Journal of Computational Physics 497 (2024) 112608

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A high-order residual-based viscosity finite element method for incompressible variable density flow

Lukas Lundgren *, Murtazo Nazarov

Department of Information Technology, Uppsala University, Sweden

ARTICLE INFO

Keywords:
Incompressible variable density flow
Residual viscosity
Artificial viscosity
Artificial compressibility
Preconditioning

ABSTRACT

In this paper, we introduce a high-order accurate finite element method for incompressible variable density flow. The method uses high-order Taylor-Hood velocity-pressure elements in space and backward differentiation formula (BDF) time stepping in time. This way of discretization leads to two main issues: (*i*) a saddle point system that needs to be solved at each time step; (*ii*) a stability issue when the viscosity of the flow goes to zero or if the density profile has a discontinuity. We address the first issue by using Schur complement preconditioning and artificial compressibility approaches. We observed similar performance between these two approaches. To address the second issue, we introduce a modified artificial Guermond-Popov viscous flux where the viscosity coefficients are constructed using a newly developed residual-based shock-capturing method. Numerical validations confirm high-order accuracy for smooth problems and accurately resolved discontinuities for problems in 2D and 3D with varying density ratios.

1. Introduction

The simulation of incompressible variable density flow plays an important role in several areas of fluid dynamics. Its importance stems from its usefulness when simulating flow largely affected by density variations. This situation occurs in many places in nature, such as stratified flow in the ocean and the mixing of fluids with distinct phases, e.g., oil and water. The governing equations that we consider in this manuscript are the incompressible Navier-Stokes equations, augmented with an advection equation for density.

The aim of this manuscript is to develop a reliable high-order accurate finite element method (FEM) for variable density flow, based on Taylor-Hood velocity-pressure elements [17]. In the finite element literature concerning variable density flow, velocity and pressure are often uncoupled using a so-called projection method [21,22,48,60]. This approach can be justified due to computationally efficiency, but as mentioned by Guermond and Minev [18,19], these methods seemingly cannot exceed second-order accuracy in time without losing unconditional stability. A challenge emerges from incorporating pressure-related boundary conditions within the projection operator, necessitating temporal extrapolation. Because of this order barrier, we instead consider a fully coupled approach that directly discretizes the weak formulation of the Navier-Stokes equations, sometimes referred to as the monolithic approach [41], resulting in a saddle point system. This approach can be considered computationally expensive and Schur complement preconditioning is typically employed [3,33,53].

* Corresponding author.
E-mail addresses: lukas.lundgren@math.su.se (L. Lundgren), murtazo.nazarov@it.uu.se (M. Nazarov).

https://doi.org/10.1016/j.jcp.2023.112608
Received 17 October 2022; Received in revised form 6 September 2023; Accepted 25 October 2023

Available online 4 November 2023

One way to speed up this process is to utilize artificial compressibility techniques [11,55,52] to relax the divergence-free constraint and regularize the saddle point system. It is also common to use artificial compressibility to decouple velocity and pressure [18,19,37,13,61,45]. Artificial compressibility methods involve a penalty parameter that determines the strength of the imposed divergence-free constraint, leading to a trade-off between accuracy and computational effort. For explicit artificial compressibility methods [61,45,39], this limitation arises from time-step restrictions. On the other hand, implicit artificial compressibility methods [18,19,37] are affected by the resulting condition number of the linear systems. Motivated by artificial compressibility and guided by the ideas presented in [14,33], we propose an approximation for the Schur complement that is well-suited for both implicit artificial compressibility methods and the monolithic method, acting as a computational link between these two approaches. We show that the performance of the proposed preconditioning is similar between the implicit artificial compressibility method and the monolithic method for all values of the artificial compressibility penalty parameter.

Another difficulty concerns stabilization since it is well-known that Galerkin discretizations are unstable for convection-dominated problems. Historically, linear stabilization techniques such as Galerkin-Least-Squares (GLS) [28] and Streamline upwind Petrov-Galerkin (SUPG) [8] methods have been immensely popular in the finite element context. Notably, Johnson et al. [32] were able to prove convergence of the GLS method for scalar conservation laws by including a residual-based artificial viscosity term. Later, Nazarov [42] recognized that the residual viscosity term was the essential piece in the convergence proof. By discarding the least-square terms, it was shown that the stabilized finite element method was still convergent. Since the least-square terms were absent, the method could be easily extended to explicit time-stepping schemes [43,44] and could be made arbitrarily high-order accurate. It also paved the way for applying the method to other spatial discretizations, such as finite-difference methods [54], spectral element methods [36] and radial basis function methods [56]. In fact, the interest in nonlinear viscosity stabilization techniques that disregard linear stabilization techniques has been growing. This trend is attributed to their ease of implementation, reduced computational requirements, potential for arbitrary high-order accuracy and compatibility with a wide range of time-stepping methods. In particular, the so-called entropy viscosity method [24,23] has been successfully applied in the compressible context [44] and also applied to the constant density incompressible context [25,59] where its role as a so-called implicit large eddy simulation (LES) has been investigated. The authors in [24,23] used the Navier-Stokes viscous flux as a viscous regularization of the compressible Euler system, while [44] used the so-called Guermond-Popov viscous flux, see e.g., [20], to regularize the system. For an incompressible flow with variable density, we have to solve an additional density equation, so in this paper, we propose a modified Guermond-Popov viscous flux to regularize our system. A semi-discrete kinetic energy estimate shows that the modified Guermond-Popov flux is energy stable and the added mass diffusion does not affect the kinetic energy balance.

We construct the viscosity parameters in the Guermond-Popov flux proportional to the residual of the system, a method commonly referred to as the residual viscosity method (RV method for short). The RV method is closely related to the entropy viscosity method: for example, for scalar conservation laws, the residual is a special case of the entropy residual, since the solution to the equation can be chosen as an entropy functional. However, a similar proof of the convergence of the RV method for general non-linear scalar conservation laws, as in [42], is not known for the entropy viscosity method. In addition, it can be difficult to choose the entropy functional so that the corresponding entropy viscosity is robust.

In this paper, we introduce a novel way of using residuals to construct artificial viscosity coefficients. We use the residual of PDE to compute a discontinuity indicator $\alpha_h \in [0,1]$, which is close to zero when the solution is smooth and is close to one when the solution has discontinuities. Then this discontinuity indicator is multiplied by the traditional Lax-Friedrichs viscous flux. The resulting scheme is discontinuity capturing since it converts to the Lax-Friedrichs scheme close to discontinuities and is high-order in smooth regions since it converts to the Galerkin method there. The discontinuity indicator function α_h can oscillate since it is constructed using the residual. We propose two additional post-processing steps: first, a smoothing step is employed, where the oscillations on α_h are removed; second, an activation function is applied to α_h which suppresses the values of α_h in the smooth regions, while it amplifies the values of α_h close to discontinuities. Scaling artificial viscosity by the means of a discontinuity indicator can be traced back to the Jameson-Schmidt-Turkel scheme [30] and also more recent research such as [26, Sec 3.4] and [47,27].

This paper is organized as follows: In Section 2 the governing equations are introduced together with some finite element preliminaries. In Section 3 we describe the full numerical method including spatial- and temporal discretization, the nonlinear viscosity method and our preconditioning approach. In Section 4 we present numerical results and in Section 5 we give concluding remarks.

2. Preliminaries

In this section, we introduce the governing equations that model variable density flow. We then discretize the equations using continuous finite element approximations. The discretization is stabilized using nonlinear viscous fluxes.

2.1. Governing equations

We consider the incompressible Navier-Stokes equation with variable density in an open polyhedral domain $\Omega \subset \mathbb{R}^d$ and finite time interval $[0, T]$

$$
\begin{aligned}
\partial_t \rho + \boldsymbol{u}\cdot\nabla\rho &= 0, \\
\partial_t(\rho\boldsymbol{u}) + \boldsymbol{u}\cdot\nabla(\rho\boldsymbol{u}) + \nabla p - \nabla\cdot(\mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)) &= \boldsymbol{f}, \qquad (\boldsymbol{x},t)\in\Omega\times(0,T], \\
\nabla\cdot\boldsymbol{u} &= 0, \\
\boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{u}^0(\boldsymbol{x}), \\
\rho(\boldsymbol{x},0) &= \rho^0(\boldsymbol{x}), \qquad \boldsymbol{x}\in\Omega,
\end{aligned}
\tag{2.1}
$$

where the density $\rho(\boldsymbol{x},t) > 0$, the velocity field $\boldsymbol{u}(\boldsymbol{x},t)$ and the pressure $p(\boldsymbol{x},t)$ are the unknowns. $\boldsymbol{f}(\boldsymbol{x},t)$ represents an external force, $\mu > 0$ is the dynamic viscosity and $\rho^0(\boldsymbol{x})$, $\boldsymbol{u}^0(\boldsymbol{x})$ are initial conditions for the density and velocity. We assume that the governing equations are supplied with well-posed boundary conditions.

2.2. Finite element preliminaries

In this section, we introduce the finite element spaces and notations that are used in this work. We denote a shape regular computational mesh by $\mathcal{T}_h$ which is a triangulation of Ω into a finite number of disjoint elements K. The global shape functions $\{\varphi_i\}_{i=1}^{N_h}$ form a basis for the space $\mathcal{M}_h$, where N_h is the total number of nodes in $\mathcal{M}_h$. We define $\mathcal{I}(i)$ as the set of all nodal points contained within the support of φ_i. The finite element spaces we use for the density, velocity and pressure are respectively given by

$$
\begin{aligned}
\mathcal{M}_h &:= \{w : w\in C^0(\Omega); w|_K \in \mathbb{P}_k, \forall K\in\mathcal{T}_h\}, \\
\boldsymbol{\mathcal{V}}_h &:= [\mathcal{M}_h]^d, \\
\mathcal{Q}_h &:= \left\{ q : q\in C^0(\Omega); q|_K \in \mathbb{P}_{k^*}, \forall K\in\mathcal{T}_h, \int_\Omega q\,\mathrm{d}\boldsymbol{x} = 0 \right\},
\end{aligned}
$$

where $\mathbb{P}_k$ and $\mathbb{P}_{k^*}$ are the set of multivariate polynomials of total degree at most $k \geq 1$ and $k^* \geq 1$ defined over K. It is well-known that to satisfy the so-called inf-sup condition [17] we require Taylor-Hood finite elements, i.e., $k > k^*$. We often use the inner products

$$
\begin{aligned}
(v,w) &:= \sum_{K\in\mathcal{T}_h}\int_K vw\ \mathrm{d}\boldsymbol{x}, \quad (\boldsymbol{v},\boldsymbol{w}) := \sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{v}\cdot\boldsymbol{w}\ \mathrm{d}\boldsymbol{x}, \\
(\nabla\boldsymbol{v},\nabla\boldsymbol{w}) &:= \sum_{K\in\mathcal{T}_h}\int_K \nabla\boldsymbol{v} : \nabla\boldsymbol{w}\ \mathrm{d}\boldsymbol{x}, \quad (v,w)_{\partial\Omega} := \sum_{K\in\mathcal{T}_h}\int_{\partial K\setminus\partial\Omega} vw\ \mathrm{d}s,
\end{aligned}
$$

with associated norms $\|\cdot\|$. For $\boldsymbol{u},\boldsymbol{w},\boldsymbol{v}\in[H^1(\Omega)]^d$ and with $\boldsymbol{u}|_{\partial\Omega} = 0$, the following relation holds due to integration by parts

$$
(\boldsymbol{u}\cdot\nabla\boldsymbol{v},\boldsymbol{w}) = -((\nabla\cdot\boldsymbol{u})\boldsymbol{v},\boldsymbol{w}) - (\boldsymbol{u}\cdot\nabla\boldsymbol{w},\boldsymbol{v}). \tag{2.2}
$$

In this paper, we limit ourselves to meshes that are quasi-uniform. We compute the nodal based mesh size $h(\boldsymbol{x})\in\mathcal{M}_h$ using L_2-projection with additional smoothing:

$$
(h,w) + (|K|^{2/d}\nabla h,\nabla w) = (|K|^{1/d}/k, w), \quad \forall w\in\mathcal{M}_h,
$$

where $|K|$ is the volume of the cell K. To facilitate the analysis, we define the following finite element space

$$
\bar{\mathcal{M}}_s := \{\overline{w}(\boldsymbol{x}) : \overline{w}\in L_2(\Omega), \overline{w}|_K \in \mathbb{P}_s, \forall K\in\mathcal{T}_h\}, \tag{2.3}
$$

where we note that $\mathcal{M}\subset\bar{\mathcal{M}}_s$ provided that $k\leq s$.

3. Numerical method

In this section, we present a numerical method approximation of (2.1). In Section 3.1 we provide the spatial discretization using a continuous FEM. Later, in Section 3.2 time is discretized using a standard high-order BDF method. In Section 3.3 we introduce the nonlinear viscosity method which is used to construct the artificial viscosity coefficients used in the spatial discretization. In Section 3.4 the preconditioning strategy is explained. Lastly, in Section 3.5 the whole numerical method is summarized.

3.1. Spatial discretization

Since this manuscript aims to develop a high-order method, we choose a pure Galerkin discretization of the governing equations (2.1). The method is derived by taking (2.1) and testing the density equation with w, the momentum equations with $\boldsymbol{v}$ and the divergence-free condition with q. After performing integration by parts on some of the terms, the Galerkin method reads as follows: Find $(\rho_h,\boldsymbol{u}_h,p_h)\in\mathcal{M}_h\times\boldsymbol{\mathcal{V}}_h\times\mathcal{Q}_h$ such that

$$\begin{aligned}
\left(\partial_t \rho_h, w\right) + \left(\boldsymbol{u}_h \cdot \nabla \rho_h, w\right) &= 0, && \forall w \in \mathcal{M}_h, \\
\left(\partial_t \left(\rho_h \boldsymbol{u}_h\right), \boldsymbol{v}\right) + \left(\boldsymbol{u}_h \cdot \nabla (\rho_h \boldsymbol{u}_h), \boldsymbol{v}\right) - \left(p_h, \nabla \cdot \boldsymbol{v}\right) & \\
+ \mu \left(\nabla \boldsymbol{u}_h + \left(\nabla \boldsymbol{u}_h\right)^T, \nabla \boldsymbol{v}\right) &= \left(\boldsymbol{f}, \boldsymbol{v}\right), && \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}}_h, \\
\left(\nabla \cdot \boldsymbol{u}_h, q\right) &= 0, && \forall q \in \mathcal{Q}_h.
\end{aligned} \tag{3.1}$$

For the time being, only space has been discretized. Time discretization is provided later in Section 3.2. The boundary conditions that we impose in this manuscript fulfill the condition $\boldsymbol{u} \cdot \boldsymbol{n}|_{\partial\Omega} = 0$ which we impose strongly. This means that many of the boundary terms inside (3.1) have been omitted.

