Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 382 (2021) 113888

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

# An efficient split-step framework for non-Newtonian incompressible flow problems with consistent pressure boundary conditions

Douglas R.Q. Pacheco[a,c,*], Richard Schussnig[b,c], Thomas-Peter Fries[b,c]

[a] Institute of Applied Mathematics, Graz University of Technology, Graz, Austria
[b] Institute of Structural Analysis, Graz University of Technology, Graz, Austria
[c] Graz Center of Computational Engineering, Graz University of Technology, Graz, Austria

Received 14 January 2021; received in revised form 21 April 2021; accepted 22 April 2021
Available online 6 May 2021

## Abstract

Incompressible flow problems with nonlinear viscosity, as they often appear in biomedical and industrial applications, impose several numerical challenges related to regularity requirements, boundary conditions, matrix preconditioning, among other aspects. In particular, standard split-step or projection schemes decoupling velocity and pressure are not as efficient for generalised Newtonian fluids, since the additional terms due to the non-zero viscosity gradient couple all velocity components again. Moreover, classical pressure correction methods are not consistent with the non-Newtonian setting, which can cause numerical artifacts such as spurious pressure boundary layers. Although consistent reformulations have been recently developed, the additional projection steps needed for the viscous stress tensor incur considerable computational overhead. In this work, we present a new time-splitting framework that handles such important issues, leading to an efficient and accurate numerical tool. Two key factors for achieving this are an appropriate explicit–implicit treatment of the viscous and convective nonlinearities, as well as the derivation of a pressure Poisson problem with fully consistent boundary conditions and finite-element-suitable regularity requirements. We present first- and higher-order stepping schemes tailored for this purpose, as well as various numerical examples showcasing the stability, accuracy and efficiency of the proposed framework.

*Keywords:* non-Newtonian fluids; Incompressible flow; Split-step schemes; Finite element methods; Pressure boundary conditions; Pressure Poisson equation

## 1. Introduction

Split-step or time-splitting methods for incompressible flow problems, the most famous of which being the so-called projection schemes, are numerical methods that decouple the computation of pressure and velocity. This is highly advantageous especially when employing iterative solvers, since the decoupled blocks reduce to simple convection–diffusion, Poisson and mass matrix problems. An excellent discussion and overview of early projection methods was performed by Guermond et al. [1,2]. The first generation of such schemes were plagued by nonphysical pressure boundary conditions (BCs) that induced severe numerical boundary layers and spoiled their order of

[*] Corresponding author at: Institute of Applied Mathematics, Graz University of Technology, Graz, Austria.
*E-mail address:* pacheco@math.tugraz.at (D.R.Q. Pacheco).

https://doi.org/10.1016/j.cma.2021.113888

accuracy. Timmermans et al. [3] solved this issue by introducing the so-called rotational pressure correction scheme that adds a correction term to the pressure update. In one of the rather scarce theoretical studies that are not restricted to pure Dirichlet problems, Guermond et al. [1] showed that in three-dimensional problems with natural BCs, the rotational projection method has an intrinsic contradiction: the factor multiplying the added correction term should be set to 1 in order to attain consistent pressure BCs, but must be taken as less than 2/3 if we wish to guarantee stability. Although this does not spoil the asymptotic order of convergence *per se*, remedies have been proposed in order to increase the accuracy of the pressure approximation on the boundaries [4].

An alternative solution was presented by Liu [5], extending the pressure Poisson method of Johnston and Liu [6] to the more realistic setting where open/traction boundaries are allowed. Liu's method completely replaces the continuity equation by a pressure Poisson problem with consistent boundary conditions. Furthermore, by treating the convective term explicitly, the scheme decouples even the velocity components, leading to a sequence of scalar problems that can be efficiently tackled. Variations of that scheme including high-order methods are available and have proven to be powerful tools in the solution of challenging incompressible flow problems [7–13].

Extending time-splitting schemes to generalised Newtonian fluids is not a trivial task, especially in a finite element framework. In fact, only very recently have the first steps in this direction been taken. Deteix and Yakoubi [14] started by extending the popular rotational projection method to fluids with non-homogeneous viscosity. A more general setting allowing natural boundary conditions was then formulated by Plasman et al. [15]. Yet, when the viscosity is not simply variable but actually dependent on the velocity gradient, the divergence of the discretised stress tensor will not be regular (smooth) enough to be incorporated directly into the pressure correction step. One way around that is to project the viscous stress tensor onto a continuous space, as done by Deteix and Yakoubi [16] in their recently introduced shear rate projection (SRP) scheme. Despite being very accurate and simple, the SRP method has a high cost when compared to standard (Newtonian) pressure correction schemes: in three dimensions, it requires the solution of a vector-valued convection–diffusion problem, two Poisson equations and at least ten scalar mass matrix problems (for adding quantities from different vector spaces). Moreover, the SRP method does not allow the use of equal-order velocity–pressure pairs, requiring a higher order for the velocity. For these reasons, the method can be rather expensive in practical applications.

In this context, we present here a new time-splitting framework with consistent Dirichlet and Neumann pressure BCs, reduced computational cost, good accuracy and stability properties, while still allowing the use of equal-order finite element pairs. By treating the rheological law and the viscous stress appropriately, we manage to decouple the velocity components and end up with a method that requires the solution of one Poisson equation, one scalar projection and two or three (i.e., the number of spatial dimensions) scalar convection–diffusion equations. The approach can be seen as an extension of the pressure Poisson method by Liu [5] to the generalised Newtonian case [17]. Also related is our recent work on pressure-Poisson-based stabilisation methods for equal-order elements [18]. First- and higher-order temporal discretisations are considered, in combination with the appropriate extrapolation rules. Various numerical examples are provided to showcase the efficiency and accuracy of our novel framework.

## 2. Strong formulation

Let us consider a spatial domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, with a Lipschitz boundary $\Gamma := \partial\Omega$ decomposed into two non-overlapping regions $\Gamma_D$ and $\Gamma_N$. The standard setting for the incompressible Navier–Stokes system reads

$$\rho\,[\partial_t \mathbf{u} + (\nabla\mathbf{u})\mathbf{u}] - \nabla\cdot\mathbb{S} + \nabla p = \mathbf{f} \quad \text{in} \quad \Omega\times(0,T], \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in} \quad \Omega\times(0,T], \tag{2}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on} \quad \Gamma_D\times(0,T], \tag{3}$$

$$(\mu\nabla\mathbf{u} - p\mathbb{I})\mathbf{n} = \mathbf{h} \quad \text{on} \quad \Gamma_N\times(0,T], \tag{4}$$

$$\mathbf{u} = \mathbf{u}_0 \quad \text{at} \quad t = 0, \tag{5}$$

where $\mathbf{u}$ is the flow velocity, $p$ is the pressure, $\rho$ is the fluid's density, $\mathbb{S}$ is the viscous stress tensor, $\mathbb{I}$ is the $d\times d$ second-order identity tensor, $\mu$ is the dynamic viscosity and the right-hand side vectors are given quantities. The symbols $\nabla$ and $\partial_t$ denote spatial and temporal differentiation, respectively. For a generalised Newtonian (often called quasi-Newtonian) fluid, the stress–strain relationship is given by

$$\mathbb{S} = 2\mu\nabla^s\mathbf{u}, \tag{6}$$

where $\nabla^s \mathbf{u}$ is the symmetric part of the velocity gradient, namely,

$$\nabla^s \mathbf{u} := \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^\top\right].$$

The viscosity is most commonly modelled through a nonlinear dependence on the shear rate $\dot{\gamma} := \sqrt{\nabla^s \mathbf{u} : \nabla^s \mathbf{u}/2}$ by a map $\eta : \mathbb{R}_+ \rightarrow \mathbb{R}_+^*$, that is,

$$\mu = \eta(\dot{\gamma}).$$

Most rheological models of industrial and biomedical interest, such as those describing polymer melts and blood, can be written in the generic form [19]

$$\eta\left(\dot{\gamma}\right) = \eta_1 + \eta_2 \left[\kappa + (\lambda \dot{\gamma})^a\right]^{\frac{n-1}{a}},$$

where $a, n, \kappa, \lambda, \eta_1, \eta_2$ are fitting parameters depending on rheological properties. The choice $\kappa = \eta_1 = 0$ gives the so-called power-law model, whereas $\kappa = 1$ leads to the Carreau–Yasuda model (or simply *Carreau model* when $a = 2$). We can recover the Newtonian case by setting $\eta_2 = 0$ or $n = 1$.

**Remark 1.** We herein focus on the setting in which the natural boundary conditions are given in terms of normal pseudo-tractions $(\mu \nabla \mathbf{u} - p\mathbb{I})\mathbf{n}$, which is the preferable one for simulating open/truncated outflow boundaries [19–22]. In certain classes of problems such as fluid–structure interaction and two-phase flows, formulations in terms of *real* tractions are more appropriate for an accurate interface coupling [23].

Even in the Newtonian case, the saddle-point structure of standard variational formulations of (1)–(5) prohibits the use of equal-order finite element spaces for pressure and velocity, as such spaces are incompatible in the Ladyzhenskaya–Babuška–Brezzi (LBB) sense. One way to circumvent that is to use residual-based stabilisation techniques such as Petrov–Galerkin-like or Galerkin least-squares methods [24,25]. Alternatively, Johnston and Liu [6] proposed replacing the explicit divergence-free constraint (2) by a $C^0$-conforming pressure Poisson equation (PPE) that enforces incompressibility implicitly. The advantage of their formulation with respect to standard stabilised methods is that, when treating the convection and pressure terms in the momentum equation explicitly in time, it is possible to decouple the computation of not only pressure and velocity, but also of the velocity components. This leads to conditionally stable and highly efficient incompressible flow solvers. That approach, however, was initially possible only for the Dirichlet problem, which limited its applicability [10]. Liu [5] then extended the PPE approach to allow natural BCs in a fully consistent way. The extension of their method to the non-Newtonian case is not trivial, though, due to the increased complexity of the corresponding PPE and also additional regularity requirements induced by the shear-dependent viscosity, as we tackle next.

We propose replacing the classical initial–boundary value problem (IBVP) (1)–(5) by an alternative one:

$$\begin{aligned}
\rho\left[\partial_t \mathbf{u} + (\nabla \mathbf{u})\mathbf{u}\right] - \mu \Delta \mathbf{u} - 2\nabla^s \mathbf{u} \nabla \mu + \nabla p &= \mathbf{f} && \text{in } \Omega \times (0, T], && (7)\\
-\Delta p + \nabla \cdot \left[2\nabla^s \mathbf{u} \nabla \mu - (\rho \nabla \mathbf{u})\mathbf{u}\right] - \left[\nabla \times (\nabla \times \mathbf{u})\right] \cdot \nabla \mu &= -\nabla \cdot \mathbf{f} && \text{in } \Omega \times [0, T], && (8)\\
\mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D \times (0, T], && (9)\\
(\mu \nabla \mathbf{u} - p\mathbb{I})\mathbf{n} &= \mathbf{h} && \text{on } \Gamma_N \times (0, T], && (10)\\
\mathbf{u} &= \mathbf{u}_0 && \text{at } t = 0, && (11)\\
\nabla \cdot \mathbf{u}_0 &= 0 && \text{in } \Omega, && (12)\\
p &= \Phi(\mathbf{u}) && \text{on } \Gamma_N \times [0, T], && (13)\\
\frac{\partial p}{\partial n} &= \Psi(\mathbf{u}) && \text{on } \Gamma_D \times [0, T], && (14)
\end{aligned}$$

where

$$\begin{aligned}
\Phi(\mathbf{u}) &:= \mu \nabla \mathbf{u} : (\mathbf{n} \otimes \mathbf{n} - \mathbb{I}) - \mathbf{h} \cdot \mathbf{n},\\
\Psi(\mathbf{u}) &:= \left[\mathbf{f} - \rho \partial_t \mathbf{g} - (\rho \nabla \mathbf{u})\mathbf{u} + 2\nabla^s \mathbf{u} \nabla \mu - \mu \nabla \times (\nabla \times \mathbf{u})\right] \cdot \mathbf{n}.
\end{aligned}$$

Although this IBVP may seem somewhat more complex than the original one, we will later show how it can be used for constructing simple finite element formulations requiring only the solution of Poisson and convection–diffusion problems. Note that, for the homogeneous Newtonian case ($\nabla\mu \equiv \mathbf{0}$), several terms vanish, recovering the formulation by Liu [5].

**Theorem 2.1.** *For sufficiently regular* $p, \mathbf{u}, \mathbf{f}, \mathbf{g}, \mathbf{h}$, *systems* (7)–(14) *and* (1)–(5) *are equivalent.*

**Proof.** We will first show that (1)–(5) imply (7)–(14). Eq. (7) comes from the original momentum balance (1), since when $\nabla\cdot\mathbf{u} = 0$ we can write

$$\begin{aligned}\nabla\cdot\mathbb{S} &= \nabla\cdot\left(2\mu\nabla^s\mathbf{u}\right)\\ &= \mu\nabla\left(\nabla\cdot\mathbf{u}\right) + \mu\Delta\mathbf{u} + 2\nabla^s\mathbf{u}\nabla\mu\\ &= \mu\Delta\mathbf{u} + 2\nabla^s\mathbf{u}\nabla\mu.\end{aligned} \tag{15}$$

Now we obtain the Neumann BC (14) by dotting Eq. (7) with the normal vector $\mathbf{n}$, restricting the result to $\Gamma_D$ and using the relation

$$\Delta\mathbf{u} \equiv \nabla\left(\nabla\cdot\mathbf{u}\right) - \nabla\times\left(\nabla\times\mathbf{u}\right) = -\nabla\times\left(\nabla\times\mathbf{u}\right). \tag{16}$$

Similarly, the PPE (8) is obtained by taking (minus) the divergence of Eq. (7), so that

$$\begin{aligned}-\Delta p + \nabla\cdot\left[2\nabla^s\mathbf{u}\nabla\mu - (\rho\nabla\mathbf{u})\mathbf{u}\right] + \nabla\cdot\mathbf{f} &= \nabla\cdot\left(\rho\partial_t\mathbf{u}\right) - \Delta\mathbf{u}\cdot\nabla\mu\\ &= \rho\partial_t\left(\nabla\cdot\mathbf{u}\right) + \left[\nabla\times\left(\nabla\times\mathbf{u}\right) - \nabla\left(\nabla\cdot\mathbf{u}\right)\right]\cdot\nabla\mu\\ &= \left[\nabla\times\left(\nabla\times\mathbf{u}\right)\right]\cdot\nabla\mu.\end{aligned}$$

To obtain the pressure Dirichlet BC (13), we first dot Eq. (4) by $\mathbf{n}$ to get

$$\begin{aligned}\mathbf{h}\cdot\mathbf{n} &= \left[\left(\mu\nabla\mathbf{u} - p\mathbb{I}\right)\mathbf{n}\right]\cdot\mathbf{n}\\ &\equiv \mu\nabla\mathbf{u} : \mathbf{n}\otimes\mathbf{n} - p\mathbf{n}\cdot\mathbf{n}\\ &= \mu\nabla\mathbf{u} : \mathbf{n}\otimes\mathbf{n} - p.\end{aligned}$$

Due to Eq. (2), we can subtract $\mu\nabla\cdot\mathbf{u}$ from the right-hand side and restrict the result to $\Gamma_N$, which finally gives us Eq. (13), since $\nabla\cdot\mathbf{u} \equiv \nabla\mathbf{u} : \mathbb{I}$. The additional condition (12) is simply the restriction of the incompressibility constraint to $t = 0$, which completes the first part of the proof.

