more appealing than (27b) from a computational point of view. Approximation (56) is also appropriate in the context of predictor-corrector methods, and could be used as well for the DPPE arising in pressure-corrector schemes.

We are now in a position to extend the velocity-correction algorithms presented in Sect. 4. As explained there, the basic idea is to extrapolate the velocity in the DPPE in order to compute an intermediate pressure, with which a corrected velocity can be computed from the momentum equation. Since the pressure is not needed in the next time step, we may take the intermediate pressure as the final one, as explained in Sect. 4. Denoting again by $\widetilde{\mathbf{U}}^{n+1}_q$ an extrapolation of order q to compute $\mathbf{U}^{n+1}$, the velocity-correction scheme using the BDF method of order k to integrate in time reads:

$$\begin{aligned} \gamma_k\delta t DM^{-1}G\mathbf{P}^{n+1} &= \gamma_k\delta t DM^{-1}(\mathbf{f}^{n+1} - \boldsymbol{K}(\widetilde{\mathbf{U}}^{n+1}_q)\widetilde{\mathbf{U}}^{n+1}_q) \\ &\quad + D\left(\sum_{i=0}^{k-1}\alpha^i_k\mathbf{U}^{n-k}\right), \end{aligned} \tag{57a}$$

$$M\frac{1}{\delta t}(\delta_k \mathbf{U}^{n+1}) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} = \mathbf{f}^{n+1}. \tag{57b}$$

The splitting error will be of order $q+1$ and the time integration error of order k. Thus, $q = k-1$ is required if one asks to have both errors of the same order.

Invoking approximation (56) for $DM^{-1}G$ we obtain the following system:

$$\begin{aligned}&\gamma_k \delta t L(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1})\\ &\quad = \gamma_k \delta t DM^{-1}(\mathbf{f}^{n+1} - K(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1} - G\widetilde{\mathbf{P}}_p^{n+1})\\ &\qquad + D\left(\sum_{i=0}^{k-1}\alpha_k^i \mathbf{U}^{n-k}\right),\end{aligned} \tag{58a}$$

$$M\frac{1}{\delta t}(\delta_k \mathbf{U}^{n+1}) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} = \mathbf{f}^{n+1}, \tag{58b}$$

where now $q = p = k-1$ in order to have all errors (splitting, pressure extrapolation and time integration) of the same order. For instance, we can obtain a first order method (in time) taking $k = 1$ and $q = p = 0$:

$$\delta t L\mathbf{P}^{n+1} = \delta t DM^{-1}\mathbf{f}^{n+1} + D\mathbf{U}^n,$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} = \mathbf{f}^{n+1}.$$

We get second order of accuracy in time with $k = 2$ and $q = p = 1$. In this case the system to be solved is:

$$\begin{aligned}&\frac{2}{3}\delta t L(\mathbf{P}^{n+1} - \mathbf{P}^n)\\ &\quad = \frac{2}{3}\delta t DM^{-1}(\mathbf{f}^{n+1} - K(\mathbf{U}^n)\mathbf{U}^n - G\mathbf{P}^n)\\ &\qquad + D\left(\frac{4}{3}\mathbf{U}^n - \frac{1}{3}\mathbf{U}^{n-1}\right),\end{aligned}$$

$$\begin{aligned}&M\frac{1}{2\delta t}(3\mathbf{U}^{n+1} - 4\mathbf{U}^n + \mathbf{U}^{n-1}) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1}\\ &\quad = \mathbf{f}^{n+1}.\end{aligned}$$

Remark 13 An interesting feature of these methods is that the splitting error is related to the accuracy of the velocity instead of the pressure (as it happens for pressure-correction methods). This is an advantage because it is known from the convergence analysis of different pressure segregation methods that the error estimates for the velocity are sharper than those for the pressure.

Similarly, third order methods can be obtained with $k = 3$, that is a BDF3 time integration scheme, and $q = p = 2$. Unfortunately, numerical experimentation indicates that this method is only conditionally stable, as it happens for third order pressure-correction methods.

Remark 14 Numerical experimentation indicates that velocity-correction methods are *unconditionally stable* for $q \leq 1$ (this is proved for some schemes in [7] and the main results are listed below). The instabilities shown by higher order schemes seem to be a common feature for different sorts of methods that segregate the calculation of the velocity and the pressure. This behavior has been pointed out for pressure-correction methods in [87]. This misbehavior is also commented in [62] for a different version of velocity-correction methods without any definitive conclusion.

6.3 Equivalent Monolithic Formulation

As for pressure-correction methods in [32], we could rewrite this system in a monolithic format in order to identify the perturbation terms introduced by the splitting. Taking the difference between (58b) after being multiplied by $\delta t DM^{-1}$ and (58a), we get the following equivalent system (in the case $k = 1$):

$$M\frac{1}{\delta t}\mathbf{U}^{n+1} - \mathbf{U}^n + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} = \mathbf{f}^{n+1},$$

$$\begin{aligned}&D\mathbf{U}^{n+1} + \beta\delta t B(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1})\\ &\quad + \delta t DM^{-1}(K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - K(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1}) = \mathbf{0},\end{aligned}$$

where $B := DM^{-1}G - L$ and $\beta = 1$ if approximation (56) is used and 0 otherwise.

As discussed in Sect. 4, the perturbed system obtained in this case is different to the one obtained for pressure-correction methods, the main advantage being that the momentum equation is not perturbed anymore. This splitting is introducing a perturbation only in the continuity equation. The term $\delta t B(\mathbf{P}^{n+1} - \widetilde{\mathbf{P}}_p^{n+1})$ (that also appears in the classical pressure-correction scheme with approximation (33)) arises now from approximation (56) and is not a splitting error. The only error term due to the splitting is $\delta t DM^{-1}(K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - K(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1})$, which is formally of order $\mathcal{O}(\delta t^{q+1})$.

