Mathematical Modelling and Analysis
Volume 20 Number 2, March 2015, 232–260
http://dx.doi.org/10.3846/13926292.2015.1021395

Publisher: Taylor&Francis and VGTU
http://www.tandfonline.com/TMMA
ISSN: 1392-6292
eISSN: 1648-3510

Numerical Solution of the Time-Dependent Navier–Stokes Equation for Variable Density–Variable Viscosity. Part I

Owe Axelsson[c,a], Xin He[b] and Maya Neytcheva[a]

[a] *Department of Information Technology, Uppsala University*
Box 337, 751 05 Uppsala, Sweden
[b] *Delft Institute of Applied Mathematics, Delft University of Technology*
2628CD Delft, The Netherlands
[c] *Institute of Geonics, AVCR, Ostrava*
Studentska 1768, 708 00 Ostrava-Poruba, Czech Republic

E-mail: owe.axelsson@it.uu.se

E-mail(*corresp.*): X.He-1@tudelft.nl

E-mail: maya.neytcheva@it.uu.se

Received June 3, 2014; revised February 1, 2015; published online March 15, 2015

Abstract. We consider methods for the numerical simulations of variable density incompressible fluids, modelled by the Navier–Stokes equations. Variable density problems arise, for instance, in interfaces between fluids of different densities in multiphase flows such as appearing in porous media problems. We show that by solving the Navier–Stokes equation for the momentum variable instead of the velocity the corresponding saddle point problem, arising at each time step, no special treatment of the pressure variable is required and leads to an efficient preconditioning of the arising block matrix. This study consists of two parts, of which this paper constitutes Part I. Here we present the algorithm, compare it with a broadly used projection-type method and illustrate some advantages and disadvantages of both techniques via analysis and numerical experiments. In addition we also include test results for a method, based on coupling of the Navier–Stokes equations with a phase-field model, where the variable density function is handled in a different way.

The theory including stability bounds and a second order splitting method is dealt with in Part II of the study.

Keywords: Navier–Stokes equations, variable density, variable viscosity, operator splitting methods, regularization, phase-field model, finite elements, iterative methods, preconditioning.

AMS Subject Classification: 65F10; 65F08; 65N30.

1 Introduction

Variable density problems arise in many complex fluid flow processes of current interest, and have been studied intensively via numerical simulations. As an example, which ranges over a large number of difficulties, we mention the general circulation model, dealing with the coupled water–atmosphere dynamics. In such a model, both the density and the viscosity of the water and the air are varying. The water is volume-wise incompressible while the air is compressible and the processes are influenced by numerous other factors, e.g. temperature, salinity, moisture etc., leading to a system where high instabilities, such as turbulence, occur.

Important variable density problems arise also in laminar flows and the work, presented in this article, deals primarily with the numerical solution of such models. Examples of the latter type are the variable density ground water flow phenomena, which have been intensively studied in the last decades. There, density-driven flow occurs and the effect of variable density becomes significant, in particular when a fluid of high density overlays a fluid of lower density (cf., e.g., [41]). Density-driven flows are of fundamental importance also when solving transport problems in porous media (see, e.g., [13]).

Another class of variable density problems arise in modelling of the interaction of several phases, for example droplet impact onto a solid or liquid surface, accurate tracking of interface surfaces between fluids of different density in multiphase flow problems, etc.

Due to its high complexity and nonlinearity, the underlying coupled system of partial differential equations is usually solved using some form of operator splitting scheme, see e.g. [34].

The research on numerical schemes for approximating viscous incompressible flow problems has been and is very active since the 1960s, when computational fluid dynamics was launched. The literature is immense and nearly inexhaustible. Without the ambition to present a comprehensive view on the numerical schemes for time-dependent NS problems and the solution of the arising algebraic systems we mention some particularly often used approaches, emphacising the most important factors that usually steer the choice of one or another numerical technique, namely, accuracy of the approximation, stability of the time discretization and the computational burden of the corresponding numerical solution procedure.

Numerical schemes for time-dependent NS equations are usually based on a semi-discretization in time, followed by a spatial discretization at each time step, e.g. the finite element method used in this paper. Typically, the methods are classified as fully implicit, semi-implicit and explicit/implicit, with various specific details and interpretations.

The fully implicit methods have as main advantages the unconditional stability and the exact fulfillment of the incompressibility constraint. At the same time these methods are computationally most expensive. Using a backward Euler differencing for the time derivative yields a nonlinear problem that has to be solved at each time step. To this end one can use either the Newton iteration or the Oseen iteration and, if the time step is not too large, the solution

in the previous time step should be a good initial guess for the next nonlinear iteration, speeding up the nonlinear converge. The backward Euler method provides only first order accuracy in time. To achieve a higher order accuracy in time one can use the Crank-Nicholson method. For a detailed theoretical assessment of that method in the context of constant density nonstationary NS we refer to [27], where second order error estimates locally in time are proven. As the method is only A-stable, it achieves a second order accuracy, but instabilities may not be sufficiently damped. As a way to overcome this, in [27] the Crank–Nicolson method is combined with a sequence of implicit Euler steps.

The straightforward use of forward Euler differencing for the time derivative yields an explicit time discretization scheme that is computationally much cheaper than the implicit methods. However, due to severe restrictions on the time step in order to satisfy the stability constrains, it is hardly ever used in pure form the present context. More to that, it is only first order accurate in time and, in addition, the incompressibility constraint cannot be satisfied exactly. Still, the ease of computing steadily attracts attention to the explicit schemes. As an example, a combination of implicit and explicit Euler schemes has been proposed in [26] that utilizes the implicit scheme in time for the linear term in the momentum equation and the explicit scheme in time for the nonlinear term. The method handles both smooth and discontinuous initial data, it is shown to be stable, and under certain condition on the stability constant, the method achieves the optimal error estimates.

The class of the semi-implicit methods is most diverse. The idea is to reduce the computational complexity at each time step by linearizing the nonlinear problem using the "frozen coefficient" approach in a similar way as is done in the Oseen problem. The method, utilized in this paper is of semi-implicit character. The idea to treat parts of the NS system using different numerical schemes has been exploited in many studies, leading to a large diversity of approximation techniques, among which are the fractional-step methods or fractional-step projection methods. These originate in the classical works of Chorin [11] and Temam [42] and through the years have been further developed in various directions: (i) fractional-step (or splitting methods) for the evolutionary equation, cf., e.g. [8] and the references therein; (ii) various projection-type methods, such as pressure projection methods [21, 24, 38], velocity-correction projection methods [25], scalar and vector penalty-projection methods [2]. The main idea of the projection methods is first to compute a velocity field without taking into account incompressibility, and then perform a pressure correction, which is a projection back to the subspace of solenoidal (divergence-free) vector fields. In those schemes, both types of time discretizations, explicit and implicit have been utilized.

As an example of a method of type (i), we refer to the so-called *fractional-step-θ* method that is a scheme of second order accuracy in time and is strongly A-stable with small dissipation. The method originates in [18] for convection-diffusion problems and is further developed and used for time-dependent flows. The method is second order accurate in time. When implementing it, the convection operator is treated separately from the diffusion operator, and the

arising hyperbolic convective problem is stabilized appropriately. More details can be found in, e.g., [12, 35, 44].

An implicit fractional step method, though ensuring only first order accuracy in time is reported in [8]. The scheme circumvents the need of applying unphysical boundary conditions for the pressure by including a diffusion term in the incompressibility step and with the price of solving a saddle point system at each iteration.

The class of the projection methods (ii) deserves a particular attention. The projection methods are attractive thanks to the lesser computational cost they require. Some recent works show that they may be efficiently implemented on high performance computer platforms [20]. However they have some insurmountable disadvantages. As already mentioned, the projection methods enforce artificial boundary conditions for the pressure, which are unphysical and can generate a numerical boundary layer. This effect is described in, e.g., in [19]. In [28] it is shown that the (scalar) incremental projection-type methods of the above type, in addition to the already mentioned artificial pressure boundary layers, exhibit large splitting errors with respect to the time and space discretization error and poor convergence for non-homogeneous flows. In a series of papers, for instance in [2], see also the references therein, it is argued that vector penalty projection methods are capable of overcoming the above drawbacks of their scalar counterparts at the price of solving saddle point systems of augmented Lagrangian type. We stress that the accuracy of the projection type of methods depends on the method parameters, which is clearly illustrated by the numerical experiments performed in this paper.

In contrast to the vast literature and variety of numerical schemes for nonstationary constant density NS problems, the experience with variable density NS is rather scarce. We refer explicitly to [24] for a projection-type scheme, to [2], for a vector penalty projection scheme, that is first order in time, second order in space and the numerical solution procedure includes saddle point systems, and to [1], where the space discretization is done using finite Differences and the scheme utilized a hierarchy of adaptively refined meshes in order to achieve a second order accurate in time projection-type method.

As already stated, a major part of the numerical simulation of nonstationary flow models is the ability to efficiently solve the nonlinear time-dependent variable density, variable-viscosity Navier–Stokes (NS) equations. NS is either combined with a mass balance equation for the density or with another equation, which describes the evolution of the interfaces between the different phases, such as the Cahn-Hilliard (CH) equations (cf., e.g., [4, 46]).

In this work we deal mostly with NS, coupled to the mass balance equation for the density, but we also include some numerical experiments within the diffuse interface approach, where NS is coupled with CH. Briefly, CH belongs to the class of *diffuse interface* methods to solve multiphase problems, where it is assumed that the interfaces have some nonzero thickness and the problem parameters vary continuously between the different flow phases. The alternative approach, the *sharp interface* methods, assumes that the interfaces have zero thickness and the aim is to resolve those exactly, taking into account the discontinuities of the problem coefficients across those interfaces. This requires

the use of proper interface boundary conditions. The advantages and disadvantages of both approaches are discussed in a vast amount of related literature (see, e.g., the references in [48]). We note that when using CH, there is no need to solve a separate equation for the density, since a special *'concentration'* variable determines automatically the relative portion of the masses and an interface of chosen thickness.

Our main focus is on the numerical solution techniques, their robustness with respect to problem, discretization and method parameters, and computational efficiency. In Section 2 we formulate the coupled Navier–Stokes-density equations to be solved and give some stability bounds. Thereby we advocate the use of a coupled system involving the momentum as primal variable instead of the velocity. In Section 3 we present further arguments for use of the momentum and the operator splitting and linearization method used. The derivation of an error estimate for the splitting error, based on these approximations is dealt with in Part II of this paper. Some of the derivations are presented in [5]. In Section 4 we discuss the necessary stabilization methods for the time-stepping scheme. Next, in Section 5 we propose a preconditioning technique for the arising block matrix system. Section 6 contains numerical results. We conclude the paper with some remarks, such as the possibility to couple with elasticity equations when solving porous media problems.

2 The Navier–Stokes Equations with Variable Density

Density-dependent incompressible NS equations are used to model the motion of a viscous incompressible non-homogeneous fluid flow. Let Ω be a smooth, bounded and connected domain in $\mathbb{R}^d$, $d = 2, 3$ and let $\mathbf{u}$, p, ρ, μ denote respectively the velocity vector, pressure, density and viscosity of a fluid in Ω.

