ORIGINAL PAPER

Consistent pressure Poisson splitting methods for incompressible multi-phase flows: eliminating numerical boundary layers and inf-sup compatibility restrictions

Douglas R. Q. Pacheco[1] · Richard Schussnig[2]

Abstract

For their simplicity and low computational cost, time-stepping schemes decoupling velocity and pressure are highly popular in incompressible flow simulations. When multiple fluids are present, the additional hyperbolic transport equation in the system makes it even more advantageous to compute different flow quantities separately. Most splitting methods, however, induce spurious pressure boundary layers or compatibility restrictions on how to discretise pressure and velocity. Pressure Poisson methods, on the other hand, overcome these issues by relying on a fully consistent problem to compute the pressure from the velocity field. Additionally, such pressure Poisson equations can be tailored so as to indirectly enforce incompressibility, without requiring solenoidal projections. Although these schemes have been extended to problems with variable viscosity, constant density is still a fundamental assumption in existing formulations. In this context, the main contribution of this work is to reformulate consistent splitting methods to allow for variable density, as arising in two-phase flows. We present a strong formulation and a consistent weak form allowing standard finite element spaces. For the temporal discretisation, backward differentiation formulas are used to decouple pressure, velocity and density, yielding iteration-free steps. The accuracy of our framework is showcased through a wide variety of numerical examples, considering manufactured and benchmark solutions, equal-order and mixed finite elements, first- and second-order stepping, as well as flows with one, two or three phases.

Keywords Two-phase flow · Split-step methods · Finite element methods · Fractional-step methods · Variable density

1 Introduction

In numerical methods for incompressible flows, the assumption that both density and viscosity are constant is often mentioned just in passing, if even mentioned at all. It is, however, a rather "fragile" assumption that can be violated, for example, in non-Newtonian, two-phase or non-isothermal flows [1]. In fact, this violation can have severe implications in both theory and numerical practice. A variable viscosity, for instance, demands a complete reformulation of projection methods decoupling velocity and pressure in time [2–4]. A non-constant density, on the other hand, requires redefining even the projection operator used in such meth-

ods [5]. In particular, for flows with more than one fluid phase, density and viscosity fields are not only variable, but discontinuous—which certainly brings additional challenges to the discretisation. Nowadays, there are various numerical techniques with different levels of sophistication to handle such issues. Sharp two-phase flow simulations can be achieved, for instance, with enriched [6], extended [7] or unfitted [8] finite element methods. For the temporal discretisation, there are also different methods to cope with the variable density, especially in split-step schemes [5,9–13].

Split-step, time-splitting or fractional-step methods are interchangeable terms to designate time-stepping schemes decoupling the computation of velocity and pressure. Their idea is to transform an incompressible flow system into a series of Poisson and convection-diffusion subproblems that are simpler to solve. Especially for multi-phase flows, which feature conservation laws of different nature, split-step methods are more popular [5,9–20] than monolithic schemes [21–23]. Among the most common time-splitting schemes are so-called *projection* or *pressure correction* methods, in

✉ Douglas R. Q. Pacheco
  douglas.r.q.pacheco@ntnu.no

[1] Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

[2] Institute of Structural Analysis, Graz University of Technology, Graz, Austria

which incompressibility is enforced by projecting the velocity onto a weakly divergence-free space. Such methods, however, have some well-known shortcomings. The earliest non-incremental variant, for example, induces completely non-physical pressure boundary conditions [24]. This can be partially remedied by so-called rotational variants [2,4,11], which in turn induce an inf-sup compatibility restriction [25,26] on the finite element spaces (equal-order discretisations of pressure and velocity become unstable). In fact, these issues are not exclusive to multi-phase flows, arising even in the Stokes problem.

For single-phase flows, Henshaw and Petersson [27], Johnston and Liu [28] and Liu [29] proposed an alternative family of split-step schemes overcoming the limitations of classical projection methods. In their version, explicit incompressibility is replaced by a consistent pressure Poisson equation (PPE) obtained directly from the balance of momentum. Their schemes do not only reduce the computational cost, but also break the finite element compatibility restrictions without inducing any spurious pressure boundary layers. The price to pay is a slightly more complicated right-hand side, which can however be handled with hardly any additional implementation effort. Nonetheless, constant viscosity and constant density are two strong assumptions in the classical derivations [28,29]. While we have recently extended consistent splitting methods to the case of variable viscosity [30], handling more than one phase requires also allowing for *density* variations, which has a stronger impact on the formulation. This is thus the main topic of the present work.

Our main goal is to devise a fully consistent pressure Poisson framework for incompressible flows with variable density, with a focus on two-phase flows. We can derive an elliptic equation for the pressure by first dividing the Navier–Stokes momentum equation by the flow density, adding a consistency term and then taking the divergence of the result. Similarly, pressure boundary conditions can be obtained from the momentum equation and its boundary data. This gives us an equation to compute the pressure from the velocity and density fields, and at the same time guarantees, on the continuous level, a divergence-free velocity—without requiring projections or corrections. This, combined with semi-implicit temporal discretisations, converts the rather complex mass-momentum system into a series of much simpler subproblems: a Poisson equation with variable coefficient, a scalar advection problem and a vector-valued convection-diffusion equation that may be decomposed into scalar components. Furthermore, when considering a finite element spatial discretisation, our schemes allow the same polynomial degree to be used for all quantities, simplifying implementation.

The rest of this article is organised in the following way. After the basic problem setup and notation are briefly described in Sect. 2, Sect. 3 presents a PPE-based reformulation that implies incompressibility without explicitly enforcing the divergence-free constraint on the velocity. When the goal is to use standard finite element elements for the spatial discretisation, we need a C^0-compatible variational formulation, which is the topic of Sect. 4. Discretisation and stabilisation aspects are addressed in Sects. 5 and 6. Finally, several numerical examples are given in Sect. 7 to test our approach in problems with increasing complexity. Let us first remark that, differently from most works on two-phase flow simulation, ours does not intend to address the challenges related to interface conditions, jumps, forces, etc. We focus instead on the time-splitting aspect of the problem, that is, on showing how one can construct simple, minimally invasive, finite-element-suitable schemes to decouple all system unknowns in time, without sacrificing boundary accuracy or inducing inf-sup compatibility restrictions. For this reason, we focus on gravity-driven flow regimes in which surface forces can be neglected. How one handles interface-related matters, however, is immaterial to the present discussion, as different techniques can be incorporated straightforwardly.

2 Problem setup and notation

Let us consider a spatial domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, with a Lipschitz boundary $\Gamma := \partial \Omega$ decomposed into three non-overlapping regions Γ_D, Γ_N and Γ_S. On the first two we prescribe velocities and normal tractions, respectively, while the latter describes a free slip boundary. Imposing slip or free-slip conditions is important in multi-phase flows to allow "wetting", since a standard no-slip condition would preclude fluid-fluid interfaces from moving along walls. Under these conditions, the standard setting for the incompressible Navier–Stokes system with non-constant density reads

$$\partial_t \rho + \boldsymbol{u} \cdot \nabla \rho = 0 \ \text{in} \ \Omega \times (0, T], \tag{1}$$

$$\rho \left[\partial_t \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{u} - \boldsymbol{g} \right] - \nabla \cdot \mathbb{S} + \nabla p = \boldsymbol{0} \ \text{in} \ \Omega \times (0, T], \tag{2}$$

$$\nabla \cdot \boldsymbol{u} = 0 \ \text{in} \ \Omega \times (0, T], \tag{3}$$

equipped with initial and boundary conditions

$$\rho = \rho_0 \ \text{at} \ t = 0, \tag{4}$$

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

$$\rho = \varrho \ \text{on} \ \Gamma_{\text{in}}(t), \ t \in (0, T], \tag{6}$$

$$\boldsymbol{u} = \mathbf{v} \ \text{on} \ \Gamma_D \times (0, T], \tag{7}$$

$$\mathbb{S}\boldsymbol{n} - p\boldsymbol{n} = \boldsymbol{t} \ \text{on} \ \Gamma_N \times (0, T], \tag{8}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0 \ \text{on} \ \Gamma_S \times (0, T], \tag{9}$$

$$\boldsymbol{n} \times (\mathbb{S}\boldsymbol{n}) = \boldsymbol{0} \ \text{on} \ \Gamma_S \times (0, T], \tag{10}$$

where $\boldsymbol{n}$ is the outward unit normal vector on $\partial\Omega$, $(\rho_0, \boldsymbol{u}_0, \varrho,$ $\mathbf{v}, \boldsymbol{t})$ are appropriate data, $\boldsymbol{g}$ denotes a gravitational field, $(\boldsymbol{u}, p, \rho)$ are the flow velocity, pressure and density, respectively, and $\mathbb{S}$ is the viscous stress tensor, which for Newtonian fluids takes the form

$$\mathbb{S} = 2\mu\nabla^s\boldsymbol{u}, \quad \nabla^s\boldsymbol{u} := \frac{1}{2}\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^\top\right], \tag{11}$$

with $\mu := \rho\nu$ and ν denoting the dynamic and kinematic viscosities, respectively. The free-slip condition can be replaced by a slip condition if we appropriately modify the right-hand side in (10), see for instance Refs. [31,32]. Also note that (10) can be written in terms of tangent vectors [33]. Since the mass equation (1) is hyperbolic, boundary conditions for ρ are only prescribed on Γ_{in}, which is the part of $\partial\Omega$ where $\boldsymbol{u} \cdot \boldsymbol{n} < 0$.

3 A consistent pressure Poisson reformulation

Regardless of the spatial discretisation method used, efficiently solving (1)–(10) in a fully coupled way is a challenging task. However, because of the saddle-point structure of the velocity-pressure subsystem, decoupling the unknowns is not straightforward here. Such structure also induces an inf-sup or Ladyzhenskaya–Babuška–Brezzi (LBB) compatibility restriction when discretising p and $\boldsymbol{u}$ with finite elements [25,26]. To overcome both limitations, our idea is to replace the incompressibility constraint (3) by a consistent pressure Poisson problem:

$$-\nabla \cdot \left[\rho^{-1}\nabla p + (\nabla\boldsymbol{u})\boldsymbol{u} - \boldsymbol{g} - \rho^{-1}\nabla\cdot\mathbb{S} + \chi\nu\nabla(\nabla\cdot\boldsymbol{u})\right]$$
$$+ \beta\nabla\cdot\boldsymbol{u} = 0 \text{ in } \Omega\times[0, T], \tag{12}$$

$$p = \zeta \text{ on } \Gamma_N \times [0, T], \tag{13}$$

$$\frac{\partial p}{\partial n} = \lambda \text{ on } \Gamma_{D,S}\times[0, T], \tag{14}$$

in which $\Gamma_{D,S} := \Gamma_D \cup \Gamma_S$, $\frac{\partial(\cdot)}{\partial n} := \boldsymbol{n}\cdot\nabla(\cdot)$, β is a nonnegative parameter to be defined later (see Sect. 5.2), and

$$\zeta := \mathbb{S} : (\boldsymbol{n}\otimes\boldsymbol{n}) - 2\mu\nabla\cdot\boldsymbol{u} - \boldsymbol{t}\cdot\boldsymbol{n}$$
$$= 2\mu\nabla^s\boldsymbol{u} : (\boldsymbol{n}\otimes\boldsymbol{n} - \mathbb{I}) - \boldsymbol{t}\cdot\boldsymbol{n}, \tag{15}$$

$$\lambda := \left[\rho\boldsymbol{g} - \rho\partial_t\mathbf{v} - (\rho\nabla\boldsymbol{u})\boldsymbol{u} + \nabla\cdot\mathbb{S} - \chi\mu\nabla(\nabla\cdot\boldsymbol{u})\right]\cdot\boldsymbol{n}, \tag{16}$$

with $\mathbf{v}\cdot\boldsymbol{n} = 0$ on Γ_S and $\mathbb{I}$ denoting the $d\times d$ identity tensor. Parameter χ in Eqs. (12) and (16) is a positive constant which we will later show that should be set as $\chi = 2$.