6.4 Stability of Velocity-Correction Methods

In this section we present a complete set of stability results for the original DPPE velocity-correction schemes obtained in [7]. The key aspects of the following analysis are:

- To identify the inherent stability related to velocity-correction methods.
- The fact that stability properties are weaker for more accurate schemes.

We analyze velocity-correction methods using BDF1 and Crank-Nicolson time integration schemes. For simplicity in the exposition we will identify the methods as BDF1-Uq-Pp, q being the order of the velocity extrapolation and p the order of the pressure extrapolation. Similarly, for the Crank-Nicolson scheme we will use CN-Uq-Pp. The parameters q and p will take the values 0 and 1 for BDF1 and $p = q = 1$ for Crank-Nicolson.

For obtaining stability bounds in this section the basic assumption in all the cases will be that

$$\sum_{n=0}^{N} \delta t \|\mathbf{f}^n\|_M^2 \le c < \infty, \tag{59}$$

for all $\delta t > 0$, that corresponds to requiring $\boldsymbol{f} \in L^2(0,T; \boldsymbol{L}^2(\Omega))$ for the continuous problem. Apart from this, *no other regularity assumptions will be required* (even though this condition is stronger than what is needed in pressure-correction schemes, given by (44)).

The main stability results for the different velocity-correction methods analyzed in this section have been highlighted in boxes, as in Sect. 5.

The first scheme to be considered is the simplest BDF1-U0-P0, together with the scheme BDF1-U0 *without pressure extrapolation*, that is to say, without making use of approximation (56). As before, we distinguish both possibilities according to the parameter β, and we denote the resulting methods by BDF1-U0-(P0). For these methods we have obtained the following stability results:

Stability of BDF1-U0-(P0):
$\{\mathbf{U}^n\} \in \ell^\infty(M) \cap \ell^2(K)$,
$\{\sqrt{\delta t}\, M^{-1} K(\mathbf{U}^n)\mathbf{U}^n\} \in \ell^2(M)$, $\quad \{\sqrt{\delta t}\, \mathbf{P}^n\} \in \ell^2(\beta B)$

Recall that matrix B is defined as $B = DM^{-1}G - L$. This estimate is optimal for the velocity. The stability for the pressure is certainly not optimal, but the important point is that we have obtained it without relying on the classical inf-sup condition for the velocity-pressure interpolation. Observe also that this stability is lost when $\beta = 0$.

In the previous estimate we have also displayed the additional control we have on the norm of the viscous plus convective terms.

The stability analysis of the method BDF1-U0-P1 follows in a straightforward way. Obviously, this method only makes sense if approximation (56) is used ($\beta = 1$, with the previous notation). The stability results for this method are:

Stability of BDF1-U0-P1:
$\{\mathbf{U}^n\} \in \ell^\infty(M) \cap \ell^2(K)$,
$\{\sqrt{\delta t}\, M^{-1} K(\mathbf{U}^n)\mathbf{U}^n\} \in \ell^2(M)$,
$\{\delta t \mathbf{P}^n\} \in \ell^\infty(B)$, $\quad \{\sqrt{\delta t}\, \delta \mathbf{P}^n\} \in \ell^2(B)$

Observe that the pressure stability now is weaker than for the BDF1-U0-P0 method, where the pressure was extrapolated only up to zero order. Control in $\ell^\infty(B)$ is obtained only for $\{\delta t \mathbf{P}^n\}$, whereas the optimal would be $\{\sqrt{\delta t}\mathbf{P}^n\}$ if $\delta t = \mathcal{O}(h^2)$. In general, a better approximation for the pressure implies less stability (which has to be found either from the use of stabilization methods or by invoking an inf-sup condition).

Now we will list the stability results for the BDF1-U1-P0 method. If approximation (56) is not used, the stability results presented in the following for the pressure disappear. Thus, we will concentrate only in the case $\beta = 1$. The results are:

Stability of BDF1-U1-P0:
$\{\mathbf{U}^n\} \in \ell^\infty(M) \cap \ell^2(K)$, $\quad \{\delta t M^{-1} K(\mathbf{U}^n)\mathbf{U}^n\} \in \ell^\infty(M)$,
$\{\sqrt{\delta t}\, \mathbf{P}^n\} \in \ell^2(B)$

The stability analysis of BDF1-U1-P1 differs from the previous one just in the pressure term to be bounded:

Stability of BDF1-U1-P1:
$\{\mathbf{U}^n\} \in \ell^\infty(M) \cap \ell^2(K)$, $\quad \{\delta t M^{-1} K(\mathbf{U}^n)\mathbf{U}^n\} \in \ell^\infty(M)$,
$\{\delta t \mathbf{P}^n\} \in \ell^\infty(B)$, $\quad \{\sqrt{\delta t}\, \delta \mathbf{P}^n\} \in \ell^2(B)$

Velocity-correction algorithms have also been analyzed for the Crank-Nicolson time integration scheme. We only list the stability results of the CN-U1-P1 method, since in order to maintain the second order accuracy of the Crank-Nicolson scheme the velocity and the pressure need to be extrapolated to first order. Again, we will consider that approximation (56) is used; otherwise, the pressure bounds presented next disappear. This method reads as follows:

$$\begin{aligned} &\delta t L(\mathbf{P}^{n+1} - \mathbf{P}^n) \\ &\quad = \delta t D M^{-1}(\mathbf{f}^{n+1} - K(\mathbf{U}^{n-1/2})\mathbf{U}^{n-1/2} - G\mathbf{P}^n) \\ &\qquad + D(\mathbf{U}^n), \end{aligned} \tag{60a}$$

$$\begin{aligned} &M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+1/2})\mathbf{U}^{n+1/2} + G\mathbf{P}^{n+1} \\ &\quad = \mathbf{f}^{n+1}. \end{aligned} \tag{60b}$$