2.1 Formulation of the coupled system

The variation of the variables $\mathbf{u}$ and ρ is described by the Navier–Stokes equations, augmented with an additional advection equation for the density. The fluid is assumed to be incompressible. As described, e.g., in [33], see also [23], the so obtained non-stationary incompressible equations take the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.1}$$

$$\frac{\partial}{\partial t}(\rho\, \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \big(\mu D(\mathbf{u})\big) + \nabla\, p = \rho\, \mathbf{f}, \tag{2.2}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.3}$$

in $\Omega \times (0, \infty)$. Here, $D(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the symmetrized deformation tensor, $\rho\, \mathbf{f}$ is a force function (per unit volume), where typically $\mathbf{f} = \mathbf{g}$, $\mathbf{g}$ being the gravity field. The incompressibility of the fluids is understood in the sense that density cannot be changed by changes in pressure (cf. e.g. [43]). The questions regarding existence, uniqueness and regularity of the solution of the model (2.1)–(2.3) have been addressed in various studies by, e.g., Ladyzhenskaya (1968), Desjardins (1997), Lions (1998), Danchin (2003, 2006).

For some more details the reader is referred to the more recent study [49] and the references therein.

Using the incompressibility of the fluid, we can rewrite the term $\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u})$ in (2.2) as follows. For the ith component it holds

$$\sum_j \partial_j(\rho u_i u_j) = \sum_j \big(u_j\partial_j(\rho u_i) + \rho u_i \partial_j u_j\big) = (\mathbf{u}\cdot\nabla)(\rho u_i) + \rho u_i \nabla\cdot\mathbf{u},$$

so $\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = (\mathbf{u}\cdot\nabla)(\rho\mathbf{u})$.

We assume that viscosity depends on density as some given continuously differentiable, positive function $\mu(\rho)$ in the interval $[0,\infty)$. It follows then from (2.1) and (2.3) that a similar equation as for the density holds also for the viscosity. Namely, we have,

$$\frac{\partial\mu}{\partial t} + \mathbf{u}\cdot\nabla\mu = \frac{\partial\mu}{\partial\rho}\left(\frac{\partial\rho}{\partial t} + \mathbf{u}\cdot\nabla\rho\right) = \frac{\partial\mu}{\partial\rho}\left(\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u})\right) = 0. \tag{2.4}$$

Hence, we can add $\frac{\partial\mu}{\partial t} + \mathbf{u}\cdot\nabla\mu = 0$ or $\frac{\partial\mu}{\partial t} + \nabla\cdot(\mu\mathbf{u}) = 0$ to the above set of equations. However, we use the relation $\mu = \mu(\rho)$ instead. Equations (2.2) takes then the form

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + (\mathbf{u}\cdot\nabla)(\rho\mathbf{u}) - \nabla\cdot\big(\mu D(\mathbf{u})\big) + \nabla p = \rho\,\mathbf{f}. \tag{2.5}$$

Introducing the momentum variable $\mathbf{v} = \rho\mathbf{u}$ and using the relation

$$\nabla\cdot\mathbf{v} \equiv \nabla\cdot(\rho\mathbf{u}) = \rho\nabla\cdot\mathbf{u} + \mathbf{u}\cdot\nabla\rho = \mathbf{u}\cdot\nabla\rho,$$

equations (2.1)–(2.3) take the form

$$\frac{\partial\rho}{\partial t} + \mathbf{u}\cdot\nabla\rho = 0, \tag{2.6}$$

$$\frac{\partial}{\partial t}(\mathbf{v}) + (\mathbf{u}\cdot\nabla)(\mathbf{v}) - \nabla\cdot\big(\mu D(\mathbf{u})\big) + \nabla p = \rho\,\mathbf{f}, \tag{2.7}$$

$$\nabla\cdot\mathbf{v} - \mathbf{u}\cdot\nabla\rho = 0, \tag{2.8}$$

where $\mathbf{u} = \frac{1}{\rho}\mathbf{v}$. The initial conditions for the system (2.6)–(2.8) are assumed to be

$$\rho|_{t=0} = \rho_0, \qquad (\rho\mathbf{u})|_{t=0} = \mathbf{v}_0, \qquad \mu|_{t=0} = \mu_0 = \mu(\rho_0).$$

The assumptions for the boundary conditions follow those from [24], namely, $\mathbf{u}|_{\Gamma_D} = \mathbf{b}$, $\mathbf{u}\cdot\mathbf{n}|_{\Gamma_N} = 0$, $\rho|_{\Gamma_{in}} = a$, $a > 0$, $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$. We note that since the advection equation (2.1) has been rewritten in the form (2.6), i.e., as a first order hyperbolic equation, it follows that the boundary conditions for ρ are given at a possible inflow boundary, $\Gamma_{in} = \{\mathbf{x}\in\Gamma,\ \mathbf{u}\cdot\mathbf{n} < 0\}$.

We note, that Γ_{in} can vary with time. In our analysis, we assume that no penetration boundary condition $\mathbf{u}\cdot\mathbf{n}|_\Gamma = 0$ holds, thus, $\Gamma_{in} = \emptyset$.

As for NS with constant density, there is no need to impose any initial or boundary conditions for the pressure variable. The pressure is uniquely

defined only up to a constant term. To make it unique, one normally imposes the additional constraint $\int_\Omega p\, d\Omega = 0$.

We assume that the given data are such that $\rho(t)$ and $\mathbf{u}(\cdot, t)$ belong to some Sobolev space, which is smoother than $H^1(\Omega)$. In the sequel we assume that μ is a known function of ρ.

For existence, uniqueness of the solution and stability estimates, see [33], [23] and [34].

2.2 Stability properties of the underlying equations

With reference to the discussion on non-zero thickness of interfaces we assume that the density is described by a continuous function. We present some relations, which turn out to be useful for showing stability of the operator splitting scheme. Usually such relations are shown to hold for the numerical solution, however we present their analogues for the continuous solution. Similar estimates can be found in the related literature, see, e.g., [33], but they are not identical.

By the assumption of the boundary condition and incompressibility, it follows that

$$\int_\Omega \rho\mathbf{u}\cdot\nabla\rho\, d\Omega = \int_\Omega \frac{1}{2}\mathbf{u}\cdot\nabla\big(\rho^2\big)\, d\Omega = -\int_\Omega \frac{1}{2}\nabla\cdot\mathbf{u}\,\rho^2\, d\Omega + \int_{\partial\Omega}\frac{1}{2}\mathbf{u}\cdot\mathbf{n}\rho^2\, ds = 0.$$

Hence, a variational formulation of the mass balance equation leads to

$$\frac{1}{2}\frac{d}{dt}\left(\int_\Omega \rho^2\, d\Omega\right) = \int_\Omega \left[\rho\frac{\partial\rho}{\partial t} + \rho\mathbf{u}\cdot\nabla\rho\right] d\Omega = 0,$$

that is,

$$\big\|\rho(\cdot,t)\big\|_0 = \|\rho_0\|_0, \quad t > 0, \tag{2.9}$$

where $\|\cdot\|_0$ denotes the $L^2(\Omega)$ norm. Hence, in this norm, the density is constant in time. In Part II of the paper it is shown that the more general relation $\|\rho(t)\|_q = \|\rho_0\|_q, t > 0, 2 \le q \le \infty$ holds, where $\|\rho(t)\|_q = \big(\int_\Omega \rho^q\big)^{\frac{1}{q}}$.

Using a variational formulation of the momentum equation (2.7), we find

$$\int_\Omega \left[\frac{\partial}{\partial t}(\mathbf{v})\cdot\mathbf{u} + (\mathbf{u}\cdot\nabla)(\mathbf{v})\cdot\mathbf{u} - \nabla\cdot(\mu D(\mathbf{u}))\cdot\mathbf{u} + \nabla p\cdot\mathbf{u}\right] d\Omega = \int_\Omega \rho\,\mathbf{f}\cdot\mathbf{u}\, d\Omega.$$

Assume that

$$\frac{\mu(\rho)}{\rho} \ge \alpha, \quad \alpha > 0, \tag{2.10}$$

and that $\int_\Omega |\nabla\mathbf{u}|^2\, d\Omega \ge \beta\int_\Omega |\mathbf{u}|^2\, d\Omega$ for some $\beta > 0$. Then, as shown in Part II of this paper and in [5], from the latter expression we obtain

$$\begin{aligned}\big\|\rho(\cdot,t)^{1/2}\mathbf{u}(\cdot,t)\big\|_0^2 &\le e^{-\alpha\beta\, t}\big\|\rho_0^{1/2}\mathbf{u}_0\big\|_0^2 + C\int_0^t e^{-\alpha\beta(t-s)}\, ds \\ &\le e^{-\alpha\beta\, t}\big\|\rho_0^{1/2}\mathbf{u}_0\big\|_0^2 + \frac{C}{\alpha\beta}\big(1 - e^{-\alpha\beta t}\big) \\ &\le e^{-\alpha\beta\, t}\big\|\rho_0^{1/2}\mathbf{u}_0\big\|_0^2 + \frac{C}{\alpha\beta}.\end{aligned} \tag{2.11}$$

If $t = O(\tau)$, i.e., if we perform just one time step, the second term above is bounded by $C\tau = |O(\tau)|$.

Thus, estimates (2.9)–(2.11) show unconditional stability of the solution of the continuous problem. Furthermore, it is seen that, under the assumption (2.10), the influence of the initial condition decreases exponentially to zero with increasing time.

3 Time Discretization, Operator Splitting Scheme and Linearization

The system of equations we consider in the sequel is the following

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0, \tag{3.1}$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{v} - \nabla \cdot \big(\mu D(\mathbf{u})\big) + \nabla p = \rho\, \mathbf{f}, \tag{3.2}$$

$$\nabla \cdot \mathbf{v} - \mathbf{u} \cdot \nabla \rho = 0 \tag{3.3}$$

with $\mathbf{u} = \frac{1}{\rho}\mathbf{v}$ and appropriate boundary and initial conditions.

Equation (3.3) replaces the classical incompressibility constraint $\nabla \cdot \mathbf{u} = 0$. We refer to (3.3) as the divergence constraint for the momentum.

Equations (3.1)–(3.3) are to be solved on a sequence of time intervals $[t_k, t_k + \tau_k]$, $k = 0, 1, \ldots$, where $t_0 = 0$. The time steps τ_k may vary. The arising systems are nonlinear. Newton's method is applicable but less convenient to use for NS, in particular for convection-dominated (high Reynolds number) flows. Among the most prohibitive factors is the repeated recomputation of the nonlinear Jacobian. For practical reasons it is better to use some form of linearization through an operator splitting method.

The splitting method used in this paper is motivated by two facts. First, since in general the initial pressure is not known, we keep the momentum equation (3.2) and divergence constraint (3.3) in their coupled form intact, which also enables the computation of the pressure without use of (artificial) pressure boundary conditions that are shown to destroy the accuracy of the computed pressure along the boundary [21]. Therefore, instead we split off the advection equation for the density (3.1), which can be handled separately. Hence, the momentum equation and the divergence constraint (3.2)–(3.3), and the advection equation (3.1) are progressed separately at each time step.

Second, for reasons of stability and to avoid the need to use very small time steps, we choose to use a stable semi-implicit time integration method, preferably of second or higher order of accuracy.