3.1.1. Stabilized scheme

Since the Galerkin scheme is unstable for hyperbolic problems (such as the density update inside (2.1)), numerical stabilization of the scheme is necessary. For this purpose, we propose adding a modified Guermond-Popov viscous flux [20]:

$$\nabla \cdot \mathbf{F}(\rho_h, \boldsymbol{u}_h) := \nabla \cdot \begin{bmatrix} \kappa_h \nabla \rho_h \\ \kappa_h \left(\rho_h \left(\nabla \boldsymbol{u}_h + (\nabla \boldsymbol{u}_h)^T\right) + \nabla \rho_h \otimes \boldsymbol{u}_h\right) \\ 0 \end{bmatrix}, \tag{3.2}$$

to the Galerkin scheme. For additional divergence cleaning we also add grad-div stabilization [46,10] $(\gamma_h \nabla \cdot \boldsymbol{u}_h, \nabla \cdot \boldsymbol{v})$ to the scheme. Here $\kappa_h \geq 0$ is artificial kinematic viscosity and γ_h is a penalty parameter. Both κ_h and γ_h are mesh-dependent and vanish as $h \to 0$. Note, that the Guermond-Popov flux is aimed to stabilize the convection field, whereas the grad-div stabilization provides additional divergence cleaning. We later show that the Guermond-Popov flux leads to kinetic energy stability. The term $\nabla \rho_h \otimes \boldsymbol{u}_h$ is key to ensure that the momentum equations are adequately compensated due to the added mass diffusion. For further details and other alternative viscous regularizations, we refer the readers to [38]. Adding the Guermond-Popov flux, grad-div stabilization to the Galerkin scheme (3.1) yields the following stabilized scheme:

$$\begin{aligned}
\left(\partial_t \rho_h, w\right) + \left(\boldsymbol{u}_h \cdot \nabla \rho_h, w\right) + \left(\kappa_h \nabla \rho_h, \nabla w\right) &= 0, && \forall w \in \mathcal{M}_h, \\
\left(\partial_t \left(\rho_h \boldsymbol{u}_h\right), \boldsymbol{v}\right) + \left(\boldsymbol{u}_h \cdot \nabla (\rho_h \boldsymbol{u}_h), \boldsymbol{v}\right) + \frac{1}{2}(\rho_h \boldsymbol{u}_h (\nabla \cdot \boldsymbol{u}_h), \boldsymbol{v}) - \left(p_h, \nabla \cdot \boldsymbol{v}\right) & \\
+ \mu (\nabla \boldsymbol{u}_h + \left(\nabla \boldsymbol{u}_h\right)^T, \nabla \boldsymbol{v}) + (\gamma_h \nabla \cdot \boldsymbol{u}_h, \nabla \cdot \boldsymbol{v}) & \\
+ (\kappa_h (\rho_h (\nabla \boldsymbol{u}_h + \left(\nabla \boldsymbol{u}_h\right)^T) + \nabla \rho_h \otimes \boldsymbol{u}_h)), \nabla \boldsymbol{v}) &= \left(\boldsymbol{f}, \boldsymbol{v}\right), && \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}}_h, \\
\left(\nabla \cdot \boldsymbol{u}_h, q\right) &= 0, && \forall q \in \mathcal{Q}_h,
\end{aligned} \tag{3.3}$$

where we also add the so-called skew-symmetric term $\frac{1}{2}(\rho_h \boldsymbol{u}_h (\nabla \cdot \boldsymbol{u}_h), \boldsymbol{v})$ which will be useful for the stability analysis. Choosing the artificial viscosity coefficient κ_h sufficiently large will lead to a stable method. We will go more into detail on how to construct κ_h and γ_h in Section 3.3 to yield a stable scheme that is also high-order accurate.

Remark 3.1. Note that it is also possible to write the momentum update as a time evolution for $\rho \partial_t \boldsymbol{u}$ or $\sqrt{\rho} \partial_t (\sqrt{\rho} \boldsymbol{u})$ instead. The reason why we chose to write the governing equations in momentum form, is that they are easily combined with the Guermond-Popov flux.

3.1.2. Semi-discrete stability estimate

In this section, we provide a semi-discrete stability estimate for the stabilized scheme (3.3). To simplify the analysis we assume that the solution has compact support, i.e., $\boldsymbol{u} = 0$ at $\partial\Omega$, and will lead to the boundary term resulting from integration by parts disappearing. The novelty of the analysis presented here is showing that the Guermond-Popov flux (3.2) fits well in the incompressible flow context. More specifically, we show that the added mass diffusion $(\kappa_h \nabla \rho_h, \nabla w)$ does not affect the discrete kinetic energy balance. The main result is presented below:

Proposition 3.1. *If $\boldsymbol{f} = 0$, the scheme satisfies the following stability estimate*

$$\frac{1}{2}\partial_t \|\sqrt{\rho_h}\boldsymbol{u}\|^2 + \mu \|\nabla \boldsymbol{u}_h\|^2 + \mu \|\nabla \cdot \boldsymbol{u}_h\|^2 + \|\sqrt{\gamma_h}\, \nabla \cdot \boldsymbol{u}_h\|^2 + \frac{1}{2}\|\sqrt{\kappa_h \rho_h}(\nabla \boldsymbol{u}_h + (\nabla \boldsymbol{u}_h)^T)\|^2 = 0. \tag{3.4}$$

Proof. We take the stabilized scheme (3.3) and set $q = p_h$ and $\boldsymbol{v} = \boldsymbol{u}_h$. Using that $\nabla \cdot (\nabla \boldsymbol{u}_h + (\nabla \boldsymbol{u}_h)^T) = \nabla \cdot \nabla \boldsymbol{u}_h + \nabla \nabla \cdot \boldsymbol{u}_h$ yields

$$\begin{aligned}
\left(\partial_t \left(\rho_h \boldsymbol{u}_h\right), \boldsymbol{u}_h\right) + \left(\boldsymbol{u}_h \cdot \nabla (\rho_h \boldsymbol{u}_h), \boldsymbol{u}_h\right) + \frac{1}{2}(\rho_h \boldsymbol{u}_h \left(\nabla \cdot \boldsymbol{u}_h\right), \boldsymbol{u}_h) + \mu \|\nabla \boldsymbol{u}_h\|^2 + \mu \|\nabla \cdot \boldsymbol{u}_h\|^2 & \\
+ \|\sqrt{\gamma_h}\, \nabla \cdot \boldsymbol{u}_h\|^2 + \left(\kappa_h \left(\rho_h \left(\nabla \boldsymbol{u}_h + \left(\nabla \boldsymbol{u}_h\right)^T\right) + \nabla \rho_h \otimes \boldsymbol{u}_h\right), \nabla \boldsymbol{u}_h\right) &= 0.
\end{aligned} \tag{3.5}$$

Using that the contraction between a symmetric and anti-symmetric matrix with zero diagonal is zero [34, Ch 11.2.1], i.e., $(A + A^T) : (A - A^T) = 0$ where A is a square matrix, one can show that

$$\begin{aligned}\left(\kappa_h\rho_h\left(\nabla\boldsymbol{u}_h+\left(\nabla\boldsymbol{u}_h\right)^T\right),\nabla\boldsymbol{u}_h\right)&=\frac{1}{2}\left(\kappa_h\rho_h\left(\nabla\boldsymbol{u}_h+\left(\nabla\boldsymbol{u}_h\right)^T\right),\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T\right)\\&\quad+\frac{1}{2}\left(\kappa_h\rho_h\left(\nabla\boldsymbol{u}_h+\left(\nabla\boldsymbol{u}_h\right)^T\right),\nabla\boldsymbol{u}_h-(\nabla\boldsymbol{u}_h)^T\right)\\&=\frac{1}{2}\left(\kappa_h\rho_h\left(\nabla\boldsymbol{u}_h+\left(\nabla\boldsymbol{u}_h\right)^T\right),\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T\right).\end{aligned}\tag{3.6}$$

Inserting (3.6) into (3.5) gives

$$\begin{aligned}&\left(\partial_t\left(\rho_h\boldsymbol{u}_h\right),\boldsymbol{u}_h\right)+(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)+\frac{1}{2}(\rho_h\boldsymbol{u}_h\left(\nabla\cdot\boldsymbol{u}_h\right),\boldsymbol{u}_h)+(\kappa_h\nabla\rho_h\otimes\boldsymbol{u}_h,\nabla\boldsymbol{u}_h)\\&\quad+\mu\|\nabla\boldsymbol{u}_h\|^2+\mu\|\nabla\cdot\boldsymbol{u}_h\|^2+\|\sqrt{\gamma_h}\,\nabla\cdot\boldsymbol{u}_h\|^2+\frac{1}{2}\|\sqrt{\kappa_h\rho_h}(\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T)\|^2=0.\end{aligned}\tag{3.7}$$

Using the integration by parts relation (2.2), one can show that

$$\begin{aligned}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)+\frac{1}{2}(\rho_h\boldsymbol{u}_h(\nabla\cdot\boldsymbol{u}_h),\boldsymbol{u}_h)&=\frac{1}{2}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)+\frac{1}{2}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)\\&\quad+\frac{1}{2}(\rho_h\boldsymbol{u}_h(\nabla\cdot\boldsymbol{u}_h),\boldsymbol{u}_h)\\&=-\frac{1}{2}(\rho_h\boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h,\boldsymbol{u}_h)-\frac{1}{2}(\rho_h\boldsymbol{u}_h(\nabla\cdot\boldsymbol{u}_h),\boldsymbol{u}_h)+\frac{1}{2}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)+\frac{1}{2}(\rho_h\boldsymbol{u}_h(\nabla\cdot\boldsymbol{u}_h),\boldsymbol{u}_h)\\&=-\frac{1}{2}(\rho_h\boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h,\boldsymbol{u}_h)+\frac{1}{2}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h).\end{aligned}$$

Expanding $\nabla(\rho_h\boldsymbol{u}_h)$ then shows that

$$\begin{aligned}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)+\frac{1}{2}(\rho_h\boldsymbol{u}_h(\nabla\cdot\boldsymbol{u}_h),\boldsymbol{u}_h)&=-\frac{1}{2}(\rho_h\boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h,\boldsymbol{u}_h)+\frac{1}{2}(\boldsymbol{u}_h\cdot\nabla(\rho_h\boldsymbol{u}_h),\boldsymbol{u}_h)\\&=-\frac{1}{2}(\rho_h\boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h,\boldsymbol{u}_h)+\frac{1}{2}\left(\boldsymbol{u}_h\cdot\nabla\rho_h,|\boldsymbol{u}_h|^2\right)+\frac{1}{2}(\rho_h\boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h,\boldsymbol{u}_h)\\&=\frac{1}{2}\left(\boldsymbol{u}_h\cdot\nabla\rho_h,|\boldsymbol{u}_h|^2\right).\end{aligned}\tag{3.8}$$

Inserting (3.8) into (3.7) gives

$$\begin{aligned}&\left(\partial_t\left(\rho_h\boldsymbol{u}_h\right),\boldsymbol{u}_h\right)+\frac{1}{2}\left(\boldsymbol{u}_h\cdot\nabla\rho_h,|\boldsymbol{u}_h|^2\right)+(\kappa_h\nabla\rho_h\otimes\boldsymbol{u}_h,\nabla\boldsymbol{u}_h)\\&+\mu\|\nabla\boldsymbol{u}_h\|^2+\mu\|\nabla\cdot\boldsymbol{u}_h\|^2+\|\sqrt{\gamma_h}\,\nabla\cdot\boldsymbol{u}_h\|^2+\frac{1}{2}\left\|\sqrt{\kappa_h\rho_h}(\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T)\right\|^2=0.\end{aligned}\tag{3.9}$$

Simplifying the term $(\kappa_h\nabla\rho_h\otimes\boldsymbol{u}_h,\nabla\boldsymbol{u}_h)$ inside (3.9) and then performing integration by parts on it yields:

$$\begin{aligned}&\left(\partial_t\left(\rho_h\boldsymbol{u}_h\right),\boldsymbol{u}_h\right)+\frac{1}{2}\left(\boldsymbol{u}_h\cdot\nabla\rho_h,|\boldsymbol{u}_h|^2\right)+\frac{1}{2}(\kappa_h\nabla\rho_h,\nabla(|\boldsymbol{u}_h|^2))\\&+\mu\|\nabla\boldsymbol{u}_h\|^2+\mu\|\nabla\cdot\boldsymbol{u}_h\|^2+\|\sqrt{\gamma_h}\,\nabla\cdot\boldsymbol{u}_h\|^2+\frac{1}{2}\left\|\sqrt{\kappa_h\rho_h}(\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T)\right\|^2=0\\&\left(\partial_t\left(\rho_h\boldsymbol{u}_h\right),\boldsymbol{u}_h\right)+\frac{1}{2}\left(\boldsymbol{u}_h\cdot\nabla\rho_h,|\boldsymbol{u}_h|^2\right)-\frac{1}{2}(\nabla\cdot(\kappa_h\nabla\rho_h),|\boldsymbol{u}_h|^2)\\&+\mu\|\nabla\boldsymbol{u}_h\|^2+\mu\|\nabla\cdot\boldsymbol{u}_h\|^2+\|\sqrt{\gamma_h}\,\nabla\cdot\boldsymbol{u}_h\|^2+\frac{1}{2}\left\|\sqrt{\kappa_h\rho_h}(\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T)\right\|^2=0.\end{aligned}\tag{3.10}$$

We define the following L_2-projection. Find $\overline{|\boldsymbol{u}_h|^2}\in\mathcal{M}$ such that

$$\left(\overline{|\boldsymbol{u}_h|^2},\overline{w}\right)=\left(|\boldsymbol{u}_h|^2,\overline{w}\right)\quad\forall\overline{w}\in\bar{\mathcal{M}}_{2k-1},\tag{3.11}$$

where $\bar{\mathcal{M}}_{2k-1}$ is defined by (2.3). Taking the density update in (3.1.1) and setting $w=-\frac{1}{2}\overline{|\boldsymbol{u}_h|^2}$ and performing integration by parts on the last term yields

$$\begin{aligned}&-\frac{1}{2}\left(\left(\partial_t\rho_h,\overline{|\boldsymbol{u}_h|^2}\right)+\left(\boldsymbol{u}_h\cdot\nabla\rho_h,\overline{|\boldsymbol{u}_h|^2}\right)+\left(\kappa_h\nabla\rho_h,\nabla\left(\overline{|\boldsymbol{u}_h|^2}\right)\right)\right)=0\\&-\frac{1}{2}\left(\left(\partial_t\rho_h,\overline{|\boldsymbol{u}_h|^2}\right)+\left(\boldsymbol{u}_h\cdot\nabla\rho_h,\overline{|\boldsymbol{u}_h|^2}\right)-\left(\nabla\cdot(\kappa_h\nabla\rho_h),\overline{|\boldsymbol{u}_h|^2}\right)\right)=0.\end{aligned}\tag{3.12}$$

Adding (3.10) and (3.12) yields

$$\begin{aligned}&\left(\partial_t\left(\rho_h\boldsymbol{u}_h\right),\boldsymbol{u}_h\right)-\frac{1}{2}\left(\partial_t\rho_h,\overline{|\boldsymbol{u}_h|^2}\right)+\mu\|\nabla\boldsymbol{u}_h\|^2+\mu\|\nabla\cdot\boldsymbol{u}_h\|^2+\|\sqrt{\gamma_h}\,\nabla\cdot\boldsymbol{u}_h\|^2\\&+\frac{1}{2}\left\|\sqrt{\kappa_h\rho_h}\left(\nabla\boldsymbol{u}_h+(\nabla\boldsymbol{u}_h)^T\right)\right\|^2+\frac{1}{2}\left(\boldsymbol{u}_h\cdot\nabla\rho_h,|\boldsymbol{u}_h|^2-\overline{|\boldsymbol{u}_h|^2}\right)-\frac{1}{2}\left(\nabla\cdot(\kappa_h\nabla\rho_h),-|\boldsymbol{u}_h|^2+\overline{|\boldsymbol{u}_h|^2}\right)=0\end{aligned}\tag{3.13}$$