At the first time step, we adopt the first order BDF1-U0-P0 scheme for simplicity. It does not affect the overall second order accuracy of the method. In the following setting this initialization is equivalent to take $\mathbf{U}^{-1/2} = \mathbf{0}$. The stability results we proved in [7] for (60) are:

Stability of CN-U1-P1:
$\{\mathbf{U}^n\} \in \ell^\infty(M) \cap \ell^2(K)$,
$\{\delta t M^{-1} K(\mathbf{U}^{n-1/2})\mathbf{U}^{n-1/2}\} \in \ell^\infty(M)$,
$\{\delta t \mathbf{P}^n\} \in \ell^\infty(B)$, $\quad \{\sqrt{\delta t}\, \delta \mathbf{P}^n\} \in \ell^2(B)$

We can easily see from the previous stability bounds that the inherent pressure stability of velocity-correction methods seems insufficient. We only have some pressure stability under approximation (56). And even in this case, the stability is under the norm associated to B (a difference between discrete Laplacians that tends to zero with h). Thus, their behavior is different from pressure-correction methods, which have a *stronger* inherent pressure stability [32, 35]. For a first order splitting error, using pressure-correction methods with approximation (56) we have control over the whole pressure gradient $\|\sqrt{\delta t}\nabla p_h^{n+1}\|_0$. For velocity-correction methods, the bound for the projection of the pressure gradient onto the finite element space (weighted with $\sqrt{\delta t}$) does not appear. Nevertheless, we can recover the control over the whole gradient by using the stability provided by the momentum equation (as shown in [7]). Summarizing, even though the stability bounds of the velocity-correction methods seem weaker, we also have stability over the whole pressure gradient (under approximation (56)).

6.5 Stabilized Velocity-Correction Methods

Again, we can obtain a DPPE from the stabilized monolithic system (49). Instead of (27b) what we now obtain, for $k = 1$, is:

$$\begin{aligned}(\delta t\widetilde{D}M^{-1}\widetilde{G} - S_{pp})\mathbf{P}^{n+1} &= \delta t\widetilde{D}M^{-1}(\mathbf{f}^{n+1} - \widetilde{K}\mathbf{U}^{n+1} + S_{uy}(\mathbf{U}^{n+1})\mathbf{Y}^{n+1}) \\ &\quad + \widetilde{D}\mathbf{U}^n - S_{py}\mathbf{Y}^{n+1}, \end{aligned}\tag{61}$$

where matrices that appear in this expression have been introduced before. Alternatively, we can consider a version of the DPPE that allows to replace the stabilized discrete Laplacian $\widetilde{D}M^{-1}\widetilde{G}$ by the non-stabilized version:

$$\begin{aligned}(\delta t DM^{-1}G - S_{pp})\mathbf{P}^{n+1} &= \delta t DM^{-1}(\mathbf{f}^{n+1} - K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - S_{uu}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} \\ &\quad - S_{up}(\mathbf{U}^{n+1})\mathbf{P}^{n+1} + S_{uy}(\mathbf{U}^{n+1})\mathbf{Y}^{n+1}) + D\mathbf{U}^n \\ &\quad + S_{pu}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - S_{py}\mathbf{Y}^{n+1}. \end{aligned}\tag{62}$$

We need to make some further approximations in order to obtain a computationally appealing stabilized velocity-correction system. As above, the velocity in the right hand side of (61) is extrapolated. Now, this extrapolation is needed not only for the viscous and convective term, but also for the stabilization terms. Further, we need to extrapolate the projection array $\mathbf{Y}$. In our numerical experimentation (see [7]) the orthogonal projection has been treated explicitly for velocity-correction methods. In the worst case, the error induced by these extrapolations is of order $\mathcal{O}(\tau\delta t)$. If we assume that $\tau \leq c\delta t$, we do not spoil the accuracy for first and second order methods. Finally, for the DPPE (62), the idea is to consider the pressure stabilization terms in an explicit way and to use DM^{-1} instead of $\widetilde{D}M^{-1}$ for the obtention of the DPPE. This last method has been used and analyzed in [7]. The scheme with (62) is the one adopted herein.

For the momentum equation no assumptions are required. Thus, the stabilized version of system (57) in the case $k = 1$ is

$$\begin{aligned}&(\delta t DM^{-1}G - S_{pp})\mathbf{P}^{n+1} \\ &\quad = \delta t DM^{-1}(\mathbf{f}^{n+1} - K(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1} - S_{uu}(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1} \\ &\qquad - S_{up}(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{P}}_p^{n+1} + S_{uy}(\widetilde{\mathbf{U}}_q^{n+1})\mathbf{Y}^n) + D\mathbf{U}^n \\ &\qquad + S_{pu}(\widetilde{\mathbf{U}}_q^{n+1})\widetilde{\mathbf{U}}_q^{n+1} - S_{py}\mathbf{Y}^n, \\ &M\frac{1}{\delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + K(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + G\mathbf{P}^{n+1} \\ &\qquad + S_{uu}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + S_{up}(\mathbf{U}^{n+1})\mathbf{P}^{n+1} - S_{uy}(\mathbf{U}^{n+1})\mathbf{Y}^n \\ &\quad = \mathbf{f}^{n+1}, \\ &M\mathbf{Y}^{n+1} - V(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - G\mathbf{P}^{n+1} = \mathbf{0}. \end{aligned}$$

At this point, approximation (56) can be applied in order to avoid dealing with $DM^{-1}G$.