Proving the other direction, i.e, that (7)–(14) implies (1)–(5), can be done by first taking the divergence of Eq. (7) and adding the result to Eq. (8), which gives us

$$\begin{aligned}0 &= \left[\nabla\times\left(\nabla\times\mathbf{u}\right)\right]\cdot\nabla\mu + \nabla\mu\cdot\Delta\mathbf{u} + \mu\nabla\cdot\left(\Delta\mathbf{u}\right) - \rho\partial_t\left(\nabla\cdot\mathbf{u}\right)\\ &= \mu\nabla\cdot\left(\Delta\mathbf{u}\right) + \nabla\mu\cdot\left[\Delta\mathbf{u} + \nabla\times\left(\nabla\times\mathbf{u}\right)\right] - \rho\partial_t\left(\nabla\cdot\mathbf{u}\right).\end{aligned} \tag{17}$$

Therefore, since $\nabla\cdot\left(\Delta\mathbf{u}\right) \equiv \Delta\left(\nabla\cdot\mathbf{u}\right)$, we get a heat-like equation on the variable $\phi := \nabla\cdot\mathbf{u}$:

$$\partial_t\phi - \nabla\cdot\left(\nu\nabla\phi\right) = 0, \tag{18}$$

where $\nu = \mu/\rho$ is the kinematic viscosity. Zero initial condition on $\phi$ is guaranteed by (12). We can obtain Neumann BCs for this equation by dotting Eq. (7) with $\mathbf{n}$, restricting the result to $\Gamma_D$ and subtracting from (14), which gives

$$\begin{aligned}0 &= \left\{\mathbf{n}\cdot\left[\Delta\mathbf{u} + \left(\nabla\times\nabla\times\mathbf{u}\right)\right]\right\}|_{\Gamma_D}\\ &= \left\{\mathbf{n}\cdot\left[\nabla\left(\nabla\cdot\mathbf{u}\right)\right]\right\}|_{\Gamma_D}\\ &= \left.\frac{\partial\phi}{\partial n}\right|_{\Gamma_D}.\end{aligned} \tag{19}$$

Dirichlet BCs for $\phi$ on $\Gamma_N$ can be obtained by dotting (10) with $\mathbf{n}$ and adding the result to (13), yielding

$$\begin{aligned}0 &= \left(\mu\nabla\mathbf{u} : \mathbb{I}\right)|_{\Gamma_N}\\ &= \left(\mu\nabla\cdot\mathbf{u}\right)|_{\Gamma_N}\\ &= \left(\mu\phi\right)|_{\Gamma_N},\end{aligned} \tag{20}$$

that is, $\phi|_{\Gamma_N} = 0$, as the viscosity is assumed to be strictly positive. We thus get $\phi \equiv 0$, that is, our modified system also enforces incompressibility. With mass conservation now proved, the equivalence between the reduced momentum equation (7) and the standard one (1) is straightforward, which concludes the proof. □

## 3. A $C^0$ variational formulation

Of course, introducing the PPE as a replacement for the simple, explicit incompressibility constraint increases the regularity requirements for both pressure and velocity. We will now show how to devise an appropriate weak formulation allowing the use of standard $C^0$ finite element spaces for the spatial discretisation.

### 3.1. Pressure Poisson equation

We begin by multiplying Eq. (8) by a test function $q \in H^1(\Omega)$, with $q|_{\Gamma_N} = 0$, and using integration by parts to yield

$$\langle \nabla q, \nabla p \rangle - \left\langle q, \frac{\partial p}{\partial n} \right\rangle_{\Gamma_D} - \langle q, [\nabla \times (\nabla \times \mathbf{u})] \cdot \nabla \mu \rangle = \left\langle q, \nabla \cdot \left[ (\rho \nabla \mathbf{u})\mathbf{u} - 2\nabla^s \mathbf{u} \nabla \mu - \mathbf{f} \right] \right\rangle,$$

where $\langle \cdot, \cdot \rangle$ and $\langle \cdot, \cdot \rangle_\Gamma$ denote the $L^2(\Omega)$ and $L^2(\Gamma)$ scalar products, and analogously for $\langle \cdot, \cdot \rangle_{\Gamma_N}$ and $\langle \cdot, \cdot \rangle_{\Gamma_D}$ on the respective parts of the boundary. Integrating the right-hand side by parts and enforcing the Neumann BC (14) yields

$$\begin{aligned} &\langle \nabla q, \nabla p + (\rho \nabla \mathbf{u})\mathbf{u} - \mathbf{f} \rangle + \langle q, \partial_t (\mathbf{n} \cdot \mathbf{g}) \rangle_{\Gamma_D} \\ &\quad = \left\langle \nabla q, \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^\top \right] \nabla \mu \right\rangle + \langle q, [\nabla \times (\nabla \times \mathbf{u})] \cdot \nabla \mu \rangle - \langle q\mathbf{n}, \mu \nabla \times (\nabla \times \mathbf{u}) \rangle_{\Gamma_D}, \end{aligned} \tag{21}$$

and

$$\begin{aligned} \langle q\mathbf{n}, \mu \nabla \times (\nabla \times \mathbf{u}) \rangle_{\Gamma_D} &= \langle q\mathbf{n}, \mu \nabla \times (\nabla \times \mathbf{u}) \rangle_\Gamma \\ &= \langle \nabla (q\mu), \nabla \times (\nabla \times \mathbf{u}) \rangle + \langle q, \mu \nabla \cdot [\nabla \times (\nabla \times \mathbf{u})] \rangle \\ &= \langle \nabla (q\mu), \nabla \times (\nabla \times \mathbf{u}) \rangle \\ &= \langle \nabla q, \mu \nabla \times (\nabla \times \mathbf{u}) \rangle + \langle q, \nabla \mu \cdot [\nabla \times (\nabla \times \mathbf{u})] \rangle . \end{aligned}$$

Therefore, the right-hand side of Eq. (21) reduces to

$$\left\langle \nabla q, \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^\top \right] \nabla \mu \right\rangle - \langle \nabla q, \mu \nabla \times (\nabla \times \mathbf{u}) \rangle .$$

Notice that we still have a second-order term which cannot be handled by $C^0$ finite element spaces. In order to allow the use of such standard discretisations, we must rewrite this second-order term using only first-order derivatives. Integration by parts gives

$$\langle \mu \nabla q, \nabla \times (\nabla \times \mathbf{u}) \rangle = \langle \nabla q \times \mathbf{n}, \mu \nabla \times \mathbf{u} \rangle_\Gamma + \langle \nabla \times (\mu \nabla q), \nabla \times \mathbf{u} \rangle ,$$

but

$$\begin{aligned} \langle \nabla \times (\mu \nabla q), \nabla \times \mathbf{u} \rangle &= \langle \nabla \mu \times \nabla q + \mu \nabla \times (\nabla q), \nabla \times \mathbf{u} \rangle \\ &= \langle \nabla \mu \times \nabla q, \nabla \times \mathbf{u} \rangle \\ &\equiv \langle \nabla q, (\nabla \times \mathbf{u}) \times \nabla \mu \rangle \\ &\equiv \left\langle \nabla q, \left[ \nabla \mathbf{u} - (\nabla \mathbf{u})^\top \right] \nabla \mu \right\rangle . \end{aligned}$$

Hence, the weak form simplifies to

$$\langle \nabla q, \nabla p \rangle = \left\langle \nabla q, \mathbf{f} - (\rho \nabla \mathbf{u})\mathbf{u} + 2(\nabla \mathbf{u})^\top \nabla \mu \right\rangle + \langle \mathbf{n} \times \nabla q, \mu \nabla \times \mathbf{u} \rangle_\Gamma - \langle q, \partial_t (\mathbf{n} \cdot \mathbf{g}) \rangle_{\Gamma_D} .$$

**Lemma 3.1.** *For $\mathbf{v} \in [L^2(\Gamma)]^d$ and $q$ such that $\mathbf{n} \times \nabla q \in [L^2(\Gamma)]^d$, with $q|_{\Gamma_N} = 0$, we have*

$$\langle \mathbf{n} \times \nabla q, \mathbf{v} \rangle_\Gamma = \langle \mathbf{n} \times \nabla q, \mathbf{v} \rangle_{\Gamma_D} . \tag{22}$$

**Proof.** The gradient $\nabla q$ can be decomposed into normal and tangential derivatives as

$$\nabla q = \left(\frac{\partial q}{\partial n}\right)\mathbf{n} + \nabla_\tau q = (\mathbf{n}\cdot\nabla q)\,\mathbf{n} + (\mathbf{n}\times\nabla q)\times\mathbf{n}.$$

Since $q|_{\Gamma_N} = 0$, the tangential derivative $\nabla_\tau q$ will also be zero on $\Gamma_N$. Thus, taking the left cross product with $\mathbf{n}$ and restricting the result to $\Gamma_N$ gives us

$$\mathbf{n}\times\nabla q|_{\Gamma_N} = \left(\frac{\partial q}{\partial n}\right)\mathbf{n}\times\mathbf{n}\bigg|_{\Gamma_N} + \mathbf{n}\times\nabla_\tau q|_{\Gamma_N} = \frac{\partial q}{\partial n}\bigg|_{\Gamma_N}\mathbf{0} + \mathbf{n}\times\mathbf{0} = \mathbf{0}.$$

Hence,

$$\langle \mathbf{n}\times\nabla q, \mathbf{v}\rangle_\Gamma = \langle \mathbf{n}\times\nabla q, \mathbf{v}\rangle_{\Gamma_D} + \langle \mathbf{n}\times\nabla q, \mathbf{v}\rangle_{\Gamma_N} = \langle \mathbf{n}\times\nabla q, \mathbf{v}\rangle_{\Gamma_D}, \tag{23}$$

as we wanted to prove. □

Thus, we can further simplify the PPE to

$$\langle\nabla q, \nabla p\rangle = \left\langle\nabla q, \mathbf{f} - (\rho\nabla\mathbf{u})\mathbf{u} + 2(\nabla\mathbf{u})^\top\nabla\mu\right\rangle + \langle\mathbf{n}\times\nabla q, \mu\nabla\times\mathbf{u}\rangle_{\Gamma_D} - \langle q, \partial_t(\mathbf{n}\cdot\mathbf{g})\rangle_{\Gamma_D}. \tag{24}$$

Although at first glance we may seem to have ended up with a weak formulation containing only first-order derivatives, that is not exactly true. Since in most applications the viscosity depends on $\nabla^s\mathbf{u}$, the presence of $\nabla\mu$ in the variational formulation would induce higher smoothness requirements on the velocity interpolant and therefore prohibit the use of standard Lagrangian finite elements. As done in least-square finite element methods, this can be overcome by introducing $\mu$ as a continuous additional unknown, and recovering it weakly through a simple $L^2$ projection. In other words, given $\mathbf{u}$, we must find $\mu \in C^0\left(\bar{\Omega}\right)$ such that for all $\upsilon \in L^2(\Omega)$

$$\langle\upsilon, \mu\rangle = \langle\upsilon, \eta(\dot\gamma)\rangle, \tag{25}$$

which at the discrete level corresponds to a simple mass matrix solve. The idea of projecting the viscosity (or even the whole viscous tensor) as a means to avoid higher-order regularity requirements has been employed in similar works on non-Newtonian fluids [16,17].

### 3.2. Momentum equation

We are now left with the simpler task of designing an appropriate variational formulation for the momentum equation (7). The only difference in comparison to the Newtonian case is the viscous term $\mu\Delta\mathbf{u} + 2\nabla^s\mathbf{u}\nabla\mu$, which in our case has an additional part due to the spatial variation of the viscosity field. In order to obtain the desired natural boundary conditions (4), we need to apply integration by parts only to the Laplacian term. For a test function $\mathbf{w}\in[H^1(\Omega)]^d$, with $\mathbf{w}|_{\Gamma_D} = \mathbf{0}$, we have

$$\begin{aligned}
-\left\langle\mathbf{w}, \mu\Delta\mathbf{u} + 2\nabla^s\mathbf{u}\nabla\mu\right\rangle &= \langle\nabla\mathbf{u}, \nabla(\mu\mathbf{w})\rangle - \langle\mathbf{w}, (\mu\nabla\mathbf{u})\mathbf{n}\rangle_{\Gamma_N} - \left\langle\mathbf{w}, 2\nabla^s\mathbf{u}\nabla\mu\right\rangle\\
&= \langle\nabla\mathbf{u}, \mu\nabla\mathbf{w} + \mathbf{w}\otimes\nabla\mu\rangle - \langle\mathbf{w}, (\mu\nabla\mathbf{u})\mathbf{n}\rangle_{\Gamma_N} - \left\langle\mathbf{w}, 2\nabla^s\mathbf{u}\nabla\mu\right\rangle\\
&\equiv \langle\nabla\mathbf{w}, \mu\nabla\mathbf{u}\rangle + \langle\mathbf{w}, \nabla\mathbf{u}\nabla\mu\rangle - \langle\mathbf{w}, (\mu\nabla\mathbf{u})\mathbf{n}\rangle_{\Gamma_N} - \left\langle\mathbf{w}, 2\nabla^s\mathbf{u}\nabla\mu\right\rangle\\
&= \langle\nabla\mathbf{w}, \mu\nabla\mathbf{u}\rangle - \left\langle\mathbf{w}, (\nabla\mathbf{u})^\top\nabla\mu\right\rangle - \langle\mathbf{w}, (\mu\nabla\mathbf{u})\mathbf{n}\rangle_{\Gamma_N}.
\end{aligned}$$

Now that second-order differentiation has been completely eliminated from our weak formulation, we are in position to define our semi-discrete variational problem based on standard $C^0$ Lagrangian finite element spaces for the interpolation of $\mathbf{u}$, $p$ and $\mu$. The finite element spaces used for velocity, pressure and viscosity will be denoted by $X_h^u$, $X_h^p$ and $X_h^\mu$, respectively. Let $(\psi^\mu, \psi^p, \psi^u)$ be the shape functions for the corresponding finite element spaces; then, we shall use subscripts to refer to nodal values: e.g., $\psi_i^p$ denotes the pressure shape function associated to the $i$th pressure node.

### 3.3. Enforcing the pressure Dirichlet BCs

In terms of regularity aspects, the last issue to be addressed is the pressure Dirichlet BC. When trying to enforce (13) on a certain section of the outflow boundary $\Gamma_N$, we face a problem: in the discrete case, the quantity

$\zeta := \mu\nabla\mathbf{u} : (\mathbf{n}\otimes\mathbf{n} - \mathbb{I}) - \mathbf{h}\cdot\mathbf{n}$ will be discontinuous and, therefore, not well defined at the pressure nodes we want to fix. The solution proposed by Liu [5] and also used in the SRP method by Plasman et al. [15] is to project $\zeta$ onto $\Gamma_N$ such that the resulting quantity $\hat{\zeta}$ is continuous on each section of $\Gamma_N$. The cost of this projection on the boundary is negligible in comparison to the other projection/solution steps in the overall scheme.

## 4. Efficient iteration-free split-step schemes

In the present work, as in standard projection methods, we are interested in schemes that decouple velocity and pressure in order to allow efficient solution methods. Therefore, the pressure term in the momentum equation will be treated explicitly. Moreover, to further improve efficiency, we will focus on schemes that decouple also the velocity components. In the Newtonian case, this can be achieved by simply treating convection explicitly, whereas the viscous term can be kept implicit or semi-implicit [6]. This does not only reduce the size of the problem, but also results in a linearised system that does not require an iterative method such as Newton–Raphson or Picard. However, in the presence of nonlinear viscosity, it is necessary to treat the viscous term appropriately in order to keep the velocity components decoupled, as we will show next.