It is still not evident why (12)–(16) would be a suitable replacement for the usual condition $\nabla\cdot\boldsymbol{u} = 0$, but for now

let us take a closer look at the equations. Equation (12) can be obtained by adding $\chi\mu\nabla(\nabla\cdot\boldsymbol{u})$ to the left-hand side of Eq. (2), dividing both sides by ρ, taking minus the divergence of the result and finally adding $\beta\nabla\cdot\boldsymbol{u}$ to the left-hand side. Both added terms are consistent with the original system, whose solution $\boldsymbol{u}$ is divergence-free. The absence of $\partial_t\boldsymbol{u}$ in Eq. (12) is also consistent, as

$$\nabla\cdot(\partial_t\boldsymbol{u}) = \partial_t(\nabla\cdot\boldsymbol{u}) = 0.$$

Moreover, the Neumann boundary condition (14,16) for the pressure is, up to an additional $\chi\mu\nabla(\nabla\cdot\boldsymbol{u})$, simply the momentum equation projected onto the normal direction $\boldsymbol{n}$ and restricted to $\Gamma_{D,S}$. The Dirichlet boundary condition (13,15) for the pressure can be obtained by dotting the traction boundary condition (8) with $\boldsymbol{n}$ and subtracting $2\mu\nabla\cdot\boldsymbol{u}$ from the left-hand side, which is again consistent. The reason for the added terms will become clear soon, as we show the equivalence between the standard Navier–Stokes system and the PPE-modified one.

The first side of the equivalence, i.e., that (12)–(16) can be derived from the momentum-incompressibility system, has just been outlined. Slightly more involved will be proving the other side of the equivalence: does the pressure Poisson problem, combined with the momentum equation, imply incompressibility? To show that, we divide Eq. (2) by ρ, apply the divergence and add the result to the PPE (12). This leaves us with

$$\beta\nabla\cdot\boldsymbol{u} - \nabla\cdot\left[\chi\nu\nabla(\nabla\cdot\boldsymbol{u}) - \partial_t\boldsymbol{u}\right] = 0,$$

or, by introducing $\varphi := \nabla\cdot\boldsymbol{u}$, with the convection-diffusion-reaction equation

$$\partial_t\varphi - \nabla\cdot(\chi\nu\nabla\varphi) + \beta\varphi = 0. \tag{17}$$

What is missing now are boundary and initial conditions for this equation. Neumann data can be obtained by dotting (2) with $\boldsymbol{n}$, restricting the result to $\Gamma_{D,S}$ and then subtracting (14), which gives $\chi\mu\boldsymbol{n}\cdot\nabla(\nabla\cdot\boldsymbol{u}) = 0$, that is,

$$\frac{\partial\varphi}{\partial n} = 0 \text{ on } \Gamma_{D,S}, \tag{18}$$

since $\chi\mu \neq 0$. Similarly, to get Dirichlet data for φ we multiply the traction boundary condition (8) by $\boldsymbol{n}$ and add the result to (13), yielding $2\mu\nabla\cdot\boldsymbol{u} = 0$, or simply

$$\varphi = 0 \text{ on } \Gamma_N. \tag{19}$$

Thus, with zero Dirichlet and Neumann boundary conditions, and provided that

$$\varphi_0 := \nabla\cdot\boldsymbol{u}_0 = 0, \tag{20}$$

the partial differential equation (17) has $\varphi \equiv 0$ as its unique solution, that is, $\nabla \cdot \boldsymbol{u} = 0$ in $\Omega \times [0, T]$, as we wanted to show.

In summary, combining the momentum equation (2) with the pressure Poisson problem (12)–(16) implies incompressibility without the need to enforce $\nabla \cdot \boldsymbol{u} = 0$ explicitly or to project $\boldsymbol{u}$ onto a divergence-free space. The main advantage of this PPE-based reformulation is that we now have an invertible operator for the pressure, as opposed to the original saddle-point structure. From the spatial discretisation standpoint, this eliminates the LBB compatibility restriction and allows equal-order finite element spaces [30]. For the temporal discretisation, the PPE gives us an invertible equation to compute p from $(\rho, \boldsymbol{u})$ without resorting to the artificial boundary conditions typical of projection methods.

The pressure Poisson equation, as presented in (12), has third-order derivatives on the velocity. Fortunately, we can eliminate them by using certain identities. Notice that

$$\nabla \cdot \left[\rho^{-1} \nabla \cdot \mathbb{S} - \chi \nu \nabla (\nabla \cdot \boldsymbol{u}) \right]$$
$$= \nabla \cdot \left[\nu (\Delta \boldsymbol{u} + \nabla (\nabla \cdot \boldsymbol{u})) + 2 \rho^{-1} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu - \chi \nu \nabla (\nabla \cdot \boldsymbol{u}) \right].$$

The vector Laplacian $\Delta \boldsymbol{u}$ obeys the identity $\Delta \boldsymbol{u} \equiv \nabla (\nabla \cdot \boldsymbol{u}) - \nabla \times (\nabla \times \boldsymbol{u})$, so that

$$\nabla \cdot \left[\rho^{-1} \nabla \cdot \mathbb{S} - \chi \nu \nabla (\nabla \cdot \boldsymbol{u}) \right]$$
$$= \nabla \cdot \left[\frac{2}{\rho} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu - \nu \nabla \times (\nabla \times \boldsymbol{u}) + (2 - \chi) \nu \nabla (\nabla \cdot \boldsymbol{u}) \right].$$

Now, by choosing $\chi = 2$, we can get rid of the third-order derivatives:

$$\nabla \cdot \left[\rho^{-1} \nabla \cdot \mathbb{S} - 2 \nu \nabla (\nabla \cdot \boldsymbol{u}) \right]$$
$$= \nabla \cdot \left[\frac{2}{\rho} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu - \nu \nabla \times (\nabla \times \boldsymbol{u}) \right] \qquad (21)$$
$$= \nabla \cdot \left(\frac{2}{\rho} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu \right) - \nabla \nu \cdot (\nabla \times \nabla \times \boldsymbol{u}) - \nu \nabla \cdot (\nabla \times \nabla \times \boldsymbol{u})$$
$$= \nabla \cdot \left(\frac{2}{\rho} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu \right) - \nabla \nu \cdot (\nabla \times \nabla \times \boldsymbol{u}) , \qquad (22)$$

since $\nabla \cdot (\nabla \times \boldsymbol{v}) = 0$ for any vector $\boldsymbol{v}$. We are then left with only first- and second-order derivatives in (22). Now, the purpose of the $\chi \nu \nabla (\nabla \cdot \boldsymbol{u})$ term artificially (but consistently) added in Eq. (12) becomes clear, and it is in fact twofold: it adds diffusion to the divergence equation (17) and allows us to eliminate the third-order derivatives in the PPE by simply setting $\chi = 2$. Although both consistency and incompressibility hold for any $\chi > 0$, this is *not* intended here as a tunable parameter. We *could* in principle choose $\chi \neq 2$, but this would only leave us with unnecessary higher-order derivatives. Since χ features only in the PPE, it adds no diffusion to the momentum equation, but only to Eq. (17), whose unique solution is $\nabla \cdot \boldsymbol{u} = \varphi \equiv 0$ for any positive χ.

4 Weak pressure Poisson formulation

The second-order velocity derivatives in (22) cannot be handled by standard Lagrangian finite elements, so we need a weak formulation containing only first-order derivatives. This is not straightforward, and in fact Li et al. [13] used finite differences to evaluate the higher-order derivatives at the nodal points, then interpolated those values in a continuous source term—which is hardly practical in unstructured meshes. To allow a more general approach suitable for C^0 finite elements in arbitrary meshes, we will next show how to reduce the order of differentiation, starting from (21) and using some vector calculus relations.

We first multiply the PPE (12) by a continuous test function q with zero value on Γ_N. Then, we integrate by parts all terms but $\beta \nabla \cdot \boldsymbol{u}$, and enforce the Neumann boundary condition (14) to get

$$\int_\Omega \rho^{-1} \nabla p \cdot \nabla q \, \mathrm{d}\Omega$$
$$= -\int_{\Gamma_D} q \, \partial_t (\boldsymbol{n} \cdot \mathbf{v}) \, \mathrm{d}\Gamma - \int_\Omega q \beta \nabla \cdot \boldsymbol{u} \, \mathrm{d}\Omega$$
$$+ \int_\Omega \nabla q \cdot \left[\boldsymbol{g} - (\nabla \boldsymbol{u}) \boldsymbol{u} + 2 \rho^{-1} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu \right.$$
$$\left. - \nu \nabla \times (\nabla \times \boldsymbol{u}) \right] \mathrm{d}\Omega \,,$$

where we can further use integration by parts to write

$$\int_\Omega \nabla q \cdot \left[\nu \nabla \times (\nabla \times \boldsymbol{u}) \right] \mathrm{d}\Omega$$
$$= \int_\Omega \nabla \times (\nu \nabla q) \cdot (\nabla \times \boldsymbol{u}) \, \mathrm{d}\Omega$$
$$+ \int_\Gamma (\nabla q \times \boldsymbol{n}) \cdot (\nu \nabla \times \boldsymbol{u}) \, \mathrm{d}\Omega \,, \qquad (23)$$

but

$$\nabla \times (\nu \nabla q) = \nabla \nu \times \nabla q + \nu \underbrace{\nabla \times (\nabla q)}_{\equiv \, \boldsymbol{0}} = \nabla \nu \times \nabla q \quad (24)$$

and

$$-(\nabla \nu \times \nabla q) \cdot (\nabla \times \boldsymbol{u}) \equiv -\nabla q \cdot \left[(\nabla \times \boldsymbol{u}) \times \nabla \nu \right]$$
$$\equiv \nabla q \cdot \left\{ \left[(\nabla \boldsymbol{u})^\top - \nabla \boldsymbol{u} \right] \nabla \nu \right\}. \qquad (25)$$

Moreover,

$$2 \rho^{-1} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \mu = 2 \rho^{-1} \nabla^{\mathrm{s}} \boldsymbol{u} (\rho \nabla \nu + \nu \nabla \rho)$$
$$= 2 \nu \rho^{-1} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \rho + \left[(\nabla \boldsymbol{u})^\top + \nabla \boldsymbol{u} \right] \nabla \nu \,.$$
$$(26)$$

Therefore, collecting the results from (23)–(26), we arrive at a weak formulation: find p satisfying the Dirichlet condition $p = \zeta$ on Γ_N (13), such that

$$
\begin{aligned}
&\int_\Omega \rho^{-1} \nabla p \cdot \nabla q \, \mathrm{d}\Omega \\
&= \int_\Gamma (\boldsymbol{n} \times \nabla q) \cdot (\nu \nabla \times \boldsymbol{u}) \, \mathrm{d}\Gamma - \int_{\Gamma_D} q \, \partial_t (\boldsymbol{n} \cdot \mathbf{v}) \, \mathrm{d}\Gamma \\
&\quad - \int_\Omega q \beta \nabla \cdot \boldsymbol{u} \, \mathrm{d}\Omega + \int_\Omega \nabla q \\
&\quad \cdot \left[\boldsymbol{g} - (\nabla \boldsymbol{u}) \boldsymbol{u} + 2\nu \rho^{-1} \nabla^{\mathrm{s}} \boldsymbol{u} \nabla \rho + 2(\nabla \boldsymbol{u})^\top \nabla \nu \right] \mathrm{d}\Omega
\end{aligned}
\tag{27}
$$

for all test functions q being zero on Γ_N. Now that we are left with only first-order derivatives, C^0 finite elements can be used to discretise all unknowns and test functions.