Let us conclude by showing which is the additional stability introduced by pressure stabilization only (not by convection stabilization). This situation makes sense for flows dominated by viscosity, for which (46) yields $\tau = \mathcal{O}(h^2)$. On top of the stability bounds given in the previous subsection, we now have:

$$\{\sqrt{\tau}\mathbf{P}^n\} \in \ell^1(L_+).$$

We refer to [7] for the proof of this result.

7 Block Preconditioners and Predictor-Corrector Schemes for the Navier-Stokes Equations

In Sect. 4 we have seen that pressure segregation algorithm can be motivated from inexact LU block factorizations of the Navier-Stokes system matrix A. These methods introduce a perturbation term that is reduced with the time step size δt. Instead of using these approximations of A as solvers, we could use them as preconditioners in an iterative process. When convergence is reached, the solution would coincide with the one of the monolithic system. One choice is to consider Richardson iteration with preconditioner A_{PC} (for pressure-correction algorithms), leading at every time step to the following iterative process (see [99]): given $\mathbf{X}^k$, solve

$$A_{PC}\mathbf{X}^{k+1} = \mathbf{b} - A\mathbf{X}^k, \tag{63}$$

where we have omitted the time step super-index for conciseness and here k denotes the iteration number. We can easily see that we can replace A_{PC} by its L-block, denoted by L_{PC}, because the array $\mathbf{U}^k$ is not used at the next iteration, only $\mathbf{P}^k$ (see Remark 16 below), which means that in fact the preconditioned system to consider is

$$L_{PC}^{-1} A\mathbf{X}^{n+1} = L_{PC}^{-1}\mathbf{b}^{n+1}. \tag{64}$$

In fact, the iterative process (63) can also be written in terms of the pressure only. When the process has converged, the velocity values can be recovered from the pressure. Equation (63) can be written as a preconditioned pressure Schur complement problem with preconditioner $\delta t DM^{-1}G$,

$$\delta t DM^{-1}G\mathbf{P}^{k+1} = DC^{-1}\left(\mathbf{f}^{n+1} + M\frac{1}{\delta t}\mathbf{U}^n\right) - DC^{-1}G\mathbf{P}^k, \tag{65}$$

when Richardson iterations are used. Let us remark that the pressure sequence $\{\mathbf{P}^k\}_{k=1,\dots}$ is identical when using (63) or (65).

In order to improve the convergence towards the monolithic system, Krylov methods, like conjugate gradients (CG), GMRES or BiCGstab, can be considered.

Remark 15 Symmetric stabilization techniques, like OSS, lead to symmetric preconditioners for the Schur complement $\delta t\widetilde{D}M^{-1}\widetilde{G}$, allowing the use of CG. However, when using other stabilization methods, like GLS or ASGS, $\delta t\widetilde{D}M^{-1}\widetilde{G}$ becomes non-symmetric, requiring the use of more expensive Krylov methods, like GMRES of BiCGstab.

The Krylov method can be applied to the preconditioned system (64) or to the Schur complement version, that is,

$$\begin{aligned}&(\delta t DM^{-1}G)^{-1}DC^{-1}G\mathbf{P}^{n+1}\\&\quad= (\delta t DM^{-1}G)^{-1}DC^{-1}\left(\mathbf{f}^{n+1} + M\frac{1}{\delta t}\mathbf{U}^n\right).\end{aligned}$$

When using the preconditioned residual as stopping criteria, as it is usually done for GMRES, the residual obtained from (65) is homogeneous, in the sense that only involves the pressure, whereas (63) leads to a heterogeneous residual that mixes velocity and pressure terms. Furthermore, system (65) requires less memory because the elements of the Krylov base have only pressure dimensions.

Remark 16 Every GMRES iteration involves a matrix-vector product $L_{PC}^{-1}A\mathbf{Y} = \mathbf{Z}$. Let us write the arrays $\mathbf{Z} = [\boldsymbol{V}, \mathbf{Q}]$ and $\mathbf{Y} = [\mathbf{W}, \mathbf{R}]$ in their velocity and pressure parts. We can write every GMRES iteration as follows: given $\mathbf{Y}$, solve $\mathbf{Z}$ such that

$$C(\boldsymbol{V} - \mathbf{W}) = G\mathbf{R}, \tag{66a}$$

$$\delta t^{-1}DM^{-1}G\mathbf{Q} = D(\boldsymbol{V} - \mathbf{W}). \tag{66b}$$

From this system, it is easily seen that a Krylov base only in terms of pressures would be enough, because $\mathbf{W}$ is not really needed for the computation of $\mathbf{Q}$. A Krylov base only in terms of pressures is very appealing since it reduces the memory requirements of the algorithm.

Remark 17 The inverse of the preconditioner needed for the matrix-vector product is used for the computation of the elements of the Krylov base. Therefore, in order to reduce the computational cost, a loose tolerance can be used for this system solution. When doing that, a FGMRES (see [83]) must be used for the external iterations.

Pressure-correction algorithms are in fact intimately related to *pressure Schur complement preconditioners* (see [44–46, 74, 99]).

The pressure-corrector preconditioner only takes into account the effect of the *reaction term* (the inertia term $\frac{1}{\delta t}M$) in the approximation of C^{-1}, neglecting viscous and convective terms. This method is very effective for *small time step sizes*. However, when the diffusive and/or the convective terms become more important, these preconditioners do not perform well.

In order to introduce viscous effects, the following preconditioner has been suggested in [23]

$$(\nu M_p^{-1} + (\delta t DM^{-1}G)^{-1})^{-1}, \tag{67}$$

where M_p is a pressure mass matrix. This method is a merge between the pressure-correction scheme and the Uzawa algorithms (see [2]). This preconditioner has turned out to be the discrete counterpart of the rotational form introduced in [98] (see [99]).