Since $\nabla\cdot(\kappa_h\nabla\rho_h)\in\bar{\mathcal{M}}_{2k-2}\subset\bar{\mathcal{M}}_{2k-1}$ and $\boldsymbol{u}_h\cdot\nabla\rho_h\in\bar{\mathcal{M}}_{2k-1}$ and $\partial_t\rho\in\mathcal{M}\subset\bar{\mathcal{M}}_{2k-1}$, we can use the L_2 projection (3.11) on (3.13) to finally yield

$$\left(\partial_t\left(\rho_h \boldsymbol{u}_h\right), \boldsymbol{u}_h\right) - \frac{1}{2}\left(\partial_t \rho_h, |\boldsymbol{u}_h|^2\right) + \mu\|\nabla \boldsymbol{u}_h\|^2 + \mu\|\nabla\cdot \boldsymbol{u}_h\|^2 + \|\sqrt{\gamma_h}\,\nabla\cdot \boldsymbol{u}_h\|^2 + \frac{1}{2}\left\|\sqrt{\kappa_h \rho_h}\left(\nabla \boldsymbol{u}_h + (\nabla \boldsymbol{u}_h)^T\right)\right\|^2 = 0,$$

$$\frac{1}{2}\partial_t\|\sqrt{\rho_h}\boldsymbol{u}\|^2 + \mu\|\nabla \boldsymbol{u}_h\|^2 + \mu\|\nabla\cdot \boldsymbol{u}_h\|^2 + \|\sqrt{\gamma_h}\,\nabla\cdot \boldsymbol{u}_h\|^2 + \frac{1}{2}\left\|\sqrt{\kappa_h \rho_h}\left(\nabla \boldsymbol{u}_h + (\nabla \boldsymbol{u}_h)^T\right)\right\|^2 = 0,$$

which concludes the proof. □

3.2. High-order time stepping

The time derivative in the variational formulation (3.3) is discretized using high-order backward differentiation formulas (BDFs). This section describes the variable time step BDF method we use. The variable time step BDF method [2] is derived by replacing $\partial_t \rho_h$, $\partial_t(\rho_h \boldsymbol{u}_h)$ with appropriate discrete approximations. The full method is as follows: Let $(\rho_h^n, \boldsymbol{u}_h^n, p_h^n) \in (\mathcal{M}_h, \boldsymbol{\mathcal{V}}_h, \mathcal{Q}_h)$ be solutions at time t_n. Let $\Delta t_{n+j} := t_{n+j+1} - t_{n+j}$ denote the time step and $\omega_i := \Delta t_i / \Delta t_{i-1}$ denote the time step ratio. Given artificial viscosity coefficients κ_h^n and γ_h^n and solutions from previous time steps, $\rho_h^{n+j}, \boldsymbol{u}_h^{n+j}$, first find $\rho_h^{n+1} \in \mathcal{M}_h$ such that

$$\left(d_t\left(\rho_h^{n+1}\right), w\right) + \left(\boldsymbol{u}_h^*\cdot\nabla \rho_h^{n+1}, w\right) + \left(\kappa_h^n \nabla \rho_h^{n+1}, \nabla w\right) = 0, \quad \forall w \in \mathcal{M}_h, \tag{3.14}$$

where $\boldsymbol{u}_h^*$ is a linearization of $\boldsymbol{u}_h$, then find $\boldsymbol{u}_h^{n+1} \in \boldsymbol{\mathcal{V}}_h$ and $p_h^{n+1} \in \mathcal{Q}_h$ such that

$$\begin{aligned}
&\left(d_t\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right), \boldsymbol{v}\right) + \left(\boldsymbol{u}_h^*\cdot\nabla\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right), \boldsymbol{v}\right) + \frac{1}{2}\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\left(\nabla\cdot \boldsymbol{u}_h^*\right), \boldsymbol{v}\right) \\
&\quad - \left(p_h^{n+1}, \nabla\cdot \boldsymbol{v}\right) + \mu\left(\nabla \boldsymbol{u}_h^{n+1} + \left(\nabla \boldsymbol{u}_h^{n+1}\right)^T, \nabla \boldsymbol{v}\right) + \left(\gamma_h^n \nabla\cdot \boldsymbol{u}_h^{n+1}, \nabla\cdot \boldsymbol{v}\right) \\
&\quad + \left(\kappa_h^n\left(\rho_h^{n+1}\left(\nabla \boldsymbol{u}_h^{n+1} + \left(\nabla \boldsymbol{u}_h^{n+1}\right)^T\right) + \nabla \rho_h^{n+1} \otimes \boldsymbol{u}_h^{n+1}\right), \nabla \boldsymbol{v}\right) = \left(\boldsymbol{f}^{n+1}, \boldsymbol{v}\right), \quad \forall \boldsymbol{v} \in \boldsymbol{\mathcal{V}}_h, \\
&\qquad \left(\nabla\cdot \boldsymbol{u}_h^{n+1}, q\right) = 0, \qquad \forall q \in \mathcal{Q}_h,
\end{aligned} \tag{3.15}$$

where $d_t\left(\rho_h^{n+1}\right)$, $d_t\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right)$ are discrete approximations of $\partial_t \rho_h$, $\partial_t(\rho_h \boldsymbol{u}_h)$, respectively. The finite-difference approximations of these for variable time step BDF methods are presented up to order 4 in the Appendix. The choice of $\boldsymbol{u}_h^*$ with respect to the BDF scheme is given in the Appendix. The time step ratio ω_i is critical to ensure that the underlying BDF scheme is zero-stable and convergent. For example, [58] showed that $\omega_i \leq 1.101$ for BDF4 and $\omega_i \leq 1.501$ for BDF3. Denote by $[s_{\min}, s_{\max}]$ the interval in which ω_i lives and ensures the zero-stability of the BDF scheme. Then, for given time step Δt_{n-1}, we calculate the next time step Δt_n using the following adaptive algorithm:

Algorithm 1 A simplified version of the time step control algorithm from [19, Sec 5.4] used to compute the next time step Δt_n given user-defined parameters CFL, $s_{\max}, s_{\min}$.

```
/* Compute time step increment based on CFL condition                    */
s_cfl = CFL min_{x∈Ω} h(x)/(‖u_h^n‖_{l^2} Δt_{n-1})
/* Make sure the next time step Δt_n is bounded by s_max Δt_{n-1}        */
s = min(s_cfl, s_max)
Δt_n = s Δt_{n-1}
if s < s_min then
  | Repeat previous time step with Δt_n instead
end
return Time step Δt_n and flag whether to repeat the time step or not
```

Remark 3.2 *(Initialization).* A common approach to initialize high-order BDF methods is to use lower-order BDF methods using small time steps. We follow this approach and take

$$\Delta t_{\text{initial}} = c_{\text{init}} \Delta t_0,$$

as initial time step, where Δt_0 is the otherwise usual initial time step given by the CFL-condition and $c_{\text{init}} \leq 1$. If c_{init} is sufficiently small, high-order accuracy is still obtained even if the method is initialized using a low-order method. If $\boldsymbol{u}_h$ is zero initially we set $\Delta t_0 = \text{CFL}\min_{\boldsymbol{x}\in\Omega} h(\boldsymbol{x}) * (1\ [\text{s/m}])$, where the last factor is included to obtain the correct unit. Another approach is to use the initialization algorithm proposed by Guermond and Minev [19, Sec 3.3].

Remark 3.3. The particular linearization used in (3.14) and (3.15) ensures that the semi-discrete estimate (3.4) still holds. To the authors' knowledge, a higher than second order fully discrete estimate [22] for variable density flow is not known for BDF methods.

3.3. Nonlinear viscosity method

In this section we present how to construct the artificial viscosity coefficients κ_h^n and γ_h^n which are used in the fully discrete approximation, i.e., (3.14) and (3.15). Briefly stated, the stabilization is performed by using a sufficient amount of artificial viscosity,

determined by κ_h^n and γ_h^n. In Sections 3.3.1 and 3.3.2 we describe how we compute κ_h^n using a residual-based viscosity approach which is one of the contributions of this work. In Section 3.3.3 we describe how we compute the grad-div stabilization penalty parameter γ_h^n in a way that is suitable for the proposed discretization. Lastly, in Section 3.3.4 we summarize the procedure.

3.3.1. A new residual viscosity method

The nonlinear viscosity κ_h^n is constructed based on the PDE residual and this technique is commonly referred to as the RV method. This technique has been successfully applied to hyperbolic problems and compressible flow [43,54,36,40,12]. The effect of the residual viscosity in this context is twofold: it provides sufficient stabilization while also acting as a large eddy simulation. Since the residual is small in smooth, resolved regions of the domain, high-order accuracy is still maintained. For a comparison between the RV method and the well-known Lilly-Smagorinsky model, we refer to Marras et al. [40].

Our approach in this work is different than the one in the references. The key idea is to use the residual to construct discontinuity indicator $\alpha_h^n \in \mathcal{M}_h \cap [0,1]$ which is close to 1 in non-smooth areas of the solution and close to zero in smooth parts of the solution. This indicator is then used to compute κ_h^n, which is chosen as first-order viscosity scaled with the indicator α_h^n

$$\kappa_h^n = \alpha_h^n h C_{\max} \|\boldsymbol{u}_h^n\|_{\ell_2}, \tag{3.16}$$

where $C_{\max}$ is a user-defined parameter determining the amount of first-order viscosity used. We have three new contributions: i) An improved localized normalization of the RV method. ii) Additional post-processing steps for increased accuracy in smooth regions and additional robustness for strong discontinuities. iii) An extension of the RV method to incompressible variable density flow. Below, we describe how to construct α_h^n.

We first define the residuals $R_{\rho_h}^n, R_{\boldsymbol{u}_h}^n, R_{\mathrm{div}\boldsymbol{u}_h}^n \in \mathcal{M}_h$ on each node i as

$$\begin{aligned}
R_{\rho_h,i}^n &:= \left| d_t(\rho_h^n) + \boldsymbol{u}_h^n \cdot \nabla \rho_h^n \right|_i, \\
R_{\boldsymbol{u}_h,i}^n &:= \left\| d_t(\rho_h^n \boldsymbol{u}_h^n) + \boldsymbol{u}_h^n \cdot \nabla(\rho_h^n \boldsymbol{u}_h^n) + \nabla p_h^n - \mu \nabla \cdot \left(\nabla \boldsymbol{u}_h^n + (\nabla \boldsymbol{u}_h^n)^T \right) - \boldsymbol{f}^n \right\|_{\ell_2,i}, \\
R_{\mathrm{div}\boldsymbol{u}_h,i}^n &:= \left| \nabla \cdot \boldsymbol{u}_h^n \right|_i,
\end{aligned} \tag{3.17}$$

where $d_t(\rho_h^n), d_t(\rho_h^n \boldsymbol{u}_h^n)$ are approximations of $\partial_t \rho_h, \partial_t(\rho_h \boldsymbol{u}_h)$ at time t_n using a BDF formula. We also define localized normalization functions on each node i as

$$\begin{aligned}
n_{\mathrm{loc},\rho_h,i}^n &:= |d_t(\rho_h^n)|_i + \|\boldsymbol{u}_h^n\|_{\ell_2,i} \|\nabla \rho_h^n\|_{\ell_2,i}, \\
n_{\mathrm{loc},\boldsymbol{u}_h,i}^n &:= \|d_t(\rho_h^n \boldsymbol{u}_h^n)\|_{\ell_2,i} + \|\boldsymbol{u}_h^n\|_{\ell_2,i} \|\nabla(\rho_h^n \boldsymbol{u}_h^n)\|_{\ell_2,i} \\
&\quad + \|\nabla p_h^n\|_{\ell_2,i} + \|\mu \nabla \cdot \left(\nabla \boldsymbol{u}_h^n + (\nabla \boldsymbol{u}_h^n)^T \right)\|_{\ell_2,i} + \|\boldsymbol{f}^n\|_{\ell_2,i}, \\
n_{\mathrm{loc},\mathrm{div}\boldsymbol{u}_h,i}^n &:= \left\| \nabla \boldsymbol{u}_h^n \right\|_{\ell_2,i},
\end{aligned}$$

where $\|\nabla \boldsymbol{u}_h^n\|_{\ell_2} := \sqrt{\sum_{j=1}^d \sum_{i=1}^d |\left(\nabla \boldsymbol{u}_h^n\right)_{i,j}|^2}$.

Let us start our discussion with the density equation. Computing $R_{\rho_h,i}^n / n_{\mathrm{loc},\rho_h,i}^n$ for each node i will yield a number between 0 and 1 which will indicate if ρ_h^n is smooth or non-smooth. If ρ_h^n is smooth $R_{\rho_h}^n$ will be small and if ρ_h^n is non-smooth both $R_{\rho_h}^n$ and $n_{\mathrm{loc},\rho_h}^n$ will be $\approx 1/h$ and thus $R_{\rho_h,i}^n / n_{\mathrm{loc},\rho_h,i}^n$ will be close to 1. Due to its construction, $R_{\rho_h,i}^n / n_{\mathrm{loc},\rho_h,i}^n$ will not exceed 1. One pitfall occurs when ρ_h^n is flat (or close to flat) since then, both $R_{\rho_h}^n$ and $n_{\mathrm{loc},\rho_h}^n$ will be small, and $R_{\rho_h,i}^n / n_{\mathrm{loc},\rho_h,i}^n$ will be close to 1. To cure this, we introduce the following improved normalization values for each node i:

$$\begin{aligned}
n_{\rho_h,i}^n &:= \begin{cases} n_{\mathrm{loc},\rho_h,i}^n, & \text{if} \quad h_i \|\nabla \rho_h^n\|_{\ell_2,i} > C_{\mathrm{flat}}, \\ \max\left(h_i^{-1}\ n_{\mathrm{glob}}(\rho_h^n \|\boldsymbol{u}_h^n\|_{\ell_2}),\ n_{\mathrm{loc},\rho_h,i}^n \right), & \text{otherwise}, \end{cases} \\
n_{\boldsymbol{u}_h,i}^n &:= \begin{cases} n_{\mathrm{loc},\boldsymbol{u}_h,i}^n, & \text{if} \quad h_i \|\nabla(\rho_h \boldsymbol{u}_h)\|_{\ell_2,i} > C_{\mathrm{flat}}, \\ \max\left(h_i^{-1}\ n_{\mathrm{glob}}(\rho_h^n \|\boldsymbol{u}_h^n\|_{\ell_2}^2),\ n_{\mathrm{loc},\boldsymbol{u}_h,i}^n \right), & \text{otherwise}, \end{cases} \\
n_{\mathrm{div}\boldsymbol{u}_h,i}^n &:= \begin{cases} n_{\mathrm{loc},\mathrm{div}\boldsymbol{u}_h,i}^n & \text{if} \quad h_i \left\| \nabla \boldsymbol{u}_h^n \right\|_{\ell_2,i} > C_{\mathrm{flat}}, \\ \max(h_i^{-1}\ n_{\mathrm{glob}}(\|\boldsymbol{u}_h^n\|_{\ell_2}),\ n_{\mathrm{loc},\mathrm{div}\boldsymbol{u}_h,i}^n), & \text{otherwise}, \end{cases}
\end{aligned} \tag{3.18}$$

where we define $n_{\mathrm{glob}}(w)$ to be a function which takes the global jump of $w \in \mathcal{M}_h$ on Ω

$$n_{\mathrm{glob}}(w) := \frac{\left|\max_\Omega w - \min_\Omega w\right|^2}{|\max_\Omega w - \min_\Omega w| + \varepsilon \|w\|_{L^\infty(\Omega \times (0,t_n))}},$$

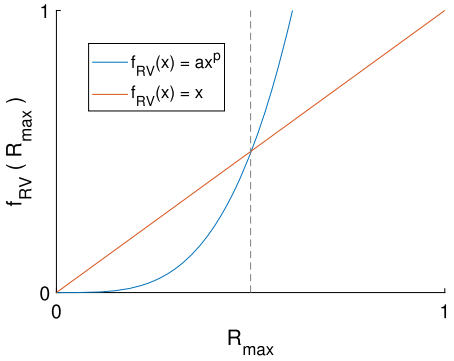

Fig. 1. Activation function f_{RV} used to suppress $R^n_{\max}$ in smooth regions and increase $R^n_{\max}$ in non-smooth regions. The solution is considered smooth to the left of the dashed grey line and non-smooth to the right of it.

with a user-defined parameter $\varepsilon \ll 1$ to avoid division by zero. In (3.18), a user-defined parameter $C_{\mathrm{flat}} \in (0.01, 1)$ is used to indicate if the solution is flat with respect to the mesh-size h. Finally, we are ready to define the global normalized residual $R^n_{\max,h} \in \mathcal{M}_h$ that is later used to construct the discontinuity indicator as

$$R^n_{\max,h,i} := \max\left(\frac{R^n_{\rho_h,i}}{n^n_{\rho_h,i}}, \frac{R^n_{\boldsymbol{u}_h,i}}{n^n_{\boldsymbol{u}_h,i}}, \frac{R^n_{\mathrm{div}\boldsymbol{u}_h,i}}{n^n_{\mathrm{div}\boldsymbol{u}_h,i}}\right). \tag{3.19}$$

Remark 3.4. Traditionally, $d_t(\rho^n_h)$ and $d_t(\rho^n_h \boldsymbol{u}^n_h)$ are computed using a sufficiently accurate BDF-formula, but other options exist, such as using a carefully constructed spatial approximation [54,36]. In this work we use BDF3 (A.1).

Remark 3.5. The standard RV method [56,36,40,12,44,25,59] for the advection equation, as expressed as $\frac{R^n_{\rho_h,i}}{n^n_{\rho_h,i}} \in [0,1]$ for each node i, is given by

$$\frac{R^n_{\rho_h,i}}{n^n_{\rho_h,i}} = \min\left(1, 2h \frac{\left|d_t\rho^n_h + \boldsymbol{u}^n_h \cdot \nabla\rho^n_h\right|_i}{\left|\boldsymbol{u}^n_h|_{\ell_2,i} \left\|\rho^n_h - \bar{\rho}\right\|_{L^\infty(\Omega)} + \epsilon}\right), \tag{3.20}$$

where $\bar{\rho} = |\Omega|^{-1} \int_\Omega \rho \, \mathrm{d}\boldsymbol{x}$ and ϵ is a small parameter to avoid division by zero. The downsides of (3.20) as a discontinuity indicator are:

- $\left\|\rho^n_h - \bar{\rho}\right\|_{L^\infty(\Omega)}$ is a global normalization used to ensure that α^n_h has the correct unit. If, for example, the domain is stretched, this normalization causes the method to artificially behave differently.
- The wave-speed $|\boldsymbol{u}^n_h|$ does not provide any information to $\frac{R^n_{\rho_h,i}}{n^n_{\rho_h,i}}$ about numerical error or smoothness for ρ^n_h.