5 Discretisation

This section will address discretisation aspects in both space and time. Although we focus on finite elements for the spatial discretisation, the fractional-step schemes presented here are also valid for other frameworks such as finite differences and finite volumes.

5.1 Temporal discretisation

As usual in split-step methods, we employ backward differentiation formulas (BDFs) to approximate the temporal derivatives, combined with matching extrapolation rules for linearisation. The idea is to have iteration-free schemes completely decoupling velocity, pressure and density. For concision, we shall limit the presentation to first- and second-order temporal discretisations, but the schemes can be straightforwardly extended to higher-order versions with possibly adaptive stepping (see Ref. [30] for details).

Let $\Delta t > 0$ be a finite time-step size and υ_n denote a certain quantity υ at time $t = t_n$. For the first-order case, the backward finite difference BDF1 and corresponding extrapolation read:

$$
\partial_t \upsilon|_{t=t_n} \approx \frac{1}{\Delta t}(\upsilon_n - \upsilon_{n-1}), \quad \upsilon_n \approx \hat{\upsilon}_n = \upsilon_{n-1} ,
\tag{28}
$$

whereas to go one order higher we use BDF2:

$$
\begin{aligned}
&\partial_t \upsilon|_{t=t_n} \approx \frac{1}{\Delta t}\left(\frac{3}{2} \upsilon_n - 2\upsilon_{n-1} + \frac{1}{2} \upsilon_{n-2} \right), \\
&\upsilon_n \approx \hat{\upsilon}_n = 2\upsilon_{n-1} - \upsilon_{n-2} .
\end{aligned}
\tag{29}
$$

For the mass equation, this results in

$$
\rho_{n+1} + \Delta t \boldsymbol{u}_n \cdot \nabla \rho_{n+1} = \rho_n
\tag{30}
$$

for BDF1, or

$$
\frac{3}{2} \rho_{n+1} + \Delta t (2\boldsymbol{u}_n - \boldsymbol{u}_{n-1}) \cdot \nabla \rho_{n+1} = 2\rho_n - \frac{1}{2} \rho_{n-1}
\tag{31}
$$

for BDF2. Either way, we are left with a linear partial differential equation that still needs to be discretised in space.

5.2 Spatial discretisation

It is well-known that the Galerkin finite element method, in its standard form, is not ideal for problems with low or zero diffusion, such as (30) or (31). Therefore, when solving the mass transport, it is common practice to employ residual-based [32,34] or entropy [11] stabilisation techniques. However, we can leverage the absence of second-order derivatives to use a least-squares (instead of Galerkin) finite element formulation, as done by Pyo and Shen [10]. For an equation of the form

$$
\mathcal{L}(u) := \alpha u + \boldsymbol{v} \cdot \nabla u = f ,
\tag{32}
$$

it consists of seeking $u \in X$ fulfilling inflow conditions, such that

$$
\int_\Omega \mathcal{L}(u)\mathcal{L}(v) \, \mathrm{d}\Omega = \int_\Omega f \mathcal{L}(v) \, \mathrm{d}\Omega \quad \text{for all } v \in X,
\tag{33}
$$

with X being an appropriate finite element space. The least-squares finite element method is suitable for hyperbolic problems without requiring stabilisation terms or tunable factors. The other two equations in our system—momentum and pressure Poisson—on the other hand, can be tackled by a standard Galerkin finite element discretisation. Yet, some stabilisation may be required for high-Reynolds flows, as addressed in Sect. 6.4.

For concision, we shall state the weak momentum equation considering $\Gamma_S = \emptyset$; details on the implementation of (free-)slip conditions can be found for instance in Refs. [31,35]. The following sets are considered:

$$
\begin{aligned}
W_\mathbf{v}^k &:= \left\{ \boldsymbol{w} \in \left[S_h^k(\Omega_h) \right]^d : \boldsymbol{w} = \mathbf{v}_h \text{ on } \Gamma_D \right\}, \\
Q_\zeta^l &:= \left\{ q \in S_h^l(\Omega_h) : q = \zeta_h \text{ on } \Gamma_N \right\}, \\
R_\varrho^m &:= \left\{ r \in S_h^m(\Omega_h) : r = \varrho_h \text{ on } \Gamma_{\mathrm{in}} \right\},
\end{aligned}
$$

where subscript h denotes the finite element interpolation of a certain quantity or geometry, and $S_h^k(\Omega_h)$ denotes the standard space of continuous, piecewise polynomial shape

functions of degree k in the triangulation of Ω_h. Notice that, in general, we will formally have $R^m_\varrho(t)$, since the inflow boundary depends instantly and locally on $\boldsymbol{n} \cdot \boldsymbol{u}(t)$.

Another discretisation-related matter to be discussed is the choice of the weighting parameter β in the pressure Poisson equation (27). On the continuous level, all we need is $\beta \geq 0$ to guarantee incompressibility, otherwise a negative β could spoil the unique solvability of (17) with non-zero eigenmodes of $\varphi = \nabla \cdot \boldsymbol{u}$. On the discrete level, however, β should be chosen appropriately if optimal convergence is desired. First, notice that the term $\beta \nabla \cdot \boldsymbol{u}$ penalises large violations of discrete incompressibility, so that β works as a penalty-like parameter. While Li [36] proposed an expression based solely on the spatial resolution, Pacheco et al. [37] showed that in fact β can be taken as the inverse of the well-known PSPG parameter:

$$\beta = \frac{1}{\tau_{\text{PSPG}}} = \sqrt{\left(\frac{\alpha_1 \nu}{h^2}\right)^2 + \left(\frac{\alpha_2 |\boldsymbol{u}|}{h}\right)^2 + \left(\frac{\alpha_3}{\Delta t}\right)^2}, \qquad (34)$$

in which h denotes the element size and $(\alpha_1, \alpha_2, \alpha_3)$ are tunable parameters. For stationary problems we take $\alpha_3 = 0$, whereas $\alpha_1 = \alpha_2 = 0$ when using LBB-compatible spaces (e.g. Taylor–Hood elements). To reduce the number of user-defined parameters, we set

$$\beta = \alpha \sqrt{\delta_{kl} \left[\left(\frac{4\nu}{h^2}\right)^2 + \left(\frac{2|\boldsymbol{u}|}{h}\right)^2\right] + \left(\frac{2}{\Delta t}\right)^2}, \qquad (35)$$

where α is a single tunable parameter, k and l are the orders of the velocity and pressure discretisations, respectively, and δ_{kl} refers to the Kronecker delta.

5.3 The Dirichlet boundary condition for the pressure

We now briefly discuss a technical issue involving the pressure Dirichlet boundary condition (13). For problems with $\Gamma_N \neq \emptyset$, we have

$$p = \zeta := 2\mu \nabla^{\text{s}} \boldsymbol{u} : (\boldsymbol{n} \otimes \boldsymbol{n} - \mathbb{I}) - \boldsymbol{t} \cdot \boldsymbol{n} \quad \text{on} \ \Gamma_N,$$

where $\boldsymbol{t}$ is some given normal traction. Since on the discrete level the stress tensor $\mathbb{S} = 2\mu \nabla^{\text{s}} \boldsymbol{u}$ will be discontinuous, so will ζ across elements. This is an issue when using continuous finite elements for the pressure, but there are various ways to circumvent that. The most common approach is to project ζ (in the $L^2(\Gamma_N)$ sense) onto a continuous space [4,29,30]. Another possibility is to notice that, although $\boldsymbol{n}$ can be discontinuous on non-planar surfaces, outflow boundaries are normally generated by truncating a larger physical domain,

so that each outlet portion of the computational domain is planar. That being the case, the tangential projector $(\boldsymbol{n} \otimes \boldsymbol{n} - \mathbb{I})$ will be constant—or piecewise constant, if there are multiple outlets—so that we could simply perform an $L^2(\Omega)$ projection of $\mathbb{S}$ onto a continuous finite element space. An $L^2(\Omega)$ projection is simpler than an $L^2(\Gamma_N)$ projection, but also more expensive. Yet, we can pre-compute and store the LDL decomposition of the mass matrix used in the projection, or even lump it [30]. As we will soon see, projecting the stress tensor may also be useful in other parts of the solver. It is also possible to compute nodal values by simply averaging ζ on the boundary patch surrounding each node, although we have not tested this option. Regardless of the approach used, we shall henceforth consider that a continuous value $\tilde{\zeta}$ is used.

6 Fractional-step schemes

The main strength of split-step methods is decoupling pressure from velocity. With PPE-based schemes, the idea is to march in time by first computing the current velocity from previous pressures, then using the updated velocity (and density) to compute the current pressure, and so on for the next steps. To do that, the pressure term in the momentum equation is treated explicitly. On the other hand, when updating the pressure we can treat all terms implicitly in the PPE, since both the current velocity and the current density will already be known at that point.

6.1 Initialisation

In general, the incompressible Navier–Stokes system is not equipped with initial conditions for the pressure. This can be an issue depending on the temporal discretisation considered, as discussed for instance by Rang [38]. In the pressure Poisson framework, on the other hand, we naturally compute this initial value by seeking $p_0 \in Q^l_{\zeta_0}$ such that

$$\begin{aligned}
\int_\Omega & \frac{1}{\rho_0} \nabla p_0 \cdot \nabla q \, \mathrm{d}\Omega \\
&= \int_\Gamma (\boldsymbol{n} \times \nabla q) \cdot (\nu_0 \nabla \times \boldsymbol{u}_0) \, \mathrm{d}\Gamma \\
&\quad - \int_{\Gamma_D} q \partial_t (\boldsymbol{n} \cdot \boldsymbol{u}_0) \, \mathrm{d}\Gamma - \int_\Omega q \beta_0 \nabla \cdot \boldsymbol{u}_0 \, \mathrm{d}\Omega + \int_\Omega \nabla q \\
&\quad \cdot \left[\boldsymbol{g}_0 - (\nabla \boldsymbol{u}_0) \boldsymbol{u}_0 + \frac{2\nu_0}{\rho_0} \nabla^{\text{s}} \boldsymbol{u}_0 \nabla \rho_0 + 2(\nabla \boldsymbol{u}_0)^\top \nabla \nu_0\right] \mathrm{d}\Omega
\end{aligned} \tag{36}$$

for all $q \in Q^l_0$. This is why it is crucial, in deriving the PPE, to divide the momentum equation by ρ before taking the divergence. If we were to apply $\nabla \cdot$ directly to the momentum

equation, the transient term would result in

$$\nabla \cdot (\rho \partial_t \boldsymbol{u}) = \rho \partial_t (\nabla \cdot \boldsymbol{u}) + \partial_t \boldsymbol{u} \cdot \nabla \rho = \partial_t \boldsymbol{u} \cdot \nabla \rho \not\equiv 0,$$

instead of $\nabla \cdot (\partial_t \boldsymbol{u}) = \partial_t (\nabla \cdot \boldsymbol{u}) \equiv 0$. Consequently, the integral on Γ_D in Eq. (36) would be replaced by a volume integral depending on the acceleration $\partial_t \boldsymbol{u}$. In that case, initialising the time-marching would not be straightforward, since the initial acceleration is not given for flow equations. That would require using a forward discretisation of $\partial_t \boldsymbol{u}$, resulting in an implicit step. Our formulation, on the other hand, requires only the time derivative of the normal data $\mathbf{v} \cdot \boldsymbol{n}$ on Γ_D, which may sound similar to the situation just described, yet is fundamentally different. Since $\mathbf{v}$ is given, its time derivative is either known exactly (e.g. when artificially ramping up inflow data), or can be approximated by a forward finite difference—which does *not* lead to an implicit coupling, since $\mathbf{v} \cdot \boldsymbol{n}$ is known *a priori* at all times. This discussion is applicable also to the initial-boundary value problem in strong form, hence it holds regardless of the spatial discretisation. Thus, using only *given* quantities, we are able to compute p_0 and start marching in time.