In the non-Newtonian case, the viscous trilinear form has two contributions: a typical weak Laplacian term $\langle\nabla\mathbf{w}, \mu\nabla\mathbf{u}\rangle$, and an asymmetric term due to the spatial gradient of the viscosity field. The latter can hinder efficiency, as it couples the velocity components even if the viscosity is linearised/extrapolated. Therefore, we treat such term explicitly, whereas for the symmetric one we use the linearised viscosity but the current velocity. When it comes to convection, there are two approaches that allow the decoupling: one is to treat the convective term in a fully explicit way [6], and the other one is to use an extrapolated convective velocity but the current velocity gradient. In the first-order (in time) case, this leads to the problem of finding $\mathbf{u}^{n+1} \in [X_h^u]^d$, with $\mathbf{u}^{n+1}|_{\Gamma_D} = \mathbf{g}^{n+1}$, such that

$$\left\langle \mathbf{w}, (\rho\nabla\mathbf{u}^{n+1})\mathbf{u}^n + \frac{\rho}{\Delta t}\mathbf{u}^{n+1}\right\rangle + \left\langle\nabla\mathbf{w}, \mu^n\nabla\mathbf{u}^{n+1}\right\rangle \\ = \left\langle \mathbf{w}, \mathbf{f}^{n+1} + \frac{\rho}{\Delta t}\mathbf{u}^n + \left(\nabla\mathbf{u}^n\right)^\top\nabla\mu^n\right\rangle + \left\langle\nabla\cdot\mathbf{w}, p^n\right\rangle + \langle\mathbf{w}, \mathbf{h}^{n+1}\rangle_{\Gamma_N} \tag{26}$$

for all $\mathbf{w} \in [X_h^u]^d$, with $\mathbf{w}|_{\Gamma_D} = \mathbf{0}$. Notice that this equation, along with the PPE and the viscosity projection, are linear algebraic systems, so that there is no need to iterate in order to find the solution. The overall first-order algorithm can be summarised as follows:

1. *Initialisation:*
   Compute the initial viscosity from Eq. (25), then the initial pressure from Eq. (24).
2. *Convection–diffusion steps:*
   For each of the $d$ velocity components, solve the convection–diffusion problem of finding $u_i^{n+1} \in X_h^u$, with $u_i^{n+1}|_{\Gamma_D} = g_i^{n+1}$, such that

$$\left\langle w, \rho\mathbf{u}^n\cdot\nabla u_i^{n+1} + \frac{\rho}{\Delta t}u_i^{n+1}\right\rangle + \left\langle\nabla w, \mu^n\nabla u_i^{n+1}\right\rangle \\ = \left\langle\frac{\partial w}{\partial x_i}, p^n\right\rangle + \left\langle w, f_i^{n+1} + \frac{\rho}{\Delta t}u_i^n + \frac{\partial\mathbf{u}^n}{\partial x_i}\cdot\nabla\mu^n\right\rangle + \langle w, h_i^{n+1}\rangle_{\Gamma_N} \\ \forall\ w \in X_h^u,\ \text{with}\ w|_{\Gamma_D} = 0. \tag{27}$$

3. *Viscosity projection:*
   Find $\mu^{n+1} \in X_h^\mu$ such that

$$\left\langle\upsilon, \mu^{n+1}\right\rangle = \left\langle\upsilon, \eta(\dot{\gamma}(\nabla^s\mathbf{u}^{n+1}))\right\rangle \quad \forall\ \upsilon \in X_h^\mu. \tag{28}$$

4. *Pressure BC projection (only if $\Gamma_N \neq \emptyset$):*
   Project $\zeta^{n+1} := \mu^{n+1}\nabla\mathbf{u}^{n+1} : (\mathbf{n}\otimes\mathbf{n} - \mathbb{I}) - \mathbf{n}\cdot\mathbf{h}^{n+1}$ such that the resulting quantity $\hat{\zeta}^{n+1}$ is continuous on $\Gamma_N$ (*cf.* Section 3.3).
5. *Pressure Poisson step:*
   Find $p^{n+1} \in X_h^p$, with $p^{n+1}|_{\Gamma_N} = \hat{\zeta}^{n+1}$, such that

$$\left\langle\nabla q, \nabla p^{n+1}\right\rangle = \left\langle\nabla q, \mathbf{f}^{n+1} - (\rho\nabla\mathbf{u}^{n+1})\mathbf{u}^{n+1} + 2(\nabla\mathbf{u}^{n+1})^\top\nabla\mu^{n+1}\right\rangle \\ + \left\langle\mathbf{n}\times\nabla q, \mu^{n+1}\nabla\times\mathbf{u}^{n+1}\right\rangle_{\Gamma_D} - \left\langle q, \mathbf{n}\cdot(\partial_t\mathbf{g})|_{t=t^{n+1}}\right\rangle_{\Gamma_D} \quad \forall\ q \in X_h^p,\ \text{with}\ q|_{\Gamma_N} = 0. \tag{29}$$

Notice that the pressure Poisson step is fully implicit, which is crucial for accuracy and stability.

### 4.1. Matrix problem

Let us denote by $\underline{u}_k^n$ the vector of nodal values of the $k$th velocity component at the $n$th time step — and analogously for $\underline{p}^n$ and $\underline{\mu}^n$. The main steps of the first-order marching scheme we just described can be written in matrix form as

$$\begin{aligned}
\left[\frac{\rho}{\Delta t}M^u + C(\mathbf{u}^n) + K^u(\mu^n)\right]\underline{u}_k^{n+1} &= \underline{f}^{n+1} + \frac{\rho}{\Delta t}M^u\underline{u}_k^n + \left[A^k(\mathbf{u}^n)\right]\underline{\mu}^n + B^k\underline{p}^n, \quad k = 1, \dots, d \\
M^\mu \underline{\mu}^{n+1} &= \underline{r}(\mathbf{u}^{n+1}), \\
K^p\underline{p}^{n+1} &= \underline{g}^{n+1} + \left[S(\mathbf{u}^{n+1}) + 2\sum_{k=1}^{d}\tilde{A}^k(\mathbf{u}^{n+1})\right]\underline{\mu}^{n+1} - \sum_{k=1}^{d}\left[\tilde{C}^k(\mathbf{u}^{n+1})\right]\underline{u}_k^{n+1},
\end{aligned}$$

where vectors $\underline{f}$ and $\underline{g}$ appropriately include the effects of body forces and BCs, and the remaining matrices are given as

$$\begin{aligned}
M_{ij}^\mu &= \left\langle \psi_i^\mu, \psi_j^\mu \right\rangle, & M_{ij}^u &= \left\langle \psi_i^u, \psi_j^u \right\rangle, \\
K_{ij}^p &= \left\langle \nabla\psi_i^p, \nabla\psi_j^p \right\rangle, & K_{ij}^u(\mu) &= \left\langle \nabla\psi_i^u, \mu\nabla\psi_j^u \right\rangle, \\
\tilde{C}_{ij}^k(\mathbf{u}) &= \left\langle \frac{\partial\psi_i^p}{\partial x_k}, \rho\mathbf{u}\cdot\nabla\psi_j^u \right\rangle, & C_{ij}(\mathbf{u}) &= \left\langle \psi_i^u, \rho\mathbf{u}\cdot\nabla\psi_j^u \right\rangle, \\
\tilde{A}_{ij}^k(\mathbf{u}) &= \left\langle \frac{\partial\psi_i^p}{\partial x_k}, \frac{\partial\mathbf{u}}{\partial x_k}\cdot\nabla\psi_j^\mu \right\rangle, & A_{ij}^k(\mathbf{u}_h) &= \left\langle \psi_i^u, \frac{\partial\mathbf{u}}{\partial x_k}\cdot\nabla\psi_j^\mu \right\rangle, \\
S_{ij}(\mathbf{u}) &= \left\langle \mathbf{n}\times\nabla\psi_i^p, \psi_j^\mu\nabla\times\mathbf{u} \right\rangle_{\Gamma_D}, & B_{ij}^k &= \left\langle \frac{\partial\psi_i^u}{\partial x_k}, \psi_j^p \right\rangle.
\end{aligned}$$

Therefore, all we need to solve are $d$ scalar convection–diffusion problems, one scalar mass matrix and one Poisson problem per time step. Also note that not only are the velocity components decoupled, but their system matrices are all identical. This simplifies computations and reduces assembly costs.

**Remark 2.** While the mass and diffusion matrices $M^u$ and $K^u$ are symmetric, the convective matrix $C$ is not. Therefore, when using iterative solvers, it may be advantageous to treat the convective term in a fully explicit manner so that the overall velocity–velocity matrix is symmetric.

### 4.2. Improving the conservation of mass

It is obvious that when the explicit incompressibility condition is replaced by the PPE, in the discrete case the resulting velocity will not be *exactly* divergence-free, which is in principle not an issue. Yet, as observed by Liu et al. [7], in problems with highly non-smooth solutions one can considerably improve stability by performing a standard Leray projection, as done in classical pressure correction methods [1,2]. After computing the velocity field $\mathbf{u}$ from Eq. (27), we can solve the Poisson problem

$$-\Delta\varphi = \nabla\cdot\mathbf{u} \quad \text{in} \quad \Omega, \tag{30}$$

$$\frac{\partial\varphi}{\partial n} = 0 \quad \text{on} \quad \Gamma_D, \tag{31}$$

$$\varphi = 0 \quad \text{on} \quad \Gamma_N, \tag{32}$$

then it is simple to verify that the modified velocity $\hat{\mathbf{u}} := \mathbf{u} + \nabla\varphi$ satisfies

$$\begin{aligned}
&\nabla\cdot\hat{\mathbf{u}} = 0 \quad \text{in} \quad \Omega, \\
&\hat{\mathbf{u}}\cdot\mathbf{n} = \mathbf{u}\cdot\mathbf{n} \quad \text{on} \quad \Gamma_D, \\
&\hat{\mathbf{u}}\cdot\mathbf{s} = \mathbf{u}\cdot\mathbf{s} \text{ on } \quad \Gamma_N,
\end{aligned}$$

for any tangential vector $\mathbf{s}$ on $\Gamma_N$. In other words, we end up with one velocity field $\mathbf{u}$ which is not divergence-free but satisfies the Dirichlet BCs fully, and another field $\hat{\mathbf{u}}$ which is divergence-free but only partially satisfies the prescribed BCs. As discussed by Guermond et al. [2], both velocity fields are expected to converge with the same rates, such that from an accuracy standpoint there is no objective reason to pick one over the other. The reason for performing the projection here is to keep divergence errors from potentially building up in time due to spatial discretisation errors (see Ref. [7] for details). Also notice that when using standard Lagrangian finite element spaces for $\varphi$, the resulting $\hat{\mathbf{u}}$ will be discontinuous. Therefore, the modified velocity has to be further projected onto a continuous space before it can be used in subsequent steps. This requires the additional solution of $d$ scalar mass matrix problems, and renders the resulting projected velocity only weakly divergence-free.

**Remark 3.** One key difference between the present framework and standard pressure correction methods is that here the projection step does not induce any artificial pressure boundary conditions, since we compute the pressure directly via the PPE, instead of using $\varphi$ to update it.

In light of the shortcomings related to the Leray projection, we use an approach that offers a compromise between the two variants discussed so far. As noted by Liu [5], if we apply the Leray projection always to the previous velocity $\mathbf{u}^n$ and use the modified velocity $\hat{\mathbf{u}}^n$ only in the acceleration term of the momentum equation, we can eliminate the computational overhead due to the projection. More precisely, for the temporally first-order scheme we write

$$\partial_t \mathbf{u}|_{t=t^{n+1}} \approx \frac{1}{\Delta t}\left(\mathbf{u}^{n+1} - \hat{\mathbf{u}}^n\right),$$

which leads to

$$\frac{\rho}{\Delta t}\left[\mathbf{u}^{n+1} - \left(\mathbf{u}^n + \nabla\varphi^n\right)\right] + \nabla p^n + \mathbf{r}^{n+1,n} = \mathbf{f}^{n+1},$$

where in $\mathbf{r}^{n+1,n}$ we group the nonlinear terms, for simplicity of presentation. We can combine $p^n$ and $\varphi^n$ appropriately to yield

$$\frac{\rho}{\Delta t}\left(\mathbf{u}^{n+1} - \mathbf{u}^n\right) + \nabla \hat{p}^n + \mathbf{r}^{n+1,n} = \mathbf{f}^{n+1},$$

with

$$\hat{p}^n := p^n - \frac{\rho}{\Delta t}\varphi^n. \tag{33}$$

Notice that this circumvents the need for projecting $\hat{\mathbf{u}}^n$ onto a continuous space, since the acceleration term has no spatial derivatives. We can further eliminate the additional Poisson step by combining Eqs. (30)–(32) with the original PPE (8)–(13) into a Poisson problem for the modified pressure $\hat{p}$:

$$-\Delta \hat{p}^n = \left[\nabla \times \left(\nabla \times \mathbf{u}^n\right)\right] \cdot \nabla\mu^n - \nabla \cdot \left[2\nabla^s\mathbf{u}^n\nabla\mu^n - (\rho\nabla\mathbf{u}^n)\mathbf{u}^n\right] - \nabla \cdot \mathbf{f}^n - \frac{\rho}{\Delta t}\nabla \cdot \mathbf{u}^n, \tag{34}$$

with the same boundary conditions as in the original PPE, since $\varphi$ has zero Dirichlet (32) and Neumann (31) BCs. It is also worth remarking that, since $\varphi$ is zero for the exact solution, the quantity $\hat{p}$ is still a consistent approximation for the pressure. Therefore, the only modification needed with respect to the basic algorithm (27)–(29) is to add to the right-hand side of the PPE the term $(-\rho/\Delta t)\, \nabla \cdot \mathbf{u}^{n+1}$, which at the matrix level becomes

$$-\frac{\rho}{\Delta t}\sum_{k=1}^{d}(B^k)^\top \underline{u}_k^{n+1}.$$

This technique is sometimes referred to as *divergence damping* [9,13], as it can be seen as penalising the PPE with large values of the velocity divergence. This simple approach considerably improves the conservation of mass and temporal stability of the overall scheme. In the next subsection we present the generalisation to higher-order, potentially adaptive temporal discretisations.