To ensure that the unconditional stability of the continuous solution holds also for the discrete solution, in [34] a method involving two projections per time step is proposed. We show that the use of such projections can be avoided if we use the equations for the momentum, $\mathbf{v} = \rho\mathbf{u}$. Note that in multiphase flows we treat interfaces between different media as having a certain thickness governed by a diffuse interface model, which means that all variables, including velocity and density, are continuous. In this way, the introduced momentum variable $\mathbf{v}$ is also continuous. In a single flow, since in general the

velocity slows down when the flow enters a thicker medium, one can expect that $\mathbf{v}$ has a smoother behaviour, i.e. less strong relative variations, than $\mathbf{u}$ and can therefore be more accurately approximated numerically. If $\mathbf{f} = \mathbf{0}$, indeed, inequality (2.11) indicates that the L_2-norm of $\mathbf{u}(\cdot, t)$ slows down if $\rho(\cdot, t)$ increases. Finally, note, that by solving a saddle point system, keeping equations (2.2)–(2.3), respectively (3.2)–(3.3) coupled, artificial pressure boundary conditions are avoided.

The equations can be solved by use of an operator splitting method such that at each time interval we first compute the density, and then solve the momentum equation together with the divergence constraint. Furthermore, we linearize the equations using a "frozen coefficient" approach in a similar way as is done in the Oseen problem. In general, this method leads to a splitting error of first order. In Part II of this paper we present an alternative method which has a splitting error of second order, cf. [5].

3.1 Time discretization scheme

To introduce notation and formalize the proposed splitting scheme, we describe the three-level backward differentiation formula (BDF2) algorithm that is of second-order of accuracy in time for the density and momentum equations.

We find the approximate sequences $\{\rho^n, \mathbf{v}^n, \mathbf{u}^n, p^n\}_{n=0,1,\cdots,N}$ with initial conditions $(\rho^0 = \rho_0, \mathbf{v}^0 = \mathbf{v}_0, \mathbf{u}^0 = \mathbf{v}_0/\rho_0)$ and for all time steps n from 0 to $N-1$.

The BDF2 scheme for constant time steps progresses as follows. First, one initializes $(\rho^0, \mathbf{v}^0, \mathbf{u}^0)$, and computes $(\rho^1, \mathbf{v}^1, \mathbf{u}^1, p^1)$ by using one step of the first-order Backward Euler scheme (for details we refer to [5]). Then for $n \geq 1$, we proceed as follows.

Algorithm 1 (BDF2)

A1-1: Set the linearly extrapolated velocity at time level $n+1$ as

$$\mathbf{u}^\star = 2\mathbf{u}^n - \mathbf{u}^{n-1}.$$

A1-2: Compute ρ^{n+1} from

$$\frac{3\rho^{n+1} - 4\rho^n + \rho^{n-1}}{2\tau} + \mathbf{u}^\star \cdot \nabla \rho^{n+1} + \frac{1}{2}\rho^{n+1} \nabla \cdot \mathbf{u}^\star = 0. \tag{3.4}$$

A1-3: Compute $\mathbf{v}^{n+1}$ and p^{n+1} from

$$\frac{3\mathbf{v}^{n+1} - 4\mathbf{v}^n + \mathbf{v}^{n-1}}{2\tau} + (\mathbf{u}^\star \cdot \nabla)\mathbf{v}^{n+1} - \nabla \cdot \left(\mu^{n+1} D\left(\frac{\mathbf{v}^{n+1}}{\rho^{n+1}} \right) \right) + \nabla p^{n+1} = \rho^{n+1} \mathbf{f}^{n+1}, \tag{3.5}$$

$$\nabla \cdot \mathbf{v}^{n+1} - \tau^2 \Delta p^{n+1} = \mathbf{u}^\star \cdot \nabla \rho^{n+1}. \tag{3.6}$$

A1-4: Finally, recover the velocity $\mathbf{u}^{n+1}$ *as* $\mathbf{u}^{n+1} = \mathbf{v}^{n+1}/\rho^{n+1}$.

In (3.4) we have added the stabilization term $\frac{1}{2}\rho^{n+1}\nabla\cdot\mathbf{u}^n$ to make the semi-implicit scheme unconditionally stable. The added term is zero if the sequence of velocities $\mathbf{u}^n{}_{n=0,\dots,N}$ satisfies the divergence free condition. In (3.6) we have additionally regularized the problem by adding the term $-\tau^2\Delta p^{n+1}$, where $-\Delta$ is the negative Laplace operator. The purpose of this regularization is to avoid unphysical oscillations in pressure. Details of all stabilisations used in this paper are presented in Section 4, including some stabilization for the momentum equation (3.5) in convection-dominant (high Reynolds number) problems. We remark, if $\nabla\cdot\mathbf{u} = 0$, that since the average,

$$\mathbf{u}\cdot\nabla\rho + \frac{1}{2}\rho\nabla\cdot\mathbf{u} = \frac{1}{2}\big(\mathbf{u}\cdot\nabla\rho + \nabla\cdot(\rho\mathbf{u})\big),$$

i.e., the average of the conservative and convective terms equals the corresponding expression in (3.4), this stabilization leads to a skew symmetric form, since if $\nabla\cdot\mathbf{u} = 0$,

$$b(\rho,\mathbf{u}) = \int_\Omega \mathbf{u}\cdot\nabla\rho = -\int_\Omega \rho\nabla\cdot\mathbf{u} = -b(\mathbf{u},\rho),$$

see [31] for further details.

If the extrapolated velocity $\mathbf{u}^\star$ satisfies the incompressibility constraint, i.e., $\nabla\cdot\mathbf{u}^\star = 0$, the advection equation (3.4) and the coupled NS equations (3.5)–(3.6) are of second order accuracy in time due to the BDF2 time discretization scheme.

3.2 Operator splitting and related splitting error

As already stated, equations (3.1)–(3.3) will be solved by use of an operator splitting method, where at each time interval we first compute the density (3.1) and then solve the momentum equation (3.2) together with the divergence constraint (3.3). Furthermore, we linearize the equations using a "frozen coefficient", i.e., a Picard iteration approach in a similar way as is done in the Oseen problem.

To this end, consider a time interval $(t_0, t_0+\tau)$. Denote the already computed values at t_0 by $\mathbf{u}_0$, μ_0 and ρ_0 and let $\widetilde{\mathbf{u}}$, $\widetilde{\mathbf{v}}$, $\widetilde{p}$, $\widetilde{\mu}$ and $\widetilde{\rho}$ be the correspondingly computed values after linearization and operator splitting. Then, for $t\in(t_0,t_0+\tau)$ we solve

$$\frac{\partial\widetilde{\rho}}{\partial t} + \nabla\cdot(\mathbf{u}_0\widetilde{\rho}) = 0, \tag{3.7}$$

$$\frac{\partial\widetilde{\mathbf{v}}}{\partial t} + \mathbf{u}_0\cdot\nabla\widetilde{\mathbf{v}} - \nabla\cdot\big(\mu_0 D(\widetilde{\mathbf{u}})\big) + \nabla\widetilde{p} = \widetilde{\rho}\,\mathbf{f}, \tag{3.8}$$

$$\nabla\cdot\widetilde{\mathbf{v}} - \mathbf{u}_0\cdot\nabla\widetilde{\rho} = 0. \tag{3.9}$$

Here we solve first (3.7) and then (3.8)–(3.9). Further, $\widetilde{\mathbf{u}} = \widetilde{\rho}^{-1}\widetilde{\mathbf{v}}$.

The relation $\widetilde{\mu} = \mu(\widetilde{\rho})$ is assumed to be known. Solving equations (3.7)–(3.9) does not provide a divergence-free velocity field. To ensure that, we use the mass conservation equation for the density once more to solve

$$\frac{\partial \widetilde{\widetilde{\rho}}}{\partial t} + \nabla \cdot (\widetilde{\mathbf{u}}\widetilde{\widetilde{\rho}}) = 0,$$

and combine this solution with already obtained values in the following way. Assuming that we use quadratic basis functions for $\mathbf{v}$, we can form a linear combination of the quadratic polynomial interpolants of the functions $\frac{1}{\rho_0}\widetilde{\mathbf{v}}$, $\frac{1}{\widetilde{\rho}}\widetilde{\mathbf{v}}$ and $\frac{1}{\widetilde{\widetilde{\rho}}}\widetilde{\mathbf{v}}$, to make in each element $\widetilde{\widetilde{\mathbf{u}}} = (\alpha\frac{1}{\rho_0} + \beta\frac{1}{\widetilde{\rho}} + \gamma\frac{1}{\widetilde{\widetilde{\rho}}})\widetilde{\mathbf{v}}$ satisfy $\nabla \cdot \widetilde{\widetilde{\mathbf{u}}} = 0$. This leads to a system with three equations to solve for the coefficients α, β and γ.

To derive error estimates of the operator splitting and linearization errors, we assume that the error in the velocity, $\mathbf{u} - \widetilde{\widetilde{\mathbf{u}}}$ is divergence-free. See Part II for details.

It can be shown that

$$\frac{\partial}{\partial t}(\rho - \tilde{\rho}) - \nabla \cdot (\mathbf{v} - \tilde{\mathbf{v}}) = 0.$$

Taking variational formulations of the corresponding basic equations and multiplying with $\mathbf{v} - \tilde{\mathbf{v}}$, $p - \tilde{p}$ and $\rho - \tilde{\rho}$ results in

$$\begin{aligned}
&\int_\Omega \left[\frac{\partial}{\partial t}\left(\frac{1}{\rho}|\mathbf{v} - \widetilde{\mathbf{v}}|^2\right) + 2\mu\big|\nabla(\mathbf{u} - \widetilde{\mathbf{u}})\big|^2\right] d\Omega \\
&\quad \le \int_\Omega \frac{1}{\rho}\Psi(\mathbf{u} - \widetilde{\mathbf{u}}) d\Omega + \int_\Omega \left[\frac{\partial}{\partial t}(\mathbf{v} - \widetilde{\mathbf{v}}) + \mathbf{u} \cdot \nabla(\mathbf{v} - \widetilde{\mathbf{v}})\right] \Phi d\Omega.
\end{aligned}$$

Here

$$\begin{aligned}
\Phi &= \frac{1}{\rho}(\mathbf{v} - \widetilde{\mathbf{v}}) - (\mathbf{u} - \widetilde{\mathbf{u}}) = \left(\frac{1}{\tilde{\rho}} - \frac{1}{\rho}\right)\underline{\mathbf{v}} + \left(\frac{1}{\rho} - \frac{1}{\tilde{\rho}}\right)(\mathbf{v} - \tilde{\mathbf{v}}), \\
\Psi &= -(\mathbf{u} - \mathbf{u}_0) \cdot \nabla \mathbf{v} + (\mathbf{u} - \mathbf{u}_0) \cdot \nabla(\mathbf{v} - \widetilde{\mathbf{v}}) + \nabla \cdot \big((\mu - \mu_0)D(\mathbf{u})\big) - \\
&\quad - \nabla \cdot \big((\mu - \mu_0)D(\mathbf{u} - \widetilde{\mathbf{u}})\big) + (\rho - \rho_0)\mathbf{f}.
\end{aligned}$$

Analogously to the derivation of (2.11) (see [5]) one obtains

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\left\|\frac{1}{\sqrt{\rho(\cdot,t)}}\big|(\mathbf{v} - \widetilde{\mathbf{v}})\big|(\cdot,t)\right\|_0^2\right) + \alpha\beta\big\|\sqrt{\rho(\cdot,t)}\big|(\mathbf{u} - \widetilde{\mathbf{u}})(\cdot,t)\big|\big\|_0^2 \le \text{rhs}, \\
&\quad t_0 < t < t_0 + \tau.
\end{aligned}$$

Let

$$e_n = \left\|\frac{1}{\sqrt{\rho(\cdot,t_n)}}\big|(\mathbf{v} - \widetilde{\mathbf{v}})(\cdot,t_n)\big|\right\|_0 + \big\|(\rho - \widetilde{\rho})(\cdot,t_n)\big\|_0.$$

Then, the following bound holds,

$$e_{n+1} \le e^{-\alpha\beta\tau} e_n + \tau^{1+\xi} C, \quad n = 0, 1, \ldots,$$

where C is a constant, depending on the regularity of the solution and $\mathbf{f}$.