6.2 First-order stepping

As mentioned in Sect. 5.1, our first-order stepping combines BDF1 with linear extrapolation ($\boldsymbol{u}_{n+1} \approx \hat{\boldsymbol{u}}_{n+1} = \boldsymbol{u}_n$ and $p_{n+1} \approx \hat{p}_{n+1} = p_n$). Given $(\rho_n, \boldsymbol{u}_n, p_n)$, we march in time by executing the following steps:

1. *Convective step*
   Find $\rho_{n+1} \in R^m_{\varrho_{n+1}}$ such that

$$\int_\Omega (r + \Delta t \boldsymbol{u}_n \cdot \nabla r)\, (\rho_{n+1} + \Delta t \boldsymbol{u}_n \cdot \nabla \rho_{n+1} - \rho_n)\, \mathrm{d}\Omega = 0 \tag{37}$$

   for all $r \in R^m_0$.
2. *Viscosity update*
   Compute ν_{n+1} based on the updated phase distribution described by ρ_{n+1}.
3. *Velocity update*
   Find $\boldsymbol{u}_{n+1} \in W^k_{\mathbf{v}_{n+1}}$ such that

$$\int_\Omega \rho_{n+1} \left\{ \boldsymbol{w} \cdot \left[(\nabla \boldsymbol{u}_{n+1}) \boldsymbol{u}_n \right] + \frac{1}{\Delta t} \boldsymbol{w} \cdot \boldsymbol{u}_{n+1} \right.$$
$$\left. + 2\nu_{n+1} \nabla^s \boldsymbol{u}_{n+1} : \nabla^s \boldsymbol{w} \right\} \mathrm{d}\Omega$$
$$= \int_\Omega p_n \nabla \cdot \boldsymbol{w}\, \mathrm{d}\Omega + \int_\Omega \rho_{n+1} \boldsymbol{w} \cdot \left(\frac{1}{\Delta t} \boldsymbol{u}_n + \boldsymbol{g}_{n+1} \right) \mathrm{d}\Omega$$
$$+ \int_{\Gamma_N} \boldsymbol{w} \cdot \boldsymbol{t}_{n+1}\, \mathrm{d}\Gamma \tag{38}$$

   for all $\boldsymbol{w} \in W^k_0$.

4. *Pressure Poisson step*
   Find $p_{n+1} \in Q^l_{\zeta_{n+1}}$ such that

$$\int_\Omega \frac{1}{\rho_{n+1}} \nabla p_{n+1} \cdot \nabla q\, \mathrm{d}\Omega$$
$$= \int_\Gamma (\boldsymbol{n} \times \nabla q) \cdot (\nu_{n+1} \nabla \times \boldsymbol{u}_{n+1})\, \mathrm{d}\Gamma$$
$$- \int_{\Gamma_D} q\, \partial_t (\boldsymbol{n} \cdot \mathbf{v}_{n+1})\, \mathrm{d}\Gamma - \int_\Omega q \beta_{n+1} \nabla \cdot \boldsymbol{u}_{n+1}\, \mathrm{d}\Omega$$
$$+ \int_\Omega \nabla q \cdot \Big[\boldsymbol{g}_{n+1} - (\nabla \boldsymbol{u}_{n+1}) \boldsymbol{u}_{n+1}$$
$$+ \frac{2\nu_{n+1}}{\rho_{n+1}} \nabla^s \boldsymbol{u}_{n+1} \nabla \rho_{n+1} + 2 (\nabla \boldsymbol{u}_{n+1})^\top \nabla \nu_{n+1} \Big] \mathrm{d}\Omega \tag{39}$$

   for all $q \in Q^l_0$.

6.3 Second-order stepping

Our second-order scheme combines BDF2 with linear extrapolation ($\boldsymbol{u}_{n+1} \approx \hat{\boldsymbol{u}}_{n+1} = 2\boldsymbol{u}_n - \boldsymbol{u}_{n-1}$ and $p_{n+1} \approx \hat{p}_{n+1} = 2p_n - p_{n-1}$). Therefore, initialisation requires, for example, using a first-order step to compute $(\rho_1, \boldsymbol{u}_1, p_1)$. Then, we can march in time by executing the following steps:

1. *Convective step*
   Find $\rho_{n+1} \in R^m_{\varrho_{n+1}}$ such that

$$\int_\Omega \left(3r + 2\Delta t \hat{\boldsymbol{u}}_{n+1} \cdot \nabla r \right) \left(3\rho_{n+1} + 2\Delta t \hat{\boldsymbol{u}}_{n+1} \right.$$
$$\left. \cdot \nabla \rho_{n+1} - 4\rho_n + \rho_{n-1} \right) \mathrm{d}\Omega = 0 \tag{40}$$

   for all $r \in R^m_0$.
2. *Viscosity update*
   Compute ν_{n+1} based on the updated phase distribution described by ρ_{n+1}.
3. *Velocity update*
   Find $\boldsymbol{u}_{n+1} \in W^k_{\mathbf{v}_{n+1}}$ such that

$$\int_\Omega \rho_{n+1} \left\{ \boldsymbol{w} \cdot \left[(\nabla \boldsymbol{u}_{n+1}) \hat{\boldsymbol{u}}_{n+1} \right] + \frac{3}{2\Delta t} \boldsymbol{w} \cdot \boldsymbol{u}_{n+1} \right.$$
$$\left. + 2\nu_{n+1} \nabla^s \boldsymbol{u}_{n+1} : \nabla \boldsymbol{w} \right\} \mathrm{d}\Omega$$
$$= \int_\Omega \hat{p}_{n+1} \nabla \cdot \boldsymbol{w}\, \mathrm{d}\Omega + \int_\Omega \rho_{n+1} \boldsymbol{w}$$
$$\cdot \left(\frac{2}{\Delta t} \boldsymbol{u}_n - \frac{1}{2\Delta t} \boldsymbol{u}_{n-1} + \boldsymbol{g}_{n+1} \right) \mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{w} \cdot \boldsymbol{t}_{n+1}\, \mathrm{d}\Gamma \tag{41}$$

   for all $\boldsymbol{w} \in W^k_0$.

4. *Pressure Poisson step*

Find $p_{n+1} \in Q^l_{\zeta_{n+1}}$ such that Eq. (39) holds for all $q \in Q^l_0$.

In both schemes, the pressure Poisson step is fully implicit, that is, the current pressure is computed directly from the current velocity and density fields. Similarly, there is no need to extrapolate the density when updating the velocity, since we already know ρ_{n+1} when computing $\boldsymbol{u}_{n+1}$. If the order of the steps is inverted, with $\boldsymbol{u}$ updated before ρ, we need to replace ρ_{n+1} by $\hat{\rho}_{n+1}$ in the velocity step, but on the other hand $\hat{\boldsymbol{u}}_{n+1}$ could be replaced by $\boldsymbol{u}_{n+1}$ in the convective step.

6.4 Convection-dominated flows

At high Reynolds numbers, the Galerkin finite element approximation of the momentum balance suffers from convective instabilities, unless a very fine mesh is used. A simple remedy for that is the streamline upwind/Petrov–Galerkin (SUPG) stabilisation. On the other hand, dominant convection also means that we can treat the viscous term explicitly (or semi-implicitly) without incurring a parabolic time-step restriction [28]. In the weak form of the momentum equation, the viscous term reads

$$\int_\Omega 2\mu \nabla^{\mathrm{s}}\boldsymbol{u} : \nabla^{\mathrm{s}}\boldsymbol{w}\,\mathrm{d}\Omega = \int_\Omega \mu \nabla\boldsymbol{u} : \nabla\boldsymbol{w}\,\mathrm{d}\Omega \\ + \int_\Omega \mu \nabla\boldsymbol{u} : (\nabla\boldsymbol{w})^\top\,\mathrm{d}\Omega\,.$$

The last term has a coupling effect over the velocity components, so that treating it explicitly decouples the velocity update step into d scalar convection-diffusion equations (see Ref. [30] for details). This reduces computational costs and allows us to apply SUPG individually to each of the scalar equations [39]. Let us, for concision, write the ith component of $\boldsymbol{u}$ at $t = t_{n+1}$ as u^i, and $\hat{v}_{n+1}$ as $\hat{v}$ for any extrapolated quantity v. In that case, our semi-discretised balance of momentum in the x_i direction reads

$$\alpha_0\rho u^i + \rho\hat{\boldsymbol{u}} \cdot \nabla u^i - \nabla \cdot (\rho v \nabla u^i) \\ = \overbrace{\nabla \cdot \left(\rho v \frac{\partial\hat{\boldsymbol{u}}}{\partial x_i}\right) + \rho g^i - \frac{\partial\hat{p}}{\partial x_i} - \sum_{j=1}^m \alpha_j \rho u^i_{n+j-1}}^{:=f_i}\,, \quad (42)$$

with α_j denoting the coefficients of the finite difference discretisation ($\alpha_0 = -\alpha_1 = 1/\Delta t$ for BDF1 or $\alpha_2 = -\alpha_1/4 = \alpha_0/3 = 0.5/\Delta t$ for BDF2). For the SUPG method, we add to the left-hand side of the weak momentum equation the

nonlinear form

$$\sum_{e=1}^{N_e} \int_{\Omega_e} \left\{\tau_{\mathrm{SUPG}}\rho\hat{\boldsymbol{u}} \cdot \nabla w\right\}\left\{\alpha_0\rho u^i + \rho\hat{\boldsymbol{u}} \cdot \nabla u^i - \nabla \cdot \right. \\ \left. \left[\rho v\left(\nabla u^i + \frac{\partial\hat{\boldsymbol{u}}}{\partial x_i}\right)\right] - f_i\right\}\mathrm{d}\Omega\,, \quad (43)$$

where τ_{SUPG} is a mesh-dependent parameter and w is a scalar test function. Notice that this stabilisation is in general consistent, as the added term is proportional to the residual of Eq. (42). However, this is not the case when using linear elements, which are unable to approximate the second-order derivatives in the viscous term. Although this is in principle not an issue for stability, the missing terms may lead to inaccuracy on flow regions where diffusion dominates. Therefore, for first-order elements we propose replacing (43) by