### *4.3. Higher-order schemes with variable time step*

It can be desirable, especially when using higher-order finite element spaces, to also increase the temporal order of discretisation, and in particular allowing for variable time steps $\Delta t^n = t^{n+1} - t^n$ for efficiency. This can be

**Table 1**
Backward differentiation and extrapolation coefficients of order $m = 2$ [26].

| $j$ | 0 | 1 | 2 |
|---|---|---|---|
| $\alpha_j^m$ | $\frac{2\Delta t^n+\Delta t^{n-1}}{\Delta t^n(\Delta t^n+\Delta t^{n-1})}$ | $-\frac{\Delta t^n+\Delta t^{n-1}}{\Delta t^n\,\Delta t^{n-1}}$ | $\frac{\Delta t^n}{\Delta t^{n-1}(\Delta t^n+\Delta t^{n-1})}$ |
| $\beta_j^m$ | – | $1+\frac{\Delta t^n}{\Delta t^{n-1}}$ | $-\frac{\Delta t^n}{\Delta t^{n-1}}$ |

attained by using a combination of higher-order backward differentiation (BDF) schemes

$$\partial_t \mathbf{u}|_{t=t^{n+1}} \approx \alpha_0^m \mathbf{u}^{n+1} + \sum_{j=1}^{m} \alpha_j^m \hat{\mathbf{u}}^{n+1-j},$$

and suitable extrapolation formulas

$$\mathbf{u}^{n+1} \approx \mathbf{u}^\star := \sum_{j=1}^{m} \beta_j^m \mathbf{u}^{n+1-j} \tag{35}$$

of order $m$, with coefficients $\alpha_j^m$ and $\beta_j^m$ given in Table 1 for $m = 2$ [26]. Again, a modified pressure is defined similar to (33) as

$$\hat{p}^\star := p^\star + \rho \sum_{j=1}^{m} \alpha_j^m \varphi^{n+1-j}, \tag{36}$$

incorporating both the extrapolated pressure $p^\star$ and Leray-projection contributions to past velocities. Let us write the right-hand side of the final PPE (24) at time $t^{n+1}$ as

$$\begin{aligned} l(\mathbf{u}^{n+1}, \mu^{n+1}, q) = {} & \left\langle \nabla q, \mathbf{f}^{n+1} - (\rho \nabla \mathbf{u}^{n+1})\mathbf{u}^{n+1} + 2(\nabla \mathbf{u}^{n+1})^\top \nabla \mu^{n+1} \right\rangle \\ & + \left\langle \mathbf{n} \times \nabla q, \mu^{n+1} \nabla \times \mathbf{u}^{n+1} \right\rangle_{\Gamma_D} - \left\langle q, \mathbf{n} \cdot \sum_{j=0}^{m} \alpha_j^m \mathbf{u}^{n+1-j} \right\rangle_{\Gamma_D}, \end{aligned}$$

with a BDF formula applied to $(\mathbf{n} \cdot \partial_t \mathbf{u})|_{\Gamma_D}$. We can insert definition (36) and linearly combine PPEs, similarly as in Eq. (34), to obtain an equation for the modified pressure $\hat{p}^\star$ to be used in the next time step:

$$\begin{aligned} \left\langle \nabla q, \nabla \hat{p}^\star \right\rangle &= \left\langle \nabla q, \nabla \left( p^\star + \rho \sum_{j=1}^{m} \alpha_j^m \varphi^{n+1-j} \right) \right\rangle \\ &= \sum_{j=1}^{m} \beta_j^m l(\mathbf{u}^{n+1-j}, \mu^{n+1-j}, q) + \left\langle q, \rho \sum_{j=1}^{m} \alpha_j^m \nabla \cdot \mathbf{u}^{n+1-j} \right\rangle. \end{aligned} \tag{37}$$

The final higher-order scheme is initialised by first computing the initial viscosity from Eq. (25) and then the initial pressure from Eq. (24). The resulting algorithm reads

1. *Initialisation:*
   Use lower-order schemes with increasing order ($\hat{m} = 1, 2, \ldots, m$) until having all the quantities required to proceed with the $m$th-order scheme.
2. *Convection–diffusion steps:*
   For each of the $d$ velocity components, solve the convection–diffusion problem of finding $u_i^{n+1} \in X_h^u$, with $u_i^{n+1}|_{\Gamma_D} = g_i^{n+1}$, such that for all $w \in X_h^u$, with $w|_{\Gamma_D} = 0$, there holds

$$\begin{aligned} & \left\langle w, \rho \mathbf{u}^\star \cdot \nabla u_i^{n+1} + \rho \alpha_0^m u_i^{n+1} \right\rangle + \left\langle \nabla w, \mu^\star \nabla u_i^{n+1} \right\rangle \\ & \quad = \left\langle \frac{\partial w}{\partial x_i}, \hat{p}^\star \right\rangle + \left\langle w, f_i^{n+1} + \rho \sum_{j=1}^{m} \left( \alpha_j^m u_i^{n+1-j} \right) + \frac{\partial \mathbf{u}^\star}{\partial x_i} \cdot \nabla \mu^\star \right\rangle + \langle w, h_i^{n+1} \rangle_{\Gamma_N}. \end{aligned} \tag{38}$$

3. *Viscosity projection:*
   Find $\mu^{n+1} \in X_h^\mu$ such that

$$\left\langle \upsilon, \mu^{n+1} \right\rangle = \left\langle \upsilon, \eta(\dot{\gamma}(\nabla^s \mathbf{u}^{n+1})) \right\rangle \quad \forall\, \upsilon \in X_h^\mu. \tag{39}$$

4. *Time-step update:*
   If using adaptive stepping, compute $\Delta t^n$ and update the coefficients $\alpha_j^m$ and $\beta_j^m$.
5. *Pressure BC projection (only if $\Gamma_N \neq \emptyset$):*
   Project $\zeta^\star := \sum_{j=1}^m \left[ \beta_j^m \left( \mu^{n+2-j} \nabla \mathbf{u}^{n+2-j} : (\mathbf{n} \otimes \mathbf{n} - \mathbb{I}) - \mathbf{n} \cdot \mathbf{h}^{n+2-j} \right) \right]$ such that the resulting quantity $\hat{\zeta}^\star$ is continuous on $\Gamma_N$.
6. *Pressure Poisson step*
   Update the modified pressure for the next time step by finding $\hat{p}^\star \in X_h^p$, with $\hat{p}^\star|_{\Gamma_N} = \hat{\zeta}^\star$, such that

$$\left\langle \nabla q, \nabla \hat{p}^\star \right\rangle = \sum_{j=1}^m \beta_j^m l(\mathbf{u}^{n+2-j}, \mu^{n+2-j}, q) + \left\langle q, \rho \sum_{j=1}^m \alpha_j^m \nabla \cdot \mathbf{u}^{n+2-j} \right\rangle$$
$$\forall\, q \in X_h^p, \text{ with } q|_{\Gamma_N} = 0. \tag{40}$$

### *4.4. Comparison with projection schemes*

It is important to state that the method we just presented – as well as its homogeneous Newtonian version proposed by Liu [5] – is *not* a projection or pressure correction method, but a PPE-based one. These two classes of methods are closely related [5], but exhibit important differences. In projection methods, incompressibility is enforced by projecting the velocity prediction onto a divergence-free space, and as a by-product one gets a scalar quantity used to update the pressure. In PPE-based schemes, incompressibility is implicitly enforced through the solution of an appropriate Poisson problem for the pressure, which reduces computational cost by eliminating a number of steps in the algorithm. Another important practical difference is that rotational pressure correction schemes require LBB-compatible velocity–pressure spaces [15], which is not the case for PPE-based methods [8]. The price we pay for having a Poisson problem for the pressure itself (rather than for the pressure correction) is having to deal with a considerably more complex right-hand side with third-order velocity derivatives. Hence, the most challenging aspect of dealing with a PPE-based system is circumventing its higher-order regularity requirements to allow standard finite element discretisations, as shown in Section 3.

## 5. Different versions of the splitting scheme and their solution

After the initialisation phase, steps (39)–(40) in the higher-order scheme require the solution of a scalar $L^2$ projection, $d$ convection–diffusion equations, a projection on each segment of the Neumann boundary, and a final scalar Poisson solve to retrieve the modified pressure $\hat{p}^\star$. The proposed scheme allows the decoupling of velocity components independent of extrapolation or time integration order, rendering the application of state-of-the-art parallel algebraic multigrid (AMG) methods even more effective. Depending on the linearisation of convective and viscous terms in the momentum equation, several possibilities arise. One may choose to

(i) linearise the viscosity via $\mu^\star$ and the convective term via $(\nabla \mathbf{u}^{n+1})\mathbf{u}^\star$,
(ii) proceed as in (i) and additionally use $(\nabla \mathbf{u}^\star)^\top \nabla \mu^\star$, or
(iii) linearise the viscous term as in (ii), but treat convection explicitly via $(\nabla \mathbf{u}^\star)\mathbf{u}^\star$.

These choices may in fact have several implications. While the presented scheme (39)–(40) – which corresponds to (ii) – decouples velocity components as opposed to (i), it also treats part of the viscous contribution explicitly, possibly implying time-step restrictions. Strategy (iii) enables, additionally to the decoupling of the velocity components, the use of the conjugate gradient (CG) method, since the resulting system matrix will be symmetric and positive definite. Variants (i) and (ii), on the other hand, are not symmetric and are therefore tackled with a flexible generalised minimal residual method (FGMRES) herein. One may further improve efficiency by lumping the coefficient matrix in the viscosity projection step (39). This virtually eliminates the added cost due to introducing the viscosity as an additional unknown, without jeopardising stability or accuracy. The methods proposed here were implemented in *deal.II* [27], interfacing *Trilinos' ML* package [28] in order to apply a single AMG V-cycle as a preconditioner in each iteration of the linear solver.

Another variant of our split-step scheme arises from considering the stress-divergence form of the viscous forces, with $(2\mu\nabla^s\mathbf{u} - p\mathbb{I})\mathbf{n} = \mathbf{t}$ as natural BC on $\Gamma_N$. In that case, the strong (8) and weak (24) forms of the PPE remain the same, and we only need to change the pressure Dirichlet BC (13) to

$$p = \mu\nabla\mathbf{u} : (2\mathbf{n}\otimes\mathbf{n} - \mathbb{I}) - \mathbf{t}\cdot\mathbf{n} \quad \text{on} \quad \Gamma_N \times [0, T], \tag{41}$$

and the weak form of the momentum equation to

$$\langle \mathbf{w}, \rho\partial_t\mathbf{u} + (\rho\nabla\mathbf{u})\mathbf{u}\rangle + \left\langle \mu\nabla\mathbf{w}, \nabla\mathbf{u} + (\nabla\mathbf{u})^\top \right\rangle - \langle \nabla\cdot\mathbf{w}, p\rangle = \langle \mathbf{w}, \mathbf{f}\rangle + \langle \mathbf{w}, \mathbf{t}\rangle_{\Gamma_N}, \tag{42}$$

where $\mathbf{t}$ corresponds to prescribed normal tractions. In order to decouple the velocity components in the split-step scheme, we must treat the term $\langle \nabla\mathbf{w}, (\mu\nabla\mathbf{u})^\top\rangle$ explicitly. Although we herein focus on the (generalised) Laplacian form due to its more appropriate natural BCs for open outlets, the stress-divergence formulation should be used whenever interface forces play a role, as in fluid–structure interaction and two-phase flow applications. It may also improve mass conservation in stabilised methods based on relaxation of incompressibility. In fact, using the stress-divergence form in combination with the PPE gives the heat equation

$$\partial_t\phi - \nabla\cdot(2\nu\nabla\phi) = 0 \tag{43}$$

for the divergence $\phi$, i.e., we get twice as much diffusion as in the Laplacian case (18). This suggests that, at the discrete level, the stress-divergence formulation might indeed result in improved incompressibility and stability, although this is not what our numerical experiments indicate (*cf.* Fig. 7). For a theoretical and numerical comparison between these two forms of the Navier–Stokes equations, we refer the reader to [29,30].

Unless when otherwise stated, we use the (generalised Laplacian) variant (ii) with full (i.e., not lumped) viscosity projection in our tests. A numerical study comparing multiple variants and different discretisation orders with respect to temporal stability and solver performance is conducted in Section 6.2. There, numerical evidence points to variant (ii) being a worthwhile compromise between fast solution and temporal stability, resulting in a standard CFL condition as observed in the Newtonian case [5].

## 6. Numerical examples

In this section, we assess the accuracy and stability of our new time-splitting framework by tackling simple problems with analytical solutions, as well as some classical benchmarks. One of the advantages of the present approach is the possibility to use finite element pressure–velocity pairs that are not LBB-compatible (see Remark 4 below). We consider first-order simplicial and tensor-product-based elements, denoted respectively as $P_1P_1$ and $Q_1Q_1$, and also quadrilateral/hexahedral Taylor–Hood elements ($Q_2Q_1$). The space chosen for the viscosity approximation is the same as the pressure space in each case.

**Remark 4.** While the formal numerical analysis for $H^1$-conforming finite element spaces (equal-order or not) is still an open question even for the Newtonian case, there is strong numerical evidence indicating the good stability properties of PPE-based schemes [7–13].

### *6.1. Manufactured solutions*

Let us start by tackling simple examples with manufactured solutions and a pure Dirichlet boundary, that is, $\Gamma_N = \emptyset$. While in most numerical examples we apply the divergence damping introduced in Section 4.2, here this term is dropped in order to yield a clearer convergence study. In the square domain $\Omega = (0, H)^2$, $H = 1$, we consider solutions of the type

$$p = 2\sin(2 - 2x_1)f(t), \quad \mathbf{u} = \begin{Bmatrix} f(t)\sin(2x_2)\sin^2(x_1) \\ -f(t)\sin(2x_1)\sin^2(x_2) \end{Bmatrix},$$

under a prototypical shear-thinning rheological law given by

$$\eta(\dot{\gamma}) = \left[1 + (10\dot{\gamma})^2\right]^{-1/4},$$

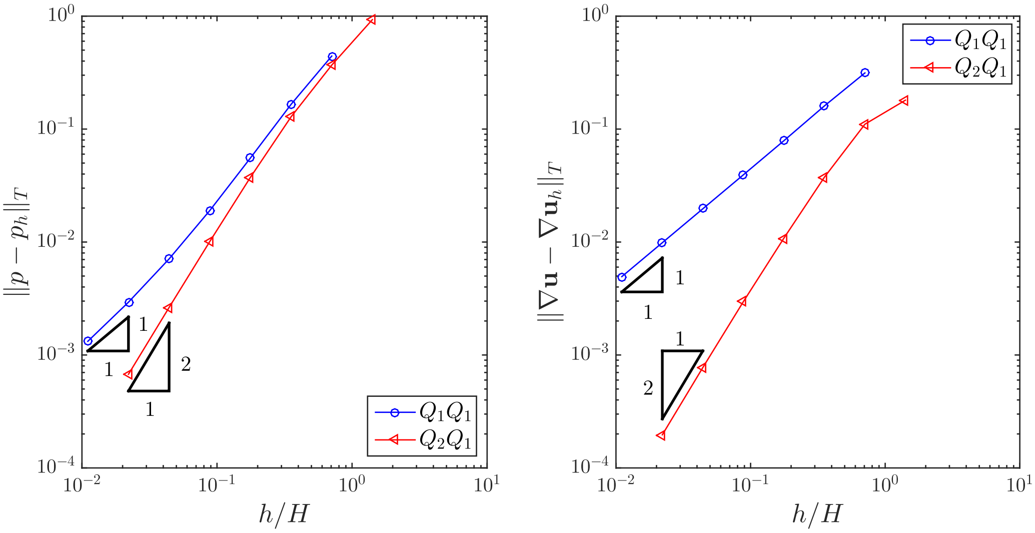

**Fig. 1.** Manufactured solution: spatial convergence study for different element types.

chosen for simplicity. We "ramp up" the solution considering different functions $f(t)$, and the resulting boundary data $\mathbf{g}$ and body force $\mathbf{f}$ are computed accordingly. First, in order to test the spatial accuracy of our method, we choose

$$f(t) = 1 - e^{-2t},$$

so that the solution tends to a steady state as $t \to \infty$. Starting with a coarse mesh containing four identical square elements, several levels of uniform spatial refinement are applied. The spatial approximation errors are computed through the relative norms

$$\|p - p_h\|_T := \left(\frac{\|p - p_h\|_{L^2(\Omega)}}{\|p\|_{L^2(\Omega)}}\right)\Bigg|_{t=T} \quad \text{and} \quad \|\nabla\mathbf{u} - \nabla\mathbf{u}_h\|_T := \left(\frac{\|\nabla\mathbf{u} - \nabla\mathbf{u}_h\|_{L^2(\Omega)}}{\|\nabla\mathbf{u}\|_{L^2(\Omega)}}\right)\Bigg|_{t=T},$$

and the spatial mesh size is denoted by $h$. The final time is selected as $T = 10$, at which point we have $1-f \approx 2\times10^{-9}$. We use the first-order temporal discretisation (BDF1) with a fixed time-step size $\Delta t = 10^{-2}$. Since we are considering a solution with negligible time residual at $t = T$, the temporal error should not contaminate the spatial order of accuracy, even when using a large (but still stable) step size and a low-order stepping scheme. The results of the spatial convergence study are shown in Fig. 1. As expected, the $Q_1Q_1$ pair yields linear spatial convergence, whereas the Taylor–Hood elements converge quadratically.

