Journal of Computational Physics 296 (2015) 113–137

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

First, second and third order fractional step methods for the three-field viscoelastic flow problem

E. Castillo, R. Codina *

Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034, Barcelona, Spain

ARTICLE INFO

Article history:
Received 8 November 2014
Received in revised form 10 April 2015
Accepted 19 April 2015
Available online 28 April 2015

Keywords:
Viscoelastic fluids
Fractional step methods
Stabilized finite element methods

ABSTRACT

In this paper, three different fractional step methods are designed for the three-field viscoelastic flow problem, whose variables are velocity, pressure and elastic stress. The starting point of our methods is the same as for classical pressure segregation algorithms used in the Newtonian incompressible Navier–Stokes problem. These methods can be understood as an inexact *LU* block factorization of the original system matrix of the fully discrete problem and are designed at the pure algebraic level. The final schemes allow one to solve the problem in a fully decoupled form, where each equation (for velocity, pressure and elastic stress) is solved separately. The first order scheme is obtained from a straightforward segregation of pressure and elastic stress in the momentum equation, whereas the key point for the second order scheme is a first order extrapolation of these variables. The third order fractional step method relies on Yosida's scheme. Referring to the spatial discretization, either the Galerkin method or a stabilized finite element formulation can be used. We describe the fractional step methods first assuming the former, and then we explain the modifications introduced by the stabilized formulation we employ and that has been proposed in a previous work. This discretization in space shows very good stability, permitting in particular the use of equal interpolation for all variables.

1. Introduction

The viscoelastic behavior of fluids can be a dominant feature in the flow of polymeric fluids, and in the injection molding flows where moving boundaries are present and the unsteady regime defines the problem [1,2]. Many applications in viscoelastic flows where the dynamics of the fluid are crucial can be found in the industry. Jets of these flows are quite broad, and include areas such as micro-dispensing of bioactive fluids through high throughput injection devices, scaffolds for tissue engineering, ink jet processes or viscoelastic blood flow past valves [3].

The flow patterns in viscoelastic fluids can be highly dynamic and in some cases chaotic, due to the elastic component of the fluid and the convective nature of the constitutive equation, even in quasi non-inertial flows. In [4], the effect of the contraction ratio in the dynamic response of the flow in square–square three-dimensional contractions is analyzed using experimental and numerical results. In [5] the instabilities and the asymmetry of flow in a symmetric domain are analyzed for flows with high Deborah number using the Leonov constitutive equations to characterize the fluid. The turbulent flow of a viscoelastic solution is a new challenge from the numerical approximation perspective, partially due to the reduced friction drag properties of this type of fluids. In [6], the four-field model proposed in [7] is modified in order to model

* Corresponding author.
E-mail addresses: ecastillo@cimne.upc.edu (E. Castillo), ramon.codina@upc.edu (R. Codina).

http://dx.doi.org/10.1016/j.jcp.2015.04.027
0021-9991/

turbulence in the channel flow of dilute polymer solutions. In all the above mentioned cases, the transient simulation is a must and the generation of efficient algorithms to solve the problem, with the unknowns highly coupled, is an important task which motivates in part our work.

The monolithic resolution of the system of equations that is obtained after discretization of the continuous problem is the most straightforward option. However, solving this system is computationally very expensive, specially in 3D. In this case, the degrees of freedom of the spatial discretization are the six independent elastic stress components, the three velocity components and the pressure. Furthermore, all these variables are coupled, some of them through nonlinear terms.

Instead of solving the monolithic system, an alternative is to use a fractional step method in time, in which different equations need to be solved for the different variables in an uncoupled way, perhaps with correction steps. The splitting of the equations introduced in fractional step methods has an additional temporal error, that has to be at least of the order of the time integration scheme used to approximate time derivatives if this order is to be preserved.

Fractional step methods can be either introduced at the continuous or at the purely algebraic levels. For overviews of both strategies in the case of the incompressible Navier–Stokes equations for Newtonian fluids, see [8] and [9], respectively. In this work we will follow the second alternative, and show how to apply it to the viscoelastic flow problem. In this case, it is necessary not only to uncouple the velocity from the pressure in the momentum equation, but also from the elastic stress. Additional equations need to be solved for these variables, followed by correction steps. We will propose three schemes of this type, of first, second and third order of (formal) accuracy in time. The first will consist of four steps, whereas five steps will be required for the second and the third order schemes.

In the viscoelastic flow context, several fractional step schemes have been proposed. One of the most popular is the Θ-method, originally proposed in [10] for Newtonian fluids and later extended to viscoelastic flows in [11,12]. In this method the problem is split in three steps: a Stokes-like system with an explicit stress value, a transport problem involving the constitutive equation and another Stokes problem which enhances the stability of the method. This Θ-method was reported as being second order in time for a proper Θ value. In [13] a priori error estimates for this Θ-method using the Oldroyd-B constitutive model were derived, and theoretical and numerical computations to demonstrate the capability of this scheme were provided. A second order in time two-step decoupling method for modeling inertialess viscoelastic fluids was presented in [14]. In [15] the two-step method was extended to solve fluid flows with inertia. In the first step of this method the elastic stress is computed at the new time level by using the velocity field at the previous time step. In the second step the viscoelastic stress just calculated is used as a force term in the momentum balance equation, which can be regarded together with the continuity equation as a Stokes system. For the two-step method, second order accuracy in time is reported in the previous references, recommending the second order semi-implicit Gear scheme instead of the Crank–Nicolson scheme to discretize the temporal derivatives. In the same fractional step context, in [16] an extension is presented of the fourth step fractional step method proposed in [17] for the standard Navier–Stokes problem, now using the Oldroyd-B constitutive model and the DEVSS-G/DG stabilization technique.

Typical applications in viscoelastic fluid flows where an efficient scheme is needed for the resolution of a large number of degrees of freedoms are, for example, the modeling of a free surface evolution and the sedimentation of particles. The first case is classical for polymeric injections and mold filling [18,19], and the second is found in the investigation of microstructure in flowing suspensions, sedimentation columns, fluidized beds, slurry transport and hydraulic fracturing [20,21]. In [18] a splitting time discretization method for free surface flows is proposed, where the interface of the fluid is solved in addition to the viscoelastic Navier–Stokes problem. The problem is solved in two steps; in the first (prediction) step the three advection problems are solved separately, and in the second (correction) step, a Stokes and constitutive Oldroyd-B problem is solved sequentially. For the case of sedimentation, in [21] an extended version of the method proposed by in [21] is described. The authors use the Marchuk–Yanenko operator splitting scheme to deal with the effect of the viscoelastic stress on the velocity, the incompressibility condition, the advection term, the constitutive equation and the constraint of rigid-body motion inside the particles.

The viscoelastic fluid flow problem presents several numerical difficulties, particularly for the spatial approximation. On the one hand, the finite element approximation used must satisfy two inf–sup conditions: the first between the interpolation of the velocity and of the pressure to control the pressure and the second between the interpolation of the velocity and of the elastic stress to control the symmetrical gradient of the velocity. On the other hand, the convective term in the momentum and in the constitutive equations require to use methods to deal with convection. Many algorithms have been developed to solve this problem: the classical SUPG method and its non-consistent counterpart, the SU method [22], the GLS method [23], and the stabilization based on the discontinuous Galerkin method [24,25], among others in the context of finite element methods. Similarly, the upwind difference scheme (UDS) [26], the second order linear upwinding scheme (LUDS) [27], the sharp and monotonic algorithm for realistic transport (SMART) the and MINMOD method [28] were introduced for the finite volume method.

The highly nonlinear nature of the constitutive equation requires proper linearization strategies, and the possibility of local peaks in the elastic stress components when the amount of elasticity is important needs to be taken into account by methods with some type of monotonicity to avoid local oscillations. A standard numerical treatment to alleviate the nonlinear iterative process is the implementation of continuation methods (see [29] for more details) clearly more used in steady state problems [29,30]. The addition of discontinuity-capturing techniques is another interesting tool, overcoat when the Weissenberg number is increased [30].

The formulation we use for the spatial approximation is a variational multiscale (VMS) stabilized finite element method detailed in [31] (which in turn relies on [30]), and we will only briefly describe it here, since our interest is the time discretization using fractional step schemes. Nevertheless, let us mention that in some cases fractional step methods have been used to obtain stable pressure interpolations, as explained in [32]. We will not consider at all the possible pressure stability provided by the splitting methods we will describe. Other stabilization methods applied to the viscoelastic flow problem can be found in [33,23,34–36], among others.

In this work we present three fractional step methods with first, second and third order splitting errors. The first step in all cases is the momentum equation to calculate an intermediate velocity. For the first order method the intermediate velocity is obtained without any pressure and elastic stress contribution. For the second order scheme, we use extrapolated values of pressure and elastic stresses to obtain a better approximation of the intermediate value of the velocity, and likewise for the third order scheme. In the second step we obtain the intermediate elastic stress values using the intermediate velocity approximation. The third step has the structure of a discrete pressure Poisson equation. The fourth step corresponds to the velocity correction step. For the first and second order schemes this step can be understood as a projection step, whereas for the third order case we use the Yosida approximation [37]. Finally, the fifth step is a projection step to obtain the final elastic stresses. In all cases, when we refer to the order of accuracy of a scheme the arguments we use are completely formal. A rigorous numerical analysis of the formulations presented is outside the scope of this paper.

The work is organized as follows. In Section 2 we introduce the continuous viscoelastic fluid flow problem. Section 3 is devoted to its numerical approximation with the Galerkin approximation and a monolithic time discretization. In Section 4, the fractional step methods we propose are presented for the Galerkin case following two approaches, namely, one based on variable extrapolation and the other on an inexact *LU* decomposition. In Section 5 we describe the stabilized finite element method we use and explain which are the modifications that need to be made to the fractional step schemes introduced earlier. Numerical results are presented in Section 6 and, finally, conclusions are drawn in Section 7.

2. The viscoelastic flow problem

2.1. Initial and boundary value problem

Let us consider a viscoelastic fluid moving in a domain Ω of $\mathbb{R}^d$ $(d = 2, 3)$ during the time interval $[0, t_f]$. Considering the flow incompressible and isothermal, the governing equations are the conservation of momentum and mass, which can be written as

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{T} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega,\ t \in]0, t_f[\tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega,\ t \in]0, t_f[\tag{2}$$

where ρ denotes the constant density, $p : \Omega \times]0, t_f[\to \mathbb{R}$ is the pressure, $\boldsymbol{u} : \Omega \times]0, t_f[\to \mathbb{R}^d$ is the velocity field, $\boldsymbol{f} : \Omega \times]0, t_f[\to \mathbb{R}^d$ is the force vector and $\boldsymbol{T} : \Omega \times]0, t_f[\to \mathbb{R}^d \otimes \mathbb{R}^d$ the deviatoric extra stress tensor, which can be defined in terms of a viscous and a viscoelastic or elastic contribution as

$$\boldsymbol{T} = 2\beta\eta_0 \nabla^s \boldsymbol{u} + \boldsymbol{\sigma} \tag{3}$$

where ∇^s is the symmetrical gradient operator and $\beta \in [0, 1]$ is a parameter to define the amount of viscous or solvent viscosity $\beta\eta_0$ and elastic or polymeric viscosity $(1 - \beta)\,\eta_0$. The constitutive behavior of the fluid needs to be completed with the expression of the elastic part of the extra stress, $\boldsymbol{\sigma}$. A large variety of approaches exist to define it (see [38,39] for a complete description). In this work, only the differential Oldroyd-B model is considered, which reads

$$\frac{\lambda}{2\eta_0}\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{1}{2\eta_0}\boldsymbol{\sigma} - (1-\beta)\,\nabla^s \boldsymbol{u} + \frac{\lambda}{2\eta_0}\left(\boldsymbol{u} \cdot \nabla \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \nabla \boldsymbol{u} - (\nabla \boldsymbol{u})^T \cdot \boldsymbol{\sigma}\right) = \boldsymbol{0} \qquad \text{in } \Omega,\ t \in]0, t_f[\tag{4}$$

where λ is the relaxation time.

Let us introduce some notation. Calling $\boldsymbol{U} = [\boldsymbol{u}, \boldsymbol{\sigma}, p]$, $\boldsymbol{F} = [\boldsymbol{f}, 0, 0]$ and defining

$$\mathcal{L}(\hat{\boldsymbol{u}}; \boldsymbol{U}) := \begin{bmatrix} \rho \hat{\boldsymbol{u}} \cdot \nabla \boldsymbol{u} - 2\beta\eta_0 \nabla \cdot \left(\nabla^s \boldsymbol{u}\right) - \nabla \cdot \boldsymbol{\sigma} + \nabla p \\ \nabla \cdot \boldsymbol{u} \\ \frac{1}{2\eta_0}\boldsymbol{\sigma} - (1-\beta)\,\nabla^s \boldsymbol{u} + \frac{\lambda}{2\eta_0}\left(\hat{\boldsymbol{u}} \cdot \nabla \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \nabla \hat{\boldsymbol{u}} - \left(\nabla \hat{\boldsymbol{u}}\right)^T \cdot \boldsymbol{\sigma}\right) \end{bmatrix}$$

and

$$\mathcal{D}_t(\boldsymbol{U}) := \begin{bmatrix} \rho \frac{\partial \boldsymbol{u}}{\partial t} \\ 0 \\ \frac{\lambda}{2\eta_0}\frac{\partial \boldsymbol{\sigma}}{\partial t} \end{bmatrix}$$

we may write (1), (2) and (4) using definition (3) as

$$\mathcal{D}_t(\boldsymbol{U}) + \mathcal{L}(\boldsymbol{u}; \boldsymbol{U}) = \boldsymbol{F} \tag{5}$$

These equations need to be complemented with initial and boundary conditions to close the problem. The conservation laws (1)–(2) and the Oldroyd-B constitutive equation (4) are a mixed parabolic–hyperbolic problem. The characteristic lines are the streamlines, and the components of the elastic stress tensor $\boldsymbol{\sigma}$ may be considered as quantities conveyed along these characteristics.

For the sake of simplicity in the exposition, we will consider the simplest boundary condition $\boldsymbol{u} = \boldsymbol{0}$ on $\partial\Omega$ for all time, even if in the numerical examples we will leave part of the boundary free, i.e., zero traction will be prescribed there, and a non-homogeneous velocity will be prescribed on a certain part of $\partial\Omega$.

The boundary conditions for the elastic stresses are more delicate. In principle, they do not need to be prescribed, but that can imply an excessively large computational domain to obtain the correct value of these stresses (see [40] for a complete description of the mathematical structure of the problem and the boundary conditions required). Therefore, in numerical applications they are often prescribed. Due to the hyperbolic nature of Eq. (4), in principle they can be fixed only on the inflow part of the boundary, $\Gamma_{\text{in}} = \{\boldsymbol{x} \in \partial\Omega \mid (\boldsymbol{u}\cdot\boldsymbol{n})(\boldsymbol{x}) < 0\}$, where $\boldsymbol{n}$ is the outward unit normal vector to $\partial\Omega$. However, in some numerical works there are examples in which elastic stresses are prescribed not only on Γ_{in}, but on the whole $\partial\Omega$ [41,42]. We will explicitly indicate in our examples where are the elastic stresses prescribed.

The problem is completely defined with the initial conditions for the velocity and the elastic stress, which are of the form $\boldsymbol{u} = \boldsymbol{u}^0$ and $\boldsymbol{\sigma} = \boldsymbol{\sigma}^0$ at time $t = 0$, with $\boldsymbol{u}^0$ and $\boldsymbol{\sigma}^0$ functions defined on the whole domain Ω.

2.2. Variational form

Let us introduce some notation in order to write the weak form of the problem. The space of square integrable functions in a domain ω will be denoted by $L^2(\omega)$ and the space of functions whose distributional derivatives of order up to $m \geq 0$ (integer) belong to $L^2(\omega)$ by $H^m(\omega)$. The space of functions in $H^1(\omega)$ vanishing on $\partial\omega$ will be written as $H^1_0(\omega)$. We shall use the symbol $\langle\cdot,\cdot\rangle_\omega$ to denote the integral of the product of two functions on ω, assuming this is well defined; subscript ω will be omitted when $\omega = \Omega$. The L^2 inner product in Ω (for scalar, vectors and tensors) will be denoted by $(\cdot,\cdot)$.