$$\sum_{e=1}^{N_e} \int_{\Omega_e} \left(\tau_{\mathrm{SUPG}}\rho\hat{\boldsymbol{u}} \cdot \nabla w\right) \\ \left(\alpha_0\rho u^i + \rho\hat{\boldsymbol{u}} \cdot \nabla u^i - \sum_{j=1}^d \frac{\partial\hat{S}_{ij}}{\partial x_j} - f_i\right)\mathrm{d}\Omega\,, \quad (44)$$

where $\hat{S}_{ij}$ denotes the components of the $L^2(\Omega)$ projection of the viscous stress tensor $\mathbb{S}$ onto a continuous finite element space. In other words, we find $\hat{S}_{ij} \in X_h$ such that

$$\int_\Omega \hat{S}_{ij}\sigma\,\mathrm{d}\Omega = \int_\Omega \left(\frac{\partial\hat{u}^i}{\partial x_j} + \frac{\partial\hat{u}^j}{\partial x_i}\right)\rho v\sigma\,\mathrm{d}\Omega$$

for all $\sigma \in X_h$, in which X_h can be taken, for instance, as the pressure finite element space (without boundary conditions). Projecting the viscous stress tensor onto a continuous finite element space is also usual in certain stabilisation [40,41] and projection methods [3]. Since $\mathbb{S}$ is symmetric, the "cost" of this projection are $d(d + 1)/2$ scalar mass-matrix systems, but there are two simple ways to basically eliminate the overhead. One is to store the LDL decomposition of the mass matrix during initialisation and re-use it for all time steps; alternatively, the mass matrix can be lumped, which is our approach herein. As for the stabilisation parameter, we use

$$\tau_{\mathrm{SUPG}} = \left(\frac{2\rho|\hat{\boldsymbol{u}}|}{h} + \rho\alpha_0 + \frac{4\rho v}{h^2}\right)^{-1}\,,$$

which is simply one among several possibilities [42].

7 Numerical examples

In this section, we assess the accuracy of our fractional-step framework through test cases with increasing complexity. One of the main advantages of our approach with respect to projection methods is allowing equal-order interpolation of all unknowns. Regarding the velocity-pressure pairing, we consider linear-linear ($\boldsymbol{P}_1 P_1$) and Taylor–Hood ($\boldsymbol{P}_2 P_1$) triangular elements; for numerical examples with other element classes, including three-dimensional ones, refer to our recent work on split-step schemes for non-Newtonian flows [30]. In all examples, a first-order discretisation (P_1) is used for the density.

7.1 Manufactured solution

We start with a simple problem having an analytical solution and a pure Dirichlet boundary. In the square domain $\Omega = (0, 1)^2$, we consider the flow described by

$$p = (2 - 2x) f(t), \quad \boldsymbol{u} = \begin{pmatrix} u(y) f(t) \\ 0 \end{pmatrix},$$

$$\rho = f(t) e^{x/u(y)}, \quad \boldsymbol{g} = \begin{pmatrix} -u(y) f^2(t) \\ 0 \end{pmatrix},$$

in which $f(t) = (1+t)^{-1}$ and $u(y) = 1 + y - y^2$. The accuracy of the first-order ($\boldsymbol{P}_1 P_1 P_1$, BDF1) and second-order ($\boldsymbol{P}_2 P_1 P_1$, BDF2) schemes is assessed through the spatial error at $t = T = 1$. Divergence penalty is not used in this example, i.e., we set $\beta = 0$ in the PPE. The velocity error is measured in the relative $\mathbf{H}^1(\Omega)$ semi-norm, whereas for pressure and density we use the relative $L^2(\Omega)$ norm. Starting with a coarse orthogonal mesh of four triangles, seven levels of uniform spatial refinement are applied, with the time-step size also halved at each level via $\Delta t = h/2$. The results of the convergence study are shown in Figure 1. As expected, the first-order elements yield linear convergence, whereas the Taylor–Hood elements converge quadratically when combined with BDF2. Notice, however, that the exact solution for this problem is smooth, so that optimal convergence is attainable with standard discretisations. In flows with more than one phase, on the other hand, the spatial convergence will in general not be optimal unless velocity kinks and possible pressure discontinuities are appropriately resolved [7].

7.2 Rayleigh–Taylor instability

One of the most popular benchmarks for variable-density flow solvers is the Rayleigh–Taylor instability with regularised interface [5,43]. The standard setup considers two fluids initially at rest in $\Omega = (0, a/2) \times (-2a, 2a)$ and sub-
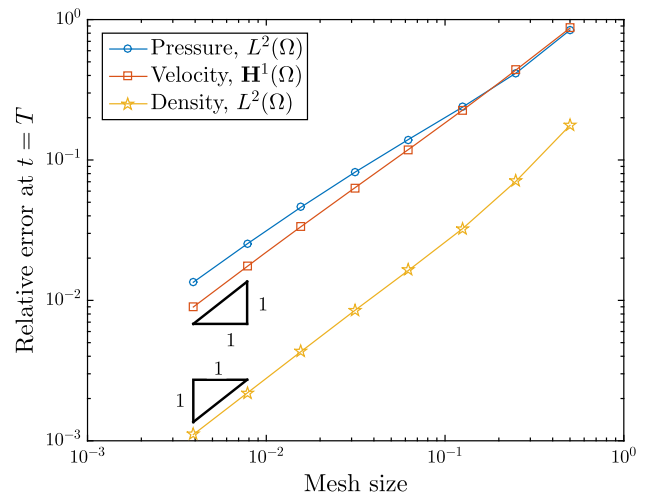
(a) First-order scheme ($\boldsymbol{P}_1 P_1 P_1$, BDF1).
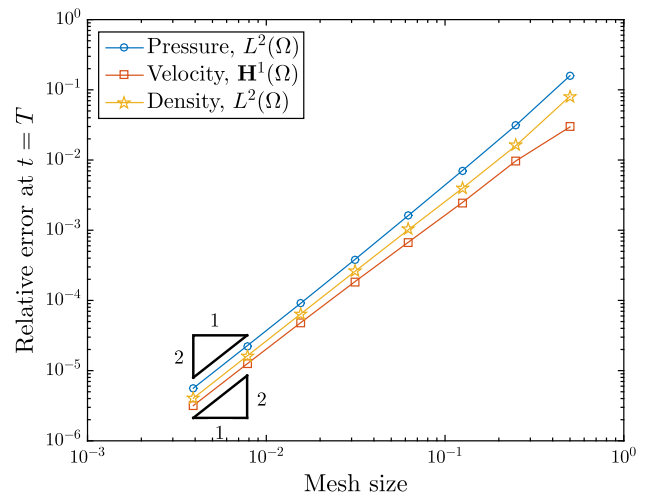
(b) Second-order scheme ($\boldsymbol{P}_2 P_1 P_1$, BDF2).

Fig. 1 Uniform refinement study considering triangular elements and a problem with smooth exact solution

ject to $\boldsymbol{g} = (0, -g)^\top$. Fluid 2 sits initially on top of fluid 1 and is denser ($\rho_2 > \rho_1$) but equally viscous ($\mu_1 = \mu_2$). The initial transition is regularised via

$$\rho|_{t=0} = \frac{\rho_2 + \rho_1}{2} + \frac{\rho_2 - \rho_1}{2} \tanh\left(\frac{y - \iota(x)}{a/100}\right),$$

where $\iota(x) = -0.1a \cos(2\pi x/a)$ describes the interface line at $t = 0$. We nondimensionalise the problem through the following reference quantities: ν_1 for viscosity, ρ_1 for density, a for length and $\sqrt{ag}$ for velocity, which gives us $\tau = \sqrt{a/g}$ as temporal scale. Then, the problem is fully parametrised through the Reynolds and Atwood numbers:
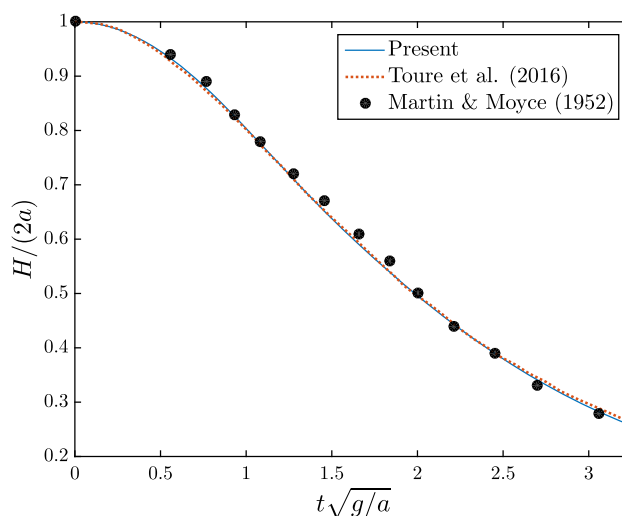
$$\text{Re} = \frac{a\sqrt{ag}}{\nu_1} \quad \text{and} \quad \text{At} = \left|\frac{\rho_2 - \rho_1}{\rho_2 + \rho_1}\right|,$$

Fig. 2 Rayleigh–Taylor instability: density field at $t/\tau = 0, 1.40, 2.10, 2.45, 2.80, 3.15$ and 3.50
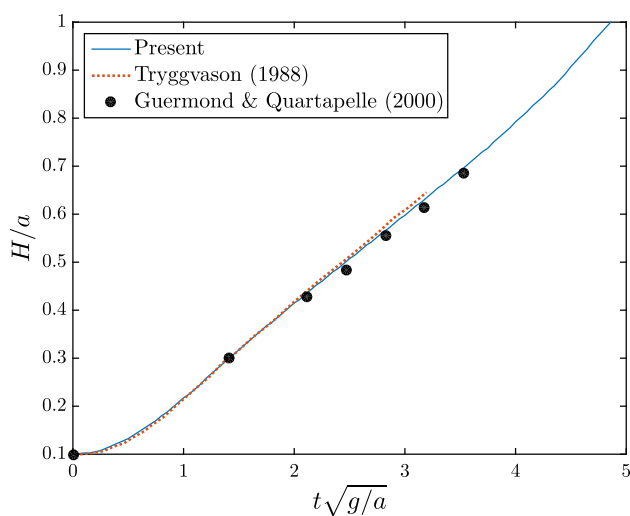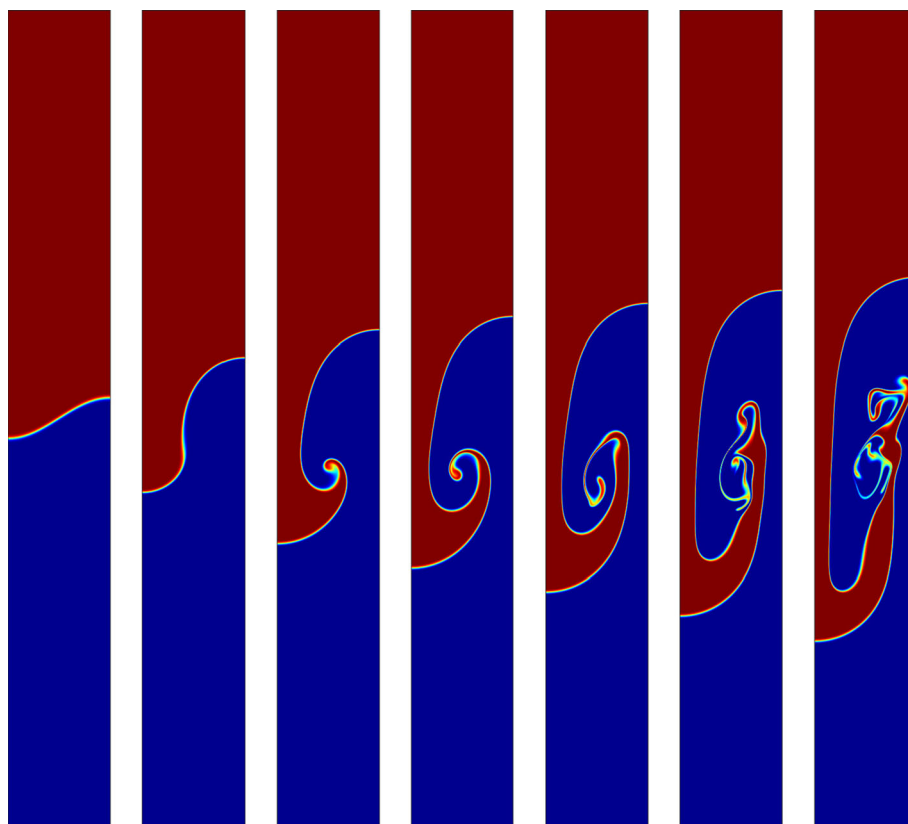
Fig. 3 Rayleigh–Taylor instability: vertical position H of the rising bubble, measured from the mid-height of the domain. The comparison with previously reported numerical results [5,43] reveals very good agreement