In order to assess the temporal accuracy of the first- and second-order stepping schemes, we now use a periodic function $f(t) = \sin^2(t)$ in the interval $t \in [0, 10]$ and consider the maximum error in the space–time domain $Q := \Omega\times(0, T]$, that is, using the $L^\infty(Q)$ norm. In this case we use the finest spatial mesh considered in the previous study, and refine the time step uniformly by halving it from $\Delta t = 0.16$ all the way down to $\Delta t = 0.00125$. The first-order temporal scheme is combined with the $Q_1Q_1$ elements, whereas for the second-order stepping scheme (BDF2) we use the $Q_2Q_1$ elements. The results are shown in Fig. 2, where the expected orders of convergence are verified. It is worth noting that in these examples the CFL number has ranged from 0.1 to almost 15, which speaks to the good stability properties of the methods used here.

Another relevant test regards how lumping the coefficient matrix $M^\mu$ in the viscosity projection affects the accuracy of the approximation. The comparison between the standard and lumped versions of the first-order splitting scheme with $Q_2Q_1$ elements is shown in Fig. 3. The pressure approximation is virtually unaffected, whereas the velocity convergence experiences a mild slope degradation at the finest levels. This indicates that using a lumped projection can be a simple way to improve efficiency in practice.

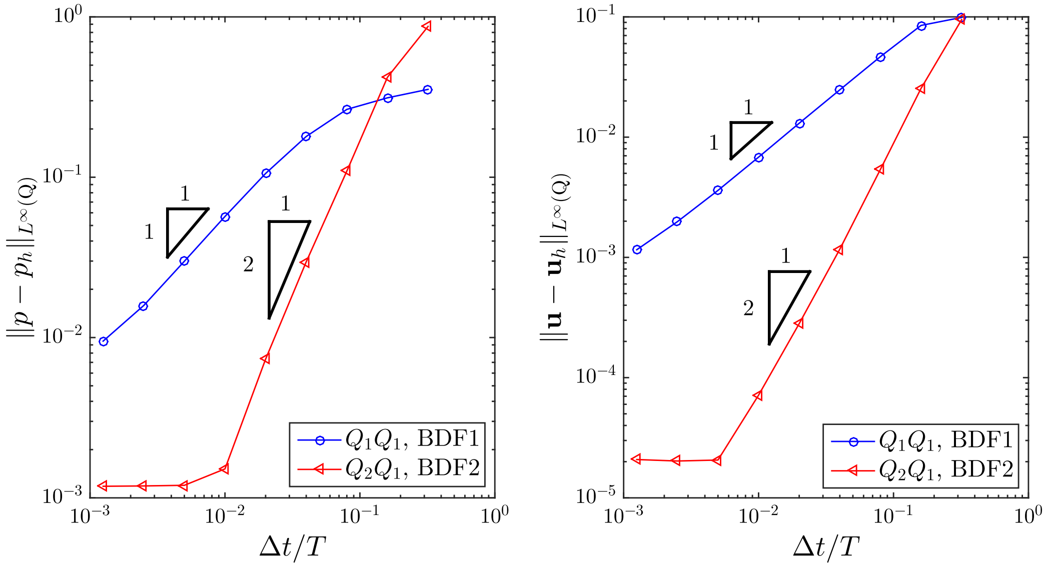

**Fig. 2.** Manufactured solution: temporal convergence study for different types of discretisation.

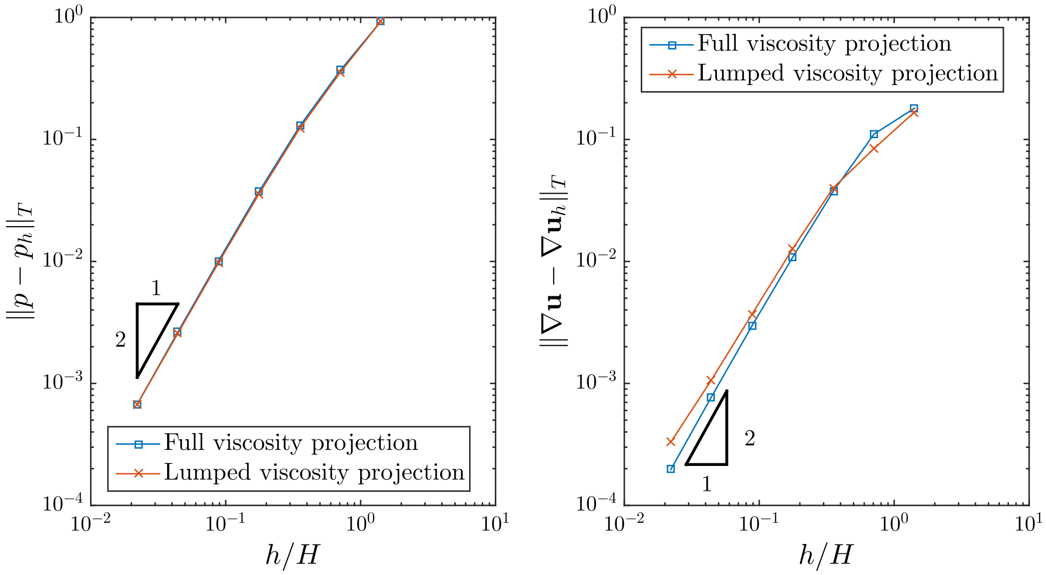

**Fig. 3.** Manufactured solution: impact on the spatial accuracy due to lumping the viscosity projection.

### *6.2. Lid-driven cavity flow*

To assess the temporal stability of the various possible linearisation approaches, the well-established regularised lid-driven cavity problem [31] in two spatial dimensions is considered. In the domain $\Omega = (0, L)^2$, the lid at $x_2 = L$ moves horizontally with a velocity $u_1 = U\xi(t)\chi(x_1)$ given by

$$\xi(t) = \begin{cases} \sin^2\left(\frac{\pi t}{2\tau}\right) & \text{for } t \leq \tau, \\ 1 & \text{otherwise,} \end{cases} \tag{44}$$

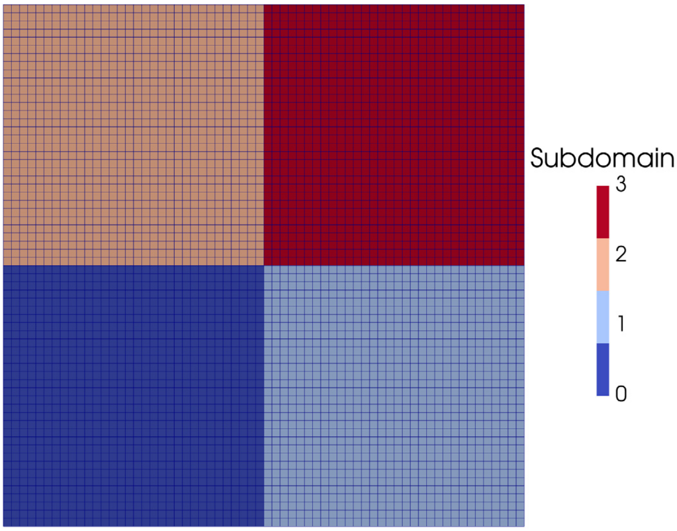

**Fig. 4.** Cavity flow benchmark: distributed computational mesh with 64 × 64 $Q_2Q_1$ elements.

$$\chi(x_1) = \begin{cases} 1-\cos^4\left(\frac{\pi x_1}{2l}\right) & \text{for } x \le l, \\ 1-\cos^4\left(\frac{\pi}{2l}(x_1-L)\right) & \text{for } x_1 \ge L-l, \\ 1 & \text{otherwise,} \end{cases} \tag{45}$$

with $\tau = 1.0$ s and the regularisation width $l = L/10$ resolving the ambiguity at the upper corner nodes. No-slip conditions are enforced on the remainder of the boundary. We use the popular Carreau model [32]

$$\eta(\dot{\gamma}) = \mu_\infty + (\mu_0 - \mu_\infty)\left[1 + (\lambda\dot{\gamma})^2\right]^{\frac{n-1}{2}}, \tag{46}$$

with parameters $\mu_\infty = 1.0$ mPa·s, $\mu_0 = 100$ mPa·s, $\lambda = 10$ s, $n = 0.25$ and fluid density $\rho = 1.0$ kg/m$^3$. Furthermore, considering $L = 1.0$ m and $U = 10$ m/s leads to a Reynolds number of $\mathrm{Re} = \rho U L/\mu_\infty \approx 10^4$. Rather coarse, uniform meshes of 128 × 128 equal-order $Q_1Q_1$ elements and 64 × 64 $Q_2Q_1$ elements are used, resulting in the same number of velocity nodes. The mesh is deliberately not tailored for boundary layers, so that the schemes can be tested in the presence of possibly large gradients. This numerical test is designed to challenge our solvers, featuring both high Reynolds numbers and having the viscosity vary by two orders of magnitude. Both configurations are subdivided into four subdomains, as depicted in Fig. 4 for the $Q_2Q_1$ case. A quasi-stationary solution as shown in Fig. 5 is recovered, showcasing here exemplarily one obtained using $Q_2Q_1$ elements and step sizes chosen such that the maximum element CFL number is approximately 5 according to the definition

$$\mathrm{CFL}_e = \max_i \left\{ \frac{|u_i|\Delta t}{h_i} \right\}, i = 1, \ldots, d, \tag{47}$$

with $h_i$ being the element length in direction $x_i$ divided by the polynomial degree of the velocity space. We also define the element Reynolds number $\mathrm{Re}_e$ as

$$\mathrm{Re}_e = \max_i \left\{ \frac{\rho|u_i|h_i}{\mu_\infty} \right\}, i = 1, \ldots, d. \tag{48}$$

We start by comparing the first-order schemes under different linearisation approaches. We plot the iteration counts $N_\mu$ of the viscosity projection step, $N_p$ of the PPE system, and either $N_{\mathbf{u}}$ (when considering the coupled system) or $\bar{N}_{u_i}$ (when solving for individual velocity components, if admissible). A CG scheme is used for all solves, except in the velocity system when the asymmetric diffusive or convective term is treated implicitly, in which cases an FGMRES solver is considered. A relative convergence criterion is defined, stopping linear iterations when the residual has decreased by a factor of $10^6$.

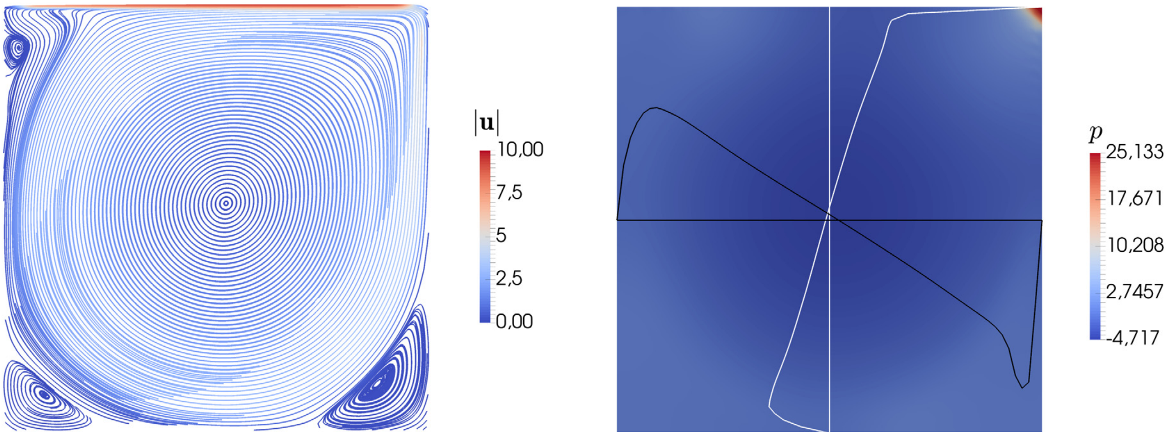

**Fig. 5.** Quasi-stationary cavity flow solution: selected streamlines coloured by velocity norm in m/s (left), and pressure $p$ in N/m$^2$ and scaled velocity components $0.05u_1$ and $0.1u_2$ in m/s (right).

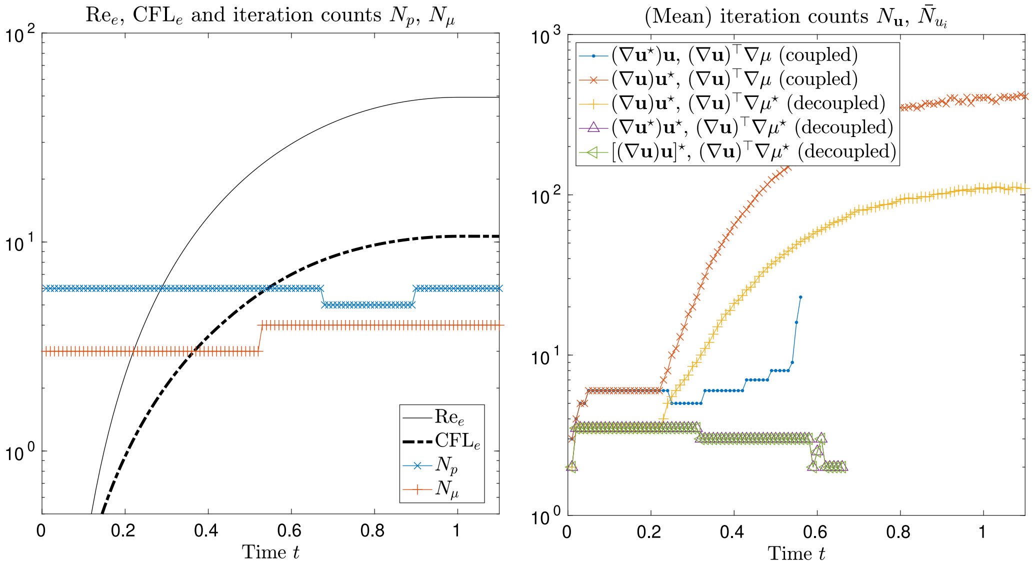

**Fig. 6.** Discretisation of the lid-driven cavity problem via BDF1 and $Q_1Q_1$ elements: local Re and CFL numbers and iteration counts of PPE ($N_p$) and viscosity projection ($N_\mu$) on the left; comparison of semi-implicit variants of the convective term $(\nabla\mathbf{u})\mathbf{u}$ and coupling viscous term $(\nabla\mathbf{u})^\top\nabla\mu$ on the right.

As can be seen in Fig. 6, all schemes remain stable under a standard CFL condition. The element CFL number reaches $\approx 10$ at $t = 1.0$ s, indicating good stability properties, while the element Reynolds number remains below 50. Iteration counts in the projection and PPE steps are very low for all considered variants, indicating the good performance of the AMG-preconditioned CG solver.