By recursion, this shows that

$$e_{n+1} \le e^{-\alpha\beta\tau} e_0 + C\tau^{1+\xi} \Sigma_{k=0}^{n} e^{-\alpha\beta k\tau} \le e^{-\alpha\beta\tau} e_0 + C\tau^{1+\xi} / \left(1 - e^{-\alpha\beta\tau}\right) \le e^{-\alpha\beta\tau} e_0 + \frac{C}{\alpha\beta} \tau^{\xi},$$

where $T = n\tau$. (A corresponding bound can be proved even if the time steps vary.)

Since by assumption $e_0 = O(\tau^{1+\xi})$, we have then

$$e_n \le \frac{C}{\alpha\beta} \tau^{\xi}.$$

Hence, if the regularity of the solution permits local errors, i.e., τ times residual, of second order, i.e. $\xi = 1$, then the splitting and linearization error is of first order, $O(\tau)$.

A similar bound for pressure p can be obtained. The above bounds are derived for the continuous equations and show that we have full control of the error due to the operator splitting and the linearization during integration over all time intervals.

4 Regularization/Stabilization Issues

Stabilization of the density equation: As seen the density equation (3.4) uses a semi-implicit scheme. In order to make it unconditionally stable, we add an additional term $\frac{1}{2}\rho^{n+1} \nabla \cdot \mathbf{u}^{\star}$ to the left-hand side of (3.4), where $\mathbf{u}^{\star}$ is the linearly extrapolated velocity defined in Algorithm 1. The stabilization term is used due to the semi-implicitness of the scheme and is zero if the sequence of velocity $\mathbf{u}^n{}_{n=0,\ldots,N}$ satisfies the divergence free condition, i,e., $\nabla \cdot \mathbf{u}^n = 0$. This stabilization method is also utilised in [23], where no theoretical analysis is given to show why it works. The appendix of this paper proves that the numerical scheme (3.4) with the stabilization term $\frac{1}{2}\rho^{n+1} \nabla \cdot \mathbf{u}^{\star}$ is unconditionally stable. This can be seen as a discrete version of the stability (conservation) property (2.9) and is one of the novelties in this paper.

There are other ways to stabilize the transport equation (3.4), among them is the so-called *entropy viscosity method* used in [24] and analysed in [22]. The *entropy viscosity method* needs more computational efforts compared to the technique used in this paper.

Stabilization of the momentum equation: For the convection dominant problems (high Reynolds numbers) we need to stabilize the convective term in equation (3.5). Which technique to use and what are the consequences of the stabilization is among the research topics, which have been dealt for years and is still not fully resolved.

In general, the flow is convection-dominated, that is, $(Re\,\mathbf{u})$ takes large values. If the local Peclet number, $Pe_k = Re\, h_k \|\mathbf{u}\|_{\infty,k}$, is larger than one, spurious node-to-node oscillations occur around steep gradients. Various techniques to avoid this can be used. Choosing the mesh size sufficiently small

to satisfy the condition regarding the local Peclet number can lead to very small elements and large linear systems to be solved. Another possibility is to use locally refined meshes where there are steep gradients. In this paper we do not discuss the global or local refinement strategy, and instead use the streamline-diffusion method [15, 30]. This involves adding an additional term $\gamma(\mathbf{u}^n \cdot \nabla \mathbf{u}^{n+1}, \mathbf{u}^n \cdot \nabla V)$ to the left-hand side of the momentum equation as (3.5), where γ is the regularization parameter and V is the test function. The local parameter γ_k is determined based on the local Peclet number defined above as

$$\begin{cases} \gamma_k = \dfrac{1}{2} h_k \left(1 - \dfrac{1}{Pe_k}\right), & \text{if } Pe_k > 1, \\ \gamma_k = 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

When using this stabilization method in Algorithm 1, we use the extrapolated velocity $\mathbf{u}^\star$ to replace $\mathbf{u}^n$ and represent the velocity $\mathbf{u}^{n+1}$ as $\frac{\mathbf{v}^{n+1}}{\rho^{n+1}}$, where ρ^{n+1} has been computed from equation (3.4). In summary, we add

$$\gamma\left(\mathbf{u}^\star \cdot \nabla \frac{\mathbf{v}^{n+1}}{\rho^{n+1}}, \mathbf{u}^\star \cdot \nabla V\right) \tag{4.2}$$

to the left-hand side of (3.5), and γ is computed based on (4.1)

Stabilizing the saddle point problem: For the coupled equations (3.5)–(3.6), space discretization by using the finite element method results in a linear system of the saddle point structure, more details in Section 5. Stabilization is required if the Ladyzhenskaya–Babuška–Brezzi (LBB) condition is not satisfied. We let V_h and P_h denote the discrete subspaces for the momentum $\mathbf{v}$ and pressure p. The LBB stability means that $\inf_{\forall p_h \in P_h} \sup_{\forall \mathbf{v}_h \in V_h} \frac{(p_h \nabla \cdot \mathbf{v}_h)_\Omega}{\|p_h\|_0 \, \|\mathbf{v}_h\|_{H^1(\Omega)}} \geq \beta > 0$ for some β, that does not depend on the space discretization parameter h.

As it turns out, see e.g. [37], also [3], in some cases unphysical pressure oscillations may still occur even though the LBB condition is satisfied. In the numerical experiments of this paper, we choose the stable finite element pair which satisfies the LBB condition and additionally stabilize the saddle point problem in order to avoid pressure oscillations as follows. We add an additional term $-\delta \Delta p^{n+1}$ to the left-hand side of (3.6), forming the perturbed equation at time level $n+1$ as

$$\nabla \cdot \mathbf{v}^{n+1} - \delta \Delta p^{n+1} = \mathbf{u}^\star \cdot \nabla \rho^{n+1}. \tag{4.3}$$

Here Δ is the Laplacian operator. The extrapolated velocity $\mathbf{u}^\star$ and the density ρ^{n+1} on time level $n+1$ have been computed, so the product $\mathbf{u}^\star \cdot \nabla \rho^{n+1}$ is taken as a right-hand side vector. To make the perturbation small, we need δ be close to zero. In this paper we choose $\delta = O(\tau^2)$ as used in equation (3.6), where τ is the time step. We note, that this kind of stabilization method for saddle point problems is quite often used if the LBB condition is not satisfied, or if the LBB condition is insufficient to avoid pressure oscillations (cf. e.g., [3] where a stationary case is considered).

We note that here the stabilization $-\delta\Delta p^{n+1}$ is added not only to avoid pressure oscillations, but also to improve the condition number of the saddle point system arising in the spacial disrectization of (3.5)–(3.6). Numerical experiments, not included in this paper, show that without this stabilization (i.e., $\delta = 0$) it takes many more iterations to solve the saddle point system by using some Krylov subspace methods.

5 Block-Matrix Structure and Preconditioning

After discretizing the coupled equations (3.5)–(3.6) in space with some proper finite element pair, such as the Taylor–Hood elements, we obtain an algebraic system at each time step of the following form

$$\mathcal{A}\begin{bmatrix}\mathbf{v}_h(t+\tau)\\ p_h(t+\tau)\end{bmatrix} = \text{rhs}, \quad \text{where } \mathcal{A} = \begin{bmatrix}A & B^T\\ B & -\tau^2 C\end{bmatrix}. \tag{5.1}$$

Here, $\mathbf{v}_h$, p_h denote the corresponding finite element approximations. The matrix block A has the form $A = O(\tau^{-1})M + E + W$, where M is the velocity mass block matrix, E arises from the discrete diffusion and convection operators and W arises from the streamline-diffusion stabilization method (4.2). The block B denotes the (negative) divergence operator and C is the discrete Laplacian operator. The nonzero sub-block $-\tau^2 C$ is caused due to the stabilization reason (4.3). Clearly, the time step τ can vary between time steps and in this paper we choose it to be constant for simplicity. We refer to [14, 47] for studying the variable time step integration methods.

Due to its large size, systems with $\mathcal{A}$ are solved by a suitable preconditioned iterative method. Among the best known preconditioners for saddle point matrices are those, based on some approximate block factorizations of $\mathcal{A}$. We recall that a well-established class of preconditioners for a general nonsingular saddle point matrix $\mathcal{A}_0 = \left[\begin{smallmatrix}A & B^T\\ B & -C\end{smallmatrix}\right]$, is based on the exact two-by-two block-factorization of $\mathcal{A}_0$, namely, $\mathcal{A}_0 = \left[\begin{smallmatrix}A & 0\\ B & -S_{\mathcal{A}_0}\end{smallmatrix}\right]\left[\begin{smallmatrix}I_1 & A^{-1}B^T\\ 0 & I_2\end{smallmatrix}\right]$, where $S_{\mathcal{A}_0} = C + BA^{-1}B^T$ is the negative Schur complement matrix, and I_1 and I_2 denotes the identity matrices with proper size. To precondition $\mathcal{A}_0$ it suffices to use an approximation of the block lower-triangular factor $\left[\begin{smallmatrix}A & 0\\ B & -S_{\mathcal{A}_0}\end{smallmatrix}\right]$, and the quality of the preconditioner depends on how accurate we solve systems with A and how well we approximate $S_{\mathcal{A}_0}$ (cf. e.g. [6]).

Due to the fact that the problem is time-dependent, the pivot block A includes a mass matrix multiplied by a factor inversely proportional to the time step, which in general improves the condition number of the block when time step is relatively small. Further, if we use a rough stopping criteria for the inner iterations to solve systems with A, it is advisable to use a flexible outer iteration method to solve the system (5.1), such as GCR (c.f. [7] and [45]) or a flexible variant of GMRES [39] which allows a variable preconditioner.

As is well-known, the most difficult problem is to find good quality approximations of the Schur complement matrices. In time-dependent applications, this task turns out to be less difficult than the stationary cases. In this paper, with the assumption that the time step τ is relatively small, the negative Schur

complement $S_{\mathcal{A}}$ of $\mathcal{A}$ in (5.1) can be approximated as follows

$$S_{\mathcal{A}} = \tau^2 C + BA^{-1}B^T \approx B\big(O\big(\tau^{-1}\big)M\big)^{-1}B^T \approx O(\tau)B\widetilde{M}^{-1}B^T,$$

where $\widetilde{M}$ denotes the diagonal of the velocity mass matrix M. The above relation gives us a good approximation of $S_{\mathcal{A}}$, namely, $\widetilde{S}_{\mathcal{A}} = O(\tau)B\widetilde{M}^{-1}B^T$. For LBB stable finite element pairs the matrix B has full rank implying a unique solution of the system (5.1). Further, the matrix $\widetilde{S}_{\mathcal{A}}$ is symmetric and positive definite.