Fig. 4 Dam break: height H of the water column on the left wall, showing good agreement with other numerical [22] and experimental [44] results

which are set as 1000 and 0.5, respectively. For the simulation, we use the first-order scheme with $\alpha = 0.01$, $\Delta t = \tau/1000$ and a uniform mesh with 6.4×10^5 elements. No SUPG stabilisation is used, which is possible due to the relatively fine mesh. The top and bottom walls are no-slip boundaries, while symmetry is enforced on the sides.

Figure 2 shows the density field for different time instants, agreeing well with the results reported by Guermond and Salgado [11] (mind that they use a different time scale $\tilde{\tau} = \tau\sqrt{\text{At}}$). For a quantitative comparison, Figure 3 shows how the height H of the rising bubble evolves in time. Our results

agree well with the simulations reported by Tryggvason [43] and Guermond and Quartapelle [5].

7.3 Dam break

We now move to a more challenging setup involving air and water. The dam break experiment by Martin and Moyce [44] is a popular benchmark for two-phase [22,45] and free-surface [46] flow simulations. This is a high-Reynolds, gravity-driven flow and, as such, suits our application scope very well. The setup considered by Landet et al. [22] consists of a tank $\Omega = (0, 5a) \times (0, 3a)$ containing a rectangular water column $(0, a) \times (0, 2a)$ surrounded by air, initially at rest, with free-slip walls and $a = 57.15$ mm. We set $\rho_{\text{air}} =$ $3\rho_{\text{water}}/2500 = 1.2$ kg/m^3 and $\nu_{\text{air}} = 15\nu_{\text{water}} = 1.5 \times 10^{-5}$ m^2/s as material properties, and $\boldsymbol{g} = (0, -9.81)^\top$ m/s^2 for gravity.

Due to the large density difference between air and water, for this example we replace the explicit mass convection (1) by the level-set equation
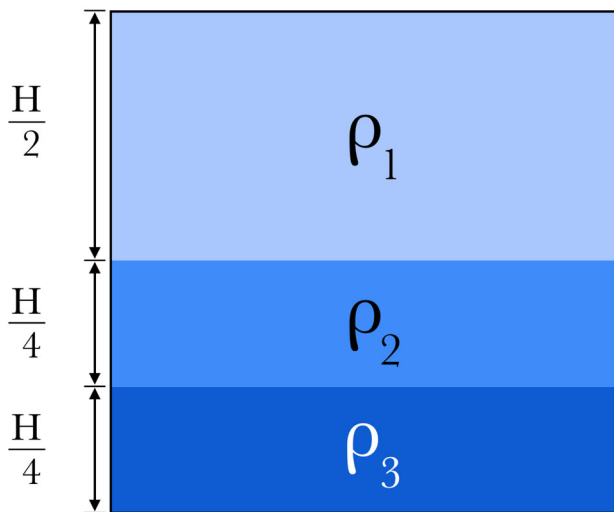
$$\partial_t \phi + \boldsymbol{u} \cdot \nabla \phi = 0 ,$$

with ϕ being a signed distance function: $\phi > 0$ corresponds to air, $\phi < 0$ to water, and $\phi = 0$ to the interface. We can therefore write
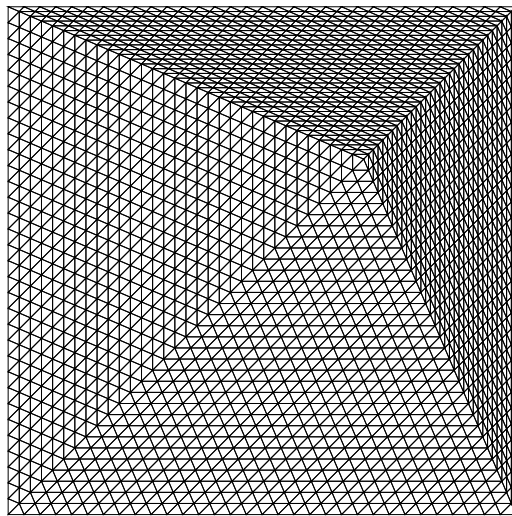
$$\rho = \frac{\rho_{\text{water}} + \rho_{\text{air}}}{2} + \frac{\rho_{\text{air}} - \rho_{\text{water}}}{2} \operatorname{sign} \phi ,$$

Fig. 5 Dam break: velocity norm and density field at $t\sqrt{g/a} \approx 0.5, 1.5, 2.5$ and 3.2
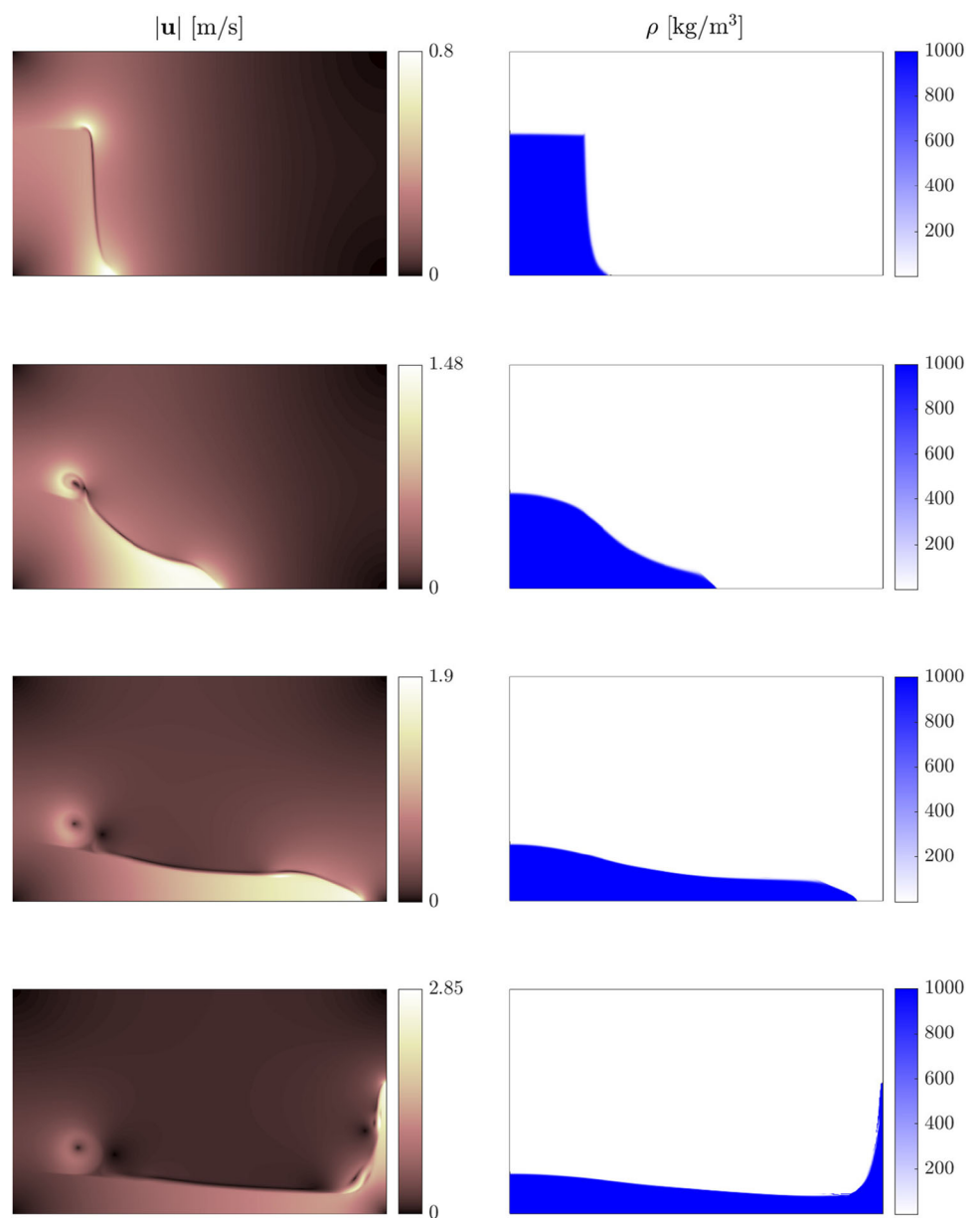
(a) Initial phase distribution with $\rho_3 > \rho_2 > \rho_1$ and uniform ν.

(b) Grid used for both $\boldsymbol{P}_2 P_1 P_1$ and $\boldsymbol{P}_1 P_1 P_1$ discretisations.

Fig. 6 Three-phase channel flow: setup and grid

and analogously for the kinematic viscosity. Since both $\nabla \rho$ and $\nabla \nu$ appear in the PPE (27), it is useful to regularise the discontinuity. A common approach is to replace the sign function by a hyperbolic tangent [47], which is also done for this example:

$$\rho = \frac{\rho_{\text{water}} + \rho_{\text{air}}}{2} + \frac{\rho_{\text{air}} - \rho_{\text{water}}}{2} \tanh k\phi, \quad k = 10^3.$$

An excellent discussion on different variants of the level-set method is provided by Olsson and Kreiss [48]. Most importantly, our formulation remains consistent regardless of how the density is treated, as the PPE is derived solely from the momentum equation.

To handle the dominant convection, especially in the aerodynamic flow, we also use SUPG stabilisation. The

discretisation is again first-order, with 3×10^5 triangular elements and $\Delta t = 10^{-4}$ s. The main benchmark quantity here is the temporal evolution of the water column's height, H. We compare our solution to the measurements by Martin and Moyce [44] and the numerical results by Touré et al. [45], who used a modified level-set method with corrected mass conservation. As shown in Figure 4, all three results are in very good agreement.