In [45, 74] a more involved preconditioner is used,

$$(\delta t DM^{-1}G)F_p^{-1}M_p,$$

where F_p is the projection of the differential operator $(\delta t^{-1} + \boldsymbol{u}\cdot\nabla - \nu\Delta)$ onto the finite element space for the pressure. For the Stokes problem, we recover (67).

In any case, using a preconditioner for the pressure Schur complement system involves to solve a problem with system matrix C at every iteration, which is very expensive. In order to avoid that, preconditioners of the kind

$$\begin{bmatrix} F_u & G \\ 0 & -\delta t DM^{-1}G \end{bmatrix} \tag{68}$$

have been considered in [44, 46], where F_u is a cheap preconditioner for C.

Remark 18 The pressure problem, with system matrix $DM^{-1}G$ (or L) is usually solved using multigrid techniques (see, e.g., [45]).

Velocity-correction methods can also be used as preconditioners, simply replacing L_{PC} by L_{VC}, the L-block of A_{VC}, in (63) or (64). This scheme has been suggested in [7] (therein denoted as predictor-corrector with nested loops) with Richardson iterations. Comparisons between the performance of pressure-correction and velocity-correction algorithms can be found also in [7].

When considering these methods as preconditioners and the nonlinear convective term is treated implicitly, we have two nested loops, one outer loop for nonlinear iterations and an inner loop for the convergence towards the monolithic system. The use of *one loop* algorithms that couple both iterative processes has been studied in [12, 35, 39, 41] for pressure-correction methods and in [7] for velocity-correction methods.

Starting from the pressure-correction method (42), we can motivate a predictor corrector scheme whose goal is to converge to the monolithic time discretized problem. We will omit the details of the motivation. Using the trapezoidal rule to integrate in time, and denoting by a superscript i the i-th iteration of the scheme, the resulting linearized system is

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1,i+1}-\mathbf{U}^{n})+K(\mathbf{U}^{n+\theta,i})\mathbf{U}^{n+\theta,i+1} + G\mathbf{P}^{n+\theta,i}=\mathbf{f}^{n+\theta}, \tag{69a}$$

$$\delta t DM^{-1}G(\mathbf{P}^{n+\theta,i+1}-\mathbf{P}^{n+\theta,i})=D\mathbf{U}^{n+\theta,i+1}. \tag{69b}$$

Apparently, this is a straightforward iteration procedure for solving the original *monolithic* problem (17) freezing the pressure gradient in the momentum equation. However, there is a term whose presence would be hardly motivated by looking only at this system, namely, the term $\delta t DM^{-1}G(\mathbf{P}^{n+\theta,i+1}-\mathbf{P}^{n+\theta,i})$. The motivation to introduce it comes from the inspection of what happens in the pressure-correction scheme.

If instead of starting from the generalized trapezoidal rule the second order BDF scheme is employed, the iterative scheme is

$$M\frac{1}{2\delta t}(3\mathbf{U}^{n+1,i+1}-4\mathbf{U}^{n}+\mathbf{U}^{n-1}) + K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i+1}+G\mathbf{P}^{n+1,i}=\mathbf{f}^{n+1}, \tag{70a}$$

$$\frac{2}{3}\delta t L(\mathbf{P}^{n+1,i+1}-\mathbf{P}^{n+1,i})=D\mathbf{U}^{n+1,i+1}. \tag{70b}$$

Both (69) and (70) are iterative schemes in which the pressure calculation is uncoupled from the velocity.

Analogously, from the monolithic scheme (27), where the mass conservation is imposed by the DPPE, we can design new predictor-corrector methods. The resulting predictor-corrector method is, for $k=1$ and using a Picard linearization of the convective term:

$$\delta t DM^{-1}G\mathbf{P}^{n+1,i+1} = \delta t DM^{-1}(\mathbf{f}^{n+1}-K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i})+D(\mathbf{U}^{n}), \tag{71a}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1,i+1}-\mathbf{U}^{n})+K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i+1} + G\mathbf{P}^{n+1,i+1}=\mathbf{f}^{n+1}, \tag{71b}$$

or, when using approximation (56),

$$\delta t L(\mathbf{P}^{n+1,i+1}-\mathbf{P}^{n+1,i}) = \delta t DM^{-1}(\mathbf{f}^{n+1}-K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i}-G\mathbf{P}^{n+1,i}) + D(\mathbf{U}^{n}), \tag{72a}$$

$$M\frac{1}{\delta t}(\mathbf{U}^{n+1,i+1}-\mathbf{U}^{n})+K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i+1} + G\mathbf{P}^{n+1,i+1}=\mathbf{f}^{n+1}. \tag{72b}$$

These methods have to be properly initialized, that is to say, it is convenient to start the process with a splitting error at least of the same order as the time integration scheme. Thus, for first order methods we could take $\mathbf{U}^{n+1,0}=\widetilde{\mathbf{U}}_q^{n+1}$ and $\mathbf{P}^{n+1,0}=\widetilde{\mathbf{P}}_p^{n+1}$, with $q=p=0$. However, it is better to use $q=p=1$, starting with a second order splitting error. This does not imply a significant additional computational cost and improves the initial guess of the iterative process. It is even useful to take $q=p=2$. This requires additional storage, but improves significantly the initial guess and thus reduces the number of iterations.

In algorithm (71) all the terms are motivated from the monolithic version of the problem, a difference with the predictor-corrector method (69) based on pressure-correction schemes. Thus, from the DPPE version of the monolithic problem a predictor-corrector scheme naturally arises, whereas it does not occur so for the typical monolithic version.