Concerning the velocity system(s), we observe that treating the convective velocity semi-implicitly (keeping it on the left-hand side) increases temporal stability, while increasing the iteration count in the FGMRES solver as the CFL number increases. This is well known from literature and therefore expected. However, considering the variant $(\nabla\mathbf{u}^\star)\mathbf{u}$ with semi-implicit viscous terms coupling the velocity components leads to sudden divergence in this specific setting and is thus considered less suitable. Linearising the asymmetric viscous contribution $\langle\mathbf{w}, (\nabla\mathbf{u})^\top\nabla\mu\rangle$ and shifting it to the right-hand side does not seem to cause any further restriction on the time-step size, but allows a decoupled solution of velocity components, hence those variants are indicated by “decoupled”. Conversely, “coupled” simply refers to schemes that do not allow solving the velocity components individually. Moreover,

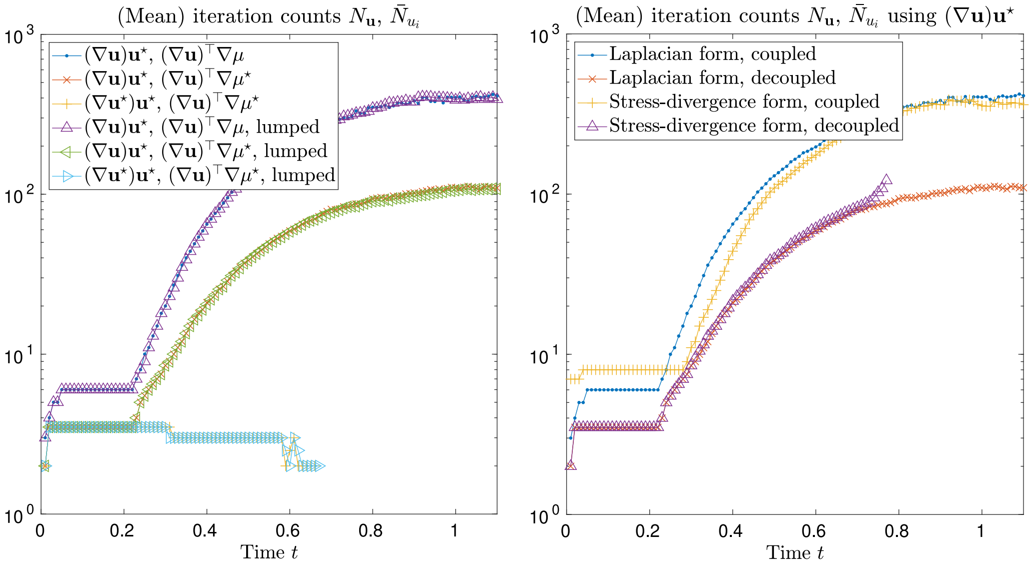

**Fig. 7.** Discretisation of the lid-driven cavity problem via BDF1 and $Q_1Q_1$ elements: influence of lumping the mass matrix (in the viscosity projection) using the Laplacian form (left); and comparison of Laplacian and stress-divergence forms (right).

variants with fully explicit convection may be of interest, using $(\nabla\mathbf{u}^\star)\mathbf{u}^\star$ or $[(\nabla\mathbf{u})\mathbf{u}]^\star$ (the superscripted star indicates extrapolation, *cf.* Section 4.3). In the current example, these variants show least temporal stability, but allow the use of a CG solver, thus yielding very low iteration counts.

The influence on stability when lumping the mass matrix in the viscosity projection step is negligible, as seen in Fig. 7. In the present example, stability is not impaired and almost identical iteration counts for all variants are attained. Thus, depending on the problem at hand, this scalar mass-matrix solve may be spared completely and replaced by a simple vector scaling.

Considering the stress-divergence form of the momentum equation, one may again derive coupled or decoupled variants by semi-implicitly treating the viscous stress terms. In the tests in Fig. 7, only the most stable variant of the convective term using $(\nabla\mathbf{u})\mathbf{u}^\star$ is considered, while the viscous term is treated such that individual velocity components are decoupled or not. All variants are stable and yield low iteration counts for small $\mathrm{CFL}_e$, but at $\mathrm{CFL}_e \approx 8.5$ the decoupled version of the stress-divergence form diverges, which does not happen with the decoupled Laplacian variant. This behaviour may be linked to the linearisations needed in order to decouple the components: in the Laplacian form, $\langle\mathbf{w}, (\nabla\mathbf{u})^\top\nabla\mu\rangle$ is treated explicitly as opposed to $\langle\nabla\mathbf{w}, (\mu\nabla\mathbf{u})^\top\rangle$ in the stress-divergence form; depending on the problem settings, the relative magnitude of these terms will differ, rendering one or the other formulation more stable.

When using a second-order temporal discretisation, i.e., BDF2 combined with linear extrapolation, the same trends are observed up to a maximum local CFL number of around 9 — by design just below the maximum CFL number reached in this problem. With BDF2, the semi-implicit schemes need smaller time-step sizes compared to their linear counterparts, but the semi-implicit viscous term does not decrease temporal stability noticeably (see Fig. 8). The scheme with explicit convection does perform rather well until $\mathrm{CFL}_e \approx 2$, but solution quality is already noticeably degraded before divergence due to temporal instability. The effects when applying mass lumping or using the stress-divergence form are similar to those already reported for BDF1 and are thus omitted.

When a second-order temporal scheme combined with $Q_2Q_1$ elements is employed, the observed trends do not differ from previous settings, as seen in Fig. 9. It is worth noting, however, that this finite element pair is stable in the inf–sup sense, which was not the case for the previously considered $Q_1Q_1$ interpolation. Summing up, these numerical tests indicate that a standard CFL condition is sufficient for temporal stability. Schemes with first order in time show better stability properties, while higher-order schemes result in smaller, yet still reasonable time-step sizes. Semi-implicit treatment of both the convective and viscous terms have not led to further stability issues, but fully

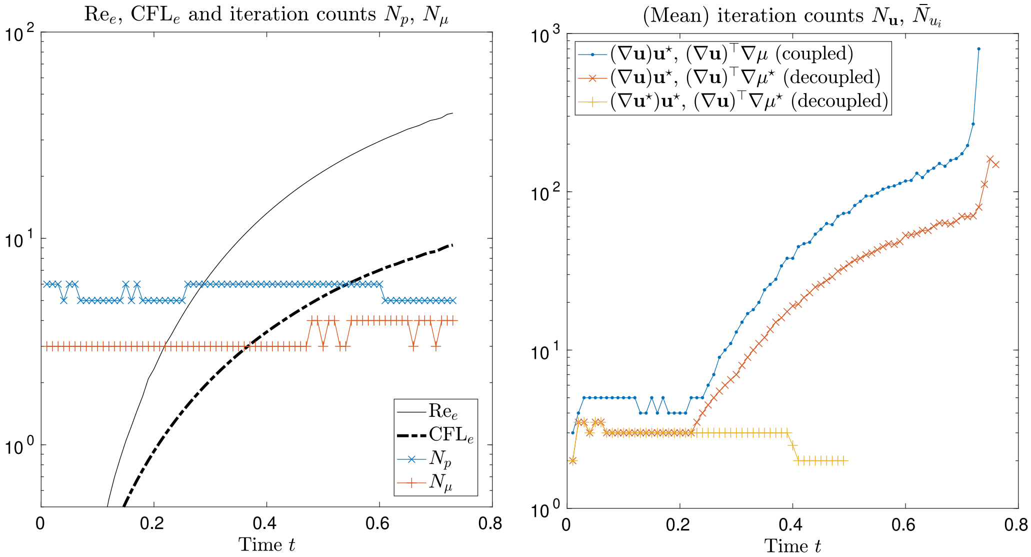

**Fig. 8.** Discretisation of the lid-driven cavity problem via BDF2 and $Q_1Q_1$ elements: local Re and CFL numbers and iteration counts of PPE ($N_p$) and viscosity projection ($N_\mu$) on the left; comparison of semi-implicit variants of the convective term $(\nabla\mathbf{u})\mathbf{u}$ and coupling viscous term $(\nabla\mathbf{u})^\top\nabla\mu$ on the right.

explicit schemes restricted the time-step size further. Lumping the mass matrix in the viscosity projection step has been found to smooth sharp gradients in viscosity and not alter stability properties in the current example. Looking at the decoupled variants, the stress-divergence form has led to divergence of the solver somewhat earlier than when employing the Laplacian form. Neither form seems to be clearly superior, though, since the slight differences are also likely influenced by the problem settings. As also observed by Liu [5], the velocity–pressure finite element pairing does not exhibit any LBB-like restriction, which is a major advantage of the method in comparison to standard rotational projection schemes. However, an in-depth theoretical analysis regarding these schemes is still a relevant open problem. The numerical evidence presented herein strongly indicates favourable stability properties, but further analysis lies beyond the scope of this contribution.

### *6.3. Carreau fluid past a backward-facing step*

We now consider another problem with less smooth solution, namely, the classical backward-facing step benchmark [33,34]. The popular setup proposed by Choi and Barakat [33] considers the Carreau rheological model (46) with the hemodynamic parameters $\rho = 1060$ kg/m$^3$, $\mu_\infty = 3.5$ mPa·s, $\mu_0 = 250$ mPa·s, $n = 0.25$ and $\lambda = 50$ s. The geometric dimensions (*cf.* Fig. 10) are $L_2 = 2L_1 = 20H$, $s = 0.9423H$ and $H = 5.2$ mm. In order to compare our results to the stationary solutions reported by Choi and Barakat [33], we use a parabolic inlet profile with flow rate $Q(t)$ ramped up smoothly from zero to $Q_{\max}$ as

$$\frac{Q(t)}{Q_{\max}} = \begin{cases} \sin^2\left(\frac{\pi t}{2\tau}\right) & \text{for } t \in [0, \tau], \\ 1 & \text{for } t \in [\tau, T], \end{cases}$$

with $\tau = 0.3$ s and $T = 2$ s. For the outlet we set the usual zero pseudo-traction condition $(\mu\nabla\mathbf{u} - p\mathbb{I})\mathbf{n} = \mathbf{0}$. Choi and Barakat [33] considered various values for the Reynolds number, which they define as $\mathrm{Re} = 2\rho Q_{\max}/\mu_\infty$.

For this example we employ a first-order method in both space and time, using triangular elements for the spatial discretisation. The mesh is structured, with the element length equal to $H/40$ in the horizontal direction. The elements above and below the step's corner have a vertical length equal to $H/40$ and $s/40$, respectively. This results in a total of 160,000 elements and 81,281 nodes. The time-step size is set as $\Delta t = 5 \times 10^{-4}$ s for all cases. The comparison between our results and the reference solution, in terms of the reattachment length $r$, is depicted in Fig. 11. The results are in general in very good agreement. For $\mathrm{Re} = 300$ there is a 5% difference, which might

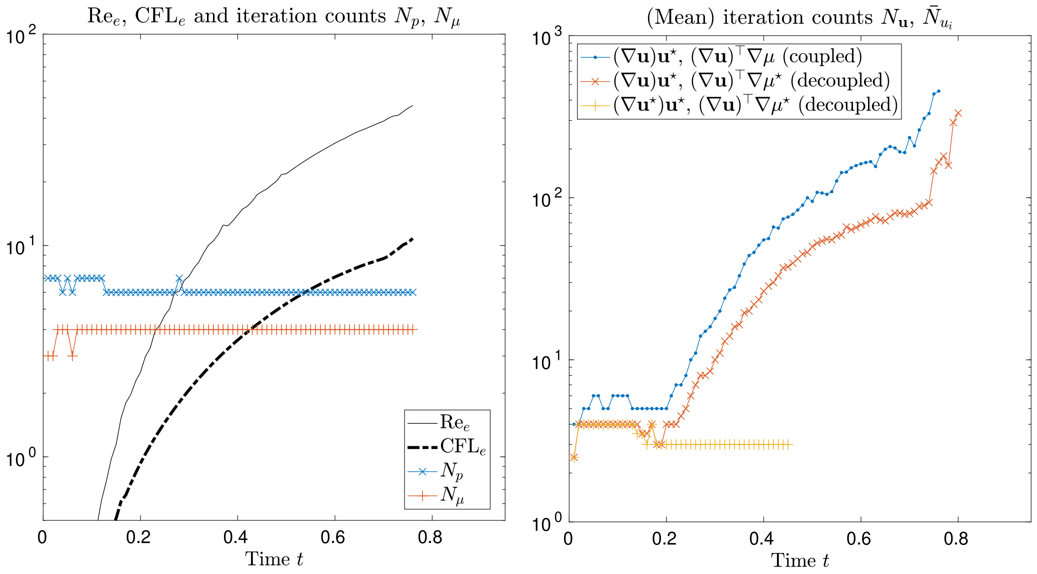

**Fig. 9.** Discretisation of the lid-driven cavity problem via BDF2 and $Q_2Q_1$ elements: local Re and CFL numbers and iteration counts of PPE ($N_p$) and viscosity projection ($N_\mu$) on the left; comparison of semi-implicit variants of the convective term $(\nabla\mathbf{u})\mathbf{u}$ and coupling viscous term $(\nabla\mathbf{u})^\top\nabla\mu$ on the right.

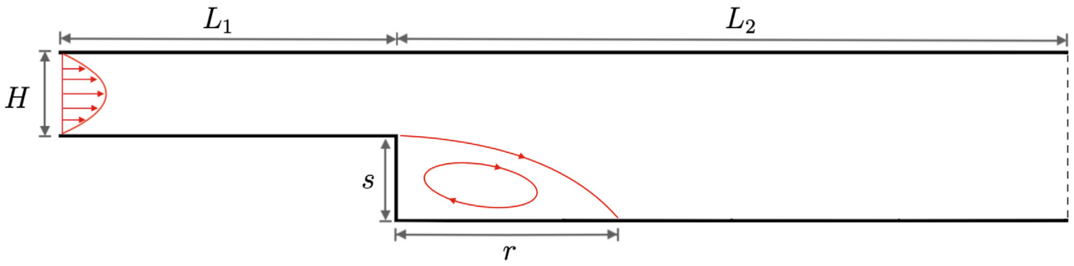

**Fig. 10.** Backward-facing step benchmark: problem setup.

indicate the need for some refinement close to the re-circulation zone (mind that we are not using any convective stabilisation such as streamline upwind Petrov–Galerkin [35]).

### 6.4. Carreau fluid flow through idealised aneurysm

As a final numerical example inspired by challenging hemodynamic applications, we consider the pulsatile flow of a shear-thinning fluid through a three-dimensional idealised (prototypical) aneurysm. Thus, fluid parameters are chosen in the physiologically relevant range, with $\rho = 1000$ kg/m$^3$, $\mu_0 = 50$ mPa·s, $\mu_\infty = 5.0$ mPa·s, $n = 0.25$ and $\lambda = 10$ s in the Carreau model (46). To generate a suitable mesh, we start off with a straight circular cylinder with central axis from $(x_1, x_2, x_3) = (0, 0, 0)$ to $(0.2, 0, 0)$, resulting in a length of $L = 0.2$ m between in- and outlet circular cross-sections of radius $R = 0.01$ m. This simple geometry is easily meshed and afterwards manipulated by transforming the radial coordinate of each nodal point according to

$$r = \left[1 + 2\sin^{10}\left(\frac{\pi x_1}{L}\right)\right]\tilde{r}.$$

Following this simple construction, the final mesh consisting of $\approx 4.9 \times 10^5$ elements and featuring boundary layers can be seen in Fig. 12. The discretised geometry is distributed across 14 processors. Another advantage of the proposed scheme, which makes heavy use of decoupling components and time-splitting, is highlighted here: a monolithic approach solving for all unknowns simultaneously would result in $\approx 12.9 \times 10^6$ DoFs using $Q_2Q_1$