To illustrate the complex flow emerging from this simple example, we show in Figure 5 the Euclidian norm of the velocity field for different instants in time. The downward motion of the water column "propels" the air upwards and, as the air goes past the column's corner, a vortex is created as in a backward-facing step flow. As the water column keeps going down, two adjacent air vortices can be seen. We also observe smaller vortices around the water surge front as it shoots upwards after meeting the right wall. In the same figure, the density plot reveals quite a sharp interface, in spite of the regularisation.

7.4 Three-phase channel flow

As our last test case, we investigate a three-phase flow in the straight channel $\Omega = (0, L) \times (0, H)$. The three fluids are initially at rest, stacked up as illustrated in Figure 6 (a). They all have the same kinematic viscosity ν, but different dynamic viscosities since $\rho_3 = 1.5\rho_2 = 3\rho_1 = 1.5$. For a constant pressure gradient $\frac{\partial p}{\partial x} = k$, we can calculate the analytical solution for the stationary Poiseuille-like profile:

$$u(y) = \begin{cases} \frac{kH^2}{2\mu_3} \left[\left(\frac{y}{H}\right)^2 - \frac{\gamma}{4}\left(\frac{y}{H}\right) \right], & \text{for } 0 \leq y \leq \frac{H}{4} \\ \frac{(1-\gamma)kH^2}{32\mu_3} + \frac{kH^2}{2\mu_2} \\ \cdot \left[\left(\frac{y}{H}\right)^2 - \frac{\gamma}{4}\left(\frac{y}{H}\right) + \frac{(\gamma-1)}{16} \right], & \text{for } \frac{H}{4} \leq y \leq \frac{H}{2} \\ \frac{(1-\gamma)kH^2}{32\mu_3} + \frac{(3-\gamma)kH^2}{32\mu_2} + \frac{kH^2}{2\mu_1} \\ \cdot \left[\left(\frac{y}{H}\right)^2 - \frac{\gamma}{4}\left(\frac{y}{H}\right) + \frac{(\gamma-2)}{8} \right], & \text{for } \frac{H}{2} \leq y \leq H \end{cases}$$

in which

$$\gamma = \frac{\mu_1\mu_2 + 3\mu_1\mu_3 + 12\mu_2\mu_3}{\mu_1\mu_2 + \mu_1\mu_3 + 2\mu_2\mu_3}.$$

For concision, we omit the derivation, but it can be done in the standard way: solving the stationary momentum equation and enforcing continuity of velocities and viscous stresses at the interfaces.

The main idea with this example is to illustrate how the penalty-like term $\beta \nabla \cdot \boldsymbol{u}$ (cf. Eq. (12)) can improve mass conservation on the discrete level. For this, we consider a test case with prescribed inflow, but an open outflow. When reaching a stationary solution, we would ideally like to see the same profile on the open outlet ($x = L$) as the one prescribed

on the inlet ($x = 0$). So, we enforce the inflow profile $\boldsymbol{u} = (u(y)f(t), 0)^\top$, with

$$f(t) = \frac{1}{2}\left[1 - \text{sign}\left(|2t - t^\star| - t^\star\right)\right]\sin^2\left(\frac{\pi t}{2t^\star}\right)$$
$$+ \frac{1}{2}\left[1 + \text{sign}(t - t^\star)\right]$$

ramping up smoothly from zero (for $t \leq 0$) to unit value (for $t \geq t^\star$). For the density, the inlet boundary condition is constant in time, obeying the initial phase distribution.

Since prescribing zero (or constant) normal traction $\boldsymbol{t}$ does not allow reproducing a developed flow, we swap the traction boundary condition (8) by a pseudo-traction condition on the outflow boundary:

$$(\mu\nabla\boldsymbol{u})\boldsymbol{n} - p\boldsymbol{n} = \tilde{\boldsymbol{t}},$$

with $\tilde{\boldsymbol{t}}$ being some normal pseudo-traction data, usually taken as zero in flows with one single outlet. To naturally impose the pseudo-traction, we switch from a stress-divergence formulation of the momentum equation to a generalised Laplacian approach [49]. In that case, the Dirichlet value (15) for the pressure has to be slightly adapted:

$$p = \mu\nabla\boldsymbol{u}^s : (\boldsymbol{n} \otimes \boldsymbol{n} - \mathbb{I}) - \tilde{\boldsymbol{t}} \cdot \boldsymbol{n} \quad \text{on } \Gamma_N.$$

Details on outflow boundary conditions and the generalised Laplacian formulation can be found in Refs. [30,49]. Then, by setting $\tilde{\boldsymbol{t}} = \boldsymbol{0}$, $\boldsymbol{g} = \boldsymbol{0}$, $T = 2.5t^\star = 1$, $L = H = 1$, $\nu = 1$ and $k = -1$, we have the complete problem setup.

Fig. 7 Three-phase channel flow: solutions obtained with ($\alpha > 0$) and without ($\alpha = 0$) divergence penalty. The density field, on the right, shows no remarkable differences between all the schemes. On the left, the (prescribed) inflow and (computed) outflow profiles, normalised by $\bar{u} := |0.5kH^2/\mu_1|$, showcase the improved mass conservation for $\alpha = 0.002$ in the first-order scheme
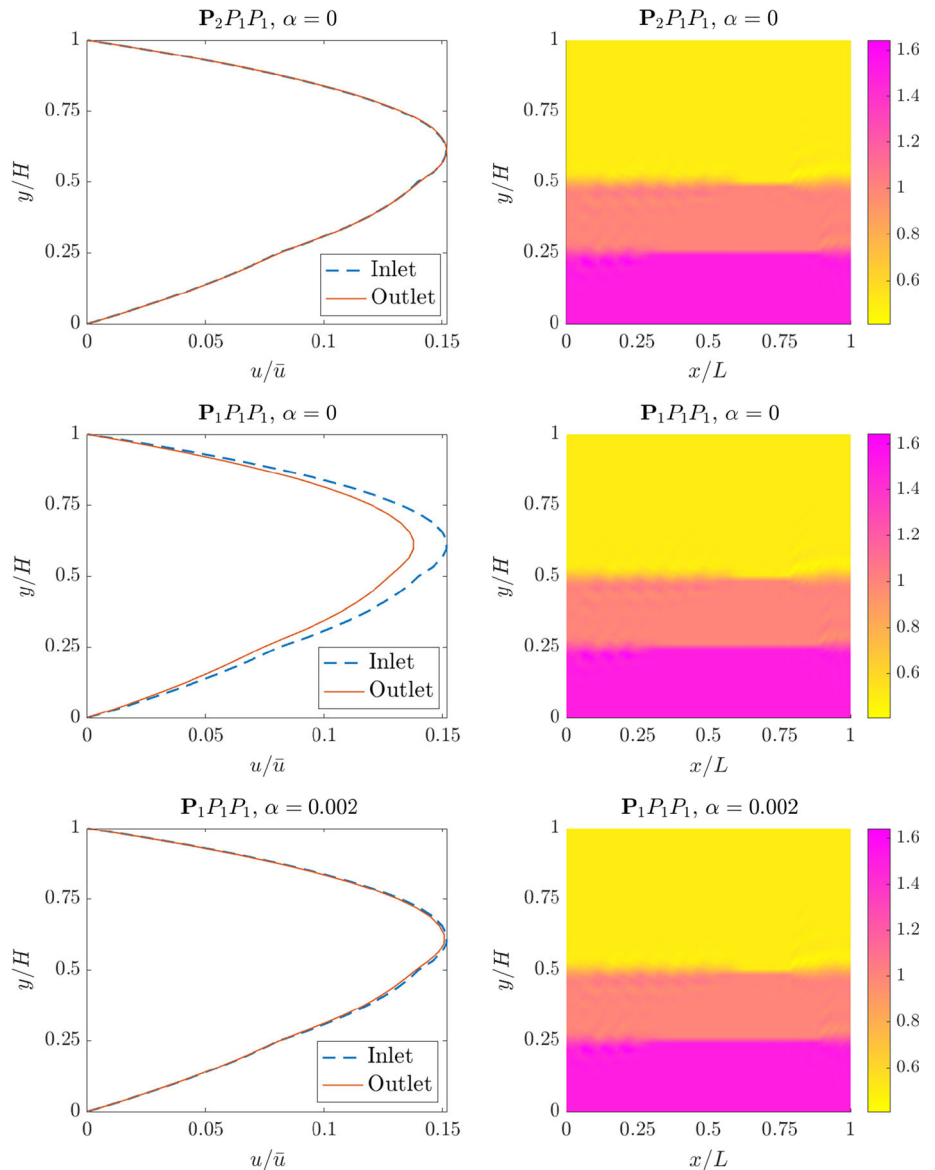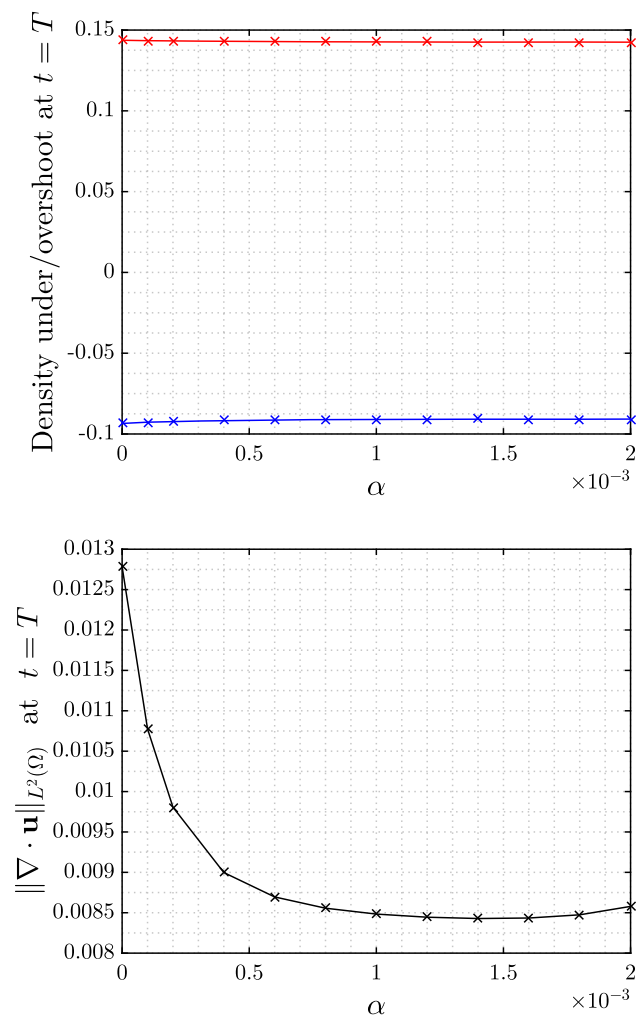
Fig. 8 Three-phase channel flow: how the divergence error and the density under- and overshoot are affected by the divergence penalty parameter α, using $\boldsymbol{P}_1 P_1 P_1$ elements

For the temporal discretisation, this time we use BDF2 with $\Delta t = 0.02$. If the interfaces were to lie completely on grid lines, the velocity kinks would be resolved by the spatial discretisation; to avoid that and be more realistic, we use the non-orthogonal mesh shown in Figure 6 (b). The first two test cases employ $\boldsymbol{P}_2 P_1 P_1$ and $\boldsymbol{P}_1 P_1 P_1$ elements, respectively, without any divergence penalty ($\alpha = 0$, *cf.* Eq. (35)), whereas the third one has $\alpha = 0.002$ and also first-order elements.