Let $\boldsymbol{\Upsilon} = H^1(\Omega)^{d\times d}_{\text{sym}}$ (symmetric second order tensors with components in $H^1(\Omega)$), $\boldsymbol{\mathcal{V}} = H^1_0(\Omega)^d$ and $\mathcal{Q} = L^2(\Omega)/\mathbb{R}$, which are the spaces where we may seek the elastic stress, the velocity and the pressure, respectively, for each fixed time t. The weak form of the problem is obtained by testing (5) against an arbitrary test function $\boldsymbol{V}$ with appropriate regularity. It can be written as: find $[\boldsymbol{u}, p, \boldsymbol{\sigma}] :]0, t_{\text{f}}[\longrightarrow \boldsymbol{\mathcal{X}} := \boldsymbol{\mathcal{V}} \times \mathcal{Q} \times \boldsymbol{\Upsilon}$ such that the initial conditions are satisfied and

$$\left(\rho\frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right) + 2\left(\beta\eta_0\nabla^s\boldsymbol{u}, \nabla^s\boldsymbol{v}\right) + \langle\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}\rangle + \left(\boldsymbol{\sigma}, \nabla^s\boldsymbol{v}\right) - (p, \nabla\cdot\boldsymbol{v}) = \langle\boldsymbol{f}, \boldsymbol{v}\rangle \tag{6}$$

$$(q, \nabla\cdot\boldsymbol{u}) = 0 \tag{7}$$

$$\left(\frac{\lambda}{2\eta_0}\frac{\partial\boldsymbol{\sigma}}{\partial t}, \boldsymbol{\tau}\right) + \left(\frac{1}{2\eta_0}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) - \left((1-\beta)\nabla^s\boldsymbol{u}, \boldsymbol{\tau}\right) + \frac{\lambda}{2\eta_0}\left(\boldsymbol{u}\cdot\nabla\boldsymbol{\sigma} - \boldsymbol{\sigma}\cdot\nabla\boldsymbol{u} - (\nabla\boldsymbol{u})^T\cdot\boldsymbol{\sigma}, \boldsymbol{\tau}\right) = 0 \tag{8}$$

for all $\boldsymbol{V} = [\boldsymbol{v}, q, \boldsymbol{\tau}] \in \boldsymbol{\mathcal{X}}$, where it is assumed that $\boldsymbol{f}$ is such that $\langle\boldsymbol{f}, \boldsymbol{v}\rangle$ is well defined.

In a compact form, problem (6)–(8) can be written as:

$$(\mathcal{D}_t(\boldsymbol{U}), \boldsymbol{V}) + B(\boldsymbol{u}; \boldsymbol{U}, \boldsymbol{V}) = \langle\boldsymbol{f}, \boldsymbol{v}\rangle \tag{9}$$

for all $\boldsymbol{V} \in \boldsymbol{\mathcal{X}}$, where

$$\begin{aligned} B(\hat{\boldsymbol{u}}; \boldsymbol{U}, \boldsymbol{V}) = {} & 2\left(\beta\eta_0\nabla^s\boldsymbol{u}, \nabla^s\boldsymbol{v}\right) + \left\langle\rho\hat{\boldsymbol{u}}\cdot\nabla\boldsymbol{u}, \boldsymbol{v}\right\rangle + \left(\boldsymbol{\sigma}, \nabla^s\boldsymbol{v}\right) - (p, \nabla\cdot\boldsymbol{v}) + (q, \nabla\cdot\boldsymbol{u}) \\ & + \left(\frac{1}{2\eta_0}\boldsymbol{\sigma}, \boldsymbol{\tau}\right) - \left((1-\beta)\nabla^s\boldsymbol{u}, \boldsymbol{\tau}\right) + \frac{\lambda}{2\eta_0}\left(\hat{\boldsymbol{u}}\cdot\nabla\boldsymbol{\sigma} - \boldsymbol{\sigma}\cdot\nabla\hat{\boldsymbol{u}} - \left(\nabla\hat{\boldsymbol{u}}\right)^T\cdot\boldsymbol{\sigma}, \boldsymbol{\tau}\right) \end{aligned}$$

3. Numerical approximation

3.1. Galerkin finite element discretization

Let us consider a finite element partition $\mathcal{T}_h$ of the computational domain Ω. The diameter of an element domain $K \in \mathcal{T}_h$ is denoted by h_K and the diameter of the partition is defined as $h = \max\{h_K \mid K \in \mathcal{T}_h\}$. From $\mathcal{T}_h$ we may construct conforming finite element spaces for the velocity, the pressure and the elastic stress, $\boldsymbol{\mathcal{V}}_h \subset \boldsymbol{\mathcal{V}}$, $\mathcal{Q}_h \subset \mathcal{Q}$ and $\boldsymbol{\Upsilon}_h \subset \boldsymbol{\Upsilon}$, respectively. Calling $\boldsymbol{\mathcal{X}}_h := \boldsymbol{\mathcal{V}}_h \times \mathcal{Q}_h \times \boldsymbol{\Upsilon}_h$, the Galerkin finite element approximation of problem (9) consists in finding $\boldsymbol{U}_h :]0, t_{\text{f}}[\longrightarrow \boldsymbol{\mathcal{X}}_h$, such that:

$$(\mathcal{D}_t(\boldsymbol{U}_h), \boldsymbol{V}_h) + B(\boldsymbol{u}_h; \boldsymbol{U}_h, \boldsymbol{V}_h) = \langle\boldsymbol{f}, \boldsymbol{v}_h\rangle \tag{10}$$

for all $\boldsymbol{V}_h = [\boldsymbol{v}_h, q_h, \boldsymbol{\tau}_h] \in \boldsymbol{\mathcal{X}}_h$, and satisfying the appropriate initial conditions.

In principle, we have posed no restrictions on the choice of the finite element spaces. However, there are restrictions that must be satisfied explicitly in the discrete formulation used. These are the same as for the three-field formulation of the Stokes problem (see [43,44] and references therein), and read as follows: there exist positive constants C_1 and C_2 such that

$$\inf_{q_h\in\mathcal{Q}_h}\sup_{\boldsymbol{v}_h\in\boldsymbol{\mathcal{V}}_h}\frac{(q_h,\nabla\cdot\boldsymbol{v}_h)}{\|q_h\|_{\mathcal{Q}_h}\|\boldsymbol{v}_h\|_{\boldsymbol{\mathcal{V}}_h}}\geq C_1,\quad \inf_{\boldsymbol{v}_h\in\boldsymbol{\mathcal{V}}_h}\sup_{\boldsymbol{\tau}_h\in\boldsymbol{\Upsilon}_h}\frac{\left(\boldsymbol{\tau}_h,\nabla^s\boldsymbol{v}_h\right)}{\|\boldsymbol{\tau}_h\|_{\boldsymbol{\Upsilon}_h}\|\boldsymbol{v}_h\|_{\boldsymbol{\mathcal{V}}_h}}\geq C_2 \tag{11}$$

where $\|\cdot\|_{\mathcal{Y}}$ denotes the norm in space $\mathcal{Y}$. Finite element interpolations that satisfy these conditions are scarce (see [45] for a 2D example and [46] for the 3D case). However, for most of our exposition we will assume that they hold, and thus that the Galerkin formulation is stable. We will explain in Section 5 the stabilized finite element formulation we favor and that allows one to use arbitrary interpolations.

3.2. Monolithic time discretization

Let us explain how to discretize in time problem (10) using a monolithic approach, i.e., solving for all the components of $\boldsymbol{U}_h$ at the same time. Any time discretization is possible, although for the sake of conciseness we will restrict ourselves to the classical backward-difference (BDF) approximations. The fractional step formulations to be presented will be based on this time discretization.

Let us consider a partition of the time interval $]0,t_f[$ into time steps of size δt, for simplicity constant. If $t^n=n\delta t$, $n=0,1,2,\ldots$, the approximation of a time dependent function $g(t)$ at t^n will be denoted by g^n. The BDF approximation to the time derivative of g of order $k=1,2,\ldots$, is given by $\frac{\delta_k g^{n+1}}{\delta t}$, which we also write $\frac{\delta_k}{\delta t}g^{n+1}$, where

$$\delta_k g^{n+1}=\frac{1}{\gamma_k}\left(g^{n+1}-\sum_{i=0}^{k-1}\varphi_k^i g^{n-i}\right) \tag{12}$$

where γ_k and φ_k^i are parameters. In particular, for the cases $k=1$, 2 and 3 we have:

$$\begin{aligned}
\delta_1 g^{n+1}&\equiv\delta g^{n+1}=g^{n+1}-g^n\\
\delta_2 g^{n+1}&=\frac{3}{2}\left(g^{n+1}-\frac{4}{3}g^n+\frac{1}{3}g^{n-1}\right)\\
\delta_3 g^{n+1}&=\frac{11}{6}\left(g^{n+1}-\frac{18}{11}g^n+\frac{9}{11}g^{n-1}-\frac{2}{11}g^{n-2}\right)
\end{aligned}$$

We will also use the extrapolation operators of order k, defined as $\widehat{g}_k^{n+1}=g^{n+1}+\mathcal{O}(\delta t^k)$, which for $k=1$, 2 and 3 are given by

$$\widehat{g}_1^{n+1}=g^n \tag{13}$$

$$\widehat{g}_2^{n+1}=2g^n-g^{n-1} \tag{14}$$

$$\widehat{g}_3^{n+1}=3g^n-3g^{n-1}+g^{n-2} \tag{15}$$

Using BDF schemes, the time discretization of problem (10) can be written, in expanded form, as: for $n=1,2,\ldots$, find $[\boldsymbol{u}_h^{n+1},p_h^{n+1},\boldsymbol{\sigma}_h^{n+1}]\in\boldsymbol{\mathcal{X}}_h$ such that

$$\left(\rho\frac{\delta_k\boldsymbol{u}_h^{n+1}}{\delta t},\boldsymbol{v}_h\right)+2\left(\beta\eta_0\nabla^s\boldsymbol{u}_h^{n+1},\nabla^s\boldsymbol{v}_h\right)+\left\langle\rho\boldsymbol{u}_h^{n+1}\cdot\nabla\boldsymbol{u}_h^{n+1},\boldsymbol{v}_h\right\rangle$$
$$+\left(\boldsymbol{\sigma}_h^{n+1},\nabla^s\boldsymbol{v}_h\right)-\left(p_h^{n+1},\nabla\cdot\boldsymbol{v}_h\right)=\left\langle\boldsymbol{f}_h^{n+1},\boldsymbol{v}_h\right\rangle \tag{16}$$

$$\left(q_h,\nabla\cdot\boldsymbol{u}_h^{n+1}\right)=0 \tag{17}$$

$$\left(\frac{\lambda}{2\eta_0}\frac{\delta_k\boldsymbol{\sigma}_h^{n+1}}{\delta t},\boldsymbol{\tau}_h\right)+\left(\frac{1}{2\eta_0}\boldsymbol{\sigma}_h^{n+1},\boldsymbol{\tau}_h\right)-\left((1-\beta)\nabla^s\boldsymbol{u}_h^{n+1},\boldsymbol{\tau}_h\right)$$
$$+\frac{\lambda}{2\eta_0}\left(\boldsymbol{u}_h^{n+1}\cdot\nabla\boldsymbol{\sigma}_h^{n+1}-\boldsymbol{\sigma}_h^{n+1}\cdot\nabla\boldsymbol{u}_h^{n+1}-\left(\nabla\boldsymbol{u}_h^{n+1}\right)^T\cdot\boldsymbol{\sigma}_h^{n+1},\boldsymbol{\tau}_h\right)=0 \tag{18}$$

for all $[\boldsymbol{v}_h,q_h,\boldsymbol{\tau}_h]\in\boldsymbol{\mathcal{X}}_h$.

The first order (BDF1) and the second order (BDF2) schemes are $\mathcal{A}$-stable. Higher order BDF methods do not keep this property, a limitation known as the *second Dahlquist barrier*. Nevertheless, BDF3 has a mild $\mathcal{A}$-stability restriction. We will use it in the numerical examples without discussing about its stability. For the time steps used, we have never observed unstable behaviors.

3.3. Algebraic system

Problem (16)–(18) yields an algebraic system for the nodal unknowns of the finite element functions $[\boldsymbol{u}_h^{n+1}, p_h^{n+1}, \boldsymbol{\sigma}_h^{n+1}] \in \boldsymbol{\mathcal{X}}_h$. The arrays of nodal unknowns will be identified by an upright case symbol, boldface for $\boldsymbol{u}_h$ and $\boldsymbol{\sigma}_h$ and light case for p_h. Matrices will be written with light case italic characters. In the case of elastic stresses, the Voight notation will be employed, so that the components of the nodal values of $\boldsymbol{\sigma}_h$ at a node a will be Σ_I^a, I running from 1 to $d_V = 3$ if $d = 2$ and from 1 to $d_V = 6$ if $d = 3$. Having this notation in mind, the algebraic structure of problem (16)–(18) is

$$M_{\boldsymbol{u}} \frac{\delta_k}{\delta t} \mathbf{U}^{n+1} + K_{\boldsymbol{u}} \left(\mathbf{U}^{n+1} \right) \mathbf{U}^{n+1} + G \mathrm{P}^{n+1} - D_{\boldsymbol{\sigma}} \boldsymbol{\Sigma}^{n+1} = \mathbf{F}^{n+1} \tag{19}$$

$$D \mathbf{U}^{n+1} = \mathbf{0} \tag{20}$$

$$M_{\boldsymbol{\sigma}} \frac{\delta_k}{\delta t} \boldsymbol{\Sigma}^{n+1} + K_{\boldsymbol{\sigma}} \left(\mathbf{U}^{n+1} \right) \boldsymbol{\Sigma}^{n+1} - S \mathbf{U}^{n+1} = \mathbf{0} \tag{21}$$

The dependence of matrices $K_{\boldsymbol{u}}$ and $K_{\boldsymbol{\sigma}}$ on $\mathbf{U}$ has been explicitly displayed. The identification of the different terms in (16)–(18) that contribute to the matrices in these expressions is straightforward.

System (19)–(21) can be written as

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & 0 \\ A_{31} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^{n+1} \\ \boldsymbol{\Sigma}^{n+1} \\ \mathrm{P}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1^{n+1} \\ \mathbf{F}_2^{n+1} \\ \mathrm{F}_3^{n+1} \end{bmatrix} \tag{22}$$

where

$$A_{11} = \frac{1}{\gamma_k \delta t} M_{\boldsymbol{u}} + K_{\boldsymbol{u}} \left(\mathbf{U}^{n+1} \right), \qquad A_{12} = -D_{\sigma}, \qquad A_{13} = G$$

$$A_{21} = -S, \qquad A_{22} = \frac{1}{\gamma_k \delta t} M_{\boldsymbol{\sigma}} + K_{\boldsymbol{\sigma}} \left(\mathbf{U}^{n+1} \right), \qquad A_{31} = D$$

$$\mathbf{F}_1 = \mathbf{F}^{n+1} + \frac{1}{\delta t \gamma_k} \left(\sum_{i=0}^{k-1} \varphi_k^i \mathbf{U}^{n-i} \right), \qquad \mathbf{F}_2 = \frac{1}{\delta t \gamma_k} \left(\sum_{i=0}^{k-1} \varphi_k^i \boldsymbol{\Sigma}^{n-i} \right), \qquad \mathrm{F}_3 = 0$$

The ordering of the equations chosen is consistent with the steps of the splitting algorithm proposed below.

4. Fractional step methods

The three fractional step methods proposed next can be viewed as extensions of pressure-segregation or pressure-correction schemes applied to the viscoelastic fluid flow problem. Instead of calculating the velocity with a guess of the pressure only, now we will need a guess of the elastic stresses as well. After computing these two fields, velocities will be corrected. We will not discuss extensions of velocity-correction methods, in which the guess is the velocity to compute the pressure instead of the opposite. Nevertheless, the ideas to be presented could also be used to design velocity-correction-type methods, or to the momentum–pressure Poisson equation formulation of the problem (see [9,47] and references therein).

Our approach here is to present the splitting at the pure algebraic level, as in [48], rather than at the space continuous level as done in the original works of Chorin [49] and Temam [50] and which is still the most common approach [8]. The algebraic viewpoint, discussed in detail in [9], obviates the discussion on pressure boundary conditions.

4.1. A first algebraic point of view: extrapolation

A first possibility to design fractional step methods is to extrapolate the variable that needs to be segregated from a certain equation, and correct the result once this variable has been computed somehow. In our case, we extrapolate the pressure and the stress in the momentum equation to compute a velocity guess, we compute these variables with this velocity guess and finally we correct the velocity.