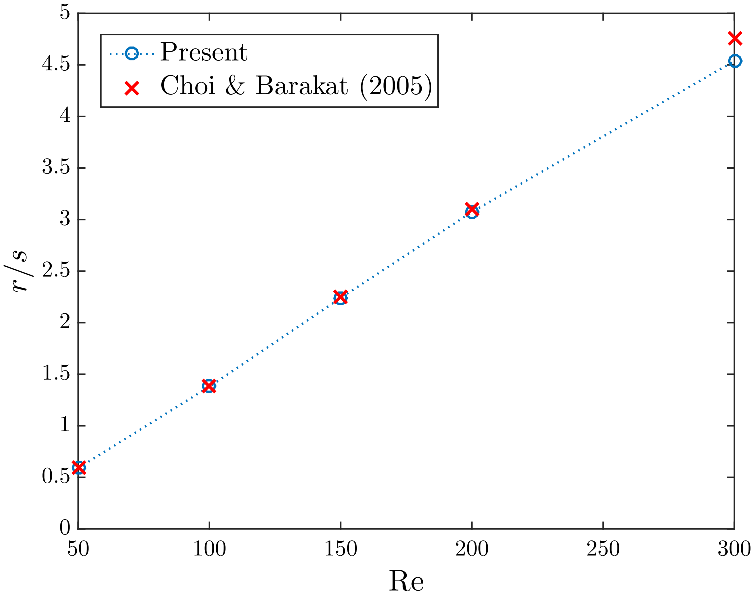

**Fig. 11.** Backward-facing step benchmark: reattachment length *versus* Reynolds number.

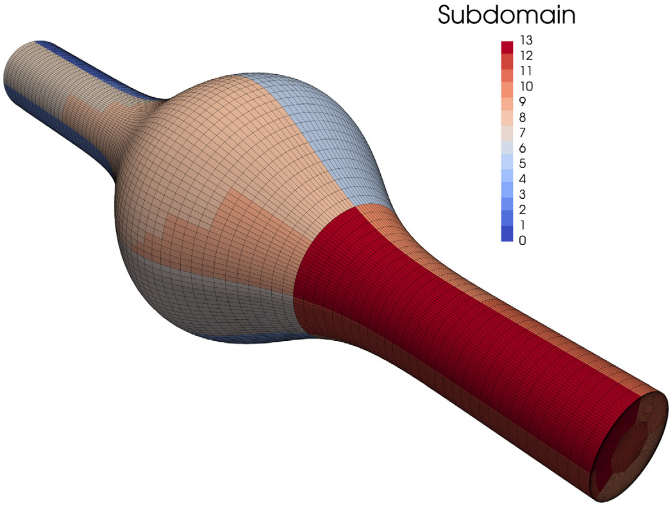

**Fig. 12.** Prototypical aneurysm mesh featuring boundary layers, distributed to 14 processors.

elements or $\approx 2.5 \times 10^6$ DoFs using stabilised $Q_1Q_1$ interpolation. However, the largest systems solved at a time in the present approach using $Q_1Q_1$ elements and decoupling the velocity components features only $\approx 5.0 \times 10^5$ DoFs, which is simply the number of mesh nodes. The time-step size in the second-order-accurate (BDF2 and linear extrapolation) method is chosen such that the maximum $\mathrm{CFL}_e$ (47) stays below 0.3. We prescribe the inlet velocity profile as

$$\mathbf{u} = \begin{pmatrix} u_I \\ 0 \\ 0 \end{pmatrix}, \quad u_I = \left(1 - \frac{r^2}{R^2}\right)\xi(t)\chi(t),$$

with

$$\chi(t) = 1 - 0.3\cos(\pi t/\tau),$$

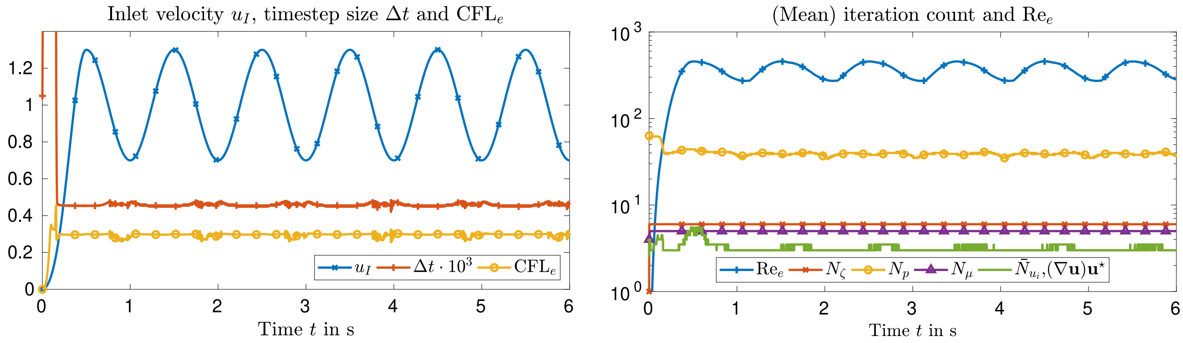

**Fig. 13.** Prototypical aneurysm example: maximum inlet velocity $u_I$, time-step size $\Delta t$ and element CFL number (left); corresponding element Reynolds number and iteration counts in the boundary projection step ($N_\zeta$), the PPE ($N_p$), the viscosity projection ($N_\mu$) and the mean velocity iteration count in momentum balance equations ($\bar{N}_{u_i}$) on the right.

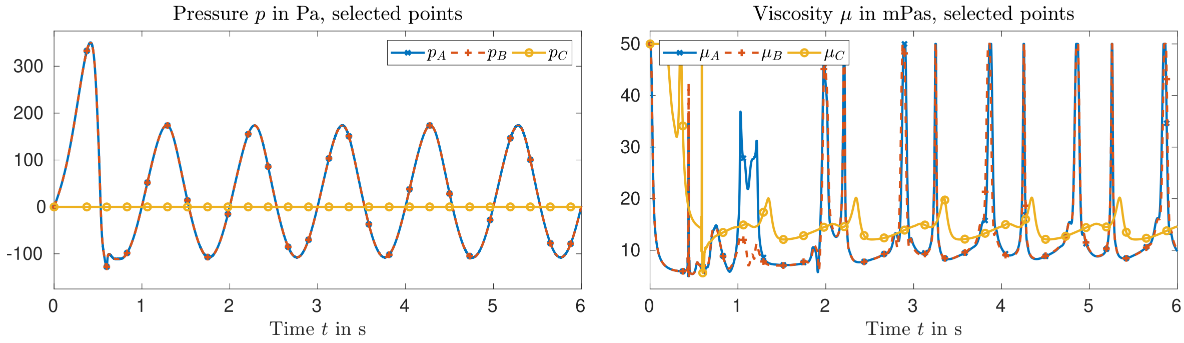

**Fig. 14.** Prototypical aneurysm example: pressure (left) and viscosity (right) at selected apex points $A = (L/2, R, 0)$, $B = (L/2, R/\sqrt{2}, R/\sqrt{2})$, and at the outlet centre $C = (L, 0, 0)$.

and $\xi(t)$ as defined in 6.3, with $\tau = 0.5$ s, resulting in time-step sizes around $5 \times 10^{-4}$ s. Then, the resulting Reynolds number considering the inlet diameter and mean inlet velocity is 2600, and the maximum element Reynolds number (48) is Re$_e \approx 450$. These quantities, namely the maximum inlet velocity, time-step size, maximum element CFL and Reynolds numbers are depicted in Fig. 13 together with the iteration counts for each of the linear solves. There, $N_p$ and $N_\mu$ again stand for the number of iterations in the AMG-preconditioned CG-solves on the linear systems arising from the PPE and viscosity-projection-step, whereas $\bar{N}_{u_i}$ is the mean number of iterations needed in the solves for the velocity components. Additionally, the iteration count $N_\zeta$ in the pressure boundary projection step is included for completeness, despite being negligible in terms of computational cost (<1% computational time).

As expected, the projection solves show excellent convergence behaviour, while $N_p$ is mildly solution-dependent due to variations in the right-hand side of the linear system. The values of $\bar{N}_{u_i}$ are remarkably low, but dependent on the time-step size and (element) Reynolds number, which is why the settings for the FGMRES-solver were conservatively set. We report the pressure and viscosity components of the solution at the apex point $A = (L/2, R, 0)$, another apex point $B = (L/2, R/\sqrt{2}, R/\sqrt{2})$ and the outlet centre point $C$ in Fig. 14. Inspecting the pressure curves, one observes a periodic solution with the mean outlet pressure being zero due to an inherent scaling in the zero pseudo-traction condition enforced. At the selected points, the viscosity is also reaching a periodic solution. With decreasing inlet velocity, strong recirculating flow is observed accompanied by rapid fluctuations in shear rate and, thus, viscosity. Moreover, the obtained solution is rotationally symmetric and reaches a periodic state. This behaviour is shown at $t \approx 5.37$ s in Fig. 15, indicated by selected streamlines coloured by velocity norm and fluid pressure. The importance of taking non-Newtonian behaviour into account becomes even more evident when inspecting local variations of viscosity as shown in Figs. 16 and 17. The viscosity is shown at two distinct points in time — strong variations with respect to both time and space are present, covering the whole range $(\mu_\infty, \mu_0]$ at time instants $t \approx 5.37$ s and $t = 5.83$ s due to strong recirculation.

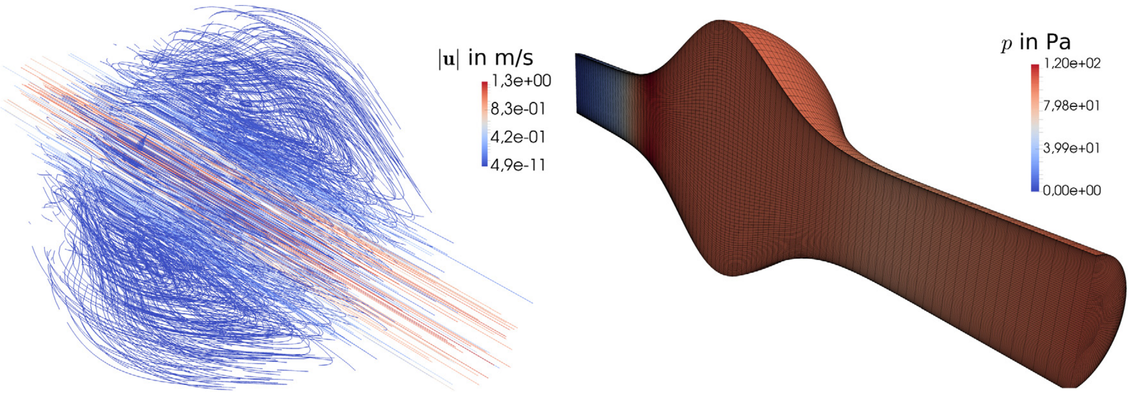

**Fig. 15.** Solution at $t = 5.37$ s showing complex flow patterns via streamlines coloured by velocity norm (left) and pressure in cut domain (right). Main flow direction from right to left.

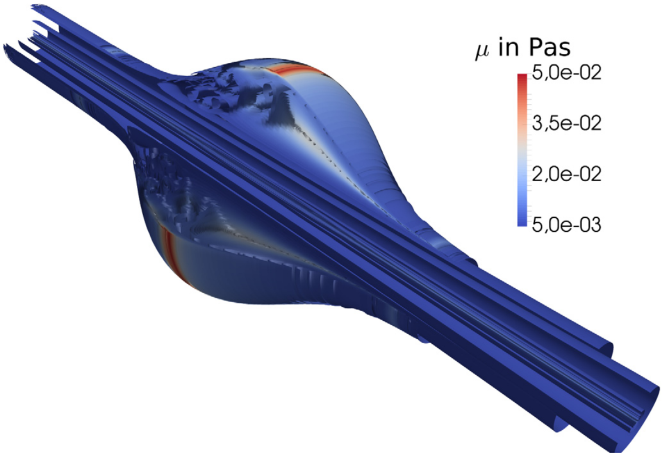

**Fig. 16.** Cut viscosity contours at $t = 5.83$ s resulting from complex recirculation pattern and large variation in the shear rate. Main flow direction from right to left.

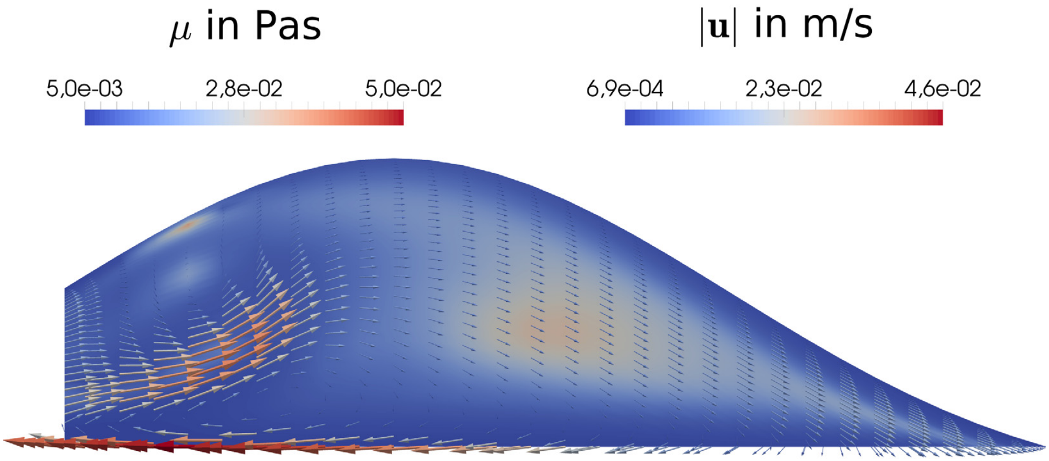

**Fig. 17.** Velocity vectors coloured by $|\mathbf{u}|$ and viscosity in close-up of expansion in the slice at $x_2 = 0$ at time $t = 5.37$ s. Main flow direction from right to left.

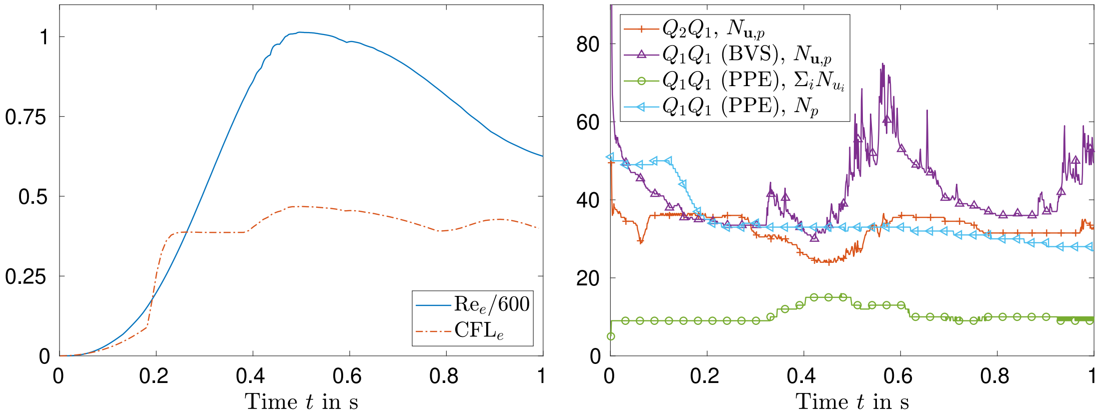

**Fig. 18.** Scaled element Reynolds and CFL numbers (left) and iteration counts $N_{\mathbf{u},p}$ for the velocity–pressure system with $Q_2Q_1$ or $Q_1Q_1$ elements compared to the PPE-based approach (right).