Figure 7 depicts the density field and the velocity profiles at $t = 1$, when the developed steady-state solution is already established. The density distribution is similar in all cases, exhibiting interface oscillations due to the relatively coarse mesh for a multi-phase flow. In spite of that, the velocity profile shows no apparent oscillations. The plots reveal how even a small α can virtually eliminate the mass loss otherwise seen for the $\boldsymbol{P}_1 P_1 P_1$ elements, whereas the second-order discretisation shows no relevant loss even without any penalty. We must remark, however, that finding a good value for α

requires some tuning: setting α too large can lead to instability, by giving excessive weight to the velocity divergence in comparison to the pressure-stabilising term $-\nabla \cdot (\rho^{-1}\nabla p)$. Figure 8 shows how α affects the divergence error and the density over- and undershoot at $t = T$. The density results show hardly any difference, as previously indicated in Figure 7. The divergence error, on the other hand, reduces by 34% when we change α from 0 to 1.4×10^{-3}. While sensitivity analysis is out of the present scope, a detailed study on how this parameter influences the accuracy of different quantities is found in our previous work for single-phase flows [37].

8 Concluding remarks

In this work, we have devised, implemented and thoroughly tested a consistent split-step framework for flows with non-constant density and viscosity. Most splitting schemes for multi-phase flows are based on pressure correction methods enforcing incompressibility via solenoidal projections. Those methods are very efficient but have well-known shortcomings, such as numerical boundary layers and/or spatial stability restrictions when using finite elements. Our method, on the other hand, is based on a consistent pressure-Poisson reformulation of the Navier–Stokes system, thereby avoiding those issues without incurring computational overhead. Moreover, the method is minimally invasive: it decomposes the variable-density Navier–Stokes system into simpler convection, diffusion and convection-diffusion problems, and the pressure Poisson step is consistent regardless of the technique—if any—applied to handle discontinuities. No spurious pressure boundary layers or finite element compatibility restrictions are induced.

As a matter of fact, consistent PPE-based schemes have recently been used in designing accurate, competitive solvers for challenging applications such as non-Newtonian flows, fluid–structure interaction and free-surface flows [30,36,50–52]. Of course, a consistent pressure Poisson equation has a more complex right-hand side than a Leray projection does, but our analytical efforts in Sects. 3 and 4 ensure that this complexity does not translate into an implementation overhead. Indeed, in the end we get a weak formulation with the exact same left-hand side as in classical projection methods [5], and a right-hand side allowing standard finite element spaces—mixed or equal-order. Although we have focused on finite element methods, our PPE-based system can be tackled within basically any other framework. It is also straightforward to consider generalised Newtonian fluids, since our equations already account for variable viscosity.

What our formulation does *not* improve on, with respect to projection methods, is the pressure operator. In both cases we have $-\nabla \cdot (\rho^{-1}\nabla p)$, which is not optimal in the presence of pressure discontinuities, as seen e.g. in surface tension mod-

els. In particular, piecewise constant finite element spaces are not allowed for the pressure, which is a recurring limitation in split-step methods. To overcome that, we are currently working on an ultra-weak reformulation of the problem, with promising results so far [53].

References

1. Helmig J, Behr M, Elgeti S (2019) Boundary-conforming finite element methods for twin-screw extruders: unsteady-temperature-dependent-non-Newtonian simulations. Comput Fluids 190:322–336

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

3. Deteix J, Yakoubi D (2019) Shear rate projection schemes for non-Newtonian fluids. Comput Methods Appl Mech Eng 354:620–636

4. Plasman L, Deteix J, Yakoubi D (2020) A projection scheme for Navier–Stokes with variable viscosity and natural boundary condition. Int J Numer Meth Fluids 92(12):1845–1865

5. Guermond J-L, Quartapelle L (2000) A projection FEM for variable density incompressible flows. J Comput Phys 165(1):167–188

6. Castillo E, Baiges J, Codina R (2015) Approximation of the two-fluid flow problem for viscoelastic fluids using the level set method and pressure enriched finite element shape functions. J Nonnewton Fluid Mech 225:37–53

7. Groß S, Reusken A (2007) An extended pressure finite element space for two-phase incompressible flows with surface tension. J Comput Phys 224(1):40–58

8. Castillo E, Cruchaga MA, Baiges J, Flores J (2019) An oil sloshing study: adaptive fixed-mesh ALE analysis and comparison with experiments. Comput Mech 63(5):985–998

9. Fraigneau Y, Guermond J-L, Quartapelle L (2001) Approximation of variable density incompressible flows by means of finite elements and finite volumes. Commun Numer Methods Eng 17(12):893–902

10. Pyo J-H, Shen J (2007) Gauge–Uzawa methods for incompressible flows with variable density. J Comput Phys 221(1):181–197

11. Guermond JL, Salgado A (2009) A splitting method for incompressible flows with variable density based on a pressure Poisson equation. J Comput Phys 228(8):2834–2846

12. Axelsson O, He X, Neytcheva M (2015) Numerical solution of the time-dependent Navier–Stokes equation for variable density-variable viscosity. Part I. Math Model Anal 20(2):232–260

13. Li M, Cheng Y, Shen J, Zhang X (2021) A bound-preserving high order scheme for variable density incompressible Navier–Stokes equations. J Comput Phys 425:109906

14. Baiges J, Codina R, Pont A, Castillo E (2017) An adaptive fixed-mesh ALE method for free surface flows. Comput Methods Appl Mech Eng 313:159–188

15. Li Y, Mei L, Ge J, Shi F (2013) A new fractional time-stepping method for variable density incompressible flows. J Comput Phys 242:124–137

16. Bonito A, Guermond J-L, Lee S (2016) Numerical simulations of bouncing jets. Int J Numer Methods Fluids 80(1):53–75

17. Saye R (2017) Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid-structure interaction, and free surface flow: part II. J Comput Phys 344:683–723

18. Wang W, Wu J, Feng X (2018) A novel pressure-correction projection finite element method for incompressible natural convection problem with variable density. Numer Heat Transf Part A Appl 74(2):1001–1017

19. Theillard M, Gibou F, Saintillan D (2019) Sharp numerical simulation of incompressible two-phase flows. J Comput Phys 391:91–118

20. Wu J, Wei L, Feng X (2020) Novel fractional time-stepping algorithms for natural convection problems with variable density. Appl Numer Math 151:64–84

21. Liu C, Walkington NJ (2007) Convergence of numerical approximations of the incompressible Navier–Stokes equations with variable density and viscosity. SIAM J Numer Anal 45:1287–1304

22. Landet T, Mardal K-A, Mortensen M (2020) Slope limiting the velocity field in a discontinuous Galerkin divergence-free two-phase flow solver. Comput Fluids 196:104322

23. Cai W, Li B, Li Y (2021) Error analysis of a fully discrete finite element method for variable density incompressible flows in two dimensions. ESAIM Math Model Numer Anal 55:S103–S147

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

25. Boffi D, Brezzi F, Fortin M (2013) Mixed finite element methods and applications, vol 44. Springer, Heidelberg

26. John V (2016) Finite element methods for incompressible flow problems. Springer, Basel

27. Henshaw WD, Petersson NA (2003) A split-step scheme for the incompressible Navier–Stokes equations. In: Numerical simulations of incompressible flows. World Scientific, pp 108–125

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

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

30. Pacheco DRQ, Schussnig R, Fries T-P (2021) An efficient split-step framework for non-Newtonian incompressible flow problems with consistent pressure boundary conditions. Comput Methods Appl Mech Eng 382:113888

31. Behr M (2004) On the application of slip boundary condition on curved boundaries. Int J Numer Methods Fluids 45(1):43–51

32. Karyofylli V, Wendling L, Make M, Hosters N, Behr M (2019) Simplex space-time meshes in thermally coupled two-phase flow simulations of mold filling. Comput Fluids 192:104261

33. Zwicke F, Eusterholz S, Elgeti S (2017) Boundary-conforming free-surface flow computations: interface tracking for linear, higher-order and isogeometric finite elements. Comput Methods Appl Mech Eng 326:175–192

34. Karyofylli V, Frings M, Elgeti S, Behr M (2018) Simplex space-time meshes in two-phase flow simulations. Int J Numer Methods Fluids 86(3):218–230

35. Engelman MS, Sani RL, Gresho PM (1982) The implementation of normal and/or tangential boundary conditions in finite element codes for incompressible fluid flow. Int J Numer Methods Fluids 2(3):225–238

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

37. Pacheco DRQ, Schussnig R, Steinbach O, Fries TP (2021) A global residual-based stabilization for equal-order finite element approximations of incompressible flows. Int J Numer Methods Eng 122(8):2075–2094

38. Rang J (2008) Pressure corrected implicit θ-schemes for the incompressible Navier–Stokes equations. Appl Math Comput 201(1–2):747–761

39. Bertoglio C, Caiazzo A, Fernández MA (2013) Fractional-step schemes for the coupling of distributed and lumped models in hemodynamics. SIAM J Sci Comput 35(3):B551–B575

40. Jansen KE, Collis SS, Whiting C, Shaki F (1999) A better consistency for low-order stabilized finite element methods. Comput Methods Appl Mech Eng 174(1–2):153–170

41. Bochev P, Gunzburger M (2004) An absolutely stable pressure-Poisson stabilized finite element method for the Stokes equations. SIAM J Numer Anal 42(3):1189–1207

42. John V, Schmeyer E (2008) Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. Comput Methods Appl Mech Eng 198(3–4):475–494

43. Tryggvason G (1988) Numerical simulations of the Rayleigh–Taylor instability. J Comput Phys 75(2):253–282

44. Martin JC, Moyce WJ (1952) Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane. Philos Trans Roy Soc Lond Ser A Math Phys Sci 244(882):312-324

45. Touré MK, Fahsi A, Soulaïmani A (2016) Stabilised finite-element methods for solving the level set equation with mass conservation. Int J Comput Fluid Dyn 30(1):38–55

46. Hansbo P (1992) The characteristic streamline diffusion method for the time-dependent incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 99(2–3):171–186

47. El Haddad M, Belhamadia Y, Deteix J, Yakoubi D (2022) A projection scheme for phase change problems with convection. Comput Math Appl 108:109–122

48. Olsson E, Kreiss G (2005) A conservative level set method for two phase flow. J Comput Phys 210:225–246

49. Pacheco DRQ, Müller TS, Steinbach O, Brenn G (2021) On outflow boundary conditions in finite element simulations of non-Newtonian internal flows. Int J Comput Vis Sci Eng 1:4

50. Bašić M, Blagojević B, Peng C, Bašić J (2021) Lagrangian differencing dynamics for time-independent non-Newtonian materials. Materials 14(20):6210

51. Schussnig R, Pacheco DRQ, Fries T-P (2022) Efficient split-step schemes for fluid–structure interaction involving incompressible generalised Newtonian flows. Comput Struct 260:106718

52. Bašić J, Degiuli N, Blagojević B, Ban D (2022) Lagrangian differencing dynamics for incompressible flows. J Comput Phys. 462:111198

53. Pacheco DRQ (2021) Stable and stabilised finite element methods for incompressible flows of generalised Newtonian fluids. In: Computation in engineering and science (CES), vol 42. Verlag der Technischen Universität Graz, Graz