To derive the methods we propose, let us start writing (19)–(21) in the *equivalent* form

$$M_{\boldsymbol{u}} \frac{\delta_k}{\delta t} \tilde{\mathbf{U}}^{n+1} + K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\mathbf{U}}^{n+1} + G \hat{\mathrm{P}}_{k'-1}^{n+1} - D_{\boldsymbol{\sigma}} \hat{\boldsymbol{\Sigma}}_{k'-1}^{n+1} = \mathbf{F}^{n+1} \tag{23}$$

$$M_{\boldsymbol{u}} \frac{1}{\gamma_k \delta t} (\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + \mathbf{N}_{\boldsymbol{u}}^{n+1} + G(\mathrm{P}^{n+1} - \hat{\mathrm{P}}_{k'-1}^{n+1}) - D_{\boldsymbol{\sigma}} (\boldsymbol{\Sigma}^{n+1} - \hat{\boldsymbol{\Sigma}}_{k'-1}^{n+1}) = \mathbf{0} \tag{24}$$

$$M_{\boldsymbol{\sigma}} \frac{\delta_k}{\delta t} \tilde{\boldsymbol{\Sigma}}^{n+1} + K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\boldsymbol{\Sigma}}^{n+1} - S \tilde{\mathbf{U}}^{n+1} = \mathbf{0} \tag{25}$$

$$M_{\boldsymbol{\sigma}}\frac{1}{\gamma_k\delta t}(\boldsymbol{\Sigma}^{n+1}-\tilde{\boldsymbol{\Sigma}}^{n+1})+\mathbf{N}_{\boldsymbol{\sigma}}^{n+1}-S(\mathbf{U}^{n+1}-\tilde{\mathbf{U}}^{n+1})=\mathbf{0} \tag{26}$$

$$-D\tilde{\mathbf{U}}^{n+1}+\gamma_k\delta t DM_{\boldsymbol{u}}^{-1}\mathbf{N}_{\boldsymbol{u}}^{n+1}+\gamma_k\delta t DM_{\boldsymbol{u}}^{-1}G(\mathrm{P}^{n+1}-\hat{\mathrm{P}}_{k'-1}^{n+1})-\gamma_k\delta t DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^{n+1}-\hat{\boldsymbol{\Sigma}}_{k'-1}^{n+1})=\mathbf{0} \tag{27}$$

where

$$\mathbf{N}_{\boldsymbol{u}}^{n+1}=K_{\boldsymbol{u}}(\mathbf{U}^{n+1})\mathbf{U}^{n+1}-K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\tilde{\mathbf{U}}^{n+1},\quad \mathbf{N}_{\boldsymbol{\sigma}}^{n+1}=K_{\boldsymbol{\sigma}}(\mathbf{U}^{n+1})\boldsymbol{\Sigma}^{n+1}-K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\tilde{\boldsymbol{\Sigma}}^{n+1}$$

In these equations, $\tilde{\mathbf{U}}^{n+1}$ and $\tilde{\boldsymbol{\Sigma}}^{n+1}$ are auxiliary variables, $\hat{g}_{k'-1}^{n+1}$ is the extrapolation of order $k'-1$ at t^{n+1} of a function g (see (13)–(15) for the cases $k'-1=1,2,3$), taking $\hat{g}_0^{n+1}=0$, and it is understood that the difference $\delta_k\tilde{g}^{n+1}$ is computed from $\tilde{g}^{n+1}$ and g^m, for time steps m previous to $n+1$, g now being either $\mathbf{U}$ or $\boldsymbol{\Sigma}$. Note that adding up (23) and (24) we recover (19), adding up (25) and (26) we recover (21), and that (27) is obtained multiplying (24) by $\gamma_k\delta t DM_{\boldsymbol{u}}^{-1}$ and using (20). This last equation (27) is a Poisson-type equation for the pressure. Note also that k gives the order of the time integration, whereas k' determines the order of the variables that are extrapolated.

In the previous equations, k determines the order of the time integration scheme and k' the order of the extrapolations to uncouple variables. They can be chosen independently, but need to be properly balanced to achieve a certain order of the final approximation. In principle, fractional step methods of order k could be designed *taking* $k'=k$ as follows:

1. Compute $\tilde{\mathbf{U}}^{n+1}$ from (23).
2. Compute $\tilde{\boldsymbol{\Sigma}}^{n+1}$ from (25).
3. Compute *an approximation* to P^{n+1} by solving (27) *neglecting* $\mathbf{N}_{\boldsymbol{u}}^{n+1}$ and *replacing* $\boldsymbol{\Sigma}^{n+1}$ by $\tilde{\boldsymbol{\Sigma}}^{n+1}$.
4. Compute an approximation to $\mathbf{U}^{n+1}$ from (24) neglecting $\mathbf{N}_{\boldsymbol{u}}^{n+1}$.
5. Compute an approximation to $\boldsymbol{\Sigma}^{n+1}$ from (26) neglecting $\mathbf{N}_{\boldsymbol{\sigma}}^{n+1}$.

Several remarks are in order:

- Steps 1 to 5 above allow one to uncouple the calculation of the different variables, which is the objective of fractional step methods.
- Matrix $DM_{\boldsymbol{u}}^{-1}G$, appearing in the pressure Poisson equation (27), has a wide stencil. Since it approximates the continuous Laplacian operator, it is often approximated by the matrix resulting from the direct approximation of this Laplacian, with a narrower stencil.
- Formally, it is easy to see that the resulting scheme is of order $\mathcal{O}(\delta t^k)$ for a given spatial discretization (if the spatial discretization is considered, the error will depend on both δt and the mesh size h). Assuming this to be true up to time step n and assuming that the scheme is stable (see next remark), from the approximation to (23)–(27) it follows that it is also true at time step $n+1$. If the order of extrapolation is higher than $k-1$, i.e. $k'>k$, the order of the error would be dominated by the time integration scheme.
- Unfortunately, the resulting scheme is only stable for $k'=1,2$. For $k'=3$, the extrapolation $\hat{\mathrm{P}}_2^{n+1}=2\mathrm{P}^n-\mathrm{P}^{n-1}$ is known to yield an unstable scheme (see [9,51] and references therein).
- For $k=1$ we have an extension to viscoelastic flows of the classical first order fractional step method, whereas for $k=2$ we have an extension of the second order method that keeps the pressure gradient at the previous time step in the momentum equation, sometimes known as the incremental projection method and whose continuous version is analyzed in [52].

The final algorithm for the first and second order schemes is displayed in Algorithm 1, where two remarks need to be made. The first is that for the first order scheme the fifth step is not needed to obtain the first order accuracy in time, and the second is that the sixth step is written only for formal reasons, with the objective of identifying this scheme with the inexact block *LU*-decomposition described in the next subsection.

Since the third order fractional step version ($k=3$) of Algorithm 1 is unstable, a third order scheme needs to be designed by other means. This third order method can be obtained by using only first order extrapolations but *without* neglecting $\mathbf{N}_{\boldsymbol{u}}^{n+1}$ in (24). As we shall see in the next subsection, this can be interpreted as a *Yosida scheme*. The steps are thus as follows:

1. Compute $\tilde{\mathbf{U}}^{n+1}$ from (23) with $k=3$ and $k'=2$.
2. Compute $\tilde{\boldsymbol{\Sigma}}^{n+1}$ from (25) with $k=3$ and $k'=2$.
3. Compute *an approximation* to P^{n+1} by solving (27) *neglecting* $\mathbf{N}_{\boldsymbol{u}}^{n+1}$, *replacing* $\boldsymbol{\Sigma}^{n+1}$ by $\tilde{\boldsymbol{\Sigma}}^{n+1}$ and taking $k=3$ and $k'=2$.
4. Compute an approximation to $\mathbf{U}^{n+1}$ from (24) *without neglecting* $\mathbf{N}_{\boldsymbol{u}}^{n+1}$.
5. Compute an approximation to $\boldsymbol{\Sigma}^{n+1}$ from (26) *neglecting* $\mathbf{N}_{\boldsymbol{\sigma}}^{n+1}$.

The resulting scheme is displayed in Algorithm 2. Even if only a first order extrapolation is used for the pressure and the elastic stresses in the momentum equation, including $\mathbf{N}_u^{n+1}$ in the fourth step allows one to obtain third order accuracy. This

Algorithm 1 First and second order fractional step schemes ($k = 1, 2$).

1. Intermediate velocity using the pressure and the elastic stress values extrapolated:

$$M_{\boldsymbol{u}} \frac{\delta_k}{\delta t} \tilde{\mathbf{U}}^{n+1} + K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\mathbf{U}}^{n+1} + G \hat{\mathrm{P}}_{k-1}^{n+1} - D_{\boldsymbol{\sigma}} \hat{\boldsymbol{\Sigma}}_{k-1}^{n+1} = \mathbf{F}^{n+1} \rightarrow \tilde{\mathbf{U}}^{n+1}$$

2. Intermediate elastic stress values using the intermediate velocity:

$$M_{\boldsymbol{\sigma}} \frac{\delta_k}{\delta t} \tilde{\boldsymbol{\Sigma}}^{n+1} + K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\boldsymbol{\Sigma}}^{n+1} - S \tilde{\mathbf{U}}^{n+1} = \mathbf{0} \rightarrow \tilde{\boldsymbol{\Sigma}}^{n+1}$$

3. Intermediate pressure calculation using the intermediate velocity and elastic stress:

$$-D \tilde{\mathbf{U}}^{n+1} + \gamma_k \delta t D M_{\boldsymbol{u}}^{-1} G(\tilde{\mathrm{P}}^{n+1} - \hat{\mathrm{P}}_{k-1}^{n+1}) - \gamma_k \delta t D M_{\boldsymbol{u}}^{-1} D_{\sigma}(\tilde{\boldsymbol{\Sigma}}^{n+1} - \hat{\boldsymbol{\Sigma}}_{k-1}^{n+1}) = \mathbf{0} \rightarrow \tilde{\mathrm{P}}^{n+1}$$

4. Velocity correction:

$$\frac{1}{\gamma_k \delta t} M_{\boldsymbol{u}}(\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + G(\tilde{\mathrm{P}}^{n+1} - \hat{\mathrm{P}}_{k-1}^{n+1}) - D_{\sigma}(\tilde{\boldsymbol{\Sigma}}^{n+1} - \hat{\boldsymbol{\Sigma}}_{k-1}^{n+1}) = \mathbf{0} \rightarrow \mathbf{U}^{n+1}$$

5. Elastic stress correction:

$$\frac{1}{\gamma_k \delta t} M_{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^{n+1} - \tilde{\boldsymbol{\Sigma}}^{n+1}) - S(\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) = \mathbf{0} \rightarrow \boldsymbol{\Sigma}^{n+1}$$

6. Pressure correction: $\mathrm{P}^{n+1} = \tilde{\mathrm{P}}^{n+1} \rightarrow \mathrm{P}^{n+1}$.

Algorithm 2 Third order fractional step scheme.

1. Intermediate velocity using the pressure and the elastic stress values extrapolated:

$$M_{\boldsymbol{u}} \frac{\delta_3}{\delta t} \tilde{\mathbf{U}}^{n+1} + K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\mathbf{U}}^{n+1} + G\mathrm{P}^n - D_{\boldsymbol{\sigma}} \boldsymbol{\Sigma}^n = \mathbf{F}^{n+1} \rightarrow \tilde{\mathbf{U}}^{n+1}$$

2. Intermediate elastic stress values using the intermediate velocity:

$$M_{\boldsymbol{\sigma}} \frac{\delta_3}{\delta t} \tilde{\boldsymbol{\Sigma}}^{n+1} + K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\boldsymbol{\Sigma}}^{n+1} - S \tilde{\mathbf{U}}^{n+1} = \mathbf{0} \rightarrow \tilde{\boldsymbol{\Sigma}}^{n+1}$$

3. Intermediate pressure calculation using the intermediate velocity and elastic stress:

$$-D \tilde{\mathbf{U}}^{n+1} + \gamma_3 \delta t D M_{\boldsymbol{u}}^{-1} G(\tilde{\mathrm{P}}^{n+1} - \mathrm{P}^n) - \gamma_3 \delta t D M_{\boldsymbol{u}}^{-1} D_{\sigma}(\tilde{\boldsymbol{\Sigma}}^{n+1} - \boldsymbol{\Sigma}^n) = \mathbf{0} \rightarrow \tilde{\mathrm{P}}^{n+1}$$

4. Velocity correction:

$$\frac{1}{\gamma_3 \delta t} M_{\boldsymbol{u}}(\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + K_{\boldsymbol{u}}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \tilde{\mathbf{U}}^{n+1} + G(\tilde{\mathrm{P}}^{n+1} - \mathrm{P}^n) - D_{\sigma}(\tilde{\boldsymbol{\Sigma}}^{n+1} - \boldsymbol{\Sigma}^n) = \mathbf{0} \rightarrow \mathbf{U}^{n+1}$$

5. Elastic stress correction:

$$\frac{1}{\gamma_3 \delta t} M_{\boldsymbol{\sigma}}(\boldsymbol{\Sigma}^{n+1} - \tilde{\boldsymbol{\Sigma}}^{n+1}) - S(\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) = \mathbf{0} \rightarrow \boldsymbol{\Sigma}^{n+1}$$

6. Pressure correction: $\mathrm{P}^{n+1} = \tilde{\mathrm{P}}^{n+1} \rightarrow \mathrm{P}^{n+1}$.

will be clearly seen in the next subsection, but it can also be formally guessed by inspecting Algorithm 2 and comparing the different steps with the exact equations (23)–(27). From the first step it is seen that $\tilde{\mathbf{U}}^{n+1}$ is expected to be a second order approximation to the exact nodal velocities at time step $n+1$, from the second step it then follows that $\tilde{\boldsymbol{\Sigma}}^{n+1}$ is a second order approximation to the exact vector of nodal stresses, and from the fourth step that $\tilde{\mathrm{P}}^{n+1}$ is a second order approximation to the nodal pressures. Then, adding up the first and the fourth steps it is seen that the error in $\tilde{\mathbf{U}}^{n+1}$ does not affect the error in $\mathbf{U}^{n+1}$, which happens to be a third order approximation to the exact nodal velocities. Taking into account $\mathbf{N}_u^{n+1}$ in the fourth step has also some advantages for the first order scheme, particularly regarding boundary conditions when it is considered at the continuous level. This is precisely the formulation whose mathematical analysis is presented in [53].

Let us remark that another alternative to obtain a third order fractional step method is to change the approach to velocity segregation methods [51,54], which permit second order extrapolation of the velocity, thus avoiding the unstable pressure extrapolation. Finally, let us mention that a third order scheme in time only makes sense for a higher order spatial approximation, since error estimates will couple spatial and temporal errors. Nevertheless, both the Galerkin approximation and the stabilized formulation presented later allow one to use any interpolation order in space.

4.2. A second algebraic point of view: inexact factorizations

An interesting approach in fractional step methods is the reinterpretation of the splitting process as an inexact LU block factorization of the matrix of the system after discretization in time and space. High order algebraic pressure segregation algorithms can be designed using this methodology. In [55] for example the authors apply this approach to solve the Navier–Stokes problem using the spectral element method.

The starting point to apply this idea is the original system (22). Let us write it as $A\mathbf{X}^{n+1} = \mathbf{B}^{n+1}$, with the obvious identifications. If we consider the LU-decomposition of A, with L a lower triangular and U an upper triangular matrix, the original system $LU\mathbf{X}^{n+1} = \mathbf{B}^{n+1}$ can be split into two systems that can be solved sequentially:

$$L\tilde{\mathbf{X}}^{n+1} = \mathbf{B}^{n+1}, \quad U\mathbf{X}^{n+1} = \tilde{\mathbf{X}}^{n+1}$$

and each of which is easy to solve by backward or forward substitution. The array $\tilde{\mathbf{X}}^{n+1}$ represents the intermediate value of the unknown. The problem of course is that both L and U involve the inversion of the block matrix A_{11}, which is computationally unaffordable. The idea of inexact factorizations is to approximate A_{11}^{-1}, this yielding approximations to L and U respectively denoted by L^* and U^*. Thus, the matrix of the approximate factorization is $A^* = L^*U^*$, and the error matrix is $E^* = A - A^*$.

The first, second and third order schemes introduced in the previous subsection, with slight modifications of higher order, are recast as inexact LU factorizations in Algorithms 3, 4 and 5. We identify there the approximate matrix A^* and the error matrix E^* of each case.

Algorithm 3 Inexact factorization for the first order fractional step method.

L-steps:

$$\frac{1}{\delta t} M_{\boldsymbol{u}} \tilde{\mathbf{U}}^{n+1} + K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\tilde{\mathbf{U}}^{n+1} = \mathbf{F}_1^{n+1} \to \tilde{\mathbf{U}}^{n+1}$$

$$\frac{1}{\delta t} M_{\boldsymbol{\sigma}} \tilde{\boldsymbol{\Sigma}}^{n+1} + K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\tilde{\boldsymbol{\Sigma}}^{n+1} - S\tilde{\mathbf{U}}^{n+1} = \mathbf{F}_2^{n+1} \to \tilde{\boldsymbol{\Sigma}}^{n+1}$$

$$-D\tilde{\mathbf{U}}^{n+1} + \delta t D M_{\boldsymbol{u}}^{-1} G\tilde{\mathrm{P}}^{n+1} - \delta t D M_{\boldsymbol{u}}^{-1} D_\sigma \tilde{\boldsymbol{\Sigma}}^{n+1} = \mathrm{F}_3^{n+1} \to \tilde{\mathrm{P}}^{n+1}$$

U-steps:

$$\mathrm{P}^{n+1} = \tilde{\mathrm{P}}^{n+1} \to \mathrm{P}^{n+1}$$

$$\boldsymbol{\Sigma}^{n+1} = \tilde{\boldsymbol{\Sigma}}^{n+1} \to \boldsymbol{\Sigma}^{n+1}$$

$$\mathbf{U}^{n+1} + \delta t M_{\boldsymbol{u}}^{-1} G\mathrm{P}^{n+1} - \delta t M_{\boldsymbol{u}}^{-1} D_\sigma \boldsymbol{\Sigma}^{n+1} = \tilde{\mathbf{U}}^{n+1} \to \mathbf{U}^{n+1}$$

L-matrix associated:

$$L^* = \begin{bmatrix} \frac{1}{\delta t} M_{\boldsymbol{u}} + K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) & 0 & 0 \\ -S & \frac{1}{\delta t} M_{\boldsymbol{\sigma}} + K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) & 0 \\ -D & -\delta t D M_{\boldsymbol{u}}^{-1} D_\sigma & \delta t D M_{\boldsymbol{u}}^{-1} G \end{bmatrix}$$

U-matrix associated:

$$U^* = \begin{bmatrix} I & -\delta t M_{\boldsymbol{u}}^{-1} D_\sigma & \delta t M_{\boldsymbol{u}}^{-1} G \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

Approximate resultant matrix:

$$A^* = \begin{bmatrix} \frac{1}{\delta t} M_{\boldsymbol{u}} + K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) & -D_\sigma - \delta t K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) M_{\boldsymbol{u}}^{-1} D_\sigma & G + \delta t K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) M_{\boldsymbol{u}}^{-1} G \\ -S & \frac{1}{\delta t} M_{\boldsymbol{\sigma}} + K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) + \delta t S M_{\boldsymbol{u}}^{-1} D_\sigma & -\delta t S M_{\boldsymbol{u}}^{-1} G \\ -D & 0 & 0 \end{bmatrix}$$

Splitting error matrix:

$$E^* = \begin{bmatrix} 0 & \delta t K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) M_{\boldsymbol{u}}^{-1} D_\sigma & -\delta t K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) M_{\boldsymbol{u}}^{-1} G \\ 0 & -\delta t S M_{\boldsymbol{u}}^{-1} D_\sigma & \delta t S M_{\boldsymbol{u}}^{-1} G \\ 0 & 0 & 0 \end{bmatrix}$$

The splitting error in the first order fractional step method (Algorithm 3) is first order in time and does not affect the continuity equation, i.e., the scheme preserves the mass conservation equation.