The second order version of (72) using BDF2 is

$$\frac{2}{3}\delta t L(\mathbf{P}^{n+1,i+1}-\mathbf{P}^{n+1,i}) = \frac{2}{3}\delta t DM^{-1}(\mathbf{f}^{n+1}-K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i}-G\mathbf{P}^{n+1,i}) + D\left(\frac{4}{3}\mathbf{U}^{n}-\frac{1}{3}\mathbf{U}^{n-1}\right),$$

$$M\frac{1}{2\delta t}(3\mathbf{U}^{n+1,i+1}-4\mathbf{U}^{n}+\mathbf{U}^{n-1})+K(\mathbf{U}^{n+1,i})\mathbf{U}^{n+1,i+1} + G\mathbf{P}^{n+1,i+1}=\mathbf{f}^{n+1},$$

with the appropriate initializations $\mathbf{U}^{n+1,0}=\widetilde{\mathbf{U}}_q^{n+1}$ and $\mathbf{P}^{n+1,0}=\widetilde{\mathbf{P}}_p^{n+1}$, with $q=p=1$ or $q=p=2$.

The stabilized version of all the schemes presented is straightforward.

8 Summary and Conclusions

The purpose of this article has been to present a state-of-the-art review of pressure segregation algorithms motivated from an algebraic point of view. Algebraic pressure segregation methods have been motivated from two different points of view:

- Extrapolation
- Inexact factorization

Depending on the extrapolated unknown, we have distinguished between pressure-correction and velocity-correction algorithms.

The Yosida method can be considered as a kind of pressure-correction algorithm. The difference between this approach and classical pressure-correction algorithms is the evaluation of the end-of-step velocity. We have also commented the possibility of getting high order pressure correction algorithms by using more accurate inexact *LU* factorizations.

The inherent pressure stability of these methods has been discussed. This stability is weaker for more accurate algorithms and deteriorates when the time step size is reduced. For these reasons, we must rely on inf-sup stable elements or stabilized formulations for getting robust algorithms. In this article we have discussed how to combine a stabilized formulation with the algorithms presented and designed stabilized pressure segregation algorithms with improved pressure stability.

The inexact *LU* factorizations that motivate pressure segregation algorithms can also be used as preconditioners. These methods are related to widely used block-preconditioners for the numerical approximation of the Navier-Stokes equations. Finally, one-loop algorithms (called predictor-corrector algorithms) that deal with nonlinear iterations and pressure correction iterations have been presented.

References

1. Armero F, Simo JC (1996) Long-term dissipativity of time-stepping algorithms for an abstract evolution equation with application to MHD and Navier-Stokes equations. Comput Methods Appl Mech Eng 131:41–90
2. Arrow K, Hurwicz L, Uzawa H (1958) Studies in nonlinear programming. Stanford University Press, Stanford
3. Babuška I (1971) Error bounds for the finite element method. Numer Math 16:322–333
4. Badia S (2006) Stabilized pressure segregation methods and their application to fluid-structure interaction problems. PhD thesis, Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports, Universitat Politècnica de Catalunya, Barcelona
5. Badia S, Codina R (2007) Convergence analysis of the FEM approximation of the first order projection method for incompressible flows with and without the inf-sup condition. Numer Math 107(4):533–557
6. Badia S, Codina R (2007) On some fluid-structure iterative algorithms using pressure segregation methods. Application to aeroelasticity. Int J Numer Methods Eng 72:46–71
7. Badia S, Codina R (2008) Pressure segregation methods based on a discrete pressure Poisson equation. An algebraic approach. Int J Numer Methods Fluids 56(4):351–382
8. Baker GA, Dougalis VA, Karakashian A (1982) On a higher order accurate fully discrete Galerkin approximation to the Navier-Stokes equations. Math Comput 39:339–375
9. Bell JB, Colella P, Glaz HM (1989) A second-order projection method for the incompressible Navier-Stokes equations. J Comput Phys 85:257–283
10. Blasco J, Codina R (2001) Space and time error estimates for a first order, pressure stabilized finite element method for the incompressible Navier-Stokes equations. Appl Numer Math 38:475–497
11. Blasco J, Codina R (2004) Error estimates for an operator splitting method for incompressible flows. Appl Numer Math 51:1–17
12. Blasco J, Codina R, Huerta A (1998) A fractional step method for the incompressible Navier-Stokes equations related to a predictor-multicorrector algorithm. Int J Numer Methods Fluids 28:1391–1419
13. Boland JM, Nicolaides RA (1983) Stability of finite elements under divergence constraints. SIAM J Numer Anal 20:722–731
14. Brenner SC, Scott LR (1994) The mathematical theory of finite element methods. Springer, Berlin
15. Brezzi F (1974) On the existence, uniqueness and approximation of saddle point problems arising from Lagrange multipliers. RAIRO Anal Numer 8:129–151
16. Brezzi F, Bathe KJ (1990) A discourse on the stability conditions for mixed finite element formulations. Comput Methods Appl Mech Eng 82:27–57
17. Brezzi F, Douglas J (1988) Stabilized mixed methods for the Stokes problem. Numer Math 53:225–235
18. Brezzi F, Fortin M (1991) Mixed and hybrid finite element methods. Springer, Berlin
19. Brezzi F, Rappaz J, Raviart PA (1981) Finite dimensional approximation of nonlinear problems. Part I: Branches of nonsingular solutions. Numer Math 36:1–25
20. Brezzi F, Rappaz J, Raviart PA (1981) Finite dimensional approximation of nonlinear problems. Part II: Limit points. Numer Math 37:1–28
21. Brezzi F, Rappaz J, Raviart PA (1981) Finite dimensional approximation of nonlinear problems. Part III: Simple bifurcation points. Numer Math 38:1–30
22. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation. Comput Methods Appl Mech Eng 32:199–259
23. Cahouet J, Chabard J-P (1988) Some fast 3D finite element solvers for the generalized Stokes problem. Int J Numer Methods Fluids 8:869–895
24. Chorin AJ (1967) A numerical method for solving incompressible viscous problems. J Comput Phys 2:12–26
25. Chorin AJ (1967) The numerical solution of the Navier-Stokes equations for an incompressible fluid. AEC Research and Development Report, NYO-1480-82. New York University, New York, 1967
26. Chorin AJ (1968) Numerical solution of the Navier-Stokes equations. Math Comput 22:745–762
27. Chorin AJ (1969) On the convergence of discrete approximation to the Navier-Stokes equations. Math Comput, 23
28. Ciarlet PG (1978) The finite element method for elliptic problems. North-Holland, Amsterdam
29. Clay Mathematics Institute (2000) http://www.claymath.org/millenium/. Millennium Problems