### *6.5. Computational performance study*

Finally, we aim to compare the presented PPE-based approach to existing methods in a slightly simpler setting than in the previous example: we choose $T = 1$ s, a fixed time-step size of $\Delta t = 10^{-3}$ s and a computational mesh with $2.1 \times 10^5$ elements for $Q_1Q_1$ interpolation and a mesh featuring the same amount of velocity DoFs for $Q_2Q_1$ interpolation; all other parameters are the same as in Section 6.4. For the comparison, let us consider coupled velocity–pressure formulations with block-systems of the form

$$\begin{pmatrix} A & B \\ \tilde{B} & C \end{pmatrix} \begin{pmatrix} \underline{\mathbf{u}}^{n+1} \\ \underline{p}^{n+1} \end{pmatrix} = \begin{pmatrix} \underline{\mathbf{f}} \\ \underline{g} \end{pmatrix},$$

solved using an FGMRES method [36] with a block-triangular right preconditioner [37]

$$\mathcal{P}^{-1} = \begin{pmatrix} A & B \\ 0 & S \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -B \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & S^{-1} \end{pmatrix},$$

with the Schur complement matrix $S = C - \tilde{B}A^{-1}B$. The velocity–velocity block in the coupled system can be interpreted as a sum of reaction, diffusion and convection terms [38]. Then, when using $Q_2Q_1$ elements, we can use the grad–div preconditioner and approximate the inverse Schur complement by [39]

$$S^{-1} \approx M_{\mu,\gamma}^{-1} + L_p^{-1},$$

while adding grad–div stabilisation, i.e., $\langle \nabla \cdot \mathbf{w}, \gamma \nabla \cdot \mathbf{u}^{n+1} \rangle$ with a parameter $\gamma = \mathcal{O}(1)$ [40]. For equal-order discretisations, on the other hand, $\gamma = \mathcal{O}(h_e)$ is the consistent choice for convective flows [40], rendering the grad–div preconditioner ineffective for equal-order pairs. Thus, we use the pressure–convection diffusion preconditioner (PCD) [41], taking

$$S^{-1} \approx M_{\mu,\gamma}^{-1} + \left(C - L_p\right)^{-1} - M_p^{-1} F_p L_p^{-1}.$$

Here, all the inverses appearing in the preconditioner are realised by single AMG V-cycles. In both cases, $M_{\mu,\gamma}$ accounts for varying viscosity and also for grad–div stabilisation when using the grad–div preconditioner. For further details, the reader is referred to [37–39,41–44]. These preconditioners for the coupled velocity–pressure systems are well established and widely used, and our main idea is to provide a simple understanding of the achievable performance based on black-box AMG methods and physics-based block-preconditioners. In order to enable the equal-order discretisation, we employ our recently introduced boundary vorticity stabilisation (BVS) [18,44], which can be seen as a simple modification of the classical PSPG method to increase parameter-robustness and eliminate spurious pressure boundary layers.

For a fair comparison, all the schemes are implemented in *deal.II* [27], using AMG methods provided by *Trilinos' ML* package [28]. All linear systems are solved until a residual reduction of $10^{-5}$ is achieved, taking the solution

**Table 2**
Number of mean AMG V-cycles per time-step (top row) needed to solve the coupled velocity–pressure systems using $Q_2Q_1$ or stabilised $Q_1Q_1$ elements or systems arising in the split-step scheme. Relative computational effort (bottom row) and absolute solver timings using 12 cores for $Q_2Q_1$ interpolation and 10 cores otherwise being the fastest possible settings.

| | $Q_2Q_1$, grad–div | Stabilised $Q_1Q_1$, PCD | $Q_1Q_1$, split-step |
|---|---|---|---|
| Mean AMG V-cycles | 65.06 (144%) | 125.29 (278%) | 45.12 (100%) |
| Linear system solve time | 9186 s (253%) | 8196 s (226%) | 3633 s (100%) |

from the last time-step as initial guess. In Fig. 18, the resulting iteration counts for the coupled velocity–pressure system, $N_{\mathbf{u},p}$, are compared with the sum of the individual velocity component solves, denoted by $\Sigma_i N_{u_i}$, and the iteration count $N_p$ in the PPE. In this numerical test, the element CFL number stays below 1, whereas the element Reynolds number is $\approx 600$. Initial iteration counts at the start of the simulation are slightly higher due to the relative convergence criterion chosen but are found rather robust with respect to the Reynolds number, where the grad–div preconditioner slightly outperforms the PCD in terms of iteration count. As before, we see very low iteration counts $N_{u_i}$ in the PPE-based approach, being basically independent of the Reynolds number. To reduce the ambiguity stemming from the specific implementations, we compare the number of AMG V-cycles needed to solve the linear systems and ignore the effort put into matrix assembly, solver setup and the actual linear solver used, since they are roughly equal or act in favour of the non-PPE solvers. Assume further that all appearing mass matrices can be lumped in the Schur complement approximation, resulting in two ($A^{-1}$ and $L_p^{-1}$) and three ($A^{-1}$ and twice $L_p^{-1}$) AMG V-cycles per FGMRES iteration using the grad–div or PCD preconditioners, respectively, and one AMG V-cycle per iteration in each of the three velocity components and the pressure Poisson system of the split-step scheme. We skip the viscosity projection in the comparison, since the mass matrix may be lumped, as already discussed. The resulting mean AMG V-cycles per time-step are listed in Table 2, showing the increased performance using the time-splitting method. Note here that the mean number of multigrid cycles per system solve does not capture the full picture: looking at the timings displayed in the second row of Table 2, it can be seen that even though a bigger number of AMG V-cycles per application of the PCD preconditioner is performed, its performance is better than expected. The grad–div preconditioner needs fewer AMG applications per outer Krylov iteration, but the denser sparsity pattern and increased cost when solving the velocity–velocity block with grad–div term (see [39]) render it more expensive. Additionally, the ratio of velocity to pressure DoFs is higher for $Q_2Q_1$ interpolation, pushing the optimal DoF count per processor for velocity and pressure sub-problems further apart. The timings underline the expected heavy influence of the actual linear system sizes rather than the overall DoF count on the total time spent in the linear solver. To sum up, the split-step scheme can be more effective than block-triangular preconditioners, especially when the approximation of the Schur complement is difficult in high Reynolds number settings.

In a last remark, we note that coupled solvers may, when suitably tailored or used in other scenarios, be at least competitive with respect to the presented split-step scheme — this is a matter of ongoing investigation. Moreover, when comparing the present scheme to a fully implicit nonlinear coupled solver, an additional speed-up by a factor of 2 to 3 can be expected depending on the problem and solver settings used (see [44]).

Despite the simplified setup, this example involves important aspects of challenging problems in hemodynamics and may serve as a reference solution. As demonstrated, applying the presented methods, substantial increases in computational efficiency can be attained. This ultimately renders even large datasets, as, e.g., present in patient-specific simulations amenable for desktop computers. Moreover, decoupling into individual solution components combined with the apparent simplicity of the present operators and resulting linear systems allows “easy” parallel computing on clusters utilising black-box solvers and preconditioners.

## 7. Concluding remarks

This work has presented a family of split-step methods for the approximation of incompressible flows of generalised Newtonian fluids. Although representing one of the simplest classes of non-Newtonian models, quasi-Newtonian fluids impose several difficulties for standard residual-based stabilisation and projection methods. By swapping the divergence-free constraint with an appropriate Poisson equation for the pressure with consistent boundary conditions, we end up with a system that enforces incompressibility without the need for the usual Helmholtz–Leray decomposition. In comparison to its original (Newtonian) version, the most challenging aspect of the present formulation is constructing a weak form suitable for standard Lagrangian finite elements. Furthermore, we have shown how to treat the nonlinearities in order to yield an efficient decoupling of the velocity components, whilst maintaining the desired temporal accuracy. Various numerical examples in different geometries and flow regimes have been presented to showcase the accuracy, stability and efficiency of the present approach. In comparison to other (only recently introduced) split-step methods for fluids with variable viscosity [16], ours has the advantages of substantially reducing the overall number of substeps and also allowing the use of equal-order finite element pairs for the discretisation of velocity and pressure. Ongoing work includes extending the present approach to fluids with pressure-dependent viscosity [45], as well as to a partitioned fluid–structure-interaction framework.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors acknowledge Graz University of Technology, Austria for the financial support of the Lead-project: Mechanics, Modeling and Simulation of Aortic Dissection.

## References

[1] J.L. Guermond, P. Minev, J. Shen, Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions, SIAM J. Numer. Anal. 43 (1) (2005) 239–258.

[2] J.L. Guermond, P. Minev, Jie Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Engrg. 195 (44–47) (2006) 6011–6045.

[3] L.J.P. Timmermans, P.D. Minev, F.N. Van de Vosse, An approximate projection scheme for incompressible flow using spectral elements, Internat. J. Numer. Methods Fluids 22 (7) (1996) 673–688.

[4] A. Poux, S. Glockner, M. Azaïez, Improvements on open and traction boundary conditions for Navier–Stokes time-splitting methods, J. Comput. Phys. 230 (10) (2011) 4011–4027.

[5] J. Liu, Open and traction boundary conditions for the incompressible Navier–Stokes equations, J. Comput. Phys. 228 (19) (2009) 7250–7267.

[6] H. Johnston, J.-G. Liu, Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term, J. Comput. Phys. 199 (1) (2004) 221–259.

[7] J.-G. Liu, J. Liu, R.L. Pego, Error estimates for finite-element Navier-Stokes solvers without standard Inf-Sup conditions, Chinese Ann. Math. Ser. B 30 (6) (2009) 743–768.

[8] J.-G. Liu, J. Liu, R.L. Pego, Stable and accurate pressure approximation for unsteady incompressible viscous flow, J. Comput. Phys. 229 (9) (2010) 3428–3453.

[9] J. Jia, J. Liu, Stable and spectrally accurate schemes for the Navier–Stokes equations, SIAM J. Sci. Comput. 33 (5) (2011) 2421–2439.

[10] D. Shirokoff, R.R. Rosales, An efficient method for the incompressible Navier–Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary, J. Comput. Phys. 230 (23) (2011) 8619–8646.

[11] Z. Sheng, M. Thiriet, F. Hecht, A high-order scheme for the incompressible Navier-Stokes equations with open boundary condition, Internat. J. Numer. Methods Fluids 73 (1) (2013) 58–73.

[12] H. Johnston, C. Wang, J.-G. Liu, A local pressure boundary condition spectral collocation scheme for the three-dimensional Navier–Stokes equations, J. Sci. Comput. 60 (3) (2014) 612–626.

[13] L. Li, A split-step finite-element method for incompressible Navier-Stokes equations with high-order accuracy up-to the boundary, J. Comput. Phys. 408 (2020) 109274.

[14] J. Deteix, D. Yakoubi, Improving the pressure accuracy in a projection scheme for incompressible fluids with variable viscosity, Appl. Math. Lett. 79 (2018) 111–117.

[15] L. Plasman, J. Deteix, D. Yakoubi, A projection scheme for Navier-Stokes with variable viscosity and natural boundary condition, Internat. J. Numer. Methods Fluids (2020) 1–21.

[16] J. Deteix, D. Yakoubi, Shear rate projection schemes for non-Newtonian fluids, Comput. Methods Appl. Mech. Engrg. 354 (2019) 620–636.

[17] D.R.Q. Pacheco, O. Steinbach, A continuous finite element framework for the pressure Poisson equation allowing non-Newtonian and compressible flow behavior, Internat. J. Numer. Methods Fluids 93 (5) (2021) 1435–1445.

[18] D.R.Q. Pacheco, R. Schussnig, O. Steinbach, T.-P. Fries, A global residual-based stabilization for equal-order finite element approximations of incompressible flows, Internat. J. Numer. Methods Engrg. 122 (8) (2021) 2075–2094.

[19] G.P. Galdi, R. Rannacher, A.M. Robertson, S. Turek, Hemodynamical Flows, in: Oberwolfach Seminars, vol. 37, Birkhäuser, Basel, 2008.

[20] R. Rannacher, On the numerical solution of the incompressible Navier-Stokes equations, J. Appl. Math. Mech. 73 (9) (1993) 203–216.

[21] J.G. Heywood, R. Rannacher, S. Turek, Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations, Internat. J. Numer. Methods Fluids 22 (5) (1996) 325–352.

[22] T. Richter, Fluid-Structure Interactions: Models, Analysis and Finite Elements, in: Lect. Notes Comput. Sci. Eng., vol. 118, Springer, Cham, 2017.

[23] M. Huber, U. Rüde, C. Waluga, B. Wohlmuth, Surface couplings for subdomain-wise isoviscous gradient based Stokes finite element discretizations, J. Sci. Comput. 74 (2) (2018) 895–919.

[24] L. John, P. Pustějovská, O. Steinbach, On the influence of the wall shear stress vector form on hemodynamic indicators, Comput. Vis. Sci. 18 (4–5) (2017) 113–122.

[25] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Vol. 44, Springer, Heidelberg, 2013.

[26] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations 1 - Nonstiff Problems, in: Springer Series in Computational Mathematics, Springer, 1993.

[27] D. Arndt, W. Bangerth, B. Blais, T.C. Clevenger, M. Fehling, A.V. Grayver, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, R. Rastak, I. Thomas, B. Turcksin, Z. Wang, D. Wells, The `deal.II` library, version 9.2, J. Numer. Math. 28 (3) (2020) 131–146.

[28] M.A. Heroux, J.M. Willenbring, A new overview of the Trilinos project, Sci. Program. 20 (2) (2012) 83–88.

[29] A. Limache, S. Idelsohn, R. Rossi, E. Oñate, The violation of objectivity in Laplace formulations of the Navier–Stokes equations, Internat. J. Numer. Methods Fluids 54 (6–8) (2007) 639–664.

[30] D.R.Q. Pacheco, T.S. Müller, O. Steinbach, G. Brenn, On outflow boundary conditions in finite element simulations of non-Newtonian internal flows, Int. J. Comput. Vis. Sci. Eng. (2021) http://dx.doi.org/10.51375/IJCVSE.2021.1.6.

[31] J. de Frutos, V. John, J. Novo, Projection methods for incompressible flow problems with WENO finite difference schemes, J. Comput. Phys. 309 (2016) 368–386.

[32] Y.I. Cho, K.R. Kensey, Effects of the non-Newtonian viscosity of blood on flows in a diseased arterial vessel. Part 1: Steady flows, Biorheology 28 (3–4) (1991) 241–262.

[33] H.W. Choi, A.I. Barakat, Numerical study of the impact of non-Newtonian blood behavior on flow over a two-dimensional backward facing step, Biorheology 42 (6) (2005) 493–509.

[34] A. Masud, J. Kwack, A stabilized mixed finite element method for the incompressible shear-rate dependent non-Newtonian fluids: Variational multiscale framework and consistent linearization, Comput. Methods Appl. Mech. Engrg. 200 (5–8) (2011) 577–596.

[35] L.P. Franca, S.L. Frey, Stabilized finite element methods: II. The incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 99 (2–3) (1992) 209–233.

[36] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. 7 (3) (1986) 856–869.

[37] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.

[38] S. Turek, Efficient Solvers for Incompressible Flow Problems - An Algorithmic and Computational Approach, in: Lecture Notes in Computational Science and Engineering, vol. 6, Springer, Berlin Heidelberg, 1999, pp. I–XV, 1–358.

[39] T. Heister, G. Rapin, Efficient augmented Lagrangian-type preconditioning for the Oseen problem using Grad-Div stabilization, Internat. J. Numer. Methods Fluids 71 (1) (2013) 118–134.

[40] V. John, Finite Element Methods for Incompressible Flow Problems, Springer, Basel, 2016.

[41] D. Kay, D. Loghin, A. Wathen, A preconditioner for the steady-state Navier–Stokes equations, SIAM J. Sci. Comput. 24 (1) (2003) 237–256.

[42] H.C. Elman, R.S. Tuminaro, Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations, Electron. Trans. Numer. Anal. 35 (2009) 257–280.

[43] J. Cahouet, J.-P Chabard, Some fast 3D finite element solvers for the generalized Stokes problem, Internat. J. Numer. Methods Fluids 8 (8) (1988) 869–895.

[44] R. Schussnig, D.R.Q. Pacheco, T.-P. Fries, Robust stabilised finite element solvers for generalised Newtonian fluid flows, J. Comput. Phys. (2021) (in press).

[45] L. Gesenhues, J.J. Camata, A.M.A. Côrtes, F.A. Rochinha, A.L.G.A. Coutinho, Finite element simulation of complex dense granular flows using a well-posed regularization of the $\mu$(I)-rheology, Comput. Fluids 188 (2019) 102–113.