For the second and the third order schemes it is more convenient to work with the increments of the variables. Suppose that we know the exact solution of the problem up to time step n and we want to compute an approximation to the solution of system $A\mathbf{X}^{n+1} = \mathbf{B}^{n+1}$, at time step $n+1$. Assume for the moment that the problem is linear and A does not depend on $\mathbf{X}$. Let $A^*\mathbf{X}^{*,n+1} = \mathbf{B}^{*,n+1}$ be the approximate system. We may write both the exact and the approximate systems

Algorithm 4 Inexact factorization for the second order fractional step method.

L-steps:

$$\frac{1}{\gamma_2\delta t}M_{\boldsymbol{u}}\delta\tilde{\mathbf{U}}^{n+1}+K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\delta\tilde{\mathbf{U}}^{n+1}=\mathbf{F}_1^{n+1}-\frac{1}{\gamma_2\delta t}M_{\boldsymbol{u}}\mathbf{U}^n-K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\mathbf{U}^n-G\mathrm{P}^n+D_{\boldsymbol{\sigma}}\boldsymbol{\Sigma}^n\rightarrow\delta\tilde{\mathbf{U}}^{n+1}$$

$$\frac{1}{\gamma_2\delta t}M_{\boldsymbol{\sigma}}\delta\tilde{\boldsymbol{\Sigma}}^{n+1}+K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\delta\tilde{\boldsymbol{\Sigma}}^{n+1}-S\delta\tilde{\mathbf{U}}^{n+1}=\mathbf{F}_2^{n+1}-\frac{1}{\gamma_2\delta t}\boldsymbol{\Sigma}^n-K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\boldsymbol{\Sigma}^n+S\mathbf{U}^n\rightarrow\delta\tilde{\boldsymbol{\Sigma}}^{n+1}$$

$$-D\delta\tilde{\mathbf{U}}^{n+1}+\gamma_2\delta tDM_{\boldsymbol{u}}^{-1}G\delta\tilde{\mathrm{P}}^{n+1}-\gamma_2\delta tDM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\delta\tilde{\boldsymbol{\Sigma}}^{n+1}=\mathrm{F}_3^{n+1}+D\mathbf{U}^n\rightarrow\delta\tilde{\mathrm{P}}^{n+1}$$

U-steps:

$$\delta\mathrm{P}^{n+1}=\delta\tilde{\mathrm{P}}^{n+1}\rightarrow\delta\mathrm{P}^{n+1}$$

$$\left[I-(\gamma_2\delta t)^2\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\right]\delta\boldsymbol{\Sigma}^{n+1}+(\gamma_2\delta t)^2\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\delta\mathrm{P}^{n+1}=\delta\tilde{\boldsymbol{\Sigma}}^{n+1}\rightarrow\delta\boldsymbol{\Sigma}^{n+1}$$

$$\delta\mathbf{U}^{n+1}+\gamma_2\delta tM_{\boldsymbol{u}}^{-1}G\delta\mathrm{P}^{n+1}-\gamma_2\delta tM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\delta\boldsymbol{\Sigma}^{n+1}=\delta\tilde{\mathbf{U}}^{n+1}\rightarrow\delta\mathbf{U}^{n+1}$$

L-matrix associated:

$$L^*=\begin{bmatrix}\frac{1}{\gamma_2\delta t}M_{\boldsymbol{u}}+K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) & 0 & 0\\ -S & \frac{1}{\gamma_2\delta t}M_{\boldsymbol{\sigma}}+K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) & 0\\ -D & -\gamma_2\delta tDM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}} & \gamma_2\delta tDM_{\boldsymbol{u}}^{-1}G\end{bmatrix}$$

U-matrix associated:

$$U^*=\begin{bmatrix}I & -\gamma_2\delta tM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}} & \gamma_2\delta tM_{\boldsymbol{u}}^{-1}G\\ 0 & I-(\gamma_2\delta t)^2\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & (\gamma_2\delta t)^2\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\\ 0 & 0 & I\end{bmatrix}$$

Approximate resultant matrix:

$$A^*=\begin{bmatrix}\frac{1}{\gamma_2\delta t}M_{\boldsymbol{\sigma}}+K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) & -D_{\boldsymbol{\sigma}}-\gamma_2\delta tK_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}} & G+\gamma_2\delta tK_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{u}}^{-1}G\\ -S & A_{22}^* & A_{23}^*\\ -D & A_{32}^* & A_{33}^*\end{bmatrix}$$

with the following components

$$A_{22}^*=\frac{1}{\gamma_2\delta t}M_{\boldsymbol{\sigma}}+K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})-(\gamma_2\delta t)^2K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)$$

$$A_{23}^*=(\gamma_2\delta t)^2K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)$$

$$A_{32}^*=(\gamma_2\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)$$

$$A_{33}^*=-(\gamma_2\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)$$

Splitting error matrix:

$$E^*=\begin{bmatrix}0 & \gamma_2\delta tK_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}} & -\gamma_2\delta tK_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{u}}^{-1}G\\ 0 & (\gamma_2\delta t)^2K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & -(\gamma_2\delta t)^2K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\\ 0 & -(\gamma_2\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & (\gamma_2\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\end{bmatrix}$$

as $A\delta\mathbf{X}^{n+1}=\delta\mathbf{B}^{n+1}:=\mathbf{B}^{n+1}-A\mathbf{X}^n$ and $A^*\delta\mathbf{X}^{*,n+1}=\delta\mathbf{B}^{*,n+1}:=\mathbf{B}^{*,n+1}-A^*\mathbf{X}^n$, respectively, where $\delta\mathbf{X}^{*,n+1}=\mathbf{X}^{*,n+1}-\mathbf{X}^n$. It is easily found that

$$\mathbf{X}^{n+1}-\mathbf{X}^{*,n+1}=(A^*)^{-1}(\delta\mathbf{B}^{n+1}-\delta\mathbf{B}^{*,n+1})-(A^*)^{-1}E^*\delta\mathbf{X}^{n+1}$$

Therefore, if $(A^*)^{-1}$ has a bounded norm (i.e., the approximate scheme is stable), $\delta\mathbf{B}^{n+1}-\delta\mathbf{B}^{*,n+1}=\mathcal{O}(\delta t^k)$ (in an appropriate norm) and $E^*=\mathcal{O}(\delta t^{k-1})$, the scheme can be formally expected to be of order k. In the case the solution up to time step n is not exact, but approximate up to the adequate order, and A depends on $\mathbf{X}$, we may proceed by induction and show (formally) the same result.

For the second and third order ($k=2,3$) fractional step schemes we propose, we will be able to write the problem in incremental form with right-hand sides whose difference with the exact ones will be of order k, assuming this is the order of the error up to time step n. Therefore, if we show that the error matrix E^* is of order $k-1$, the solution at time step $n+1$ will have an error of order k. Again, the dependence of A on the unknown has to be treated by assuming this unknown has a certain order of approximation and then checking it a posteriori. As mentioned earlier, these arguments are merely formal, but serve to understand why the methods have the order we claim and that is observed in numerical experiments.

The inexact factorization point of view of the second and third order schemes is presented in Algorithm 4 and Algorithm 5 respectively. Considering first the second order case, it is seen from Algorithm 4 that the splitting error for the momentum equation is the dominant error. The splitting error for the constitutive equation is formally of third order in

Algorithm 5 Inexact factorization for the third order fractional step method.

L-steps:

$$\frac{1}{\gamma_3\delta t}M_{\boldsymbol{u}}\delta\tilde{\mathbf{U}}^{n+1}+K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\delta\tilde{\mathbf{U}}^{n+1}=\mathbf{F}_1^{n+1}-\frac{1}{\gamma_3\delta t}M_{\boldsymbol{u}}\mathbf{U}^n-K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\mathbf{U}^n-G\mathrm{P}^n+D_{\boldsymbol{\sigma}}\boldsymbol{\Sigma}^n\to\delta\tilde{\mathbf{U}}^{n+1}$$

$$\frac{1}{\gamma_3\delta t}M_{\boldsymbol{\sigma}}\delta\tilde{\boldsymbol{\Sigma}}^{n+1}+K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\delta\tilde{\boldsymbol{\Sigma}}^{n+1}-S\delta\tilde{\mathbf{U}}^{n+1}=\mathbf{F}_2^{n+1}-\frac{1}{\gamma_3\delta t}M_{\boldsymbol{\sigma}}\boldsymbol{\Sigma}^n-K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\boldsymbol{\Sigma}^n+S\mathbf{U}^n\to\delta\tilde{\boldsymbol{\Sigma}}^{n+1}$$

$$-D\delta\tilde{\mathbf{U}}^{n+1}+\gamma_3\delta tDM_{\boldsymbol{u}}^{-1}G\delta\tilde{\mathrm{P}}^{n+1}-\gamma_3\delta tDM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\delta\tilde{\boldsymbol{\Sigma}}^{n+1}=\mathrm{F}_3^{n+1}+D\mathbf{U}^n\to\delta\tilde{\mathrm{P}}^{n+1}$$

U-steps:

$$\delta\mathrm{P}^{n+1}=\delta\tilde{\mathrm{P}}^{n+1}\to\delta\mathrm{P}^{n+1}$$

$$\left[I-(\gamma_3\delta t)^2\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\right]\delta\boldsymbol{\Sigma}^{n+1}+(\gamma_3\delta t)^2\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\delta\mathrm{P}^{n+1}=\delta\tilde{\boldsymbol{\Sigma}}^{n+1}\to\delta\boldsymbol{\Sigma}^{n+1}$$

$$\delta\mathbf{U}^{n+1}+\left(\frac{1}{\gamma_3\delta t}M_{\boldsymbol{u}}+K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\right)^{-1}G\delta\mathrm{P}^{n+1}-\left(\frac{1}{\gamma_3\delta t}M_{\boldsymbol{u}}+K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\right)^{-1}D_{\boldsymbol{\sigma}}\delta\boldsymbol{\Sigma}^{n+1}=\delta\tilde{\mathbf{U}}^{n+1}\to\delta\mathbf{U}^{n+1}$$

L-matrix associated:

$$L^*=\begin{bmatrix}C_{\boldsymbol{u}} & 0 & 0\\ -S & C_{\boldsymbol{\sigma}} & 0\\ -D & -\gamma_3\delta tDM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}} & \gamma_3\delta tDM_{\boldsymbol{u}}^{-1}G\end{bmatrix}$$

$$C_{\boldsymbol{u}}=\frac{1}{\gamma_3\delta t}M_{\boldsymbol{u}}+K_{\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}),\quad C_{\boldsymbol{\sigma}}=\frac{1}{\gamma_3\delta t}M_{\boldsymbol{\sigma}}+K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})$$

U-matrix associated:

$$U^*=\begin{bmatrix}I & -C_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}} & C_{\boldsymbol{u}}^{-1}G\\ 0 & \left[I-(\gamma_3\delta t)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(C_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\right] & (\gamma_3\delta t)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(C_{\boldsymbol{u}}^{-1}G\right)\\ 0 & 0 & I\end{bmatrix}$$

Approximate resultant matrix:

$$A^*=\begin{bmatrix}C_{\boldsymbol{u}} & -D_{\boldsymbol{\sigma}} & G\\ -S & C_{\boldsymbol{\sigma}}-(\gamma_3\delta t)^2\left(K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & (\gamma_3\delta t)^2\left(K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\\ -D & (\gamma_3\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & -(\gamma_3\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\end{bmatrix}$$

Splitting error matrix:

$$E^*=\begin{bmatrix}0 & 0 & 0\\ 0 & (\gamma_3\delta t)^2\left(K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & -(\gamma_3\delta t)^2\left(K_{\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\\ 0 & -(\gamma_3\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right) & (\gamma_3\delta t)^3\left(DM_{\boldsymbol{u}}^{-1}D_{\boldsymbol{\sigma}}\right)\left(M_{\boldsymbol{\sigma}}^{-1}S\right)\left(M_{\boldsymbol{u}}^{-1}G\right)\end{bmatrix}$$

time, and the splitting error in the continuity equation is also of third order, because the inversion of A^* introduces a factor $1/\delta t$ (the matrix multiplying the pressure increments has a factor δt). From this observation it follows that in order to design a third order scheme in fact it is only necessary to improve the accuracy of the splitting of the momentum equation. This is precisely what can be accomplished with the Yosida scheme introduced in the previous subsection. From Algorithm 5 it is observed that in fact in this case there is no error in the momentum equation. That the third order scheme can be interpreted as a Yosida factorization (with an extrapolation of the pressure and the elastic stresses in the momentum equation) comes from the expression of matrices L^* and U^* in Algorithm 5.

Let us finally remark that the second and third order schemes proposed in this subsection are not identical to those proposed in the previous one. The equations for the intermediate variables are exactly the same, and the intermediate pressure is taken in all cases as the end-of-step pressure. The difference is in the order of calculation of the end-of-step velocities and elastic stresses. In Algorithms 1 (for $k=2$) and 2, the end-of-step velocities are first computed and then the elastic stresses are updated, whereas in Algorithm 4 and Algorithm 5 the order is reversed. This change in the order is given by the *LU* decomposition structure. Note that solving for the elastic stresses without knowing the end-of-step velocities implies solving two systems, with matrices $M_{\boldsymbol{u}}$ and $M_{\boldsymbol{\sigma}}$, and is therefore computationally more expensive. Nevertheless, the difference can be shown to be of fourth order by using Maclaurin expansions in terms of δt, and thus the splitting error is the same for the methods of this subsection and of the previous one.

5. Stabilized finite element formulation

Up to this point, we have assumed that the spatial approximation has been carried out using the Galerkin finite element method or, in fact, any spatial discretization that yields the same matrix structure as this method. However, apart from methods to stabilize convection dominated flows, it may be convenient to use formulations that allow one to use arbitrary

interpolations, not necessarily satisfying the inf–sup conditions (11). Let us remark that sometimes pressure stability can be obtained from the pressure splitting, even if that implies a proper selection of the time step size and it cannot be used in methods of order higher than one. See [32] for a discussion about this point and the analysis of pressure stability for first and second order fractional step schemes for Newtonian flows.

In this section we present the stabilized finite element formulation that we use, which is described in detail in [31] for the stationary problem. Here we will only write the resulting discrete equations, and after this we will present the final algebraic structure and the linearization of the problem. The essential difference with respect to (22) is the appearance of a non-zero A_{33} block in matrix A. We will explain how this modifies the fractional step methods presented heretofore.

5.1. Stabilized monolithic formulation

Stabilized finite element methods consist in modifying the discrete Galerkin formulation of the problem by adding terms designed to enhance stability without upsetting accuracy. The method we propose, and that replaces (10), consists in finding $\boldsymbol{U}_h :]0, t_f[\longrightarrow \boldsymbol{\mathcal{X}}_h$, such that:

$$(\mathcal{D}_t(\boldsymbol{U}_h), \boldsymbol{V}_h) + B(\boldsymbol{u}_h; \boldsymbol{U}_h, \boldsymbol{V}_h) + S_1^{\perp}(\boldsymbol{u}_h; \boldsymbol{U}_h, \boldsymbol{V}_h) + S_2^{\perp}(\boldsymbol{U}_h, \boldsymbol{V}_h) + S_3(\boldsymbol{u}_h; \boldsymbol{U}_h, \boldsymbol{V}_h) = \langle \boldsymbol{f}, \boldsymbol{v}_h \rangle \tag{28}$$

for all $\boldsymbol{V}_h = [\boldsymbol{v}_h, q_h, \boldsymbol{\tau}_h] \in \boldsymbol{\mathcal{X}}_h$, and satisfying the appropriate initial conditions.