The residual indicator proposed in this work (3.19) avoids these two issues.

3.3.2. *Further post-processing and nonlinear transformations of the discontinuity indicator*

In this section, we describe how to further enhance the properties of our proposed discontinuity indicator using some nonlinear transformations. We aim to make the method less diffusive in smooth areas of the solution and make the method more robust for strong discontinuities. To this end, we propose using a so-called activation function f_{RV} so that $f_{\mathrm{RV}}(R^n_{\max,h}) < R^n_{\max,h}$ when $R^n_{\max,h}$ is small and $f_{\mathrm{RV}}(R^n_{\max,h}) > R^n_{\max,h}$ when $R^n_{\max,h}$ is large. There are many ways to choose a suitable activation function, see e.g., [27], and in this work we choose

$$f_{\mathrm{RV}}(x) = ax^p,$$

with $p > 1$ and $a > 0$, which is a simple activation function that has the aforementioned properties. This is illustrated in Fig. 1, where we see that $f_{\mathrm{RV}}(R^n_{\max,h})$ suppresses $R^n_{\max,h}$ to the left of the gray dashed line, where the solution is considered smooth and increases $R^n_{\max,h}$ to the right of it.

We propose solving the following projection problem: find $\tilde{\alpha}^n_h \in \mathcal{M}_h$ such that

$$(\tilde{\alpha}^n_h, w) + (h^2 \nabla\tilde{\alpha}^n_h, \nabla w) = (f_{\mathrm{RV}}(R^n_{\max,h}), w), \quad \forall w \in \mathcal{M}_h, \tag{3.21}$$

which is a standard L_2-projection with additional smoothing. Since the PDE residual is oscillative by its nature, the added smoothing helps remove oscillations from $\tilde{\alpha}^n_h$. In Section 4.5.1 we compare the performance of the method with and without the added smoothing.

Finally, we define the discontinuity indicator

$$\alpha_h^n := \min(1, |\tilde{\alpha}_h^n|), \tag{3.22}$$

so that the values of α_h^n are still between 0 and 1.

Remark 3.6. The added smoothing in (3.21) serves a similar role as the smoother proposed by Reisner et al. [50], Barter and Darmofal [5] and Ramani et al. [49]. They construct a smoother variation of artificial viscosity by solving an additional scalar reaction-diffusion equation. This approach removes small nonphysical oscillations introduced by non-smooth viscosity and does not pollute the solution in the downstream region. The difference between their approach and ours is that we solve an L_2-projection problem at each time step instead of solving an additional scalar reaction-diffusion equation.

3.3.3. An efficient grad-div stabilization parameter

Grad-div stabilization is frequently added to Galerkin discretizations of the Navier-Stokes equations and is an effective way of reducing divergence errors in the solution [46,10,31]. A larger γ_h^n will lead to smaller divergence errors present in the numerical solution. As $\gamma_h^n \to \infty$ the solution converges to a pointwise divergence-free solution as demonstrated by Case et al. [10]. The optimal value of γ_h^n (in terms of discretization error) is problem dependent and varies depending on which error norm is utilized and can vary from $\gamma_h^n \in [h, 10000]$ [31,10]. The downside with a high value of γ is that the linear system becomes very hard for an iterative Krylov method to solve. We choose γ_h^n in a Galerkin-Least-Squares fashion [57, Sec 3.1]

$$\gamma_h^n = C_{\max,\gamma} h |\rho_h^n|_{\infty,K} |\boldsymbol{u}_h^n|_{\ell_2}, \tag{3.23}$$

where $C_{\max,\gamma}$ is a user-defined parameter. As demonstrated in Section 4.5.2, choosing $C_{\max,\gamma} = 0.5$ gives additional divergence cleaning without making the iterative method significantly more expensive.

3.3.4. Summary of the nonlinear viscosity method

To summarize, the artificial viscosity coefficients κ_h^n and γ_h^n used in the stabilized FEM ((3.14) and (3.15)) are computed as follows:

1. Compute the residuals (3.17) and normalization functions (3.18).
2. Compute the node-wise maximum of each normalized residual (3.19).
3. Solve the projection problem with smoothing (3.21).
4. Set α_h^n to (3.22).
5. Compute κ_h^n combining first-order viscosity and α_h^n (3.16).
6. Compute γ_h^n using (3.23).

3.4. Preconditioning strategy

In this section, we describe the preconditioning strategy we use. Since the density update has been decoupled from the momentum update, two linear systems need to be solved at each time step. The density update (3.14) is easily handled by, for example, standard block Jacobi preconditioning. We, therefore, focus our discussion on the preconditioning technique used for the linear system associated with the momentum update. When writing (3.15) as a linear system we obtain the following block system for velocity and pressure

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+1} \end{bmatrix} = \begin{bmatrix} RHS \\ 0 \end{bmatrix}, \tag{3.24}$$

where U^{n+1} and P^{n+1} are the solution vectors for velocity and pressure, F is the momentum block, which contains mass, convection and stiffness matrices, B is the matrix associated with the pressure and velocity coupling and RHS is the right-hand side associated with (3.15). The block system (3.24) is a classical saddle point system that is difficult to solve unless a suitable preconditioning technique is chosen. In this manuscript, we use a preconditioner based on the Schur complement. We also investigate artificial compressibility as a technique to regularize the saddle point problem.

3.4.1. Artificial compressibility

The main idea with artificial compressibility is to regularize the incompressibility constraint using a penalty parameter $\epsilon > 0$ such that

$$\epsilon p_t + \nabla \cdot \boldsymbol{u} = 0, \tag{3.25}$$

to impose the divergence-free condition less strong [11,55]. In the discrete setting, (3.25) is discretized using backward Euler leading to

$$p^{n+1} = p^n - \lambda \nabla \cdot \boldsymbol{u}^{n+1}, \tag{3.26}$$

where $\lambda := \Delta t_n/\epsilon$ is a penalty parameter. Employing a Galerkin approach to the regularized divergence-free constraint (3.26) gives rise to a slightly different linear system than doing the same with the usual strong constraint $\nabla\cdot\boldsymbol{u}^{n+1}$. The singular linear system from before (3.24) has been changed into a non-singular matrix

$$\begin{bmatrix} F & B \\ B^T & \frac{1}{\lambda}M_p \end{bmatrix}\begin{bmatrix} U^{n+1} \\ P^{n+1} \end{bmatrix} = \begin{bmatrix} RHS \\ \frac{1}{\lambda}P^n \end{bmatrix}, \tag{3.27}$$

where M_p is the pressure mass matrix. The block system (3.27) converges to (3.24) as $\lambda \to \infty$. The above system (3.27) has an improved condition number compared to (3.24), but with the tradeoff that the divergence-free condition is imposed less strongly, leading to loss of accuracy.

One of the goals of this manuscript is to investigate this. The research questions we pose are the following: How large should λ be to obtain high-order accuracy? For which λ is the implicit artificial compressibility method (3.27) faster than the classical saddle point system (3.24)? Can we find a preconditioning strategy that shows similar performance between solving (3.24) and (3.27)?

Remark 3.7. Another approach is to impose artificial compressibility in the strong form as used by Guermond and Minev [18], Layton and McLaughlin [35]. This leads to a completely different system involving solving a grad-div ($\lambda\nabla\nabla\cdot\boldsymbol{u}_h$) dominated linear system which is notoriously difficult to solve when $\lambda \gg 1$ (which is required for high-order accuracy). This was circumvented by the time stepping technique by Guermond and Minev [18,19], Lundgren and Nazarov [37] which allowed using $\lambda = 1$ to achieve high-order accuracy.

Remark 3.8. Instead of an implicit discretization of the artificial compressibility constraint (3.26) it is also possible to discretize the constraint explicitly. Significant computational benefits arise from employing explicit time-stepping in conjunction with artificial compressibility [45,61], eliminating the need to solve an expensive linear system. However, as the references indicate, explicit artificial compressibility imposes a time-restriction criterion of $\Delta t \leq \lambda^{-1/2}h$. As mentioned by Guermond and Minev [18, Remark 2.2] we require $\lambda = (\Delta t)^{-1}$ to be formally first-order accurate leading to $\Delta t \leq h^2$. This restriction becomes worse for higher-order accuracy, requiring $\lambda = (\Delta t)^{-p}$ where p represents the desired order of accuracy. Notably, DeCaria et al. [13] demonstrates that a stability condition can be circumvented by introducing a grad-div operator ($\nabla\nabla\cdot\boldsymbol{u}_h$) to the momentum equations. However, this approach can be considered computationally intensive. For this reason, we opt to benchmark our results against implicit artificial compressibility, given its practical considerations.

3.4.2. *Schur complement preconditioning*

In this section we propose a preconditioning strategy suitable for both (3.24) and (3.27). We use flexible GMRES with right preconditioning [51]. The preconditioner we use is

$$P = \begin{bmatrix} F & B \\ 0 & S \end{bmatrix}, \tag{3.28}$$

where the Schur complement S is defined as

$$S = \frac{1}{\lambda}M_p - B^T F^{-1} B.$$

If the exact Schur complement is used as a preconditioner, iterative solvers such as GMRES converge in at most two iterations [15]. The challenge, however, is to design a good approximation of the Schur complement since the exact Schur complement is a dense matrix. There have been various approaches to construct this throughout the years, see e.g., [15,3,33,53].

Let M_u and A_u denote the velocity mass and stiffness matrices and let M_p and A_p denote the pressure mass and stiffness matrices. In the Stokes context, one very appealing idea comes from the fact that $B^T M_u^{-1} B \approx A_p$ and $B^T A_u^{-1} B \approx M_p$ which allows the construction of a Schur complement of viscosity-dominated or reaction-dominated problems. Unfortunately, to the authors' knowledge, a similar approximation is not known for the convection operator. If we assume that the time step follows a CFL condition and that μ is relatively small (in this manuscript $\mu \leq 0.01$), F is dominated by M_u. We, therefore, propose the following approximation of the Schur complement

$$\left(S_{approx}\right)_{i,j} = \max\left(\frac{1}{\lambda}, C_\lambda\right)\left(\varphi_i^p, \varphi_j^p\right) - \frac{\Delta t_n}{\alpha_{4,n}}\left(\frac{1}{\rho_{h,i}^{n+1}}\nabla\varphi_i^p, \nabla\varphi_j^p\right), \tag{3.29}$$

where the first matrix is the pressure mass matrix, the second matrix is a variable coefficient pressure stiffness matrix, $C_\lambda \ll 1$ is a user-defined parameter and $\alpha_{4,n}$ is the leading coefficient of the BDF4-method, see the Appendix. A similar approximation of the Schur complement has previously been used by Kronbichler et al. [33] for flow at low Reynolds numbers. The difference between their approach and ours is that we have neglected the viscous contribution of F and assumed that F is dominated by M_u. The idea of including $\frac{1}{\lambda}\left(\varphi_i^p, \varphi_j^p\right)$ in our Schur complement approximation came from the work by Dorostkar et al. [14].

We use PETSc [4] as linear algebra backend which provides many routines for iterative methods and preconditioners. In particular, PETSc has routines for flexible GMRES together with preconditioning of the form (3.28) where the user can supply their own approximation of the Schur complement. To solve the inner systems, we use conjugate gradient as an iterative method, where we limit inner iterations to 1. As preconditioner for the momentum block we use block Jacobi preconditioning and we use algebraic

multigrid as a preconditioner when solving the Schur complement block. More specifically, we use the package hypre [29] using its default settings.

We use (3.29) as an approximation for the Schur complement, both when solving the saddle point system (3.24) and the regularized saddle point system (3.27). For (3.24), one can observe that $\lambda = \infty$. We have noticed that removing the pressure mass matrix entirely leads to slow convergence of the iterative method. Instead of removing the pressure mass matrix entirely, we propose scaling the number in front of the pressure mass matrix to $\approx [10^{-6}, 10^{-14}]$ for the monolithic method by setting $C_\lambda \approx [10^{-6}, 10^{-14}]$. Choosing a number in this range gives good results and the performance is not sensitive to numbers within this range. Because of this similar performance is observed between the implicit artificial compressibility method (3.27) and the monolithic approach (3.24) for all values of λ.

3.4.3. Generating a good initial guess

Choosing a good initial guess can drastically reduce the number of iterations needed for an iterative method to converge. We present a simple way of generating an initial guess which, in the ideal case, can decrease the iteration number up to four times at a minimal computational cost. We propose generating an initial guess by using linear extrapolation from previous time steps. We denote u^* as an initial guess for the next time step. Below some extrapolation formulas for constant time steps are presented, from first-order to eight-order accuracy

$$u^* = u^n = u^{n+1} + \mathcal{O}(\Delta t), \tag{3.30}$$

$$u^* = 2u^n - u^{n-1} = u^{n+1} + \mathcal{O}\left(\Delta t^2\right),$$

$$u^* = 3u^n - 3u^{n-1} + u^{n-2} = u^{n+1} + \mathcal{O}\left(\Delta t^3\right), \tag{3.31}$$

$$u^* = 4u^n - 6u^{n-1} + 4u^{n-2} - u^{n-3} = u^{n+1} + \mathcal{O}\left(\Delta t^4\right),$$

$$u^* = 5u^n - 10u^{n-1} + 10u^{n-2} - 5u^{n-3} + u^{n-4} = u^{n+1} + \mathcal{O}\left(\Delta t^5\right),$$

$$u^* = 6u^n - 15u^{n-1} + 20u^{n-2} - 15u^{n-3} + 6u^{n-4} - u^{n-5} = u^{n+1} + \mathcal{O}\left(\Delta t^6\right),$$

$$u^* = 7u^n - 21u^{n-1} + 35u^{n-2} - 35u^{n-3} + 21u^{n-4} - 7u^{n-5} + u^{n-6} = u^{n+1} + \mathcal{O}\left(\Delta t^7\right),$$

$$u^* = 8u^n - 28u^{n-1} + 56u^{n-2} - 70u^{n-3} + 56u^{n-4} - 28u^{n-5} + 8u^{n-6} - u^{n-7} = u^{n+1} + \mathcal{O}\left(\Delta t^8\right). \tag{3.32}$$

There are many extrapolation orders to choose from, and the performance of the initial guess varies depending on the discretization used. In Section 4.4 we investigate which orders perform best on the discretization we use in this work. Extrapolation formulas for variable time steps (taken from [58]) are found in the Appendix. The formulas in the Appendix reduce to (3.30)-(3.32) when the time step is constant. We note that the projection-based initial guess is a viable alternative [16].

3.5. Summary of the method

To summarize, our stabilized finite element solution of the variable density Navier-Stokes equations (2.1) is obtained by the following steps. In each time step, the solution is advanced as

1. Compute artificial viscosity coefficients according to Section 3.3.4.
2. Compute the next time step Δt_n using Algorithm 1.
3. Compute an initial guess using one of the extrapolation formulas ((3.30)-(3.32) for constant Δt or the Appendix for variable Δt).
4. Assemble and solve density update (3.14).
5. Assemble the momentum equations (3.15) as represented by the block-system (3.24).
6. Assemble the Schur complement approximation (3.29).
7. Solve the block-system (3.24) using flexible GMRES with right preconditioning using (3.28) as a preconditioner.

To initialize the method, see Remark 3.2.

4. Numerical examples

In this section, we test the method against some benchmarks from the literature. A summary of the method is described in Section 3.5. Our code is implemented in FEniCS [1], an open-source finite element library. In the current work, we carry out experiments using $\mathbb{P}_3$ finite elements in space and the variable time step BDF4 method in time. To satisfy the inf-sup condition we use $\mathbb{P}_2$ elements for pressure. In all computations, the BDF-method is initialized using the initial time step factor $c_{init} = 0.1$. We also set CFL $= 0.15$, $s_{\max} = 1.01$ and $s_{\min} = 0.99$ in the time step control routine. In the Krylov solver, we use conjugate gradient for both the F solve and S solve. The F solve is preconditioned using block Jacobi and the S solve is preconditioned with algebraic multigrid using hypre [29] with default settings. We fix the number of inner iterations to one.