In summary, the precondition $\mathcal{P}$ used in this paper for the coefficient matrix $\mathcal{A}$ in (5.1) is of a form

$$\mathcal{P} = \begin{bmatrix} \widetilde{A} & 0 \\ B & -\widetilde{S}_{\mathcal{A}} \end{bmatrix}, \tag{5.2}$$

where $\widetilde{A}$ denotes an approximation of A defined implicitly through an inner iterative method with a proper stopping tolerance. To solve systems with A and $\widetilde{S}_{\mathcal{A}}$, in this paper we use an algebraic multigrid solution method and more details are presented in Section 6.

6 Numerical Illustrations

We consider two test problems.

Problem 1 [Problem with a known analytic solution]. We solve the variable density Navier–Stokes equations in the square domain $[-0.5, 0.5]^2$, having the following analytical solution,

$$\begin{aligned} &\rho(x,y,t) = 2 + x\cos\big(\sin(t)\big) + y\sin\big(\sin(t)\big), \\ &\mathbf{u}(x,y,t) = \big[-y\cos(t), x\cos(t)\big]^T, \quad p(x,y,t) = \sin(x)\sin(y)\sin(t). \end{aligned} \tag{6.1}$$

These functions satisfy the mass conservation equation. Correspondingly, the right-hand side of the momentum equation has the form

$$\rho\mathbf{f} = \begin{bmatrix} \rho(x,y,t)(y\sin(t) - x\cos(t)^2) + \cos(x)\sin(y)\sin(t) \\ -\rho(x,y,t)(x\sin(t) + y\cos(t)^2) + \sin(x)\cos(y)\sin(t) \end{bmatrix}.$$

This problem has been used as a benchmark test in several other related works, such as [24].

Problem 2 [Rayleigh–Taylor instability]. We compute the development of a Rayleigh- Taylor instability (RTI) in the rectangular domain $[0,1] \cup [0,4]$, consisting of two immiscible liquids. At $t = 0$ the heavier liquid is located above the lighter, and for $t > 0$, the system is driven by the action of the downward gravity force, i.e., the force term in the momentum equation is directed downward and is equal to $\rho\,\mathbf{g}$.

Both problems are discretized in space using a Q_2-Q_2-Q_1 finite element discretization for the density-velocity-pressure. Here Q_1 and Q_2 stand for 'bilinear' and 'biquadratic' basis functions, correspondingly. The FEM pair Q_2-Q_1

is one of the Taylor–Hood elements and is LBB-stable. All results in this paper are carried out in `Matlab` 7.13 (R2011b), and performed on a Linux-64 platform with 4 Intel(R) Core i5 CPUs, 660@3.33GHz. The reported execution time is in seconds.

Table 1. Time discretization error, BDF2.

	Velocity		Density		Pressure	
τ	Inf norm	Rate	Inf norm	Rate	Inf norm	Rate
0.0312	3.28e−4	3.96	4.66e−4	4.16	1.38e−3	3.81
0.0156	8.28e−5	3.96	1.12e−4	4.34	3.63e−4	3.85
0.0078	2.09e−5	3.93	2.58e−5	4.44	9.42e−5	5.12
0.0039	5.32e−6	–	5.80e−6	–	1.84e−5	–
τ	L^2 norm	Rate	L^2 norm	Rate	L^2 norm	Rate
0.0312	3.47e−2	4.08	3.53e−2	4.36	2.67e−2	3.88
0.0156	8.50e−3	4.05	8.90e−3	4.04	6.89e−3	3.81
0.0078	2.10e−3	4.01	2.20e−3	3.91	1.81e−3	3.76
0.0039	5.24e−4	–	5.63e−4	–	4.81e−4	–

We first present some tests for Problem 1. The discrete problem is solved using the regularized splitting schemes of Sections 3 and 4. The mesh size is chosen small enough so that the error from the discretization in space is negligible compared to the time stepping error.

Since when using BDF2 (Algorithm 1) the overall discretization error is $O(\tau^2) + O(h^2)$, the time steps tested have been chosen in the range $8h \geq \tau \geq h$, where h is the mesh size. The results are shown in Table 1, where the parameters are fixed as $h = 0.0039$, $\mu = 0.01$, $T = 3.14$, and the error is measured using the maximum norm. As can be seen, the time discretization error is of second-order for all the quantities.

To check whether Algorithm 1 can reasonably well resolve the pressure, we plot in Figure 1 the difference between the analytical and the computed pressure. Figure 1 shows that the pressure is captured quite well and the small difference between the computed and analytical pressure holds everywhere in the domain. This is a contrast to the plot in [21] where an artificial boundary condition is imposed for the pressure unknowns causing a larger error along the boundaries of the computational domain.

As can be seen from Table 1, for the density the convergence rates are larger than four in the infinite and L^2 norms. Also, the convergence rate in the infinite norm increases for smaller time steps, and decreases for smaller time steps in the L^2 norm.

Perhaps, it might be due to that the local errors where the maximal errors occur, are damped more efficiently for smaller time steps than the average L^2 errors are. Currently, no fully satisfactory explanation for these observations is found and they need more detailed analysis. In a test where Crank–Nicolson method was used instead of BDF, see [5], this behaviour was not seen.

As a check of the quality of the preconditioner $\mathcal{P}$ in (5.2) for Problem 1, Table 2 presents the iteration counts when solving the system (5.1) by GCR and the iterations when solving systems with $\widetilde{S}_{\mathcal{A}}$ and A by the off-the-shelf

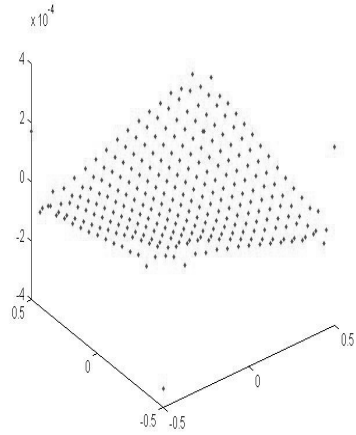


Figure 1. Difference between exact and computed pressure, $\tau = h = 0.0156$ and $T = 1.57$.

aggregation-based multigrid solver AGMG from [36]. All three stopping tolerances are relative and chosen to be 10^{-6}. For a fixed ratio τ/h, i,e., looking through the columns, one can see that the AGMG and the GCR iterations are essentially independent of the mesh refinement. If we fix the mesh size h, i.e., looking through the rows, one can see that AGMG and the GCR iterations depend slightly on the time steps τ. Table 2 shows that the preconditioner $\mathcal{P}$ with AGMG as the inner solver for the sub-blocks has an optimal order of computational complexity.

Table 2. Number of iterations.

τ	Iterations			Iterations			Iterations		
	$\widetilde{S}_{\mathcal{A}}$	A	$\mathcal{A}$	$\widetilde{S}_{\mathcal{A}}$	A	$\mathcal{A}$	$\widetilde{S}_{\mathcal{A}}$	A	$\mathcal{A}$
		$h = 0.031$			$h = 0.015$			$h = 0.007$	
h	8	5	5	8	5	5	8	7	7
$2h$	8	7	7	8	7	7	8	8	9
$4h$	8	9	8	8	9	10	8	8	8
$8h$	8	7	11	8	10	10	8	10	10

As mentioned, the results in Table 2 are obtained by setting all three stopping tolerances as 10^{-6}. The purpose of choosing such a tight tolerance for AGMG is to illustrate its efficiency. In practice, however, it is not necessary to choose such a small tolerance. For the $\widetilde{S}_{\mathcal{A}}^{-1}$, only one AGMG V-cycle is enough to obtain an accurate solution.

We turn next to Problem 2. RTI describes the phenomena of mixing of two materials, one lighter than the other, seeking to reduce their combined potential energy. This is a potentially unstable hydrodynamic configuration, involving linear, nonlinear and turbulent regimes. RTI, namely, the occurrence

of interfacial instabilities, has been first considered in 1883 by Lord Rayleigh in the idealized setting of two incompressible immiscible fluids in a constant gravitational field.

Due to the fact that over longer time the processes become turbulent and need to be resolved down to the Kolmogorov scale, some studies take the approach to consider the materials compressible (cf. [17]). The references we have used are [17,24,29,32,43]. In some of the works, the simulations are performed within the sharp interface framework, referred also as 'direct numerical simulations' (DNS) of the NS equations, as in [17, 24]. The advantage of DNS, compared to the other option introduced below, is its simplicity since one does not have to couple other mathematical models with the Navier–Stokes equations. The main drawback is that DNS method highly depends on the computer resources. Because the Navier–Stokes equations do not straightforwardly describe the interface and the surface tension force is not considered at all, the mesh must namely be sufficiently fine to guarantee the resolution, especially within the interface. In some other works, such as in [43], Euler equations are solved. A paper rich with experimental observations and numerical simulations, based on NS but assuming compressibility, is [29].

The framework we follow here is to couple NS to CH. The CH equation is one of the modelling tools within the diffuse interface approach and avoids the need to solve the density equation. This approach is followed by [4,9,16,32,40] and others.

The coupled NS-CH system in dimensionless form reads as follows

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{v} - \frac{1}{Re}\Delta\mathbf{u} + \nabla p &= \frac{\rho}{Fr}\mathbf{g}, \\ \nabla\cdot\mathbf{v} &= \mathbf{u}\cdot\nabla\rho, \\ \eta - \Psi'(C) + \epsilon^2\Delta C &= 0, \\ -\frac{1}{Pe}\Delta\eta + \frac{\partial C}{\partial t} + (\mathbf{u}\cdot\nabla)C &= 0 \end{aligned} \tag{6.2}$$

with $\frac{\partial C}{\partial n} = 0$, $\frac{\partial \eta}{\partial n} = 0$, and $C(\mathbf{x}, 0) = C_0(\mathbf{x})$, together with some appropriate boundary and initial conditions for the NS equations. Here the momentum variable $\mathbf{v}$ is the same as defined before, namely, $\mathbf{v} = \rho\mathbf{u}$.

In Problem 2, we assume that the two immiscible and incompressible phases share the same viscosity value but have different densities. This assumption is used only for simplicity, and there is no hesitation to extend the numerical scheme proposed for (6.2) in this paper to variable density and variable viscosity applications. In the above equations, the term C is referred to as the concentration and the term η denotes the chemical potential. The function $C(\mathbf{x}, t)$ attains a different constant value in each flow phase and rapidly but smoothly changes within the interface between the phase flows, enabling in this way to detect the position of the interface. In this paper we let C vary between -1 and 1 and, accordingly, choose the double well function $\Psi(C)$ as $\Psi(C) = \frac{1}{4}(C^2 - 1)^2$. The dimensionless physical parameters are the Reynolds number Re, Froude number Fr, Peclet number Pe and the Cahn number ϵ. The derivation of the dimensionless form (6.2) and the expressions of these parameters are clearly given in [32].

System (6.2) is solved again using an operator splitting technique. One solves NS to determine the velocity, then solves CH to determine the position of the interface, namely, density ρ, and repeats. In [16], convergence of the NS-CH splitting is derived. We point out that, in general, the coupling between NS and CH is not only via velocity but also via an additional term in the right hand side vector of NS, that depends on the concentration variable and describes the so-called surface tension forces. As explained in [43], and implemented in [24], this force term can be ignored when RTI is simulated.