In order to give the expression of the stabilization terms $S_1^{\perp}$, $S_2^{\perp}$ and S_3 in (28), let P_h denote the L^2 projection onto the appropriate finite element space, either of velocities, elastic stresses or pressures (the case will be clear by the argument of this projection), *without* boundary conditions. Let also $P_h^{\perp} = I - P_h$ be the orthogonal projection, I being now the identity. We then define

$$\begin{aligned} S_1^{\perp}(\hat{\boldsymbol{u}}_h; \boldsymbol{U}_h, \boldsymbol{V}_h) = & \sum_K \alpha_1 \left\langle P_h^{\perp}\left[\rho \hat{\boldsymbol{u}}_h \cdot \nabla \boldsymbol{u}_h\right], \rho \hat{\boldsymbol{u}}_h \cdot \nabla \boldsymbol{v}_h \right\rangle_K + \sum_K \alpha_1 \left\langle P_h^{\perp}\left[\nabla p_h\right], \nabla q_h \right\rangle_K \\ & + (1-\beta) \sum_K \alpha_1 \left\langle P_h^{\perp}\left[\nabla \cdot \boldsymbol{\sigma}_h\right], \nabla \cdot \boldsymbol{\tau}_h \right\rangle_K \end{aligned} \tag{29}$$

$$S_2^{\perp}(\boldsymbol{U}_h, \boldsymbol{V}_h) = \sum_K \alpha_2 \left\langle P_h^{\perp}\left[\nabla \cdot \boldsymbol{u}_h\right], \nabla \cdot \boldsymbol{v}_h \right\rangle_K \tag{30}$$

$$\begin{aligned} S_3(\hat{\boldsymbol{u}}_h; \boldsymbol{U}_h, \boldsymbol{V}_h) = & \sum_K \alpha_3 \left\langle \tilde{P}_h \Big[\frac{\lambda}{2\eta_0} \left(\frac{\partial}{\partial t}\boldsymbol{\sigma}_h + \hat{\boldsymbol{u}}_h \cdot \nabla \boldsymbol{\sigma}_h - \boldsymbol{\sigma}_h \cdot \nabla \hat{\boldsymbol{u}}_h - \left(\nabla \hat{\boldsymbol{u}}_h\right)^T \cdot \boldsymbol{\sigma}_h \right) + \frac{1}{2\eta_0}\boldsymbol{\sigma}_h \right. \\ & \left. - (1-\beta)\nabla^s \boldsymbol{u}_h \Big], -\frac{1}{2\eta_0}\boldsymbol{\tau}_h - \nabla^s \boldsymbol{v}_h + \frac{\lambda}{2\eta_0}\left(\hat{\boldsymbol{u}}_h \cdot \nabla \boldsymbol{\tau}_h - \boldsymbol{\tau}_h \cdot \left(\nabla \hat{\boldsymbol{u}}_h\right)^T - \nabla \hat{\boldsymbol{u}}_h \cdot \boldsymbol{\tau}_h \right) \right\rangle_K \end{aligned} \tag{31}$$

where in (31) the projection $\tilde{P}_h$ can be taken as the identity or the orthogonal projection $P_h^{\perp}$, and the parameters $\alpha_i, i = 1, 2, 3$, are computed within each element K as

$$\alpha_1 = \left[c_1 \frac{\eta_0}{h_1^2} + c_2 \frac{\rho\,|\boldsymbol{u}_h|}{h_2} \right]^{-1}$$

$$\alpha_2 = \frac{h_1^2}{c_1 \alpha_1}$$

$$\alpha_3 = \left[c_3 \frac{1}{2\eta_0} + c_4 \left(\frac{\lambda}{2\eta_0}\frac{|\boldsymbol{u}_h|}{h_2} + \frac{\lambda}{\eta_0}|\nabla \boldsymbol{u}_h| \right) \right]^{-1}$$

In these expressions, h_1 corresponds to a characteristic element length calculated as the square root of the element area in the 2D case and the cubic root of the element volume in 3D, and h_2 corresponds to another characteristic length calculated as the element length in the streamline direction. The term $|\boldsymbol{u}_h|$ is the Euclidean norm of the velocity, and $|\nabla \boldsymbol{u}_h|$ the Frobenius norm of the velocity gradient. The constants c_i, $i = 1, 2, 3, 4$, are algorithmic parameters in the formulation. The values used in this work are $c_1 = 12.0$, $c_2 = 2.0$, $c_3 = 4.0$ and $c_4 = 0.125$. We have found better convergence in the nonlinear iterative scheme described later using the modified values c_1 and c_4, with respect to the original values presented in [31, 30] ($c_1 = 4.0$ and $c_4 = 0.25$), with similar accuracy (the values of the algorithmic constants affect the results, but not the mesh convergence rate to be expected). For higher order elements, the characteristic lengths h_1 and h_2 should be divided respectively by r^2 and r, r being the order of the finite element interpolation.

The stabilizing mechanism introduced by the terms $S_1^{\perp}$, $S_2^{\perp}$ and S_3 are the following. The first component of the $S_1^{\perp}$ gives control on the convective term, the second component gives control on the pressure gradient, and the third term gives control on the divergence of the viscoelastic stress. The term $S_2^{\perp}$ is not a must but in some cases it improves stability of the problem. Finally, the term S_3 ensures stability of the constitutive equation. Note that some of the components of this last term are the convective–convective term of the viscoelastic stress tensor and an equivalent EVSS-structure component,

among others cross local inner-product terms (see [31] for more details of this spatial stabilized formulation). The addition of these three terms permit the resolution of convection dominant problems both in velocity and in stress, and the implementation of equal order interpolation for all the unknowns. The orthogonal projections introduce consistency errors, but of optimal order. For stationary problems, the resulting formulation turns out to have optimal order of convergence, as checked numerically in [31] and proved in some simpler settings, for example in [44]. If r is the order of the finite element interpolation, velocity convergence in $L^2(\Omega)$ turns out to be of order $r+1$, whereas pressure and stress convergence in $L^2(\Omega)$ and velocity convergence in $H^1(\Omega)$ are of order r.

The time discretization of (28) can be performed as for the Galerkin method. Using BDF schemes of order k, partial time derivatives have to be replaced by incremental quotients $\delta_k/\delta t$ and all unknowns need to be evaluated at time step $n+1$. We consider the resulting fully discrete problem in the following.

5.2. Algebraic formulation and fractional step scheme

From Eq. (28) it is easy to see that the algebraic system to be solved at each time step will be of the form

$$\begin{bmatrix} A_{s,11} & A_{s,12} & A_{13} \\ A_{s,21} & A_{s,22} & 0 \\ A_{31} & 0 & A_{s,33} \end{bmatrix} \begin{bmatrix} \mathbf{U}^{n+1} \\ \mathbf{\Sigma}^{n+1} \\ \mathrm{P}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathrm{F}_3 \end{bmatrix} \tag{32}$$

where subscript s has been introduced in the matrices that have contributions from the stabilization terms. Comparing this system with (22), we see that:

- From the point of view of the algebraic structure, the only difference between Eqs. (32) and (22) is the presence of the term $A_{s,33}$, which comes from $\sum_K \alpha_1 \left\langle P_h^{\perp}[\nabla p_h], \nabla q_h \right\rangle_K$. We will write it as $A_{s,33} = L_s^{\perp}$.
- The fact that $A_{s,23} = A_{s,32} = 0$ simplifies the introduction of fractional step methods. The orthogonal projections $P_h^{\perp}$ in (29) and (30) (and maybe also in (31)) allows one to introduce a sort of term-by-term stabilization while maintaining optimal accuracy. This is also why matrices A_{13} and A_{31} of the Galerkin method are unaltered. See [56,57] for the analysis in the Newtonian case.
- The time derivative of the velocity does not appear in the stabilization terms, and therefore the only matrix which is multiplied by $1/\delta t$ in the first row of (32) is $M_{\boldsymbol{u}}$, which is symmetric and positive definite. An analogous comment can be made about the second row if $\tilde{P}_h = P_h^{\perp}$, since the orthogonal projection of the time derivative is zero. On the other hand, if $\tilde{P}_h = I$ there is a contribution of the elastic stress time derivative in the stabilization terms of the constitutive equation, which may deteriorate the behavior of iterative solvers.
- All stabilization terms are nonlinear, as the stabilization parameters depend on the velocity $\boldsymbol{u}_h$.

With these observations in mind, the first, second and third order fractional step methods proposed for the Galerkin method can be easily extended to the stabilized finite element method. The only difference will be to take into account that matrices may have a contribution coming from the stabilization terms and that there will be a new contribution to the discrete equation for the pressure. For example, the third order scheme is given in Algorithm 6. Matrices with a contribution from the stabilization terms have been identified with a subscript s. The resulting matrix L^* is as in Algorithm 5, only adding $L_s^{\perp}$ in the 33-block and accounting for the stabilization contributions. This is also the only difference in matrix A^*, while matrices U^* and E^* are the same as in Algorithm 5, except for the modification in the matrices due to stabilization. This in particular implies that the splitting error will be of the same order in both the Galerkin and the stabilized finite element formulations.

Algorithm 6 Inexact factorization for the third order fractional step method using the stabilized finite element formulation.

L-steps:

$$\frac{1}{\gamma_3 \delta t} M_{\boldsymbol{u}} \delta \tilde{\mathbf{U}}^{n+1} + K_{s,\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \delta \tilde{\mathbf{U}}^{n+1} = \mathbf{F}_1^{n+1} - \frac{1}{\gamma_3 \delta t} M_{\boldsymbol{u}} \mathbf{U}^n - K_{s,\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \mathbf{U}^n - G\mathrm{P}^n + D_{s,\boldsymbol{\sigma}} \mathbf{\Sigma}^n$$

$$\frac{1}{\gamma_3 \delta t} M_{s,\boldsymbol{\sigma}} \delta \tilde{\mathbf{\Sigma}}^{n+1} + K_{s,\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) \delta \tilde{\mathbf{\Sigma}}^{n+1} - S_s \delta \tilde{\mathbf{U}}^{n+1} = \mathbf{F}_2^{n+1} - \frac{1}{\gamma_3 \delta t} M_{s,\boldsymbol{\sigma}} \mathbf{\Sigma}^n - K_{s,\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1}) \mathbf{\Sigma}^n + S_s \mathbf{U}^n$$

$$-D \delta \tilde{\mathbf{U}}^{n+1} + (\gamma_3 \delta t D M_{\boldsymbol{u}}^{-1} G + L_s^{\perp}) \delta \tilde{\mathrm{P}}^{n+1} - \gamma_3 \delta t D M_{\boldsymbol{u}}^{-1} D_{s,\boldsymbol{\sigma}} \delta \tilde{\mathbf{\Sigma}}^{n+1} = \mathrm{F}_3^{n+1} + D\mathbf{U}^n - L_s^{\perp} \mathrm{P}^n$$

U-steps:

$$\delta \mathrm{P}^{n+1} = \delta \tilde{\mathrm{P}}^{n+1}$$

$$\left[I - (\gamma_3 \delta t)^2 \left(M_{s,\boldsymbol{\sigma}}^{-1} S_s \right) \left(M_{\boldsymbol{u}}^{-1} D_{s,\boldsymbol{\sigma}} \right) \right] \delta \mathbf{\Sigma}^{n+1} + (\gamma_3 \delta t)^2 \left(M_{s,\boldsymbol{\sigma}}^{-1} S_s \right) \left(M_{\boldsymbol{u}}^{-1} G \right) \delta \mathrm{P}^{n+1} = \delta \tilde{\mathbf{\Sigma}}^{n+1}$$

$$\delta \mathbf{U}^{n+1} + \left(\frac{1}{\gamma_3 \delta t} M_{\boldsymbol{u}} + K_{s,\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \right)^{-1} G \delta \mathrm{P}^{n+1} - \left(\frac{1}{\gamma_3 \delta t} M_{\boldsymbol{u}} + K_{s,\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1}) \right)^{-1} D_{s,\boldsymbol{\sigma}} \delta \mathbf{\Sigma}^{n+1} = \delta \tilde{\mathbf{U}}^{n+1}$$

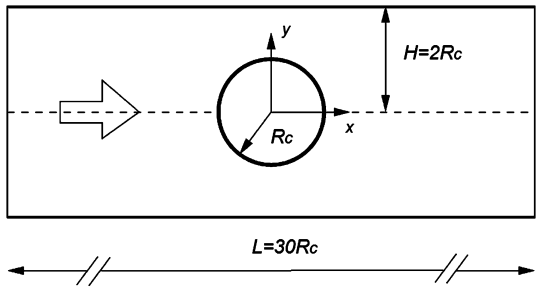


Fig. 1. Schematic representation of the flow past a cylinder.

5.3. Linearized problem

The equations to be solved in the methods proposed are nonlinear and an iterative scheme is required to deal with this nonlinearity. In the case of the stabilized finite element formulation, it is also important to explain how to treat the orthogonal projections $P_h^\perp$. Let us consider for example the third order method in Algorithm 6. The three ingredients we use to design an iterative scheme are the following:

- A fixed-point-type method is used to evaluate all the stabilization terms. They need to be calculated with a known velocity $\boldsymbol{u}_h$, which may taken either as the velocity of the previous iteration of the iterative scheme (the intermediate or the end-of-step one) or as the velocity of the previous time step. This does not affect the order of convergence of the formulation, neither in space nor in time, but only the stability and the nonlinear convergence behavior. We have found effective both choices, and used the former in the numerical examples.
- The nonlinearity in $K_{s,\boldsymbol{u}}(\tilde{\mathbf{U}}^{n+1})\delta\tilde{\mathbf{U}}^{n+1}$ due to the convective term may be treated using either a fixed-point or a Newton–Raphson method. As in any fractional step scheme, δt cannot be taken very large for the method to be effective, compared for example to the critical time step of an explicit time integration scheme. Thus, only a few nonlinear iterations are required and the fixed-point option is usually enough. Note that the term $K_{s,\boldsymbol{\sigma}}(\tilde{\mathbf{U}}^{n+1})\delta\tilde{\boldsymbol{\Sigma}}^{n+1}$ does not need to be linearized, as $\tilde{\mathbf{U}}^{n+1}$ is already known when $\delta\tilde{\boldsymbol{\Sigma}}^{n+1}$ needs to be computed.
- The treatment of $P_h^\perp$ deserves a special comment. For any function g, we may compute $P_h^\perp(g)$ as $P_h^\perp(g) = g - P_h(g)$. Matrices resulting from the orthogonal projection of the unknowns have a wider stencil, when compared to that of the Galerkin method. In order to avoid this, at the i-th iteration of the n-th time step we may approximate $P_h^\perp(g^{n,i}) \approx g^{n,i} - P_h(g^{n,i-1})$ or $P_h^\perp(g^{n,i}) \approx g^{n,i} - P_h(g^{n-1})$. Numerical experiments show that both options are effective, the former being chosen in the numerical examples presented next.

6. Numerical results

In this section, some numerical tests are conducted to show the numerical performance of the proposed fractional step methods, and to compare them with the monolithic approach. The first example (Section 6.1) is a convergence test with a manufactured solution to check the time integration errors associated with each of the fractional steps algorithms. The second example (Section 6.2) is the typical flow over a confined cylinder, which serves to show that the proposed methods converge accurately to a steady state solution. Finally, in Section 6.3 we present the numerical results of a dynamic lid-driven cavity flow problem in 2D and 3D, and we discuss the efficiency of the stabilized fractional step methods. In all cases, either linear (P_1) or multilinear (Q_1) elements are used, and the same continuous interpolation is employed for all variables. All the results are obtained using the proposed fractional step methods and the stabilized formulation presented in Section 5.

The geometry of the viscoelastic fluid flow past a confined cylinder in a channel problem is shown in Fig. 1, whereas the unstructured mesh used to discretize this domain with linear triangles is shown in Fig. 2. For the lid-driven cavity problem, the geometry and the mesh, formed by bi-linear elements in 2D and by tri-linear elements in 3D, are presented in Fig. 3 for the 2D case and in Fig. 4 for the 3D case.

In all the numerical examples the discrete and linearized problem is solved by using an iterative solver based on the BiCGStab (Stabilized version of BiConjugate Gradient Squared) method of van der Vorst [58], with the additive Schwarz preconditioner. The version of the second and third order splitting schemes used is the one presented in Algorithms 1 (for $k = 2$) and 2.

6.1. Convergence test

In this first example we consider a simple convergence test whose goal is to check numerically the rate of convergence in time for the three proposed fractional step algorithms. We recall that as time integration scheme we use backward differences, of the same order as the fractional step to be tested.

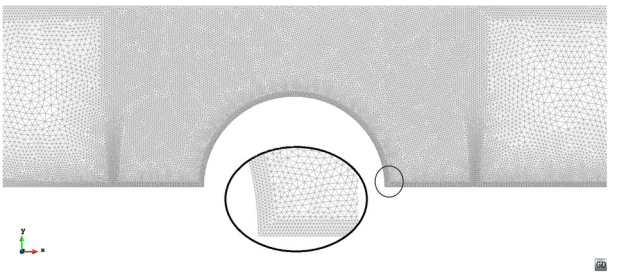


Fig. 2. Computational mesh (M2) used in the flow past a cylinder.