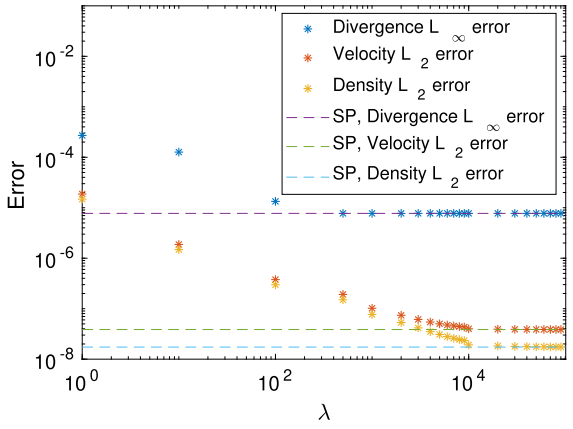

Fig. 2. Manufactured solution. Discretization error as a function of artificial compressibility penalty parameter λ for a given mesh resolution. Comparison with saddle point approach (denoted SP in the figure) which corresponds to $\lambda = \infty$.

We use the following parameters for the nonlinear viscosity method: $C_{\text{flat}} = 0.1$, $C_{\max} = 1.0$, $f_{\text{RV}}(x) = 15x^2$, $C_{\max,\gamma} = 0.5$. We chose these parameters to make the method as robust as possible and use them for all benchmark problems in this work. In particular, the benchmark problem in Section 4.7.2 proved particularly challenging and $C_{\max}$ had to be set to 1 to be able to solve the problem without getting negative density.

4.1. Manufactured solution

In this section, we verify the accuracy of the proposed method by using a manufactured solution on a unit disk. We follow the setup from [21] where the forcing function $\boldsymbol{f}$ is chosen to obtain the following exact solution

$$\rho(\boldsymbol{x}, t) = 2 + x\cos(\sin(t)) + y\sin(\sin(t)),$$
$$\boldsymbol{u}(\boldsymbol{x}, t) = \begin{bmatrix} -y\cos(t) \\ x\cos(t) \end{bmatrix},$$
$$p(\boldsymbol{x}, t) = \sin(x)\sin(y)\sin(t).$$

Dirichlet boundary conditions are imposed for the velocity. We perform a convergence study using a series of unstructured meshes. The dynamic viscosity is set to $\mu = 0.01$. The termination time is set to $T = 10$ and we set CFL $= 0.1$ to see the convergence rates as clearly as possible.

The convergence results are presented in Table 1. The L_1, L_2 and L_∞ errors are presented for all components. All the errors are computed using a high-order quadrature and are relative, i.e., they are normalized with their corresponding norm. Overall, the density and velocity errors converge with the expected convergence rate 4. Similarly, the pressure errors converge with the expected rate 3. The stabilized method also converges with its expected accuracy. We mention that the results of our modified RV method are surprisingly accurate. Typically, the errors of the coarse mesh of the RV-method can be around one order of magnitude less accurate than the unstabilized scheme [54], especially if the simulation is run for a long time (like in this case).

4.2. The impact of artificial compressibility on accuracy and computational effort

In this section, we investigate how the artificial compressibility penalty parameter λ impacts computational effort and discretization errors. We use the same setup as in Section 4.1 with an unstructured mesh consisting of 25327 $\mathbb{P}_3$ nodes. In Fig. 2 the maximum divergence error over space and time, i.e., $\|\nabla\cdot\boldsymbol{u}_h\|_{L_\infty(\Omega\times(0,T))}$, is plotted against λ. Moreover, the relative L_2 error of the velocity and density at time $T = 10$ is plotted against λ. Overall the figure indicates that a sufficiently large λ is required to ensure that the errors of the artificial compressibility method have comparable errors to solving the classical saddle point method.

4.3. Performance of the preconditioner

To evaluate our proposed preconditioning technique we take the manufactured problem from before and measure the mean number of outer iterations necessary to converge to a specified tolerance when solving (3.24). Since we keep the number of inner iterations fixed to one, the number of outer iterations is the key number when it comes to measuring the performance of the proposed preconditioning technique. The results are presented in Table 2. The relative tolerance is set to 10^{-9} and we use the solution from the previous timestep as an initial guess. The results show that the preconditioner scales well when the problem size increases for all values of λ tested. Interestingly, when λ takes on larger values, the performance of solving both the saddle point problem (3.24) and the regularized saddle point problem (3.27) becomes indistinguishable.

Table 1
Manufactured solution. Convergence study for $T = 10$, $\mu = 0.01$, CFL$=0.1$, $(\rho_h, \boldsymbol{u}_h, p_h) \in \mathbb{P}_3\mathbb{P}_3\mathbb{P}_2$.

	Velocity						
	# dofs	L_1	rate	L_2	rate	L_∞	rate
Galerkin	794	9.76E-05	0.00	1.15E-04	0.00	3.39E-04	0.00
	3314	7.38E-06	3.61	8.52E-06	3.65	3.27E-05	3.28
	13184	5.38E-07	3.79	6.21E-07	3.79	3.11E-06	3.41
	50654	3.83E-08	3.93	4.34E-08	3.95	2.21E-07	3.93
	165998	3.67E-09	3.95	4.07E-09	3.99	3.03E-08	3.35
RV	794	9.80E-05	0.00	1.16E-04	0.00	3.36E-04	0.00
	3314	7.38E-06	3.62	8.52E-06	3.65	3.28E-05	3.26
	13184	5.38E-07	3.79	6.21E-07	3.79	3.11E-06	3.41
	50654	3.83E-08	3.93	4.34E-08	3.95	2.21E-07	3.93
	165998	3.53E-09	4.02	3.94E-09	4.04	2.97E-08	3.39
	Density						
	# dofs	L_1	rate	L_2	rate	L_∞	rate
Galerkin	397	9.53E-05	0.00	1.24E-04	0.00	4.91E-04	0.00
	1657	4.16E-06	4.38	5.58E-06	4.35	3.57E-05	3.67
	6592	2.59E-07	4.02	3.59E-07	3.97	1.76E-06	4.36
	25327	1.42E-08	4.31	1.94E-08	4.33	1.50E-07	3.65
	82999	1.15E-09	4.24	1.51E-09	4.30	1.33E-08	4.09
RV	397	9.47E-05	0.00	1.24E-04	0.00	4.40E-04	0.00
	1657	4.18E-06	4.37	5.61E-06	4.34	3.40E-05	3.58
	6592	2.59E-07	4.03	3.59E-07	3.98	1.75E-06	4.29
	25327	1.42E-08	4.31	1.94E-08	4.33	1.50E-07	3.65
	82999	1.08E-09	4.35	1.45E-09	4.37	1.33E-08	4.09
	Pressure						
	# dofs	L_1	rate	L_2	rate	L_∞	rate
Galerkin	184	6.89E-04	0.00	8.18E-04	0.00	2.47E-03	0.00
	751	7.56E-05	3.14	9.40E-05	3.08	4.65E-04	2.38
	2958	8.45E-06	3.20	1.09E-05	3.15	4.53E-05	3.40
	11311	1.05E-06	3.11	1.38E-06	3.07	4.50E-06	3.44
	36987	1.72E-07	3.06	2.25E-07	3.06	8.02E-07	2.91
RV	184	6.89E-04	0.00	8.18E-04	0.00	2.47E-03	0.00
	751	7.56E-05	3.14	9.40E-05	3.08	4.65E-04	2.38
	2958	8.45E-06	3.20	1.09E-05	3.15	4.53E-05	3.40
	11311	1.05E-06	3.11	1.38E-06	3.07	4.50E-06	3.44
	36987	1.72E-07	3.06	2.25E-07	3.06	8.10E-07	2.90

Table 2
Manufactured solution. Outer iterations necessary to reach tolerance 10^{-9}.

# $\mathbb{P}_3$ nodes	397	1657	6592	26092	82999
$\lambda = 1$	10.2	8.9	8.2	7.1	6
$\lambda = 10$	11.5	9.7	8.5	7.3	6.2
$\lambda = 100$	11.9	10	8.8	7.3	6.3
$\lambda = 1000$	12.1	10	8.8	7.3	6.3
$\lambda = 10000$	12.1	10	8.9	7.4	6.3
Saddle-point ($\lambda = \infty$)	12.1	10	8.9	7.4	6.3
Saddle-point ($\lambda = \infty$), with initial guess	6.6	6	6	5.5	6.5

This convergence can be attributed to the role of λ and the parameter C_λ within the Schur complement approximation (3.29). For our proposed saddle-point method, we recommend setting C_λ within the range of approximately 10^{-6} to 10^{-14}. This leads to comparable performance between the saddle-point problem and its regularized counterpart and performance is not sensitive to numbers within this range.

4.4. Investigation of initial guess

In this section we investigate which of the extrapolation formulas (3.30)-(3.32) performs best as an initial guess when solving the linear system (3.24). We take the same problem setup from before, but only solve the problem with a constant time step since the extrapolation formulas (3.30)-(3.32) are given for constant time steps. Variable time step extrapolation formulas are presented up to

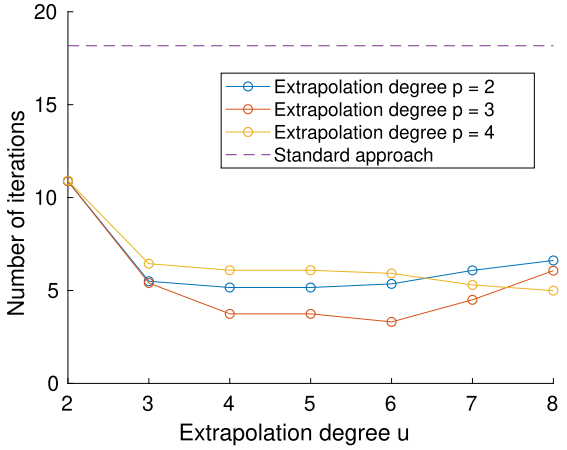

Fig. 3. Manufactured solution. Mean number of iterations to reach tolerance 10^{-14} as a function of initial guess.

Table 3
Rayleigh-Taylor instability at $Re = 5000$ and density ratio 3. Outer iterations necessary to reach tolerance 10^{-9}.

# $\mathbb{P}_3$ nodes	7471	29341	116281	462961
$\lambda = 1$	20.5	20.7	20.3	20.3
$\lambda = 10$	22.2	22.1	21.7	22
$\lambda = 100$	22.5	22.3	21.9	22.4
$\lambda = 1000$	22.5	22.4	22	22.3
$\lambda = 10000$	22.5	22.4	22	22.2
Saddle-point ($\lambda = \infty$)	22.5	22.4	22	22.1
Saddle-point ($\lambda = \infty$), with initial guess	18	17.8	17.1	16.7

order 4 in the Appendix. The mesh is chosen as an unstructured mesh containing 25327 $\mathbb{P}_3$ nodes and the relative tolerance is set very low (10^{-14}) so that the difference between the different extrapolation formulas is more pronounced.

In Fig. 3 the mean number of iterations needed to converge is presented for different extrapolation orders. We vary the extrapolation orders for both pressure and velocity. Using the solution from the previous time step as an initial guess, i.e., (3.30), is referred to as the standard approach in the figure. The results indicate that, in some sense, an optimal initial guess can be generated using order 3 for pressure and order 4 to 6 for velocity, which are similar to the order of the scheme.

4.5. Rayleigh-Taylor instability

Now we will focus on solving more complex problems. Consider the so-called Rayleigh-Taylor instability in 2D. The Rayleigh-Taylor instability occurs when a fluid accelerates into another fluid with a different density. The classical setup is that a lighter fluid supports a heavier fluid in a gravitational field. Any small perturbation to the system forces it out of equilibrium since the initial equilibrium state is unstable. We follow the same setup as [21]. The solution is computed in a rectangular domain $\Omega = \{(x, y) \in (-L/2, L/2) \times (-2L, 2L)\}$ and the characteristic velocity scale is set to $\sqrt{Lg}$ which gives the Reynolds number $Re = \rho_2 L^{3/2} g^{1/2}/\mu$. The forcing function is set to $\boldsymbol{f} = (0, -\rho g)$ to achieve a downward gravitational force, where g is the gravitational acceleration. The density jump is initially regularized using a hyperbolic tangent function and is given by

$$\rho^0(\boldsymbol{x}) = \frac{\rho_1 + \rho_2}{2} + \frac{\rho_1 - \rho_2}{2} \tanh\left(\frac{y - \eta(\boldsymbol{x})}{0.01d}\right),$$

where $\eta(\boldsymbol{x}) = -0.1d\cos(2\pi x/d)$, $d = 2$. Slip boundary conditions are enforced at the boundaries. The following parameters are used: $L = 1$, $\rho_1 = 3$, $\rho_2 = 1$, $g = 1$.

We present the time evolution of the computed density field and the computed discontinuity indicator α_h, in the time-scale of Tryggvason ($t = \sqrt{2} t_{Tryg}$), for $Re = 1000$ in Fig. 4 and $Re = 5000$ in Fig. 5. The results use 462961 $\mathbb{P}_3$ nodes. Overall, the results are in agreement with the result obtained by Guermond and Salgado [21] where a second-order accurate Taylor-Hood FEM was used.

To evaluate our preconditioning strategy, we include the number of outer iterations necessary to converge to the relative tolerance 10^{-9} in Table 3. The solution from the previous time step is used as an initial guess. In the last row of Table 3, we present results when the initial guess strategy from Section 3.4.3 is used. Motivated by the results from Section 4.4, we choose third-order extrapolation for pressure and fourth-order extrapolation for velocity to generate our initial guess. The proposed preconditioning technique shows similar performance between the regularized saddle-point problem (3.27) and the saddle-point problem (3.24).

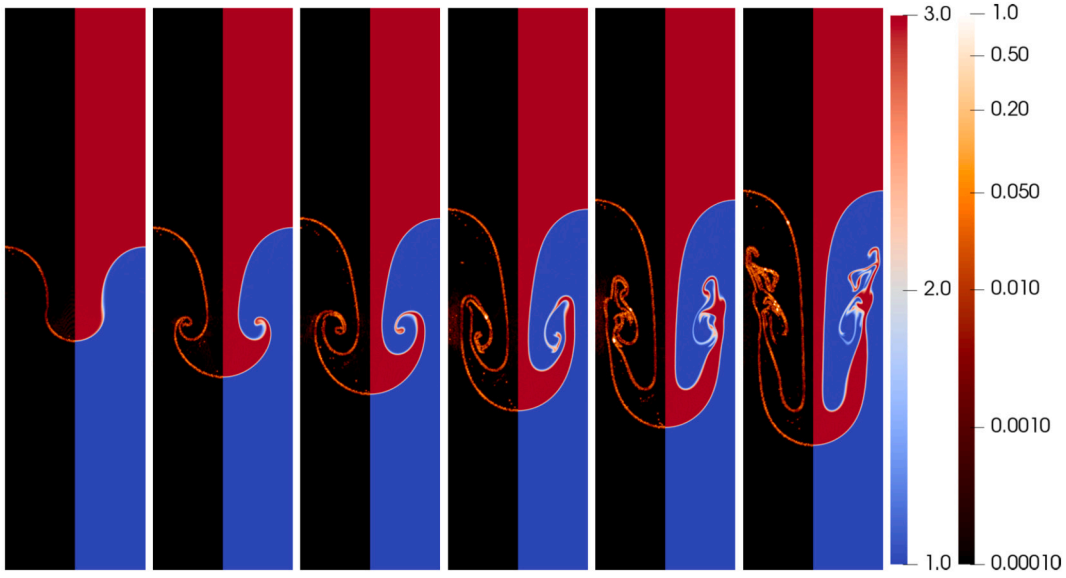

Fig. 4. Rayleigh-Taylor instability at $Re = 1000$ and density ratio 3. Density ρ_h (right) and discontinuity indicator α_h in logarithmic scale (left) at times $t = 1, 1.5, 1.75, 2, 2.25$ and 2.5.