As can be seen, in contrast to the DNS method (3.1)–(3.3), in the coupled system (6.2) there is no need to solve the density equation anymore. The reason is that when the interface is known, the density can be straightforwardly obtained. After computing C at the next time step, the density at the next time step is recovered as a weighted harmonic average of the densities of the different phases (two in this case),

$$\rho(C) = 2\,\big/\big((1-C)/\rho_1 + (1+C)/\rho_2\big)\,. \tag{6.3}$$

The expression (6.3) follows naturally since $(1-C)/2$ and $(1+C)/2$ are the mass fractions of the fluids and a unit volume $1/\rho$ consists of the corresponding volumes of the two fluids.

The numerical scheme to solve (6.2) is described as follows:

Algorithm 2 (Coupled NS-CH) *Given initial* $(C_0, \eta_0, \mathbf{u}_0)$*, for* $n \geq 1$*, proceed:*

A2-1: Compute (C^{n+1}, η^{n+1}) *by solving the Cahn-Hilliard equations*

$$\begin{aligned} \eta^{n+1} - \Psi'\big(C^{n+1}\big) + \epsilon^2 \Delta C^{n+1} &= 0, \\ -\frac{1}{Pe}\Delta\eta^{n+1} + \frac{\partial C^{n+1}}{\partial t} + \big(\mathbf{u}^n \cdot \nabla\big) C^{n+1} &= 0. \end{aligned} \tag{6.4}$$

A2-2: Recover ρ^{n+1} *as* (6.3)*.*

A2-3: Compute $(\mathbf{v}^{n+1}, p^{n+1})$ *by solving*

$$\begin{aligned} &\frac{3\mathbf{v}^{n+1} - 4\mathbf{v}^n + \mathbf{v}^{n-1}}{2\tau} + (\mathbf{u}^\star \cdot \nabla)\mathbf{v}^{n+1} - \frac{1}{Re}\Delta\frac{\mathbf{v}^{n+1}}{\rho^{n+1}} + a_{sd}(\mathbf{u}^\star, V) + \nabla p^{n+1} \\ &\quad = \frac{\rho^{n+1}}{Fr}\mathbf{g}, \end{aligned}$$

$$\nabla \cdot \mathbf{v}^{n+1} - \tau^2 \Delta p^{n+1} = \mathbf{u}^\star \cdot \nabla \rho^{n+1}, \quad \mathbf{u}^\star = 2\mathbf{u}^n - \mathbf{u}^{n-1}. \tag{6.5}$$

A2-4: Recover $\mathbf{u}^{n+1} = \mathbf{v}^{n+1}/\rho^{n+1}$*.*

Here τ is the time step. For $n = 1$, $(\mathbf{v}^1, \mathbf{u}^1, p^1)$ can be computed by using one step of the first-order Backward Euler scheme. As already introduced in Section 4 the additional term $a_{sd}(\mathbf{u}^\star, V) = \sum_k \gamma_k(\mathbf{u}^\star \cdot \nabla \frac{\mathbf{v}^{n+1}}{\rho^{n+1}}, \mathbf{u}^\star \cdot \nabla V)$ arises

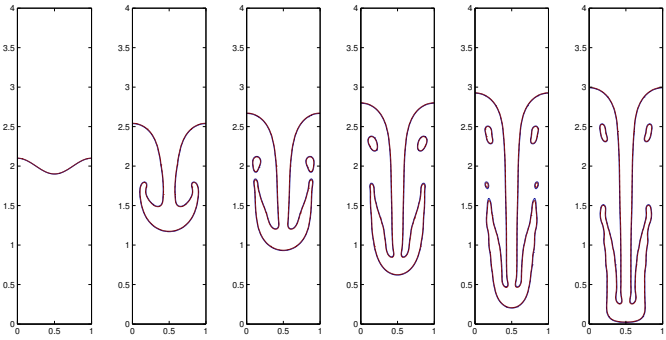
from the streamline diffusion stabilization and the added term $-\tau^2 \Delta p^{n+1}$ is used to avoid pressure oscillations.

The nonlinear term in the CH part is $\Psi'(C^{n+1}) = (C^{n+1})^3 - C^{n+1}$. In order to make the CH equations numerically stable, here we use an operator splitting method, namely, $\Psi'(C^{n+1}) = (C^{n+1})^3 - C^n$. This stabilization method is quite often used [30, 32]. How to efficiently solve the linear systems arising from the Cahn-Hilliard equations (6.4) falls out of the scope of the present paper. We follow the details in [4, 10]. For the solution of the Navier–Stokes equations (6.5) we use the already discussed preconditioning techniques in Section 5.

The settings of the problem parameters for Problem 2 are fixed as follows. The density difference is represented by the Atwood number $At = (\rho_2 - \rho_1)/(\rho_2 + \rho_1)$. Here we choose the density ratio to be $\rho_2/\rho_1 = 3$ and 7, i.e., the corresponding Atwood numbers $At = 0.5$ and 0.75. At $t = 0$, the initial location of the two immiscible and incompressible flows are imposed as

$$C(\mathbf{x}, 0) = tanh\Big(\frac{y - 2 - 0.1cos(2\pi x)}{\sqrt{2}\epsilon}\Big),$$

where $C = 1$ denotes the flow with density ρ_2, $C = -1$ corresponds to the flow with density ρ_1, and $-1 \ll C \ll 1$ describe the location of the interface. The computational domain is $[0, 1] \cup [0, 4]$ and we simulate on a space mesh $m \times 4m$, i.e., the mesh size is $h = 1/m$. The Cahn number ϵ, that determines the thickness of the interface, is set to be $\epsilon = 0.08\sqrt{2h}$. The reason is that we want to have a sufficient number of grid points across the interface. The Peclet number is chosen to be $Pe = 1/\epsilon$ and the effect of the Peclet number is numerically illustrated in [32]. The Reynolds number is set to be $Re = 1000$ and $Re = 5000$. The time step is chosen as $\tau = 0.16\sqrt{2}h$.

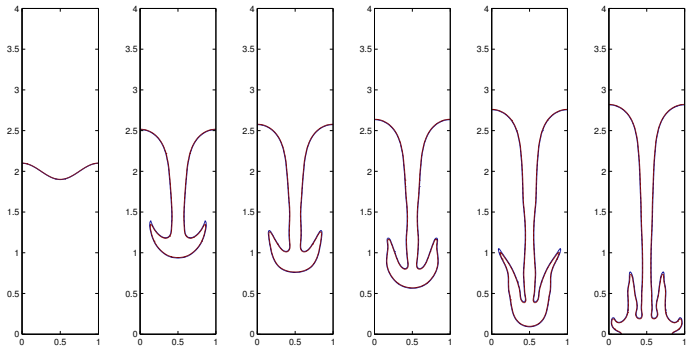


(a) $t = 0$ (b) $t = 2.0$ (c) $t = 2.5$ (d) $t = 3.0$ (e) $t = 3.5$ (f) $t = 3.75$

Figure 2. The interface of Problem 2 for $Re = 1000$ and $At = 0.5$.

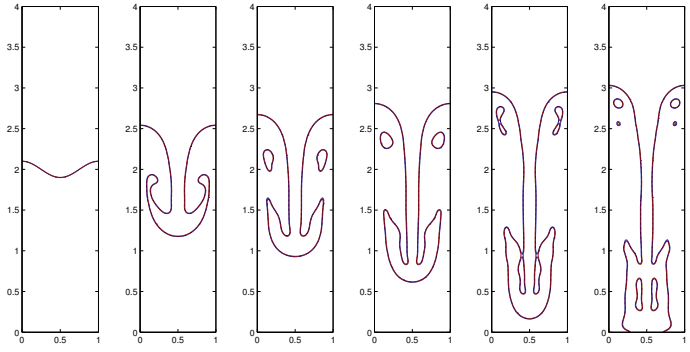
The evolution of the interface is plotted in Figures 2–4 in the time scale of Tryggvason [43], which is related to ours by $t_{\text{Tryg}} = t\sqrt{At}$. Here we choose the mesh size as $h = 1/64$. Due to the reasons given before, we stop the

simulations when the heavier fluid reaches the bottom. As can be seen in Figures 2–3, for the same Reynolds number a larger density ratio results in a faster evolution, namely the heavier flow reaches the bottom within a shorter time. The Reynolds number is another factor that influences the position and figure of the interface, comparing Figures 2 and 4. For a comparison, here we choose Figure 6 in [32] as a reference. The reasons are that we both use the NS-CH scheme and choose similar parameters. Figure 2 resembles well that in [32], and this means that Algorithm 2 introduced in this paper can accurately capture the physical properties in the RTI problem.



(a) $t = 0$ (b) $t = 2.0$ (c) $t = 2.25$ (d) $t = 2.5$ (e) $t = 3.0$ (f) $t = 3.25$

Figure 3. The interface of Problem 2 for $Re = 1000$ and $At = 0.75$.



(a) $t = 0$ (b) $t = 2.0$ (c) $t = 2.5$ (d) $t = 3.0$ (e) $t = 3.5$ (f) $t = 3.75$

Figure 4. The interface of Problem 2 for $Re = 5000$ and $At = 0.5$.

In Algorithm 2 [Coupled NS-CH], at each time step the variables $(\mathbf{v}^{n+1}, p^{n+1})$ are coupled together and computed simultaneously by solving

Table 3. Number of iterations for computing $(\mathbf{v}^{n+1}, p^{n+1})$ in Algorithm 2.

h	GCR-Iter.-$\mathcal{A}$	AGMG-Iter.-A	AGMG-Iter.-$\widetilde{S}_A$	CPU time
1/64	8	2	6	0.28
1/128	8	2	6	1.14
1/256	8	2	6	4.22

the equation (6.5). After space discretization, equation (6.5) can be rewritten as a linear system in a block structure as (5.1). Here we choose GCR method to solve the arising system. The relative stopping tolerance is chosen to be 10^{-6} and the number of GCR iterations is denoted as GCR-Iter.-$\mathcal{A}$. The preconditioner used for the matrix $\mathcal{A}$ in (5.1) is $\mathcal{P}$, as given in (5.2). In the preconditioner, we use AGMG solution method for two sub-systems with A and $\widetilde{S}_A$ and the relative stopping tolerance is 10^{-2}. The number of iterations is denoted as AGMG-Iter.-A and AGMG-Iter.-$\widetilde{S}_A$. In Table 3 we give the number of these three iterations and the computational CPU time for computing $(\mathbf{v}^{n+1}, p^{n+1})$ at each time step. The computational grids are 64×256, 128×512 and 256×1024. The corresponding mesh sizes are $h = 1/64$, $h = 1/128$ and $h = 1/256$. Here we also choose the time step as $\tau = 0.16\sqrt{2}h$ and the Reynolds number as $Re = 3000$. As seen in Table 3, all the three iteration counts are independent of the mesh size and the computational time shows that the complexity is of the optimal order.

There is another class of powerful solution methods for the variable density Navier–Stokes equations, namely, the projection methods. Among many variants, for comparison in this paper we choose the BDF second-order rotational projection method [24]. This method gained popularity since the pressure is determined by just solving a Poisson equation with constant coefficient per time step. The comparison is based on the Rayleigh–Taylor instability problem, and coupled with the Cahn-Hilliard equations. The computational procedure is presented as follows.