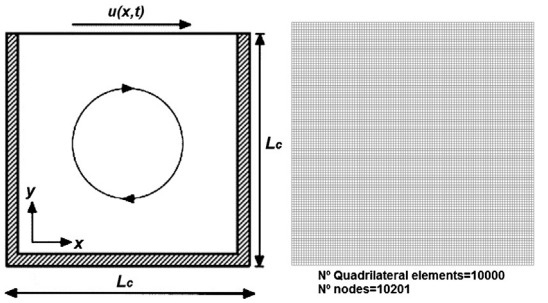


Fig. 3. Schematic representation of the 2D lid-driven cavity flow problem.

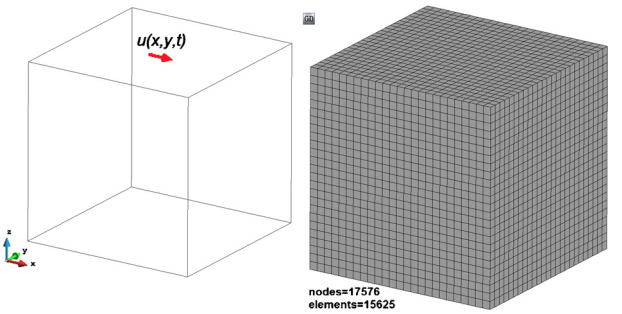


Fig. 4. Schematic representation of the 3D lid-driven cavity flow problem.

The computational domain is the unit square, discretized using a uniform triangular mesh with 2500 linear elements. The boundary and initial conditions and the force term are prescribed so that the analytical solution is given by:

$$u_x(x,y) = (4x+6)\,f(t)\,, \quad u_y(x,y) = -(4y-6)\,f(t)$$
$$\sigma_{xx}(x,y) = (2x+3)\,f(t)\,, \quad \sigma_{xy}(x,y) = (x+y)\,f(t)\,, \quad \sigma_{yy}(x,y) = (2y+3)\,f(t)$$
$$p(x,y) = x$$

with $f(t) = \cos(4\pi t)\exp(-t)$ and subscripts x and y referring to the Cartesian axes. Obviously these velocity and elastic stress fields do not satisfy the constitutive equation (4) with a zero right-hand side. We have added a forcing term in this equation given by

$$\boldsymbol{f}_c = \frac{\lambda}{2\eta_0}\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{1}{2\eta_0}\boldsymbol{\sigma} - (1-\beta)\,\nabla^s \boldsymbol{u} + \frac{\lambda}{2\eta_0}\left(\boldsymbol{u}\cdot\nabla\boldsymbol{\sigma} - \boldsymbol{\sigma}\cdot\nabla\boldsymbol{u} - (\nabla\boldsymbol{u})^T\cdot\boldsymbol{\sigma}\right) \text{ in } \Omega$$

Similarly, the force term in the momentum equation is computed so that the solution is the one given above.

Note that for each time t the exact solution belongs to the finite element space, and thus the only source of numerical error is the time discretization, avoiding this way the possibility of a mix of space and time errors. The stabilized finite element formulation used in the work was already tested in [31] for the stationary problem using linear and quadratic elements, and it is not the objective of this paper to test it again, but to check the performance of the fractional step schemes we propose.

Results are shown in Fig. 5. The error E is measured in two different norms, $\ell^\infty(L^2(\Omega))$ (maximum of the sequence of spatial L^2-norms of the solution) and $\ell^2(H^1(\Omega))$ (ℓ^2-norm of the sequence of spatial H^1-norms of the solutions). It is seen that the three methods show the expected rate of convergence. In all cases shown in Fig. 5 we use the same

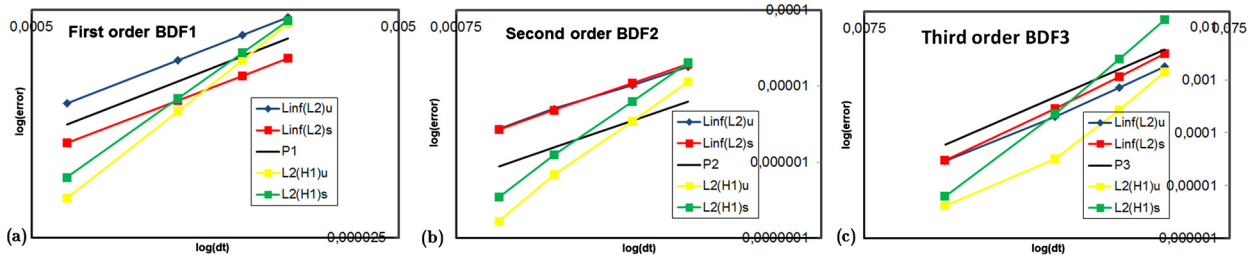


Fig. 5. Converge test: (a) first order scheme, (b) second order scheme and (c) third order scheme.

nomenclature: Linf(L2)z corresponds to the $\ell^\infty(L^2(\Omega))$-error, where z represents the measured variable, L2(H1)z corresponds to the $\ell^2(H^1(\Omega))$-error and P_i, $i = 1, 2, 3$, is the reference line with slope equal to i in logarithmic scale.

6.2. Viscoelastic fluid flow past a confined cylinder in a channel

The flow over a cylinder is a classical benchmark to prove the accuracy of new formulations in viscoelastic fluid flows. We use it here to check the behavior of the proposed fractional step methods to reach a steady state. Note that this steady state has also an error of the order of the scheme itself, it is not independent of the time step size δt.

Let us describe the boundary conditions of the problem. For the velocity, no-slip conditions are imposed on the top wall $y = H$, and symmetry conditions are prescribed along the axis $y = 0$, which in this case means that the y-velocity is set to zero. A fully developed parabolic velocity profile and the associated elastic stress profile are prescribed at the inlet $y = 0$. These are given by

$$u_x = \frac{3Q}{2R_c}\left(1 - \frac{y^2}{(2R_c)^2}\right), \qquad u_y = 0$$

$$\sigma_{xx} = 2\lambda\,(1-\beta)\,\eta_0\left(\frac{\partial u_x}{\partial y}\right)^2, \quad \sigma_{xy} = (1-\beta)\,\eta_0\left(\frac{\partial u_x}{\partial y}\right), \quad \sigma_{yy} = 0$$

where Q is the flow rate, taken as $Q = 1$, $\eta_0 = 1$, $\beta = 0.59$ and R_c is the cylinder radius. For the outlet, the horizontal velocity is left free, the vertical velocity is taken equal to zero and the pressure is prescribed to zero, constant. The boundary term that comes from integration by parts of the divergence of the elastic stress, and that now is not zero on the outlet because the test function is not zero there, is evaluated using the elastic stresses of the time step previous to the one in which the solution is computed.

As initial condition, we start with a Newtonian ($\lambda = 0$) stationary solution, and then we solve the problem for different increasing Weissenberg numbers $\mathrm{We} = \lambda U/L_0$ (U characteristic velocity, taken as $U = 3Q/2R_c$, L_0 characteristic length of the problem, taken as $L_0 = R_c$). The initial condition for one case is taken as the stationary solution of the previous one. In all the calculations, the convective term in the momentum equation is neglected, the standard option in this benchmark problem.

In viscoelastic fluids the high Weissenberg number problem (HWNP) has been the major stumbling block in computational rheology for the last decades. The term HWNP refers to the empirical observation that all numerical methods break down when the Weissenberg number exceeds a critical value. The precise critical value at which computations break down varies with the problem (including the constitutive model), the method and the mesh used [59].

The viscoelastic problem can be solved using the standard constitutive equation given by (4) or a log-conformation formulation, permitting the latter to solve fluids with higher elasticity. In this work we use the former, but the log-conformation formulation could also be used with the fractional step schemes we have presented. Using (4), in the literature one finds important differences in the obtention of drag coefficient values in fluids with $\mathrm{We} > 0.6$. In [60] the authors solve the flow over a cylinder problem with the simplest upwind differencing scheme (UDS) and with high order resolution schemes (SMART and MINMOD) to evaluate the elastic stress derivatives in the constitutive equation. A finite volume formulation is employed in that work. The authors find that the UDS fails to obtain converged solutions for $\mathrm{We} > 0.43$ on fine meshes, whereas they could go up to $\mathrm{We} = 1.5$ using "moderate" meshes (mesh M60 in that article). Using the high resolution schemes the authors found more accuracy, but a lower range of working values of Weissenberg numbers ($\mathrm{We} \leq 1.0$). In general, what can be found in the literature is that standard formulations can solve the problem for We up to a value between 0.7 and 1.0, depending on the reference [60,44,33], whereas the log-conformation formulation allows one to obtain solutions for We up to a value between 2.0 and 2.5, depending on the source [61,59,62]. In all cases mentioned, the Oldroyd-B constitutive model is used and the width of the channel is twice the radius of the cylinder. Using our formulations, we have been able to solve the problem up to $\mathrm{We} = 1.3$, both using the second order and the third order fractional step schemes we have proposed.

We have used three meshes in order to check the independence of the results with respect to the spatial approximation. The number of elements and nodes and the minimum element size $h_{\min}$ of these meshes are detailed in Table 1. All the results to be presented have been computed with mesh M2, unless otherwise stated.

Table 1
Meshes used in the flow over a cylinder problem.

Mesh	Nodes	Elements	h_{min}
M1	21 330	41 514	0.025
M2	29 152	56 905	0.015
M3	35 167	67 856	0.005

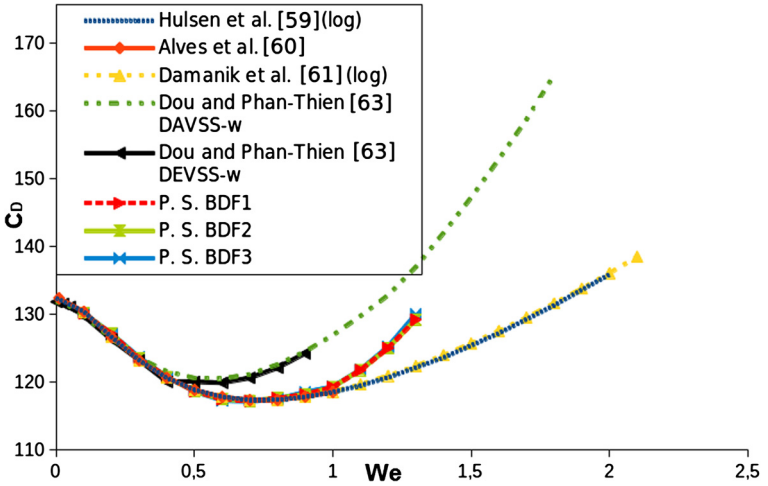


Fig. 6. Drag coefficient validation.

Table 2
Drag force coefficient for the flow over a confined cylinder.

We	[60]	[59] (log)	[62]	[62] (log)	P. S. (M2) (BDF2)	P. S. (M2) (BDF3)
0.1	130.355	130.363	–	–	130.1	130.128
0.2	126.632	126.626	–	–	127.01	127.003
0.3	123.210	123.193	–	–	123.48	123.437
0.4	120.607	120.596	–	–	120.74	120.742
0.5	118.838	118.836	118.821	118.818	118.626	118.598
0.6	117.797	117.792	117.776	117.774	117.438	117.446
0.7	117.323	117.340	117.324	117.323	117.222	117.229
0.8	117.357	117.373	117.370	117.364	117.62	117.504
0.9	117.851	117.787	not solved	117.817	118.15	118.28
1.0	118.518	118.501	not solved	118.680	119.25	119.27
1.1	not solved	119.466	not solved	119.780	122.07	121.8
1.2	not solved	120.650	not solved	121.051	125.02	125.2
1.3	not solved	–	not solved	–	129.2	130.02
1.4	not solved	123.801	not solved	124.092	not solved	not solved

In Fig. 6, the drag coefficients obtained with our formulations are compared with the values that can be found in the literature [59–61,63]. The agreement is good up to We = 1.0, with differences lower than 1% with respect to the accepted values of reference, both for the second and for the third order fractional step methods. For more elastic cases the differences with respect to the log-conformation approach begin to increase, which is possibly associated to the HWNP. The results obtained with meshes M1 and M2 are very close.

Table 2 presents some published drag coefficients and the values obtained in the present work with the second and third order fractional step schemes. It is important to note that in some cases the references use the log-conformation elastic stress tensor formulation, and are identified with the label "log". In [62] the authors use the standard and the log-conformation formulation to solve the same problem using a finite volume method. With our formulations the problem can be solved with the same accuracy as the high order finite volume methods (SMART and MINMOD), but with the possibility to solve more elastic cases. However, for We > 1.0 the accuracy could be questioned, as we have seen.

In terms of the elastic stress convergence for We $\geq$ 0.7, significant discrepancies are encountered among the results found in the literature (see [62] and references therein), especially in the maximum peak of normal elastic stresses at the rear of the wake. In Fig. 7, we show a comparison of the elastic stress profile of the component σ_{xx} along the cylinder wall and downstream the plane $y = 0$ that we obtained against other published results (once again, results obtained with the log-conformation formulation are labeled "log"). For the We = 0.6 case the agreement with the reported values is very good independently of the used mesh. For the other two cases, the results of the mesh M3 are also presented, showing a

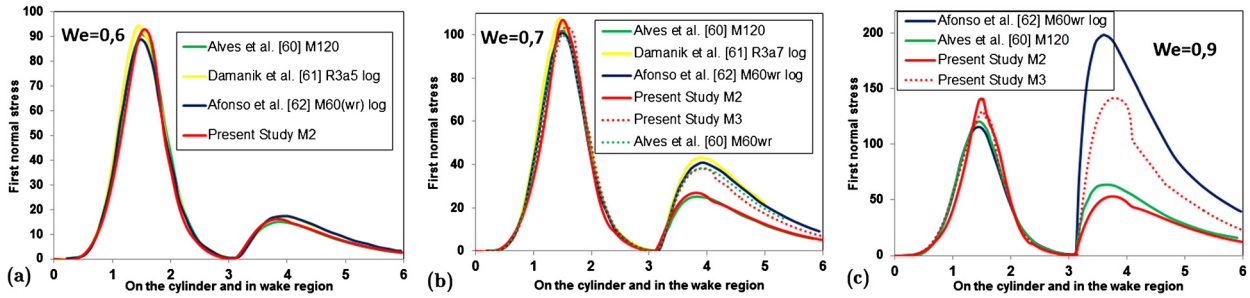


Fig. 7. Stress profile along the cylinder wall and downstream the plane $y = 0$ for: (a) We = 0.6, (b) We = 0.7 and (c) We = 0.9.

better approximation to the log-conformation formulations, especially in the wake region of the cylinder. A similar behavior was reported by Alves et al. in [60] using an extremely refined mesh behind the cylinder. Is important to note that the drag coefficients presented above are practically identical using the mesh M2 or the mesh M3. In [59] the authors do not find any sign of convergence of the elastic stress in the wake beyond the cylinder as the mesh is refined, even using the log-conformation formulation and the Oldroyd-B model for We ≈ 1.0. The authors relate this phenomenon to the behavior of the constitutive model, which models an unlimited extension of the fluid at finite extension rates.

Fig. 8 shows the contour lines for the We = 1.3 case of velocity, pressure and elastic stress. The convective nature of the constitutive equation can be appreciated clearly. Furthermore, no oscillations appear, neither in pressure nor in elastic stress components, which are the most difficult results to obtain in this problem.

6.3. Lid-driven cavity flow problem

6.3.1. 2D cavity

A viscoelastic fluid can exhibit quite different flow behavior from a Newtonian one in many aspects, such as rod climbing, siphoning, and secondary flows. The elastic stresses of the viscoelastic fluid depend not only on the actual stresses, but also on all the stresses to which they have been subjected during their previous deformation history. The lid-driven cavity flow is a clear example of the differences that can be generated by the viscoelastic contribution in the fluid, even in non-inertial (Stokes) flows.

In the following example, the fluid is confined in the unit square $[0, 1]^2$, bounded by solid walls, with the top boundary moving to the x^+ axis. For Newtonian fluids, a discontinuity of the flow field at the upper corners causes the pressure to diverge, without affecting the well-posedness of the problem. A viscoelastic fluid cannot sustain deformations at a stagnation point, and therefore the motion of the lid needs to be smooth, so that $\nabla \boldsymbol{u}$ vanishes at the corners [64]. To satisfy this condition, we use the following horizontal velocity profile at the lid of the cavity ($y = 1$) as boundary condition for the velocity [64,65]:

$$u_x(x, 1, t) = 8\,[1 + \tanh(8\,(t - 0.5))]\,x^2\,(1 - x)^2$$

Function $1 + \tanh(8(t - 0.5))$ has a smooth transition, being almost zero at $t = 0$ and growing rapidly up to two. The lid velocity attains its maximum ($u_x = 1$) at the center of the wall ($x = 0.5$). The inflow boundary conditions for the elastic stress tensor are not needed since there is no inflow boundary in this problem.

In Table 3 the location of the primary vortex of the Stokes viscoelastic flow that we have obtained is compared with the result published in [65]. In this reference the authors use the log-conformation tensor instead of the elastic stress tensor with a first order accurate operator-splitting technique to solve the system of equations in a finite element formulation. In our results the mesh used for all the cases is a structured mesh with 10 000 (100×100) bilinear elements (Q_1), with a constant time step $\delta t = 0.0025$.