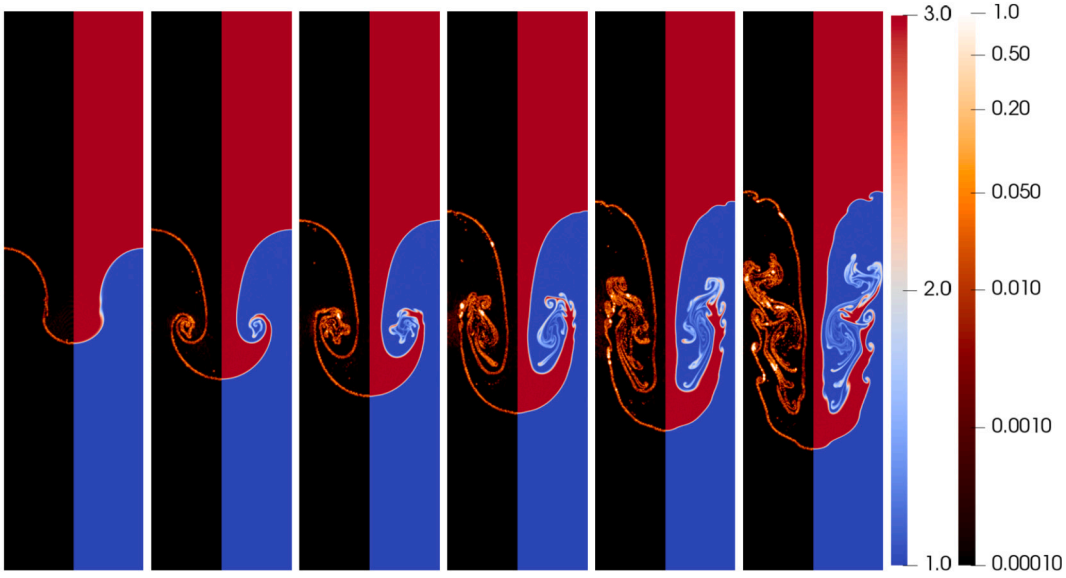

Fig. 5. Rayleigh-Taylor instability at $Re = 5000$ and density ratio 3. Density ρ_h (right) and discontinuity indicator α_h in logarithmic scale (left) at times $t = 1, 1.5, 1.75, 2, 2.25$ and 2.5.

4.5.1. *Effect of post-processing the discontinuity indicator*

In this section, we briefly investigate the impact of the post-processing proposed in Section 3.3.2. One of the proposed ideas was to use an activation function f_{RV} such that f_{RV} suppresses the residual in the smooth region, but amplifies the residual in non-smooth regions. The motivation was to increase accuracy in smooth regions while also improving discontinuity detection when needed. The other proposed idea was to add the additional elliptic smoothing in (3.21) to reduce oscillations in α_h. We consider the same setup as in Section 4.5.

In Fig. 6 we compare the results obtained at $T = 2.5$ with two activation functions: $f_{\mathrm{RV}} = 15x^2$ and $f_{\mathrm{RV}} = x$, and with or without elliptic smoothing $(h^2\nabla\tilde{\alpha}_h^n, \nabla w)$ when solving the L_2 projection step (3.21). We see that using $f_{\mathrm{RV}} = 15x^2$ results in a higher value of the discontinuity indicator at the discontinuities and a smaller value in the smooth region. The computed densities are overall quite similar with slightly more structure in the flow when using $f_{\mathrm{RV}} = 15x^2$. As evident from the figure, the added smoothing reduces oscillations in the discontinuity indicator and makes the interface of the density sharper.

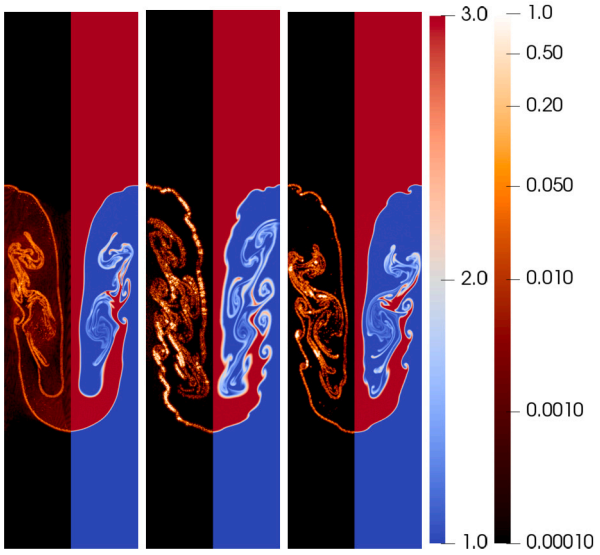

Fig. 6. Rayleigh-Taylor instability at $Re = 5000$ and density ratio 3. Discontinuity indicator α_h and density ρ_h for different activation functions and smoothing post-processing: using $f_{\mathrm{RV}} = x$ with no smoothing (left), using $f_{\mathrm{RV}} = 15x^2$ with no smoothing (middle), and using $f_{\mathrm{RV}} = 15x^2$ with smoothing (right).

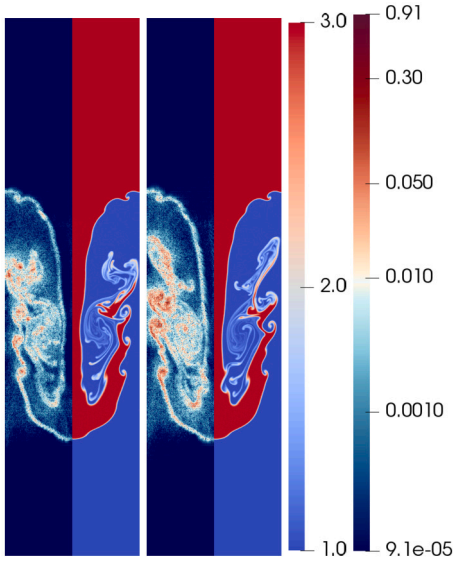

Fig. 7. Rayleigh-Taylor instability at $Re = 5000$ and density ratio 3. Comparison between (left column) grad-div stabilization using $C_{\mathrm{max},\gamma} = 0.5$ and (right column) no grad-div stabilization, i.e., $C_{\mathrm{max},\gamma} = 0$. Computed density ρ_h (right) and divergence error $\nabla \cdot \boldsymbol{u}_h$ in logarithmic scale (left).

Table 4
Rayleigh-Taylor instability at $Re = 5000$ and density ratio 3. Outer iterations necessary to reach tolerance 10^{-9} with and without grad-div stabilization.

# $\mathbb{P}_3$ nodes	7471	29341	116281	462961
Saddle-point, $C_{\mathrm{max},\gamma} = 0.0$	18	17.8	17.1	16.7
Saddle-point, $C_{\mathrm{max},\gamma} = 0.5$	18.5	18.4	17.9	17.5
Saddle-point, $C_{\mathrm{max},\gamma} = 1$,	19.7	19.6	19.1	18.7

4.5.2. *Effect of grad-div stabilization*

In this section, we illustrate the effects of adding grad-div stabilization. In Fig. 7 the results of our proposed discretization are presented at $T = 2.5$, both with and without grad-div stabilization. The effect on the number of iterations is presented in Table 4 showing that choosing $C_{\mathrm{max},\gamma} \approx 0.5$ has minimal effect on the number of iterations necessary for the iterative Krylov method to converge. Third-order extrapolation for pressure and fourth-order extrapolation for velocity was used to generate an initial guess.

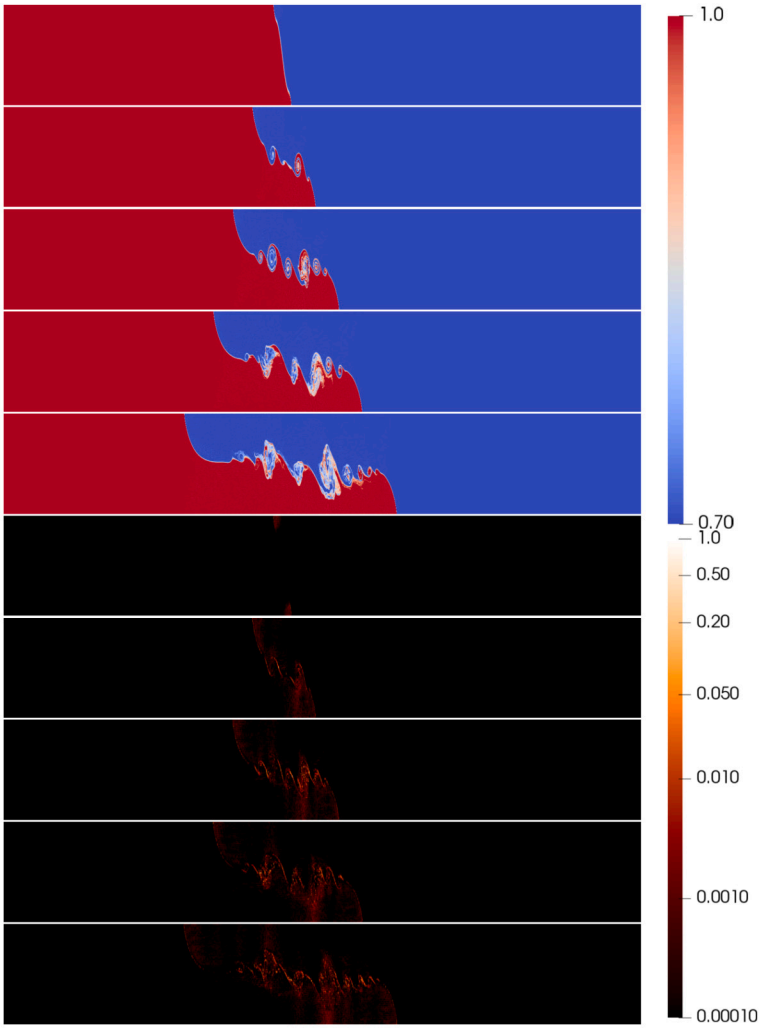

Fig. 8. Lock-exchange problem at $Re = 4000$ and density ratio 1/0.7. Density ρ_h (top) and discontinuity indicator α_h in logarithmic scale (bottom) at times $t = 1, 3, 5, 7, 10$.

Since $\gamma_h = \mathcal{O}(h)$, the divergence cleaning effect is quite small as evident by the figure. However, it still does have some effect on the flow feature and the performance of the iterative method is only slightly slower due to the added grad-div stabilization. In conclusion, the proposed scaling of the added grad-div stabilization adds additional divergence cleaning without adversely affecting the performance of the method.

4.6. Lock-exchange

This benchmark problem was studied extensively by Bartholomew and Laizet [6] and Birman et al. [7]. The classical setup is a heavy fluid with density ρ_1 and a lighter fluid with density ρ_2 separated by a vertical barrier. The heavy fluid is located to the left of the barrier and the lighter fluid is to the right. The barrier is removed at $t = 0$ and then the heavy gas moves to the right and the lighter fluid moves to the left. Following the setup from [6,7], the computational domain is set to $\Omega = \{(x, y) \in (-L/2, L/2) \times (0, 32L)\}$ with a characteristic velocity scale set to $u = \sqrt{g(\rho_1 - \rho_2)/\rho_1 L}$ and the Reynolds number is defined as $Re = \rho_1 L^{3/2} \sqrt{g(\rho_1 - \rho_2)/\rho_1}/\mu$. Following the reference we set $L = 1$, $\rho_1 = 1$ and $\rho_2 = 0.7$. The Reynolds number is set to 4000 yielding a constant dynamic viscosity coefficient μ. The forcing function is set to $\boldsymbol{f} = (0, -\rho g)$ to yield a downward gravitational force. The initial condition is initially regularized using the error function and is given by

$$\rho^0(\boldsymbol{x}) = \frac{1}{2}\left(\frac{\rho_2}{\rho_1} + 1\right) - \frac{1}{2}\left(1 - \frac{\rho_2}{\rho_1}\right) \operatorname{erf}\left(x_0 \sqrt{Re}\right),$$

where $x_0 = 14$ is the location of the barrier at $t = 0$. The computed density and discontinuity indicator are presented in Fig. 8 at times $T = 1, 3, 5, 7, 10$ (at time-scale $L^{1/2} \left(g(\rho_1 - \rho_2)/\rho_1\right)^{-1/2}$). The results use 2889901 $\mathbb{P}_3$ nodes. The main difference between the results presented in this work and [6,7] is that the authors in [6,7] added a constant diffusion term to the density update to model

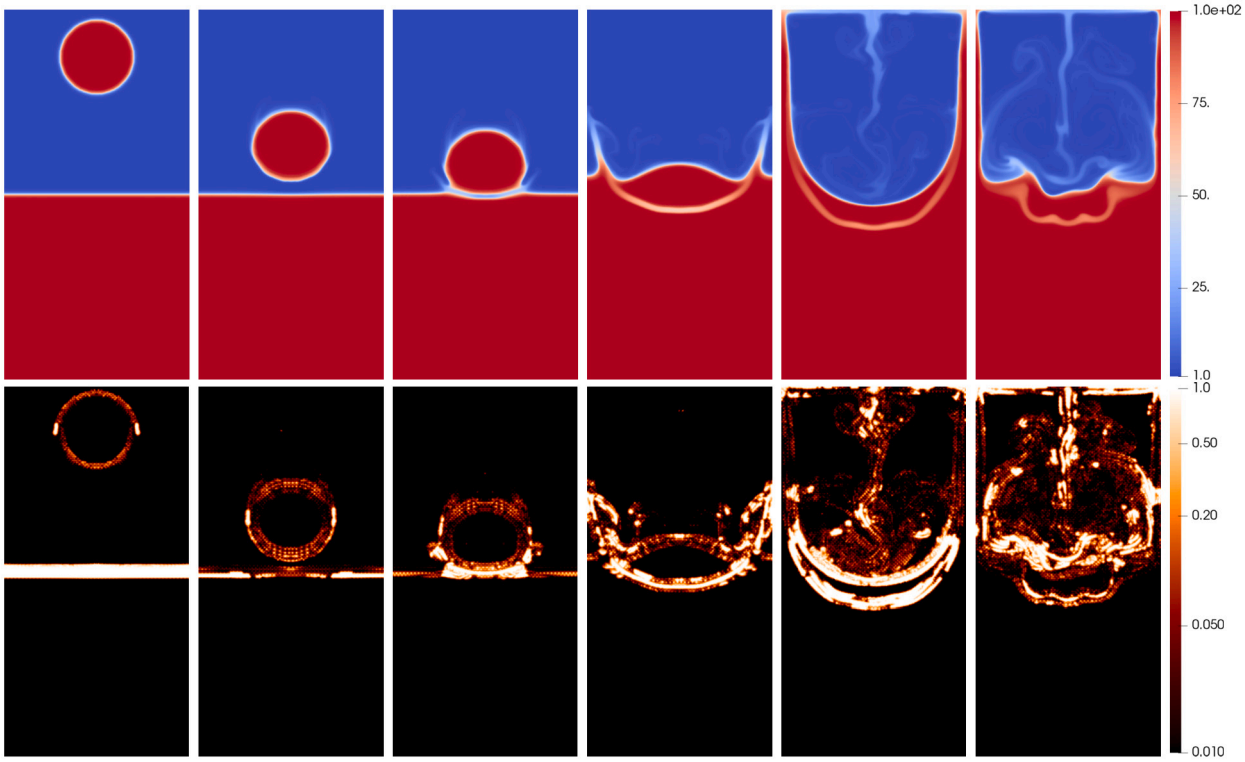

Fig. 9. Falling bubble in 2D at $Re = 3132$ and density ratio 100. Density ρ_h (top) and discontinuity indicator α_h (bottom) at times $t = 0.1, 1, 1.1, 1.3, 2.2, 3$.

molecular diffusivity. In this work, the only added mass diffusivity comes from our proposed stabilization in Section 3.3. In this way, more structure is evident in Fig. 8 compared to the results from [6,7]. Another difference is that Bartholomew and Laizet [6], Birman et al. [7] used constant kinematic viscosity instead of constant dynamic viscosity in their simulations.

4.7. Robustness at large density ratios

In this section, we verify the robustness of the proposed method by solving benchmark problems with density ratios of 100 and 1000 at high Reynolds numbers. To this end, we first consider the benchmark problem proposed Calgaro et al. [9] which simulates a heavy droplet that falls through a light fluid into a planar heavy fluid in Section 4.7.1. We later extend this benchmark problem to 3D in Section 4.7.2.

The point of these benchmarks is not to simulate a realistic fluid. As mentioned by Calgaro et al. [9], there is no surface tension present in the model (2.1) and a level-set approach, a Cahn-Hillard approach or a volume of fluid approach, see e.g., references in [9], is preferred if the goal is to model multiphase flow of liquids. The good thing about this benchmark is that it serves as a challenging robustness test when the density ratio becomes very large.