Algorithm 3 (Coupled NS-CH) *Given initial* $(C_0, \eta_0, \mathbf{u}_0, p_0, \phi_0)$*, for* $n \geq 1$*:*

A3-1: Compute (C^{n+1}, η^{n+1}) *by solving the Cahn-Hilliard equations* (6.4)*.*

A3-2: Recover ρ^{n+1} *as* (6.3)*.*

A3-3: Compute $\mathbf{u}^{n+1}$ *by solving*

$$\rho^{n+1}\frac{3\mathbf{u}^{n+1}-4\mathbf{u}^{n}+\mathbf{u}^{n-1}}{2\tau}+\rho^{n+1}(\mathbf{u}^{\star}\cdot\nabla)\mathbf{u}^{n+1}-\frac{1}{Re}\Delta\mathbf{u}^{n+1}+\nabla\left(p^{n}+\frac{4}{3}\phi^{n}-\frac{\phi^{n-1}}{3}\right)=\frac{\rho^{n+1}}{Fr}\mathbf{g}, \tag{6.6}$$

where $\mathbf{u}^{\star} = 2\mathbf{u}^{n} - \mathbf{u}^{n-1}$*.*

A3-4: A pressure correction is evaluated by solving

$$\Delta\phi^{n+1} = \frac{3\chi}{2\tau}\nabla\cdot\mathbf{u}^{n+1}, \quad \partial_n\phi^{n+1}|_{\partial\Omega}. \tag{6.7}$$

A3-5: Finally, the pressure is updated as

$$p^{n+1} = p^n + \phi^{n+1} - \frac{1}{Re}\nabla\cdot\mathbf{u}^{n+1}.$$

For $n = 1$, $(\mathbf{u}^1, p^1, \phi^1)$ can be computed by using one step of the first-order incremental projection method in [24]. The method parameter χ is assumed to satisfy

$$\forall n = 0, ..., N, \quad \chi \le \rho^n(\mathbf{x}) \quad \text{in } \Omega. \tag{6.8}$$

In practice χ is chosen as $\chi = \min\rho^0$ [24].

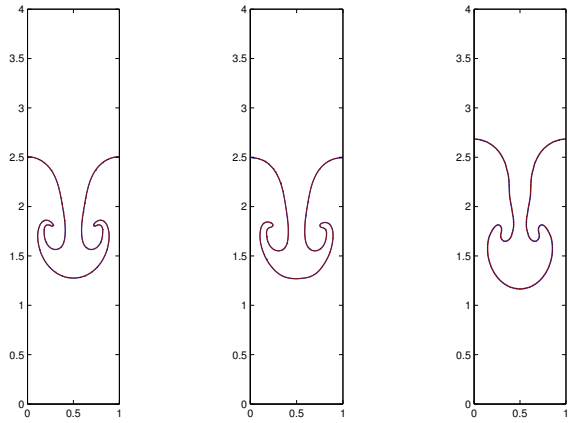
As seen in steps A3-3 and A3-4, the velocity $\mathbf{u}^{n+1}$ and pressure p^{n+1} are decoupled and computed separately. To solve the linear systems arising in the space discretization of (6.6) and (6.7), we use the AGMG solution method and the relative stopping tolerance is 10^{-6}. The iteration count numbers are denoted as AGMG-Iter.-$\mathbf{u}$ and AGMG-Iter.-p. In Table 4 we show the two iteration counts and the computational CPU time each time step for computing $(\mathbf{u}^{n+1}, p^{n+1})$. Here we use the same mesh size h, the time step τ and other physical parameters as used in Table 3. As seen in Table 4, the two iteration counts are also independent of the mesh size and the computational time is of the optimal order too.

Comparing the CPU time in Tables 3 and 4, we see that the projection scheme is somewhat more efficient than the coupled scheme introduced in this paper. However, the projection scheme needs artificial boundary conditions, as in (6.7), and the initial condition for the pressure which is not known in general.

Table 4. Number of iterations for computing $(\mathbf{u}^{n+1}, p^{n+1})$ in Algorithm 3.

h	AGMG-Iter.-$\mathbf{u}$	AGMG-Iter.-p	CPU time
1/64	6	13	0.12
1/128	6	14	0.51
1/256	8	14	2.12

Furthermore, the accuracy of the rotational projection method in Algorithm 3 depends on the method parameter χ. As analysed in [24], χ should satisfy the constraint (6.8) and all experiments are carried out by choosing $\chi = \min\rho^0$ in [24]. However, there is no guarantee that $\chi = \min\rho^0$ fully satisfies (6.8). A more safe choice is that we take χ much smaller. We conduct numerical experiments that show that the accuracy of the projection method is destroyed by choosing $\chi = 0.001\min\rho^0$. In Figure 5 we plot the interfaces simulated by Algorithm 2 and Algorithm 3 with $\chi = \min\rho^0$ and $\chi = 0.001\min\rho^0$. As compared, the projection method used in Algorithm 3 with $\chi = \min\rho^0$ can



(a) Algorithm 2

(b) Algorithm 3 with $\chi = \min \rho^0$

(c) Algorithm 3 with $\chi = 0.001 \min \rho^0$

Figure 5. $Re = 3000$, the interface of Problem 2 at dimensionless times 1.75 simulated by different schemes, $h = 1/128$.

compute nearly the same interface as the coupled scheme in Algorithm 2, see Figures 5 (a) and 5 (b). Also, these two figures are almost the same as that plotted in [32]. By decreasing χ the computed interface is much less accurate, as shown in Figure 5 (c). This means that the accuracy of the projection method introduced in [24] depends strongly on the method parameter.

To summarize, comparing the projection and the coupled schemes, based on the numerical experiments in this paper, we conclude that the projection method is more efficient in terms of solution time. The reason is that the velocity and pressure are decoupled and computed by solving two elliptic equations. On the other hand, its accuracy depends on a method parameter. For the coupled scheme introduced in this paper, the momentum and the pressure are coupled and computed simultaneously. There is no additional method parameter involved. A further development of efficient preconditioners in the coupled scheme will make it faster and more attractive for large scale numerical simulations. This is considered as a future research topic.

7 Concluding Remarks

We have studied various aspects of the numerical solution of the variable density Navier–Stokes equations – discretization, operator splitting and linearization, and the interplay between the related errors. We have also considered preconditioning techniques, suitable for the arising linear systems, in order to enable fast and robust numerical simulations of the underlying flow phenomena.

We have investigated the impact of using the momentum instead of the velocity as a primal unknown variable, based on the argument that in a single media flow the momentum can be better behaved than the velocity and, thus, is easier to compute numerically. Due to the diffusion interface model used in multiphase flow problems, the momentum variable is continuous within the whole computational domain, in particular across the interfaces between different media. We have also considered the efficiency of an operator splitting, where no artificial boundary conditions have to be imposed for the pressure, enabling in this way the use of well-known high quality preconditioners for saddle point matrices. The divergence free vector field is computed without the need to use projections onto the divergence-free vector space. No lower-order errors due to splitting and linearization arise.

We also construct an efficient stabilization method for the mass balance equation, namely, the density equation. This stabilization makes the semi-implicit scheme for the density equation unconditionally stable, and needs less computational complexity than other methods. Also, other stabilizations are used in this paper to avoid unphysical oscillations in the momentum and pressure. The efficiency of the proposed preconditioner, for the arising saddle point system at each time step, is tested via numerical experiments. The results presented in this paper show that the computational complexity by using the proposed preconditioner is of optimal order.

A complete testing and comparison of the performance of the proposed solution scheme with other, projection-based operator splitting schemes, would be very insightful and is a subject of a separate study.

The methods used in this paper can be applied also for more coupled equations, such as Navier–Stokes equations for a free fluid coupled with poroelasticity equations, modelled by Darcy flow equations and equations for a porous elastic medium.

Acknowledgements

The authors are indebted to the anonymous reviewer for the constructive comments and suggestions that lead to improvments in the contents of the paper and the presentation of the results.

Appendix

Consider the density equation $\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0$, where we use the following numerical scheme to compute ρ^{n+1} for time level $n \geq 1$

$$\frac{3\rho^{n+1} - 4\rho^n + \rho^{n-1}}{2\tau} + \mathbf{u}^\star \cdot \nabla \rho^{n+1} + \frac{1}{2}\rho^{n+1} \nabla \cdot \mathbf{u}^\star = 0. \tag{7.1}$$

Here $\mathbf{u}^\star$ denotes the linearly extrapolated velocity at time level $n+1$ as $\mathbf{u}^\star = 2\mathbf{u}^n - \mathbf{u}^{n-1}$. The time step is denoted as τ.

Theorem 1. *Assume that the sequence of the velocities* $\mathbf{u}^n{}_{n=0,\ldots,N}$ *satisfies the boundary conditions* $\mathbf{u}^n \cdot \mathbf{n}|_{\partial\Omega} = 0$*, the solutions of* (7.1) *satisfy*

$$3\left\|\rho^N\right\|^2 + \left\|\rho^0\right\|^2 + 2\left\|\rho^N - \rho^{N-1}\right\|^2 \leq 4\left\|\rho^1\right\|^2 + 2\left\|\rho^1 - \rho^0\right\|^2.$$

Proof. We multiply (7.1) by $4\tau\rho^{n+1}$ and integrate over Ω. We have

$$4\int_\Omega 2\rho^{n+1}\big(\rho^{n+1}-\rho^n\big)-\int_\Omega 2\rho^{n+1}\big(\rho^{n+1}-\rho^{n-1}\big) + 2\tau\int_\Omega\big(2\rho^{n+1}\mathbf{u}^\star\cdot\nabla\rho^{n+1}+\big(\rho^{n+1}\big)^2\nabla\cdot\mathbf{u}^\star\big)=0. \tag{7.2}$$

Using the boundary condition on $\mathbf{u}^\star$ and integrating by parts, we have

$$2\tau\int_\Omega\big(2\rho^{n+1}\mathbf{u}^\star\cdot\nabla\rho^{n+1}+\big(\rho^{n+1}\big)^2\nabla\cdot\mathbf{u}^\star\big)=2\tau\int_\Omega\nabla\cdot\big(\big(\rho^{n+1}\big)^2\mathbf{u}^\star\big)=0. \tag{7.3}$$

Using the identities $2a(a-b)=a^2-b^2+(a-b)^2$ and (7.3), we can rewrite the identity (7.2) as

$$4\big(\|\rho^{n+1}\|^2-\|\rho^n\|^2+\|\rho^{n+1}-\rho^n\|^2\big)-\big(\|\rho^{n+1}\|^2-\|\rho^{n-1}\|^2+\|\rho^{n+1}-\rho^{n-1}\|^2\big)=0. \tag{7.4}$$

Summing up the identity (7.4) from $n=1$ to $n=N-1$ we have

$$4\bigg(\|\rho^N\|^2-\|\rho^1\|^2+\sum_{k=1}^{N-1}\|\rho^{k+1}-\rho^k\|^2\bigg) -\bigg(\|\rho^N\|^2-\|\rho_0\|^2+\sum_{k=1}^{N-1}\|\rho^{k+1}-\rho^{k-1}\|^2\bigg)=0,$$

furthermore

$$3\|\rho^N\|^2+\|\rho^0\|^2+4\sum_{k=1}^{N-1}\|\rho^{k+1}-\rho^k\|^2=4\|\rho^1\|^2+\sum_{k=1}^{N-1}\|\rho^{k+1}-\rho^{k-1}\|^2.$$

Using the inequality $2ab\le a^2+b^2$ we have

$$\begin{aligned}\|\rho^{k+1}-\rho^{k-1}\|^2&=\|(\rho^{k+1}-\rho^k)+(\rho^k-\rho^{k-1})\|^2\\&\le\big(\|\rho^{k+1}-\rho^k\|+\|\rho^k-\rho^{k-1}\|\big)^2\\&\le 2\big(\|\rho^{k+1}-\rho^k\|^2+\|\rho^k-\rho^{k-1}\|^2\big).\end{aligned}$$

Then we have

$$3\|\rho^N\|^2+\|\rho^0\|^2+2\sum_{k=1}^{N-1}\|\rho^{k+1}-\rho^k\|^2\le 4\|\rho^1\|^2+2\sum_{k=1}^{N-1}\|\rho^k-\rho^{k-1}\|^2.$$

Finally we have

$$3\|\rho^N\|^2+\|\rho^0\|^2+2\|\rho^N-\rho^{N-1}\|^2\le 4\|\rho^1\|^2+2\|\rho^1-\rho^0\|^2.$$

References

[1] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell and M.L. Welcome. A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *J. Comput. Phys.*, **142**(1):1–46, 1998. http://dx.doi.org/10.1006/jcph.1998.5890.