30. Codina R (1998) Comparison of some finite element methods for solving the dif fusion-convection-reaction equation. Comput Methods Appl Mech Eng 156:185–210
31. Codina R (2000) Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. Comput Methods Appl Mech Eng 190:1579–1599
32. Codina R (2001) Pressure stability in fractional step finite element methods for incompressible flows. J Comput Phys 170:112–140
33. Codina R (2002) Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. Comput Methods Appl Mech Eng 191:4295–4321
34. Codina R, Badia S (2004) Second order fractional step schemes for the incompressible Navier-Stokes equations. Inherent pressure stability and pressure stabilization. In: Proceedings of WCCM VI, Beijing, China, 2004
35. Codina R, Badia S (2006) On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. Comput Methods Appl Mech Eng 195:2900–2918
36. Codina R, Blasco J (1997) A finite element formulation for the Stokes problem allowing equal velocity-pressure interpolation. Comput Methods Appl Mech Eng 143:373–391
37. Codina R, Blasco J (2000) Analysis of a pressure-stabilized finite element approximation of the stationary Navier-Stokes equations. Numer Math 87:59–81
38. Codina R, Blasco J (2002) Analysis of a stabilized finite element approximation of the transient convection-diffusion-reaction equation using orthogonal subscales. Comput Vis Sci 4:167–174
39. Codina R, Folch A (2004) A stabilized finite element predictor–corrector scheme for the incompressible Navier-Stokes equations using a nodal based implementation. Int J Numer Methods Fluids 44:483–503
40. Codina R, Principe J, Guasch O, Badia S (2007) Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. Comput Methods Appl Mech Eng 196:24132–2430
41. Codina R, Soto O (2004) Approximation of the incompressible Navier-Stokes equations using orthogonal-subscale stabilization and pressure segregation on anisotropic finite element meshes. Comput Methods Appl Mech Eng 193:1403–1419
42. de Rham G (1973) Variétés différentiables formes, courants, formes harmoniques. Hermann, Paris
43. E W, Liu JG (1995) Projection method I: Convergence and numerical boundary layers. SIAM J Numer Anal 32:1017–1057
44. Elman HC (2002) Preconditioners for saddle point problems arising in computational fluid dynamics. Appl Numer Math 43:75–89
45. Elman HC, Howle VE, Shadid JN, Tuminaro RS (2003) A parallel block multi-level preconditioner for the 3D incompressible Navier-Stokes equations. J Comput Phys 187:504–523
46. Elman HC, Silvester DJ, Wathen AJ (2002) Block preconditioners for the discrete incompressible Navier-Stokes equations. Int J Numer Methods Fluids 40:333–344
47. Ern A, Guermond JL (2004) Theory and practice of finite elements. Springer, Berlin
48. Franca L, Stenberg R (1991) Error analysis of some Galerkin least-squares methods for the elasticity equations. SIAM J Numer Anal 28:1680–1697
49. Fried I, Malkus DS (1975) Finite element mass matrix lumping by numerical integration with no convergence rate loss. Int J Solids Struct 11:461–466
50. Gervasio P (2006) Convergence analysis of high order algebraic fractional step schemes for time-dependent Stokes equations. Technical report, Quaderno del Seminario Matematico di Brescia, 2006
51. Gervasio P, Saleri F (2006) Algebraic fractional-step schemes for time-dependent incompressible Navier-Stokes equations. J Sci Comput 27(1–3):257–269
52. Gervasio P, Saleri F, Veneziani A (2006) Algebraic fractional-step schemes with spectral methods for the incompressible Navier-Stokes equations. J Comput Phys 214(1):347–365
53. Girault V, Raviart PA (1986) Finite element methods for Navier-Stokes equations. Springer, Berlin
54. Gresho PM, Sani RL (2000) Incompressible flow and the finite element method. Wiley, New York
55. Gresho PM (1990) On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part I: Theory. Int J Numer Methods Fluids 11:587–620
56. Gresho PM, Chan ST, Christon MA, Hindmarsh AC (1995) A little more on stabilized q_1q_1 for transient viscous incompressible flow. Int J Numer Methods Fluids 21:837–856
57. Guermond JL (1994) Remarques sur les méthodes de projection pour l'approximation des équations de Navier-Stokes. Numer Math 67:465–473
58. Guermond JL, Minev P, Shen J (2006) An overview of projection methods for incompressible flows. Comput Methods Appl Mech Eng 195:6011–6045
59. Guermond JL, Quartapelle L (1998) On stability and convergence of projection methods based on pressure Poisson equation. Int J Numer Methods Fluids 26:1039–1053
60. Guermond JL, Quartapelle L (1998) On the approximation of the unsteady Navier-Stokes equations by finite element projection methods. Numer Math 80:207–238
61. Guermond JL, Shen J (2003) A new class of truly consistent splitting schemes for incompressible flows. J Comput Phys 192:262–276
62. Guermond JL, Shen J (2003) Velocity-correction projection methods for incompressible flows. SIAM J Numer Anal 41:112–134
63. Guermond JL, Shen J (2004) On the error estimates for the rotational pressure-correction projection methods. Math Comput 73:1719–1737
64. Heywood JG, Rannacher R (1982) Finite element approximation of the nonstationary Navier-Stokes problem. I: Regularity of solutions and second-order error estimates for spatial discretization. SIAM J Numer Anal 19:275–311
65. Hinton E, Rock T, Zienkiewicz OC (1976) A note on mass lumping and related processes in the finite element method. Earthquake Eng Struct Dyn 4:245–249
66. Hughes TJR (1995) Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. Comput Methods Appl Mech Eng 127:387–401
67. Hughes TJR, Franca LP, Hulbert GM (1987) A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces. Comput Methods Appl Mech Eng 65:85–96
68. Hughes TJR, Franca LP, Hulbert GM (1989) A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. Comput Methods Appl Mech Eng 73:173–189
69. Hundsdorfer W, Verwer JG (2003) Numerical solution of time-dependent advection-diffusion-reaction equations. Springer, Berlin
70. Karniadakis GE, Israeli M, Orszag SE (1991) High order splitting methods for the incompressible Navier-Stokes equations. J Comput Phys 59:414–443