4.7.1. Falling bubble in 2D

This benchmark was first proposed in [9] which simulates a heavy droplet that falls through a light fluid into a planar heavy fluid. The computational domain is $\Omega = \{(x, y, z) \in (0, L) \times (0, 2L)\}$ with $L = 1$ and at $t = 0$ the fluid is at rest. The forcing function is set to $\boldsymbol{f} = (0, -\rho g)$ with $g = 1$. Similar to Section 4.5, the Reynolds number is defined as $Re = \rho_{min} L^{3/2} g^{1/2} / \mu$ which we set to 3132. The initial density profile is given by

$$\rho^0(\boldsymbol{x}) = \begin{cases} 100, & \text{if } 0 \le y \le 1 \text{ or } 0 \le r \le 0.2, \\ 1, & \text{if } 1 < y \le 2 \text{ or } 0.2 < r, \end{cases}$$

where $r = \sqrt{(x - 0.5)^2 + (y - 1.75)^2}$. Since the initial condition $\rho^0(\boldsymbol{x})$ is discontinuous, it will lead to negative density when interpolated to $\mathbb{P}_3$ space. We, therefore, perform a smoothing step on the initial condition by solving the following L_2-projection problem

$$(\tilde{\rho}^0, w) + \sigma(h^2 \nabla \tilde{\rho}^0, \nabla w) = (\rho^0, w), \quad \forall w \in \mathcal{M}_h, \tag{4.1}$$

to ensure that there are no oscillations present in the initial condition. We set the smoothing weight to $\sigma = 7$. The computed density field and discontinuity indicator are presented in Fig. 9. The results use 180901 $\mathbb{P}_3$ nodes. Calgaro et al. [9] solved this problem using a hybrid finite-volume FEM and observed a similar deformation of the bubble.

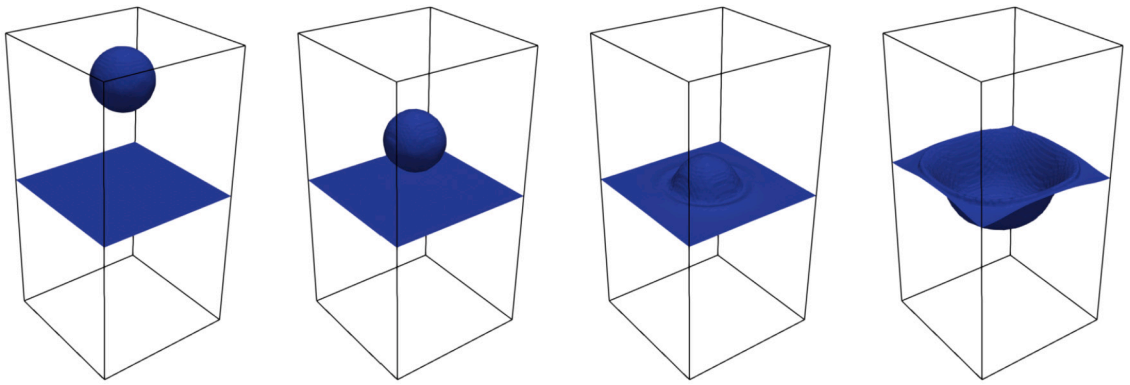

Fig. 10. Falling bubble in 3D at $Re = \infty$ and density ratio 1000. The density threshold $\rho_h \geq 500$ is plotted at times $t = 0.0, 0.9, 1.2, 1.8$.

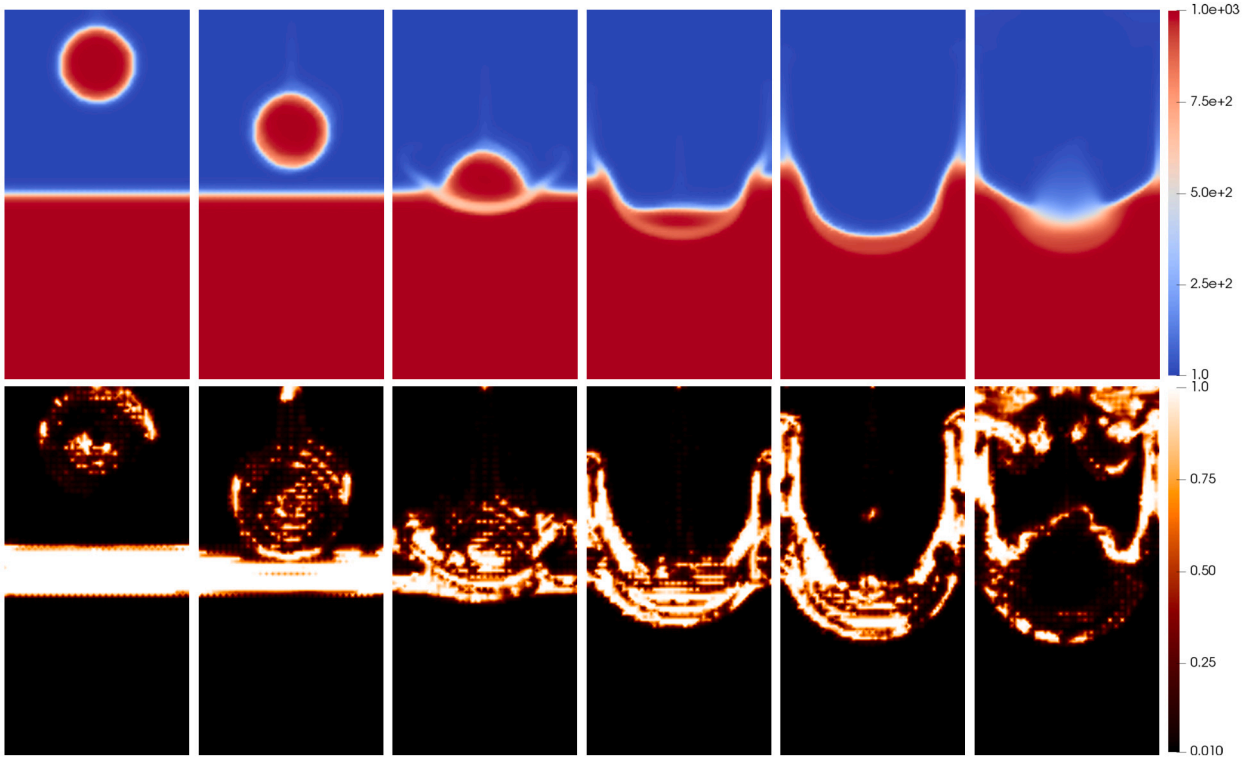

Fig. 11. Falling bubble in 3D at $Re = \infty$ and density ratio 1000. Density ρ_h (top) and discontinuity indicator α_h (bottom) at $z = 0.5$ and at times $t = 0.3, 0.9, 1.2, 1.5, 1.8, 2.4$.

4.7.2. Falling bubble in 3D

Inspired by the 2D falling bubble test in Section 4.7.1 we aim to construct a more challenging benchmark by increasing the density ratio, moving the domain to 3D and setting $\mu = 0$. The computational domain is $\Omega = \{(x, y, z) \in (0, L) \times (0, 2L) \times (0, L)\}$ with $L = 1$ and at $t = 0$ the fluid is at rest. The forcing function is set to $\boldsymbol{f} = (0, -\rho g, 0)$ with $g = 1$. The initial condition is set to a 3D bubble

$$\rho(x, y, z, 0) = \begin{cases} 1000, & \text{if } 0 \leq y \leq 1 \text{ or } 0 \leq r \leq 0.2, \\ 1, & \text{if } 1 < y \leq 2 \text{ or } 0.2 < r, \end{cases}$$

where $r = \sqrt{(x - 0.5)^2 + (y - 1.75)^2 + (z - 0.5)^2}$. As in the 2D case, we solve (4.1) with $\sigma = 7$ to ensure that the initial condition is oscillation free. The computed density field and discontinuity indicator are presented in Fig. 11. To better visualize the solution we only plot the heavy fluid in Fig. 10 where we set the threshold value to 500. The results use 1498861 $\mathbb{P}_3$ nodes. The combination of a coarse mesh together with the large density ratio leads to quite diffusive results. The results do, however, confirm the robustness of the method.

5. Conclusion and further work

The purpose of this work was to develop a reliable high-order FEM approximation of the variable density Navier-Stokes equations. To this end, we considered a monolithic approach and the proposed discretization leads to a saddle point system that needs to be solved each time step. We investigated artificial compressibility as a technique to regularize the saddle point system. A Schur complement preconditioning technique was employed to solve the system and we provided a link between artificial compressibility methods and the solution of classical saddle point systems. The performance of the proposed preconditioning shows similar performance to solving the classical saddle point system and using the artificial compressibility method for all values of the penalty parameter λ.

We proposed stabilizing the method using a modified Guermond-Popov viscous flux [20] scaled with mesh-dependent artificial viscosity and grad-div stabilization which we showed satisfies a semi-discrete kinetic energy balance. More specifically, we showed that the added mass diffusion does not affect the kinetic energy balance. The artificial viscosity coefficients were constructed using high-order residual-based viscosity. We explored a new variant of the RV method where the residual was self-normalized to construct a unit-free discontinuity indicator.

In the current work, $\mathbb{P}_3\mathbb{P}_3\mathbb{P}_2$ continuous finite elements were used and in time a BDF4 time stepping method was used. Several benchmark problems in 2D and 3D were solved which confirms the expected convergence rate for smooth problems and shows accurately resolved discontinuities in the presence of sharp gradients. The method was also shown to handle density ratios up to 1000 in 3D without encountering issues with positivity of density. Even though the method performs well on the benchmarks considered in this manuscript, the solution is not oscillation free and positivity of density can not be guaranteed. Additional work is required to achieve this and is one of the future goals of the authors.

CRediT authorship contribution statement

Lukas Lundgren: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Murtazo Nazarov:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Supervision, Visualization, Writing – review & editing.

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.

Data availability

No data was used for the research described in the article.

Acknowledgements

The computations were enabled by resources in project SNIC 2022/22-428 provided by the Swedish National Infrastructure for Computing (SNIC) at UPPMAX, partially funded by the Swedish Research Council through grant agreement no. 2018-05973. The first author was partly supported by the Center for Interdisciplinary Mathematics, Uppsala University. We express our gratitude to the anonymous reviewers whose constructive and useful comments significantly improved the quality of this manuscript.

Appendix A

In this section we provide the coefficients used to approximate $\partial_t(\rho_h \boldsymbol{u}_h)$, $\partial_t \rho_h$ and $\boldsymbol{u}_h^*$ for commonly used variable time step BDF-methods [58]. Since these are used to initialize the BDF4 method used in this work, we present these for completeness. The extrapolation formulas used for $\boldsymbol{u}_h^*$ presented in this section reduce to (3.30)-(3.31) if the time steps are constant. Given solutions from previous time steps $\rho_h^{n+j}, \boldsymbol{u}_h^{n+j}, j = -(N_{\text{BDF}} - 1), \dots 0$, where N_{BDF} is the order of the BDF, find $\rho_h^{n+1} \in M_h$ such that (3.14) holds. Then find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ and $p_h \in Q_h$ such that (3.15) holds. The necessary coefficients are provided in the following sections.

A.1. Variable BDF1 method

$$d_t\left(\rho_h^{n+1}\right) = \frac{1}{\Delta t_n}(\rho_h^{n+1} - \rho_h^n), \quad d_t\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right) = \frac{1}{\Delta t_n}(\rho_h^{n+1}\boldsymbol{u}_h^{n+1} - \rho_h^n\boldsymbol{u}_h^n), \quad \boldsymbol{u}_h^* = \boldsymbol{u}_h^n.$$

A.2. Variable BDF2 method

$$d_t\left(\rho_h^{n+1}\right) = \frac{1}{\Delta t_n}\sum_{j=0}^{2}\alpha_{j,n}\rho_h^{n+j-1}, \quad d_t\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right) = \frac{1}{\Delta t_n}\sum_{j=0}^{2}\alpha_{j,n}\rho^{n+j-1}\boldsymbol{u}^{n+j-1}, \quad \boldsymbol{u}_h^* = \sum_{j=0}^{1}\beta_{j,n}\boldsymbol{u}_h^{n+j-1},$$

where

$$\alpha_{0,n} = \frac{\omega_n^2}{1+\omega_n},$$
$$\alpha_{1,n} = -(1+\omega_n),$$
$$\alpha_{2,n} = \frac{1+2\omega_n}{1+\omega_n},$$
$$\beta_{0,n} = -\omega_n,$$
$$\beta_{1,n} = 1+\omega_n.$$

A.3. Variable BDF3 method

$$d_t\left(\rho_h^{n+1}\right) = \frac{1}{\Delta t_n}\sum_{j=0}^{3}\alpha_{j,n}\rho_h^{n+j-2}, \quad d_t\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right) = \frac{1}{\Delta t_n}\sum_{j=0}^{3}\alpha_{j,n}\rho^{n+j-2}\boldsymbol{u}^{n+j-2}, \quad \boldsymbol{u}_h^{*} = \sum_{j=0}^{2}\beta_{j,n}\boldsymbol{u}_h^{n+j-2}, \tag{A.1}$$

where

$$\alpha_{0,n} = -\frac{\omega_{n-1}^3\omega_n^2(1+\omega_n)}{(1+\omega_{n-1})(1+\omega_{n-1}+\omega_{n-1}\omega_n)},$$
$$\alpha_{1,n} = \omega_n^2\left(\omega_{n-1}+\frac{1}{1+\omega_n}\right),$$
$$\alpha_{2,n} = -1-\omega_n-\frac{\omega_{n-1}\omega_n(1+\omega_n)}{1+\omega_{n-1}},$$
$$\alpha_{3,n} = 1+\frac{\omega_n}{1+\omega_n}+\frac{\omega_{n-1}\omega_n}{1+\omega_{n-1}(1+\omega_n)},$$
$$\beta_{0,n} = \frac{\omega_{n-1}^2\omega_n(1+\omega_n)}{1+\omega_{n-1}},$$
$$\beta_{1,n} = -\omega_n\left(1+\omega_{n-1}(1+\omega_n)\right),$$
$$\beta_{2,n} = \frac{(1+\omega_n)(1+\omega_{n-1}(1+\omega_n))}{1+\omega_{n-1}}.$$

A.4. Variable BDF4 method

$$d_t\left(\rho_h^{n+1}\right) = \frac{1}{\Delta t_n}\sum_{j=0}^{4}\alpha_{j,n}\rho_h^{n+j-3}, \quad d_t\left(\rho_h^{n+1}\boldsymbol{u}_h^{n+1}\right) = \frac{1}{\Delta t_n}\sum_{j=0}^{4}\alpha_{j,n}\rho^{n+j-3}\boldsymbol{u}^{n+j-3}, \quad \boldsymbol{u}_h^{*} = \sum_{j=0}^{3}\beta_{j,n}\boldsymbol{u}_h^{n+j-3},$$

where

$$\alpha_{0,n} = \frac{1+\omega_n}{1+\omega_{n-2}}\frac{A_2}{A_1}\frac{\omega_{n-2}^4\omega_{n-1}^3\omega_n^2}{A_3},$$
$$\alpha_{1,n} = -\omega_{n-1}^3\omega_n^2\frac{1+\omega_n}{1+\omega_{n-1}}\frac{A_3}{A_2},$$
$$\alpha_{2,n} = \omega_n\left(\frac{\omega_n}{1+\omega_n}+\omega_{n-1}\omega_n\frac{A_3+\omega_{n-2}}{1+\omega_{n-2}}\right),$$
$$\alpha_{3,n} = -1-\omega_n\left(1+\frac{\omega_{n-1}(1+\omega_n)}{1+\omega_{n-1}}\left(1+\frac{\omega_{n-2}A_2}{A_1}\right)\right),$$
$$\alpha_{4,n} = 1+\frac{\omega_n}{1+\omega_n}+\frac{\omega_{n-1}\omega_n}{A_2}+\frac{\omega_{n-2}\omega_{n-1}\omega_n}{A_3},$$
$$\beta_{0,n} = -\omega_{n-2}^3\omega_{n-1}^2\omega_n\frac{1+\omega_n}{1+\omega_{n-2}}\frac{A_2}{A_1},$$
$$\beta_{1,n} = \omega_{n-1}^2\omega_n\frac{1+\omega_n}{1+\omega_{n-1}}A_3,$$
$$\beta_{2,n} = -A_2A_3\frac{\omega_n}{1+\omega_{n-2}},$$
$$\beta_{3,n} = \frac{\omega_{n-1}(1+\omega_n)}{1+\omega_{n-1}}\frac{(1+\omega_n)(A_3+\omega_{n-2})+\frac{1+\omega_{n-2}}{\omega_{n-1}}}{A_1},$$

$$A_1 = 1 + \omega_{n-2}(1 + \omega_{n-1}),$$

$$A_2 = 1 + \omega_{n-1}(1 + \omega_n),$$

$$A_3 = 1 + \omega_{n-2} A_2.$$

References

[1] M. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. Rognes, G. Wells, The FEniCS project version 1.5, Arch. Numer. Softw. 3 (2015) 9–23, https://doi.org/10.11588/ans.2015.100.20553.