The lid-driven cavity flow problem without inertia leads to a symmetrical horizontal location of the vortex for a Newtonian fluid. From Table 3 one can see how the symmetry is broken when the Weissenberg number is increased. It is observed that the vortex is displaced in the direction opposite to the one expected when we increase the Reynolds number in a flow with inertia.

The cases considered so far yield stationary solutions, even for higher values of the Weissenberg number [64]. We have started with them to compare our results with those existing in the literature. In order to solve a truly dynamic problem, we force the lid velocity to be time dependent. In particular, let us consider now that it is given by

$$u_x(x, 1, t) = 16x^2\,(1 - x)^2 \sin(\pi t)$$

Using this lid velocity we obtain again a smooth initial boundary condition, as well as a time dependent behavior. The time interval used in all the simulations presented next is $[0, 2]$, which represents a complete period in the boundary condition. The time step size has been taken as $\delta t = 0.0025$.

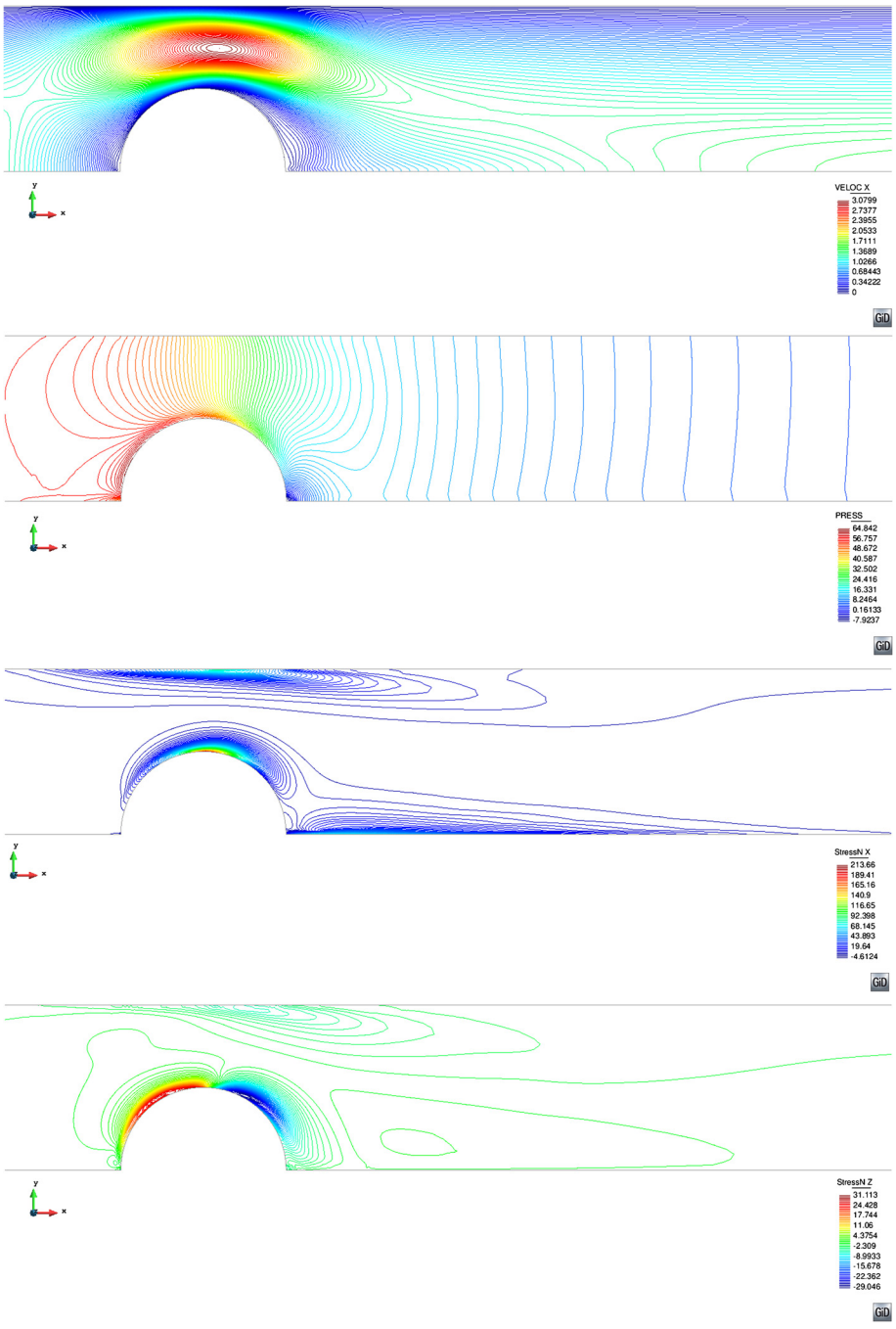


Fig. 8. Results for the flow over a cylinder for the case We = 1.3. From the top to the bottom: isolines of u_x, p, σ_{xx} and σ_{xy}.

Table 3
Vortex location in lid driven cavity problem.

We	(x, y)	
	Pan et al. [65]	Present study
0.0	–	(0.50, 0.765)
0.5	(0.4692383, 0.7981873)	(0.47, 0.80)
1.0	(0.4395693, 0.8159704)	(0.44, 0.82)
1.5	–	(0.41, 0.83)

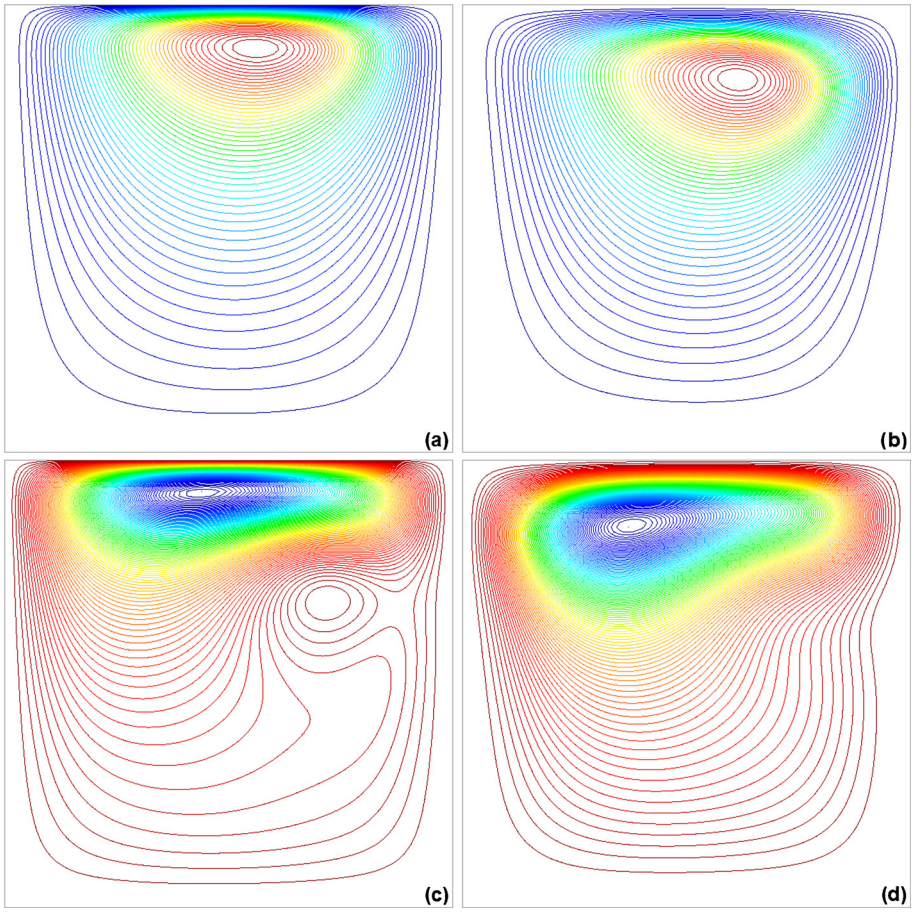


Fig. 9. Streamlines in the lid-driven cavity problem using the third order fractional step algorithm. From the top to the bottom and from the left to the right: $t = 0.5$, 1, 1.5 and 2.0.

We wish to compare now the performance of the monolithic time integration against the fractional step methods we have proposed, particularly in terms of CPU time. We take the Weissenberg number $\mathrm{We} = 1$ and two values of the Reynolds number, $\mathrm{Re} = UL_0/\beta\eta_0 = 1$ (quasi non-inertial case) and $\mathrm{Re} = 100$ (convective case). The characteristic velocity to compute Re and We has been taken as the maximum lid velocity and the characteristic length as $L_0 = 1$. Figs. 9 and 10 show the flow patterns obtained by the third order fractional step method in the convective case (both the other two fractional step schemes and the monolithic case give practically identical results).

Table 4 shows the saves in CPU time of the fractional step algorithms using the different temporal integrators with respect to the monolithic formulation. A simple sequential implementation has been used in all cases. The saves are presented as the quotient between the CPU time of the fractional step scheme over the CPU time of the corresponding monolithic scheme. The mesh used is the same as in the stationary problem.

For the quasi non-inertial problem the saves obtained are not very important, particularly for the third order fractional step scheme. However, these saves are more relevant for the convective case. Apart from the fact that the linear systems to be solved in the fractional step method are smaller, the drawback of the monolithic formulation is that the total number of iterations of the linear system solver is driven by the slowest variable. In fractional step schemes, each variable requires a different number of iterations to solve the corresponding linear system. As for Newtonian flows, the slowest variable in incompressible flows is the pressure. Table 5 illustrates this point. There, $\overline{\mathrm{nni}}$ is the average number of nonlinear iterations and $\overline{\mathrm{nsi}}$ is the average number of solver iterations. Subscripts are used in the fractional step case to identify the variable.

From the information of Table 5 we observe that the number of nonlinear iterations needed for the monolithic approaches is always greater than when using fractional step methods, and that in this last case the number of solver iterations in the momentum and in the constitutive equations is drastically smaller than for the pressure equation, as expected. This fact is a general trend, also encountered in Newtonian flows. Moreover, the subsystems to be solved when using fractional step methods are in general better conditioned, and solvers specifically designed for each of them can be used. Even if in our examples we have used the same solver for all subsystems, specific solvers could be exploited to improve the performance of fractional step schemes.

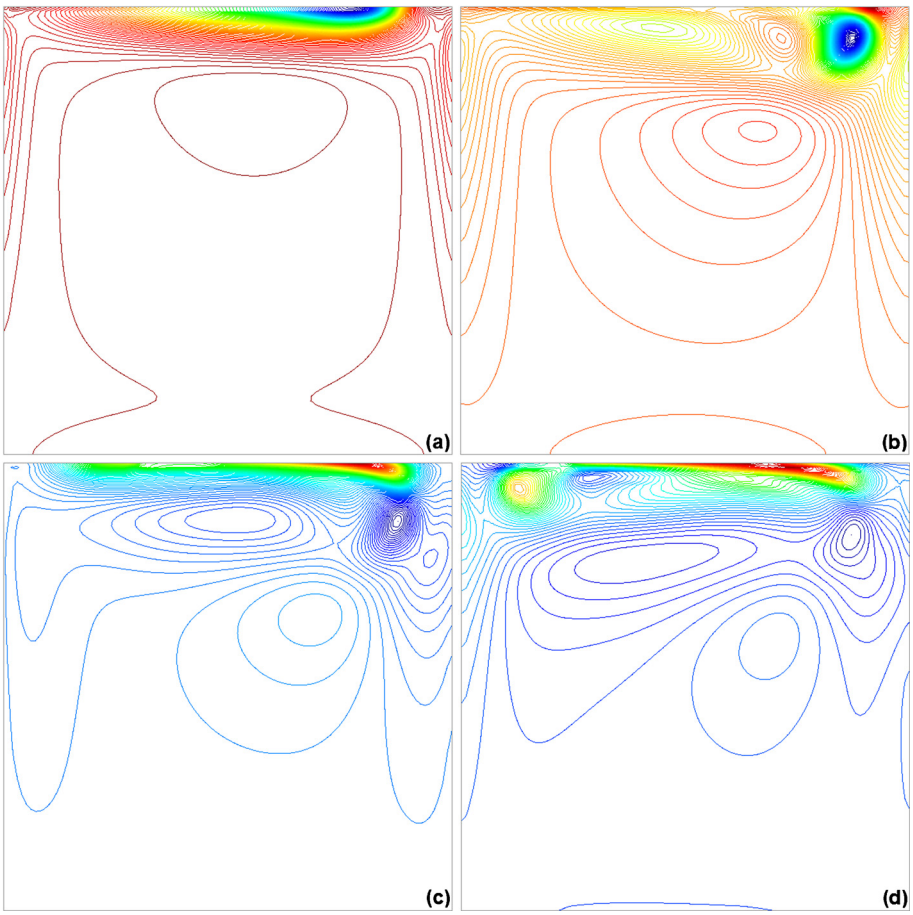


Fig. 10. σ_{xy} component of the elastic stress tensor in the lid-driven cavity problem using the third order fractional step algorithm. From the top to the bottom and from the left to the right: $t = 0.5$, 1, 1.5 and 2.0.

Table 4
Saves in CPU time of the fractional step schemes over the monolithic formulations.

$M_1 = 100 \times 100$	FS-BDF1	FS-BDF2	FS-BDF3
We = 1.0, Re = 1.0	0.64	0.57	0.88
We = 1.0, Re = 100	0.338	0.1818	0.2589

Table 5
Comparison between the number of iterations of the monolithic and of the second order fractional step algorithm using the BDF2 time integrator.

Case	Monolithic	Fractional-step		
	$\overline{\text{nni}}/\overline{\text{nsi}}$	$\overline{\text{nni}}_{\boldsymbol{u}}/\overline{\text{nsi}}_{\boldsymbol{u}}$	$\overline{\text{nni}}_{\boldsymbol{\sigma}}/\overline{\text{nsi}}_{\boldsymbol{\sigma}}$	$\overline{\text{nsi}}_p$
We = 1.0, Re = 1.0	13/57	11/2	9/2	108
We = 1.0, Re = 100	22/107	13/3	12/3	115

6.3.2. 3D cavity

To check the efficiency of fractional step algorithms in the 3D case, we consider now the three-dimensional lid-driven cavity problem, only using the second order algorithm and its monolithic counterpart for the sake of brevity. The problem is an extension of the 2D case presented above, now solved in the unit cube $[0, 1]^3$. On the lid $z = 1$ the x-component of the velocity is prescribed to

$$u_x(x, y, 1, t) = 256x^2 (1 - x)^2 y^2 (1 - y)^2 \sin(\pi t)$$

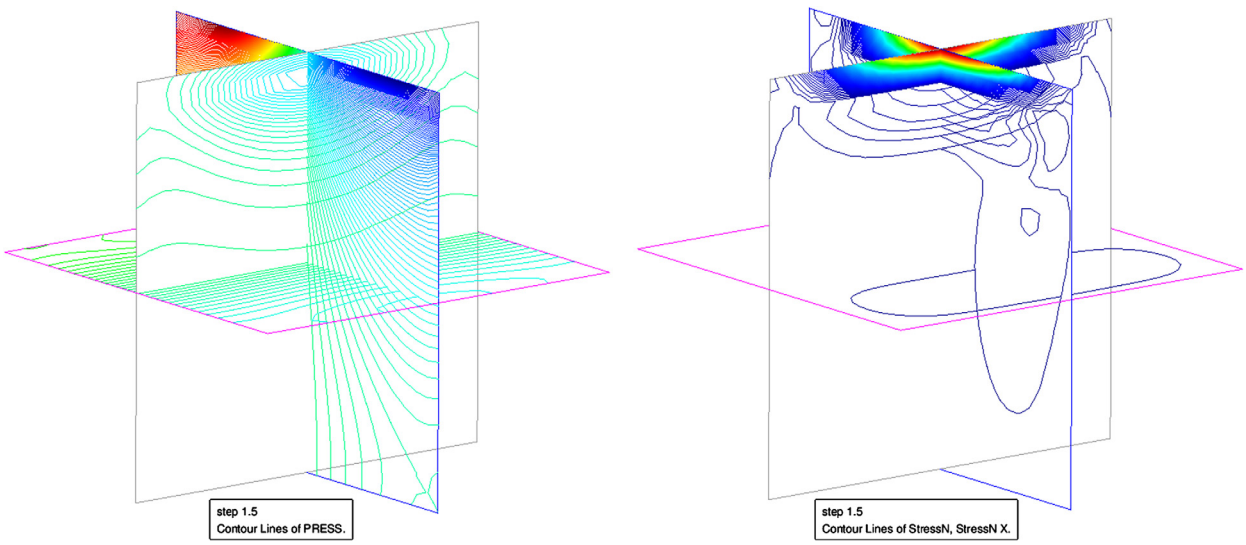
and $u_y = u_z = 0$, whereas $\boldsymbol{u} = \boldsymbol{0}$ is prescribed on the rest of the boundaries. No boundary conditions are required for the elastic stress tensor. Two cases are analyzed, one with Reynolds number Re = 0 (Stokes flow) and the other with Re = 100, and in both cases We = 0.5. The time interval of analysis is $[0, 2]$.

Table 6
Saves in CPU time of the fractional step schemes over the monolithic formulations in the 3D case (second order schemes).