71. Kim J, Moin P (1985) Application of the fractional step method to incompressible Navier-Stokes equations. J Comput Phys 59:308–323
72. Ladyzhenskaya O (1969) The mathematical theory of viscous incompressible flow. Gordon and Breach, New York
73. Leray J (1934) Essai sur les mouvements d'un liquide visqueux emplissant l'espace. Acta Math 63:193–248
74. Loghin D, Wathen AJ (2002) Schur complement preconditioners for the Navier-Stokes equations. Int J Numer Methods Fluids 40:403–412
75. Oden JT, Demkowicz LZ (1996) Applied functional analysis. CRC Press, Boca Raton
76. Orszag SA, Israeli M, Deville M (1986) Boundary conditions for incompressible flows. J Sci Comput 1:75–111
77. Perot JB (1993) An analysis of the fractional step method. J Comput Phys 108:51–58
78. Prohl A (1997) Projection and quasi-compressibility methods for solving the incompressible Navier-Stokes equations. Teubner, Stuttgart
79. Quarteroni A, Sacchi Landriani G, Valli A (1991) Coupling viscous and inviscid Stokes equations via a domain decomposition method for finite elements. Numer Math 59:831–859
80. Quarteroni A, Saleri F, Veneziani A (1999) Analysis of the Yosida method for the incompressible Navier-Stokes equations. J Math Pures Appl 78:473–503
81. Quarteroni A, Saleri F, Veneziani A (2000) Factorization methods for the numerical approximation of Navier-Stokes equations. Comput Methods Appl Mech Eng 188:505–526
82. Rannacher R (1992) On Chorin's projection method for incompressible Navier-Stokes equations. Lecture notes in mathematics, vol 1530. Springer, Berlin, pp 167–183
83. Saad Y (1996) Iterative methods for sparse linear systems. PWS-Kent, Boston
84. Saleri F, Veneziani A (2005) Pressure correction algebraic splitting methods for the incompressible Navier-Stokes equations. SIAM J Numer Anal 43(1):174–194
85. Shen J (1992) On error estimates for some higher order projection and penalty-projection methods for Navier-Stokes equations. Numer Math 62:49–73
86. Shen J (1992) On error estimates of projection methods for Navier-Stokes equations: first order schemes. SIAM J Numer Anal 29:57–77
87. Shen J (1993) A remark on the projection-3 method. Int J Numer Methods Fluids 16:249–253
88. Shen J (1994) Remarks on the pressure error estimates for the projection methods. Numer Math 67:513–520
89. Shen J (1996) On error estimates of the projection methods for the Navier-Stokes equations: second-order schemes. Math Comput 65:1039–1065
90. Simo JC, Armero F (1994) Unconditional stability and long term behavior of transient algorithms for the incompressible Navier-Stokes equations. Comput Methods Appl Mech Eng 111:111–154
91. Strang G, Fix J (1973) An analysis of the finite element method. Prentice Hall, Englewood Cliffs
92. Temam R (1968) Sur la stabilité et la convergence de la méthode des pas fractionaires. Ann Math Pures Appl LXXIV:191–380
93. Temam R (1968) Une méthode d'approximations de la solution des equations de Navier-Stokes. Bull Soc Math Fr 98:115–152
94. Temam R (1969) Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionaires (I). Arch Ration Mech Anal 32:135–153
95. Temam R (1969) Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionaires (II). Arch Ration Mech Anal 33:377–385
96. Temam R (1984) Navier-Stokes equations. North-Holland, Amsterdam
97. Temam R (1991) Remark on the pressure boundary condition for the projection method. Theor Comput Fluid Dyn 3:181–184
98. Timmermans LJP, Minev PD, Van de Vosse FN (1996) An approximate projection scheme for incompressible flow using spectral elements. Int J Numer Methods Fluids 22:673–688
99. Turek S (1999) Efficient solvers for incompressible flow problems. Lecture notes in computational science and engineering. Springer, Berlin
100. van Kan J (1986) A second-order accurate pressure correction scheme for viscous incompressible flow. SIAM J Sci Stat Comput 7:870–891