[2] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. (ISSN 0036-1429) 32 (3) (1995) 797–823, https://doi.org/10.1137/0732037.

[3] O. Axelsson, X. He, M. Neytcheva, Numerical solution of the time-dependent Navier-Stokes equation for variable density–variable viscosity. Part I, Math. Model. Anal. (ISSN 1392-6292) 20 (2) (2015) 232–260, https://doi.org/10.3846/13926292.2015.1021395.

[4] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, a. Dener, v. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, D. May, L. McInnes, R. Mills, T. Munson, K. Rupp, P. Sanan, B. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc users manual, 2019.

[5] G.E. Barter, D.L. Darmofal, Shock capturing with PDE-based artificial viscosity for DGFEM. I. Formulation, J. Comput. Phys. (ISSN 0021-9991) 229 (5) (2010) 1810–1827, https://doi.org/10.1016/j.jcp.2009.11.010.

[6] P. Bartholomew, S. Laizet, A new highly scalable, high-order accurate framework for variable-density flows: application to non-Boussinesq gravity currents, Comput. Phys. Commun. (ISSN 0010-4655) 242 (2019) 83–94, https://doi.org/10.1016/j.cpc.2019.03.019.

[7] V.K. Birman, J.E. Martin, E. Meiburg, The non-Boussinesq lock-exchange problem. II. High-resolution simulations, J. Fluid Mech. (ISSN 0022-1120) 537 (2005) 125–144, https://doi.org/10.1017/S0022112005005033.

[8] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, in: FENOMECH "81, Part I, Stuttgart, 1981, Comput. Methods Appl. Mech. Eng. (ISSN 0045-7825) 32 (1–3) (1982) 199–259, https://doi.org/10.1016/0045-7825(82)90071-8.

[9] C. Calgaro, E. Creusé, T. Goudon, An hybrid finite volume-finite element method for variable density incompressible flows, J. Comput. Phys. (ISSN 0021-9991) 227 (9) (2008) 4671–4696, https://doi.org/10.1016/j.jcp.2008.01.017.

[10] M.A. Case, V.J. Ervin, A. Linke, L.G. Rebholz, A connection between Scott-Vogelius and grad-div stabilized Taylor-Hood FE approximations of the Navier-Stokes equations, SIAM J. Numer. Anal. (ISSN 0036-1429) 49 (4) (2011) 1461–1481, https://doi.org/10.1137/100794250.

[11] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. (ISSN 0021-9991) 2 (1) (1967) 12–26, https://doi.org/10.1016/0021-9991(67)90037-X.

[12] T.A. Dao, M. Nazarov, A high-order residual-based viscosity finite element method for the ideal MHD equations, J. Sci. Comput. (ISSN 0885-7474) 92 (3) (2022) 77, https://doi.org/10.1007/s10915-022-01918-4.

[13] V. DeCaria, W. Layton, M. McLaughlin, A conservative, second order, unconditionally stable artificial compression method, Comput. Methods Appl. Mech. Eng. (ISSN 0045-7825) 325 (2017) 733–747, https://doi.org/10.1016/j.cma.2017.07.033.

[14] A. Dorostkar, M. Neytcheva, B. Lund, Numerical and computational aspects of some block-preconditioners for saddle point systems, Parallel Comput. (ISSN 0167-8191) 49 (2015) 164–178, https://doi.org/10.1016/j.parco.2015.06.003.

[15] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, second edition, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, ISBN 978-0-19-967880-8, 2014, https://doi.org/10.1093/acprof:oso/9780199678792.001.0001.

[16] P.F. Fischer, Projection techniques for iterative solution of $A\underline{x} = \underline{b}$ with successive right-hand sides, Comput. Methods Appl. Mech. Eng. (ISSN 0045-7825) 163 (1–4) (1998) 193–204, https://doi.org/10.1016/S0045-7825(98)00012-7.

[17] V. Girault, P.-A. Raviart, Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms, Springer Series in Computational Mathematics, vol. 5, Springer-Verlag, Berlin, ISBN 3-540-15796-4, 1986, https://doi.org/10.1007/978-3-642-61623-5.

[18] J.-L. Guermond, P. Minev, High-order time stepping for the incompressible Navier-Stokes equations, SIAM J. Sci. Comput. (ISSN 1064-8275) 37 (6) (2015) A2656–A2681, https://doi.org/10.1137/140975231.

[19] J.-L. Guermond, P. Minev, High-order adaptive time stepping for the incompressible Navier-Stokes equations, SIAM J. Sci. Comput. (ISSN 1064-8275) 41 (2) (2019) A770–A788, https://doi.org/10.1137/18M1209301.

[20] J.-L. Guermond, B. Popov, Viscous regularization of the Euler equations and entropy principles, SIAM J. Appl. Math. (ISSN 0036-1399) 74 (2) (2014) 284–305, https://doi.org/10.1137/120903312.

[21] J.-L. Guermond, A. Salgado, A splitting method for incompressible flows with variable density based on a pressure Poisson equation, J. Comput. Phys. (ISSN 0021-9991) 228 (8) (2009) 2834–2846, https://doi.org/10.1016/j.jcp.2008.12.036.

[22] J.-L. Guermond, A.J. Salgado, Error analysis of a fractional time-stepping technique for incompressible flows with variable density, SIAM J. Numer. Anal. (ISSN 0036-1429) 49 (3) (2011) 917–944, https://doi.org/10.1137/090768758.

[23] J.-L. Guermond, M. Nazarov, B. Popov, Implementation of the entropy viscosity method, Technical Report 4015, KTH, Numerical Analysis, NA, 2011, QC 20110720.

[24] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. (ISSN 0021-9991) 230 (11) (2011) 4248–4267, https://doi.org/10.1016/j.jcp.2010.11.043.

[25] J.L. Guermond, A. Larios, T. Thompson, Validation of an entropy-viscosity model for large eddy simulation, in: J. Fröhlich, H. Kuerten, B.J. Geurts, V. Armenio (Eds.), Direct and Large-Eddy Simulation IX, Springer International Publishing, Cham, ISBN 978-3-319-14448-1, 2015, pp. 43–48.

[26] J.-L. Guermond, M. Nazarov, B. Popov, I. Tomas, Second-order invariant domain preserving approximation of the Euler equations using convex limiting, SIAM J. Sci. Comput. (ISSN 1064-8275) 40 (5) (2018) A3211–A3239, https://doi.org/10.1137/17M1149961.

[27] J.-L. Guermond, M. Nazarov, B. Popov, Finite element based invariant-domain preserving approximation of hyperbolic systems: beyond second-order accuracy in space, Comput. Methods Appl. Mech. Eng. 418 (2024) 116470, https://doi.org/10.1016/j.cma.2023.116470, ISSN 0045-7825, 1879-2138.

[28] T.J.R. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics. VIII. The Galerkin/least-squares method for advective-diffusive equations, Comput. Methods Appl. Mech. Eng. (ISSN 0045-7825) 73 (2) (1989) 173–189, https://doi.org/10.1016/0045-7825(89)90111-4.

[29] hypre, *hypre*: high performance preconditioners, https://llnl.gov/casc/hypre, https://github.com/hypre-space/hypre.

[30] A. Jameson, Origins and further development of the Jameson–Schmidt–Turkel scheme, AIAA J. 55 (2017) 1–23, https://doi.org/10.2514/1.J055493.

[31] E.W. Jenkins, V. John, A. Linke, L.G. Rebholz, On the parameter choice in grad-div stabilization for the Stokes equations, Adv. Comput. Math. (ISSN 1019-7168) 40 (2) (2014) 491–516, https://doi.org/10.1007/s10444-013-9316-1.

[32] C. Johnson, A. Szepessy, P. Hansbo, On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws, Math. Comput. (ISSN 0025-5718) 54 (189) (1990) 107–129, https://doi.org/10.2307/2008684.

[33] M. Kronbichler, A. Diagne, H. Holmgren, A fast massively parallel two-phase flow solver for microfluidic chip simulation, Int. J. High Perform. Comput. Appl. 32 (2) (2018) 266–287, https://doi.org/10.1177/1094342016671790.

[34] M.G. Larson, F. Bengzon, The Finite Element Method: Theory, Implementation, and Applications, Texts in Computational Science and Engineering, vol. 10, Springer, Heidelberg, ISBN 978-3-642-33286-9, 2013, https://doi.org/10.1007/978-3-642-33287-6, 978-3-642-33287-6.
[35] W. Layton, M. McLaughlin, Doubly-adaptive artificial compression methods for incompressible flow, J. Numer. Math. (ISSN 1570-2820) 28 (3) (2020) 179–196, https://doi.org/10.1515/jnma-2019-0015.
[36] L. Lu, M. Nazarov, P. Fischer, Nonlinear artificial viscosity for spectral element methods, C. R. Math. Acad. Sci. Paris (ISSN 1631-073X) 357 (7) (2019) 646–654, https://doi.org/10.1016/j.crma.2019.07.006.
[37] L. Lundgren, M. Nazarov, A high-order artificial compressibility method based on Taylor series time-stepping for variable density flow, J. Comput. Appl. Math. (ISSN 0377-0427) (2022) 114846, https://doi.org/10.1016/j.cam.2022.114846, https://www.sciencedirect.com/science/article/pii/S0377042722004447.
[38] L. Lundgren, M. Nazarov, A fully conservative and shift-invariant formulation for Galerkin discretizations of incompressible variable density flow, https://arxiv.org/abs/2305.04813, 2023.
[39] J. Manzanero, G. Rubio, D.A. Kopriva, E. Ferrer, E. Valero, An entropy-stable discontinuous Galerkin approximation for the incompressible Navier-Stokes equations with variable density and artificial compressibility, J. Comput. Phys. (ISSN 0021-9991) 408 (2020) 109241, https://doi.org/10.1016/j.jcp.2020.109241.
[40] S. Marras, M. Nazarov, F.X. Giraldo, Stabilized high-order Galerkin methods based on a parameter-free dynamic SGS model for LES, J. Comput. Phys. (ISSN 0021-9991) 301 (2015) 77–101, https://doi.org/10.1016/j.jcp.2015.07.034.
[41] R. Milani, Compatible Discrete Operator schemes for the unsteady incompressible Navier–Stokes equations, Theses, Université Paris-Est, Dec. 2020, https://tel.archives-ouvertes.fr/tel-03080530.
[42] M. Nazarov, Convergence of a residual based artificial viscosity finite element method, Comput. Math. Appl. (ISSN 0898-1221) 65 (4) (2013) 616–626, https://doi.org/10.1016/j.camwa.2012.11.003.
[43] M. Nazarov, J. Hoffman, Residual-based artificial viscosity for simulation of turbulent compressible flow using adaptive finite element methods, Int. J. Numer. Methods Fluids (ISSN 0271-2091) 71 (3) (2013) 339–357, https://doi.org/10.1002/fld.3663.
[44] M. Nazarov, A. Larcher, Numerical investigation of a viscous regularization of the Euler equations by entropy viscosity, Comput. Methods Appl. Mech. Eng. (ISSN 0045-7825) 317 (2017) 128–152, https://doi.org/10.1016/j.cma.2016.12.010.
[45] T. Ohwada, P. Asinari, Artificial compressibility method revisited: asymptotic numerical method for incompressible Navier-Stokes equations, J. Comput. Phys. 229 (5) (2010) 1698–1723, https://doi.org/10.1016/j.jcp.2009.11.003, ISSN 0021-9991, 1090-2716.
[46] M.A. Olshanskii, A. Reusken, Grad-div stabilization for Stokes equations, Math. Comput. (ISSN 0025-5718) 73 (248) (2004) 1699–1718, https://doi.org/10.1090/S0025-5718-03-01629-6.
[47] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, https://doi.org/10.2514/6.2006-112, https://arc.aiaa.org/doi/abs/10.2514/6.2006-112.
[48] J.-H. Pyo, J. Shen, Gauge-Uzawa methods for incompressible flows with variable density, J. Comput. Phys. (ISSN 0021-9991) 221 (1) (2007) 181–197, https://doi.org/10.1016/j.jcp.2006.06.013.
[49] R. Ramani, J. Reisner, S. Shkoller, A space-time smooth artificial viscosity method with wavelet noise indicator and shock collision scheme, part 1: the 1-D case, J. Comput. Phys. (ISSN 0021-9991) 387 (2019) 81–116, https://doi.org/10.1016/j.jcp.2019.02.049.
[50] J. Reisner, J. Serencsa, S. Shkoller, A space-time smooth artificial viscosity method for nonlinear conservation laws, J. Comput. Phys. (ISSN 0021-9991) 235 (2013) 912–933, https://doi.org/10.1016/j.jcp.2012.08.027.
[51] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. (ISSN 1064-8275) 14 (2) (1993) 461–469, https://doi.org/10.1137/0914028.
[52] J. Shen, On a new pseudocompressibility method for the incompressible Navier-Stokes equations, Appl. Numer. Math. (ISSN 0168-9274) 21 (1) (1996) 71–90, https://doi.org/10.1016/0168-9274(95)00132-8.
[53] D. Silvester, H. Elman, D. Kay, A. Wathen, Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow, in: Numerical Analysis 2000, vol. VII, Partial Differ. Equ. 128 (2001) 261–279, https://doi.org/10.1016/S0377-0427(00)00515-X.
[54] V. Stiernström, L. Lundgren, M. Nazarov, K. Mattsson, A residual-based artificial viscosity finite difference method for scalar conservation laws, J. Comput. Phys. (ISSN 0021-9991) 430 (2021) 110100, https://doi.org/10.1016/j.jcp.2020.110100.
[55] R. Temam, Une méthode d'approximation de la solution des équations de Navier-Stokes, Bull. Soc. Math. Fr. (ISSN 0037-9484) 96 (1968) 115–152, http://www.numdam.org/item?id=BSMF_1968_96_115_0.
[56] I. Tominec, M. Nazarov, Residual viscosity stabilized RBF-FD methods for solving nonlinear conservation laws, J. Sci. Comput. (ISSN 0885-7474) 94 (1) (2023) 14, https://doi.org/10.1007/s10915-022-02055-8.
[57] R. Vilela de Abreu, N. Jansson, J. Hoffman, Computation of aeroacoustic sources for a Gulfstream G550 nose landing gear model using adaptive FEM, in: Special Issue for ICMMES-2014, Comput. Fluids (ISSN 0045-7930) 124 (2016) 136–146, https://doi.org/10.1016/j.compfluid.2015.10.017, https://www.sciencedirect.com/science/article/pii/S0045793015003515.
[58] D. Wang, S.J. Ruuth, Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations, J. Comput. Math. (ISSN 0254-9409) 26 (6) (2008) 838–855.
[59] Z. Wang, M.S. Triantafyllou, Y. Constantinides, G.E. Karniadakis, An entropy-viscosity large eddy simulation study of turbulent flow in a flexible pipe, J. Fluid Mech. (ISSN 0022-1120) 859 (2019) 691–730, https://doi.org/10.1017/jfm.2018.808.
[60] J. Wu, J. Shen, X. Feng, Unconditionally stable Gauge-Uzawa finite element schemes for incompressible natural convection problems with variable density, J. Comput. Phys. (ISSN 0021-9991) 348 (2017) 776–789, https://doi.org/10.1016/j.jcp.2017.07.045.
[61] L. Yang, S. Badia, R. Codina, A pseudo-compressible variational multiscale solver for turbulent incompressible flows, Comput. Mech. (ISSN 0178-7675) 58 (6) (2016) 1051–1069, https://doi.org/10.1007/s00466-016-1332-9.