	$T_{cputime}$	$B_{cputime}$	$S_{cputime}$
We = 0.5, Re = 0.0	0.2439	0.3817	0.1072
We = 0.5, Re = 100	0.0673	0.156	0.0129

Table 7
Comparison between the number of iterations of the monolithic and the second order fractional step algorithm in the 3D case.

Case	Monolithic	Fractional-step		
	$\overline{nni}/\overline{nsi}$	$\overline{nni}_{\boldsymbol{u}}/\overline{nsi}_{\boldsymbol{u}}$	$\overline{nni}_{\sigma}/\overline{nsi}_{\sigma}$	$\overline{nsi}_{p}$
We = 0.5, Re = 100	70/36	18/2	17/2	48



(a) (b)

Fig. 11. Lid-driven cavity in 3D: (a) pressure isolines and (b) σ_{xx} component of the elastic stress tensor.

Referring to the numerical discretization, a mesh of 15 625 linear hexahedral elements (Q_1) is employed. The time step size has been taken as $\delta t = 0.01$.

The comparison between the monolithic and fractional step methods is made in terms of the total CPU time ($T_{cputime}$), the CPU time used to build the algebraic system ($B_{cputime}$) and the time needed by the solver ($S_{cputime}$), all given in Table 6, as well as in terms of the number of iterations needed by the solver to solve the system of equations and by the number of nonlinear iterations used to obtain converged results, which are given in Table 7.

As for the 2D case, the reduction in the CPU time needed to solve the same problem with respect to the monolithic approach is very important and justifies the design of accurate fractional step algorithms. In terms of the number of iterations needed by each scheme (Table 7), the same comments as for the 2D case can be made for this 3D problem. The number of iterations needed by the monolithic method is greater both for the linear solver and for the nonlinear algorithm. Again we can see as the pressure equation is the bottleneck of the problem, with the greatest number of solver iterations, which in the monolithic case delays the convergence of the total system of equations. Likewise, the greater number of nonlinear iterations needed by the monolithic method confirms the better treatment of the nonlinearity when the problem is solved in a decoupled manner.

Fig. 11 shows some graphical results of the 3D cavity at time $t = 1.5$. Figs. 11(a) and 11(b) show respectively the pressure isolines and the isolines of the σ_{xx} component of the elastic stress tensor. No oscillations appear in neither of these variables. Finally, some streamlines at $t = 2$ are shown in Fig. 12 to visualize the flow pattern.

7. Conclusions

In this paper we have presented three different fractional step methods of first, second and third order in time to solve the viscoelastic fluid flow problem. The three methods have been constructed at the pure algebraic level, presenting in all cases the formal design via extrapolation and the associated approximate *LU* decomposition, with the corresponding matrix

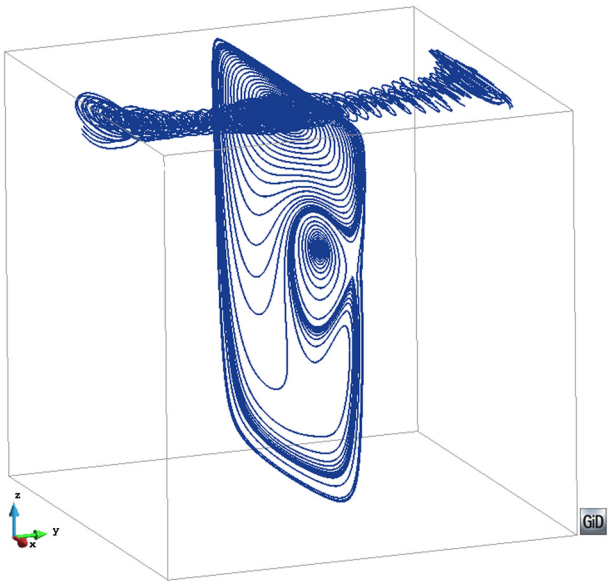


Fig. 12. Lid-driven cavity in 3D: some streamlines at $t = 2$.

errors. The third order scheme, which to our knowledge had not been tested before even for Newtonian flows, has been obtained by applying a Yosida splitting to the problem written in terms of the increments of the variables from one time step to the next. The accuracy of all the resulting methods has been checked numerically using an analytical manufactured solution, showing optimal convergence properties for smooth solutions.

Both the Galerkin and the stabilized finite element formulation presented in [31] have been considered for the spatial discretization. It has been shown that the latter introduces slight modifications of the former when applied in conjunction with the fractional step schemes. The resulting formulation has been found to be very robust, reaching Weissenberg numbers greater than those found for other formulations using the standard form of the constitutive equation, and close to those reached using the log-conformation version of it.

The efficiency of the fractional step formulations has been tested in a dynamic problem in 2D and 3D, showing an important reduction in the CPU time with respect to the monolithic case for the three formulations, particularly in problems where convection is more important. The splitting of the equations yields a problem that requires less nonlinear iterations per time step than the monolithic method, and the maximum number of linear solver iterations is only needed for the slowest variable, pressure in the incompressible flows considered.

As a general comment, the methods we have proposed for the viscoelastic flow problem inherit the properties of fractional step methods for Newtonian flows, and can be in fact designed by extending the same conceptual ingredients. Likewise, they could also be used as preconditioners of the monolithic problem, or as the starting point of predictor–corrector schemes, topics not touched in this work.

Acknowledgements

E. Castillo gratefully acknowledges the support received from CONICYT (Programa de Formación de Capital o Humano Avanzado/Becas-Chile Folio 72120080), and from the Universidad de Santiago de Chile. R. Codina gratefully acknowledges the support received from the Catalan Government through the ICREA Acadèmia Research Program.

References

[1] J. Mewis, N.J. Wagner, Current trends in suspension rheology, J. Non-Newton. Fluid Mech. 157 (2009) 147–150.
[2] D.M. Hall, T. Lookman, G.H. Fredrickson, S. Banerjee, Numerical method for hydrodynamic transport of inhomogeneous polymer melts, J. Comput. Phys. 224 (2007) 681–698.
[3] J.-D. Yu, S. Sakai, J. Sethian, Two-phase viscoelastic jetting, J. Comput. Phys. 220 (2007) 568–585.
[4] P. Sousa, P. Coelho, M. Oliveira, M. Alves, Effect of the contraction ratio upon viscoelastic fluid flow in three-dimensional square–square contractions, Chem. Eng. Sci. 66 (2011) 998–1009.
[5] Y. Kwon, Numerical aspects in modeling high Deborah number flow and elastic instability, J. Comput. Phys. 265 (2014) 128–144.
[6] M. Masoudian, K. Kim, F. Pinho, R. Sureshkumar, A viscoelastic turbulent flow model valid up to the maximum drag reduction limit, J. Non-Newton. Fluid Mech. 202 (2013) 99–111.
[7] F. Lien, P. Durbin, Non-linear modeling with application to high-lift, in: Proceedings of the Summer Program, Stanford University, 1996.
[8] J. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006) 6011–6045.
[9] S. Badia, R. Codina, Algebraic pressure segregation methods for the incompressible Navier–Stokes equations, Arch. Comput. Methods Eng. 15 (2007) 1–52.

[10] R. Glowinsky, O. Pironneau, Finite-element methods for Navier–Stokes equations, Annu. Rev. Fluid Mech. 24 (1992) 164–204.
[11] P. Singh, L. Leal, Finite-element simulation of the start-up problem for a viscoelastic fluid in an eccentric rotating cylinder geometry using a third-order upwind scheme, Theor. Comput. Fluid Dyn. 5 (1993) 107–137.
[12] P. Saramito, A new θ-scheme algorithm and incompressible fem for viscoelastic fluid flows, Modél. Math. Anal. Numér. 28 (1994) 1–35.
[13] J. Chrispell, V. Ervin, E. Jenkins, A fractional step θ-method approximation of time-dependent viscoelastic fluid flow, J. Comput. Appl. Math. 232 (2009) 159–175.
[14] G. D'Avino, M. Hulsen, Decoupled second-order transient schemes for the flow of viscoelastic fluids without a viscous solvent contribution, J. Non-Newton. Fluid Mech. 165 (2010) 1602–1612.
[15] G. D'Avino, M. Hulsen, P. Maffettone, Decoupled transient schemes for viscoelastic fluid flow with inertia, Comput. Fluids 66 (2012) 183–193.
[16] J.M. Kim, C. Kim, J.H. Kim, C. Chung, K.H. Ahn, S.J. Lee, High-resolution finite element simulation of 4:1 planar contraction flow of viscoelastic fluid, J. Non-Newton. Fluid Mech. 129 (2005) 23–37.
[17] H. Choi, H. Choi, J. Yoo, A fractional four-step finite element formulation of the unsteady incompressible Navier–Stokes equations using SUPG and linear equal-order element methods, Comput. Methods Appl. Mech. Eng. 143 (1997) 333–348.
[18] A. Bonito, M. Picasso, M. Laso, Numerical simulation of 3D viscoelastic flows with free surfaces, J. Comput. Phys. 215 (2006) 691–716.
[19] M. Tomé, A. Castelo, A. Afonso, M. Alves, F. Pinho, Application of the log-conformation tensor to three-dimensional time-dependent free surface flows, J. Non-Newton. Fluid Mech. 175–176 (2012) 44–54.
[20] P. Singh, D. Joseph, T. Hesla, R. Glowinski, T.-W. Pan, A distributed Lagrange multiplier/fictitious domain method for viscoelastic particulate flows, J. Non-Newton. Fluid Mech. 91 (2000) 165–188.
[21] Z. Yu, N. Phan-Thien, Y. Fan, R.I. Tanner, Viscoelastic mobility problem of a system of particles, J. Non-Newton. Fluid Mech. 104 (2002) 87–124.
[22] J. Marchal, M. Crochet, A new mixed finite element for calculating viscoelastic flow, J. Non-Newton. Fluid Mech. 26 (1987) 77–114.
[23] Y. Fan, R. Tanner, N. Phan-Thien, Galerkin/least-square finite-element methods for steady viscoelastic flows, J. Non-Newton. Fluid Mech. 84 (1999) 233–256.
[24] M. Fortin, A. Fortin, A new approach for the FEM simulation of viscoelastic flows, J. Non-Newton. Fluid Mech. 32 (1989) 295–310.
[25] F.P. Baaijens, Application of low-order discontinuous Galerkin methods to the analysis of viscoelastic flows, J. Non-Newton. Fluid Mech. 52 (1994) 37–57.
[26] J.Y. Yoo, Y. Na, A numerical study of the planar contraction flow of a viscoelastic fluid using the SIMPLER algorithm, J. Non-Newton. Fluid Mech. 39 (1991) 89–106.
[27] P. Oliveira, F. Pinho, Plane contraction flows of upper convected Maxwell and Phan-Thien–Tanner fluids as predicted by a finite-volume method, J. Non-Newton. Fluid Mech. 88 (1999) 63–88.
[28] M. Alves, F. Pinho, P. Oliveira, Effect of a high-resolution differencing scheme on finite-volume predictions of viscoelastic flows, J. Non-Newton. Fluid Mech. 93 (2000) 287–314.
[29] J.S. Howell, Computation of viscoelastic fluid flows using continuation methods, J. Comput. Appl. Math. 225 (2009) 187–201.
[30] E. Castillo, R. Codina, Stabilized stress–velocity–pressure finite element formulations of the Navier–Stokes problem for fluids with non-linear viscosity, Comput. Methods Appl. Mech. Eng. 279 (2014) 554–578.
[31] E. Castillo, R. Codina, Variational multi-scale stabilized formulations for the stationary three-field incompressible viscoelastic flow problem, Comput. Methods Appl. Mech. Eng. 279 (2014) 579–605.
[32] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Comput. Phys. 170 (2001) 112–140.
[33] J. Kwack, A. Masud, A three-field formulation for incompressible viscoelastic fluids, Int. J. Eng. Sci. 48 (2010) 1413–1432 (Special Issue in Honor of K.R. Rajagopal).
[34] O.M. Coronado, D. Arora, M. Behr, M. Pasquali, Four-field Galerkin/least-squares formulation for viscoelastic fluids, J. Non-Newton. Fluid Mech. 140 (2006) 132–144.
[35] X. Li, X. Han, X. Wang, Numerical modeling of viscoelastic flows using equal low-order finite elements, Comput. Methods Appl. Mech. Eng. 199 (2010) 570–581.
[36] H.-C. Lee, A nonlinear weighted least-squares finite element method for the Oldroyd-B viscoelastic flow, Appl. Math. Comput. 219 (2012) 421–434.
[37] A. Quarteroni, F. Saleri, A. Veneziani, Analysis of the Yosida method for the incompressible Navier–Stokes equations, J. Math. Pures Appl. 78 (1999) 473–503.
[38] R.B. Bird, R.C. Amstrong, O. Hassager, Dynamics of Polymeric Liquids, vol. 1: Fluid Mechanics, 2nd ed., Wiley, New York, 1987.
[39] R.B. Bird, C.F. Curtiss, R.C. Amstrong, O. Hassager, Dynamics of Polymeric Liquids, vol. 2: Kinetic Theory, 2nd ed., Wiley, New York, 1987.
[40] E. Fernández-Cara, F. Guillén, R. Ortega, Mathematical modeling and analysis of viscoelastic fluids of the Oldroyd kind, in: Handbook of Numerical Analysis, vol. VIII, North-Holland, 2002.
[41] K. Yapici, B. Karasozen, Y. Uludag, Finite volume simulation of viscoelastic laminar flow in a lid-driven cavity, J. Non-Newton. Fluid Mech. 164 (2009) 51–65.
[42] J. Su, J. Ouyang, X. Wang, B. Yang, W. Zhou, Lattice Boltzmann method for the simulation of viscoelastic fluid flows over a large range of Weissenberg numbers, J. Non-Newton. Fluid Mech. 194 (2013) 42–59.
[43] J. Bonvin, M. Picasso, R. Stenberg, GLS and EVSS methods for a three-field Stokes problem arising from viscoelastic flows, Comput. Methods Appl. Mech. Eng. 190 (2001) 3893–3914.
[44] R. Codina, Finite element approximation of the three-field formulation of the Stokes problem using arbitrary interpolations, SIAM J. Numer. Anal. 47 (2009) 699–718.
[45] J. Marchal, M. Crochet, A new mixed finite element for calculating viscoelastic flow, J. Non-Newton. Fluid Mech. 26 (1987) 77–114.
[46] A.C. Bogaerds, W.M. Verbeeten, G.W. Peters, F.P. Baaijens, 3D viscoelastic analysis of a polymer solution in a complex flow, Comput. Methods Appl. Mech. Eng. 180 (1999) 413–430.
[47] R. Codina, S. Badia, On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems, Comput. Methods Appl. Mech. Eng. 195 (2006) 2900–2918.
[48] J. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[49] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[50] R. Temam, Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionaires (I), Arch. Ration. Mech. Anal. 32 (1969) 135–153.
[51] S. Badia, R. Codina, Pressure segregation methods based on a discrete pressure Poisson equation. An algebraic approach, Int. J. Numer. Methods Fluids 56 (2008) 351–382.
[52] J. Shen, On error estimates of some higher order projection and penalty-projection methods, Numer. Math. 62 (1992) 49–73.
[53] J. Blasco, R. Codina, Error estimates for an operator splitting method for incompressible flows, Appl. Numer. Math. 51 (2004) 1–17.
[54] H. Owen, R. Codina, A third-order velocity correction scheme obtained at the discrete level, Int. J. Numer. Methods Fluids 69 (2012) 57–72.
[55] P. Gervasio, F. Saleri, A. Veneziani, Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 214 (2006) 347–365.

[56] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, Comput. Methods Appl. Mech. Eng. 190 (2000) 1579–1599.
[57] R. Codina, Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales, Appl. Numer. Math. 58 (2008) 264–283.
[58] H.A. van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.
[59] M.A. Hulsen, R. Fattal, R. Kupferman, Flow of viscoelastic fluids past a cylinder at high Weissenberg number: stabilized simulations using matrix logarithms, J. Non-Newton. Fluid Mech. 127 (2005) 27–39.
[60] M. Alves, F. Pinho, P. Oliveira, The flow of viscoelastic fluids past a cylinder: finite-volume high-resolution methods, J. Non-Newton. Fluid Mech. 97 (2001) 207–232.
[61] H. Damanik, J. Hron, A. Ouazzi, S. Turek, A monolithic FEM approach for the log-conformation reformulation LCR of viscoelastic flow problems, J. Non-Newton. Fluid Mech. 165 (2010) 1105–1113.
[62] A. Afonso, P. Oliveira, F. Pinho, M. Alves, The log-conformation tensor approach in the finite-volume method framework, J. Non-Newton. Fluid Mech. 157 (2009) 55–65.
[63] H.-S. Dou, N. Phan-Thien, The flow of an Oldroyd-B fluid past a cylinder in a channel: adaptive viscosity vorticity DAVSS-w formulation, J. Non-Newton. Fluid Mech. 87 (1999) 47–73.
[64] R. Fattal, R. Kupferman, Time-dependent simulation of viscoelastic flows at high Weissenberg number using the log-conformation representation, J. Non-Newton. Fluid Mech. 126 (2005) 23–37.
[65] T.-W. Pan, J. Hao, R. Glowinski, On the simulation of a time-dependent cavity flow of an Oldroyd-B fluid, Int. J. Numer. Methods Fluids 60 (2009) 791–808.