[2] P. Angot, J.-P. Caltagirone and P.A. Fabrie. A fast vector penalty-projection method for incompressible non-homogeneous or multiphase Navier–Stokes problems. *Appl. Math. Lett.*, **25**(11):1681–1688, 2012. http://dx.doi.org/10.1016/j.aml.2012.01.037.

[3] O. Axelsson, V.A. Barker, M. Neytcheva and B. Polman. Solving the Stokes problem on a massively parallel computer. *Math. Model. Anal.*, **6**(1):7–27, 2001. http://dx.doi.org/10.1080/13926292.2001.9637141.

[4] O. Axelsson, P. Boyanova, M. Kronbichler, M. Neytcheva and X. Wu. Numerical and computational efficiency of solvers for two-phase problems. *Comput. Math. Appl.*, **65**(3):301–314, 2013. http://dx.doi.org/10.1016/j.camwa.2012.05.020.

[5] O. Axelsson, X. He and M. Neytcheva. Numerical solution of the time-dependent Navier–Stokes equation for variable density-variable viscosity. Technical Report 2012-019, Institute for Information Technology, Uppsala University, The Sweden, 2014.

[6] O. Axelsson and M. Neytcheva. A general approach to analyse preconditioners for two-by-two block matrices. *Numer. Linear Algebra Appl.*, **20**(5):723–742, 2013. http://dx.doi.org/10.1002/nla.830.

[7] O. Axelsson and P.S. Vassilevski. A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning. *SIAM J. Matrix Anal. Appl.*, **12**(4):625–644, 1991. http://dx.doi.org/10.1137/0612048.

[8] J. Blasco, R. Codina and A. Huerta. A fractional-step method for the incompressible Navier–Stokes equations related to a predictor-multicorrector algorithm. *Int. J. Numer. Meth. Fluids*, **28**(10):1391–1419, 1998.

[9] P. Boyanova, M. Do-Quang and M. Neytcheva. Efficient preconditioners for large scale binary Cahn–Hilliard models. *Comput. Meth. Appl. Math.*, **12**(1):1–22, 2012. http://dx.doi.org/10.2478/cmam-2012-0001.

[10] P. Boynanova. *On numerical solution methods for block-structure discrete systems.* PhD thesis, Institute for Information Technology, Uppsala University, The Sweden, 2012.

[11] A.J. Chorin. Numerical solution of the Navier–Stokes equations. *Math. Comp.*, **22**(104):745–762, 1968. http://dx.doi.org/10.1090/S0025-5718-1968-0242392-2.

[12] J.C. Chrispell, V.J. Ervin and E.W. Jenkins. A fractional step θ-method for convection–diffusion problems. *J. Math. Anal. Appl.*, **333**(1):204–218, 2007. http://dx.doi.org/10.1016/j.jmaa.2006.11.059.

[13] H.-J.G. Diersch and O. Kolditz. Variable-density flow and transport in porous media: approaches and challenges. *Adv. Water Resour.*, **25**(8):899–944, 2002. http://dx.doi.org/10.1016/S0309-1708(02)00063-5.

[14] S. Eckert, H. Baaser, D. Gross and O. Scherf. A BDF2 integration method with step size control for elasto-plasticity. *Comput. Mech.*, **34**(5):377–386, 2004. http://dx.doi.org/10.1007/s00466-004-0581-1.

[15] H.C. Elman, D.J. Silvester and A.J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014.

[16] X. Feng. Fully discrete finite element approximations of the Navier–Stokes–Cahn–Hilliard diffuse interface model for two-phase fluid flows. *SIAM J. Numer. Anal.*, **44**(3):1049–1072, 2006. http://dx.doi.org/10.1137/050638333.

[17] S. Gauthier and B. Le Creurer. Compressibility effects in Rayleigh–Taylor instability-induced flows. *Phil. Trans. R. Soc. A*, **368**(1916):1681–1704, 2010.

[18] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*, volume 4. Springer, 1984.

[19] P.M. Gresho. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *Int. J. Numer. Meth. Fluids*, **11**(5):587–620, 1990. http://dx.doi.org/10.1002/fld.1650110509.

[20] J.-L. Guermond and P. Minev. Efficient Parallel Algorithms for Unsteady Incompressible Flows. In *Numerical Solution of Partial Differential Equations: Theory, Algorithms, and Their Applications*, pp. 185–201. Springer, 2013.

[21] J.-L. Guermond, P. Minev and J. Shen. An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Eng.*, **195**(44):6011–6045, 2006. http://dx.doi.org/10.1016/j.cma.2005.10.010.

[22] J.-L. Guermond, R. Pasquetti and B. Popov. Entropy viscosity method for nonlinear conservation laws. *J. Comput. Phys.*, **230**(11):4248–4267, 2011. http://dx.doi.org/10.1016/j.jcp.2010.11.043.

[23] J.-L. Guermond and L. Quartapelle. A projection FEM for variable density incompressible flows. *J. Comput. Phys.*, **165**(1):167–188, 2000. http://dx.doi.org/10.1006/jcph.2000.6609.

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

[25] J.-L. Guermond and J. Shen. Velocity-correction projection methods for incompressible flows. *SIAM J. Numer. Anal.*, **41**(1):112–134, 2003. http://dx.doi.org/10.1137/S0036142901395400.

[26] Y. He. The Euler implicit/explicit scheme for the 2D time-dependent Navier-Stokes equations with smooth or non-smooth initial data. *Math. Comput.*, **77**(264):2097–2124, 2008. http://dx.doi.org/10.1090/S0025-5718-08-02127-3.

[27] J.G. Heywood and R. Rannacher. Finite-element approximation of the nonstationary Navier–Stokes problem. Part IV: Error analysis for second-order time discretization. *SIAM J. Numer. Anal.*, **27**(2):353–384, 1990. http://dx.doi.org/10.1137/0727022.

[28] M. Jobelin, C. Lapuerta, J.C. Latché, Ph. Angot and B. Piar. A finite element penalty-projection method for incompressible flows. *J. Comput. Phys.*, **217**(2):502–518, 2006. http://dx.doi.org/10.1016/j.jcp.2006.01.019.

[29] D.D. Joseph and Y.Y. Renardy. *Fluid Dynamics of Two Miscible Liquids with Diffusion and Gradient Stresses*. Springer, 1993.

[30] D. Kay and R. Welford. Efficient numerical solution of Cahn–Hilliard–Navier–Stokes fluids in 2D. *SIAM J. Sci. Comput.*, **29**(6):2241–2257, 2007. http://dx.doi.org/10.1137/050648110.

[31] D. Kuzmin. *A Guide to Numerical Methods for Transport Equations*. University Erlangen-Nuremberg, Germany, 2010. http://www.mathematik.uni-dortmund.de/ kuzmin/cfdbook.html

[32] H.G. Lee, K. Kim and J. Kim. On the long time simulation of the Rayleigh–Taylor instability. *Int. J. Numer. Meth. Eng.*, **85**(13):1633–1647, 2011. http://dx.doi.org/10.1002/nme.3034.

[33] P.L. Lions and R. Farwig. Mathematical topics in fluid mechanics. Vol. 1: Incompressible models. *SIAM Rev.*, **40**(2):402–403, 1998.

[34] Ch. Liu and N.J. Walkington. Convergence of numerical approximations of the incompressible Navier–Stokes equations with variable density and viscosity. *SIAM J. Numer. Anal.*, **45**(3):1287–1304, 2007. http://dx.doi.org/10.1137/050629008.

[35] S. Müller-Urbaniak. *Eine Analyse dex Zweischritt-θ-Verfahrens zur Lösung der instationären Navier–Stokes-Gleichungen*. PhD thesis, University of Heidelberg, German, 1994.

[36] Y. Notay. The software package AGMG. http://homepages.ulb.ac.be/ ynotay/

[37] R. Pierre. Simple C^0 approximations for the computation of incompressible flows. *Comp. Meth. Appl. Mech. Eng.*, **68**(2):205–227, 1988. http://dx.doi.org/10.1016/0045-7825(88)90116-8.

[38] R. Rannacher. *On Chorin's Projection Method for the Incompressible Navier–Stokes Equations*. Springer, 1992.

[39] Y. Saad. A flexible inner–outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, **14**(2):461–469, 1993. http://dx.doi.org/10.1137/0914028.

[40] J. Shen and X. Yang. A phase-field model and its numerical approximation for two-phase incompressible flows with different densities and viscosities. *SIAM J. Sci. Comp.*, **32**(3):1159–1179, 2010. http://dx.doi.org/10.1137/09075860X.

[41] C.T. Simmons. Variable density groundwater flow: From current challenges to future possibilities. *Hydrogeol. J.*, **13**(1):116–119, 2005. http://dx.doi.org/10.1007/s10040-004-0408-3.

[42] R. Temam. Une méthode d'approximation de la solution des équations de Navier–Stokes. *Bull. Soc. Math. France*, **96**(1):115–152, 1968.

[43] G. Tryggvason. Numerical simulations of the Rayleigh–Taylor instability. *J. Comput. Phys.*, **75**(2):253–282, 1988. http://dx.doi.org/10.1016/0021-9991(88)90112-X.

[44] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approache*. Springer, 1999.

[45] P.S. Vassilevski. *Multilevel Block Factorization Preconditioners*. Springer, 2008.

[46] W. Villanueva and G. Amberg. Some generic capillary-driven flows. *Int. J. Multiphase Flow*, **32**(9):1072–1086, 2006. http://dx.doi.org/10.1016/j.ijmultiphaseflow.2006.05.003.

[47] D. Wang and S.J. Ruuth. Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations. *J. Comput. Math.*, **26**(6):838–855, 2008.

[48] S. Zahedi. *ONumerical Methods for Fluid Interface Problems*. PhD thesis, KTH Royal Institute of Technology, The Sweden, 2011.

[49] K. Zhao. Large time behavior of density-dependent incompressible Navier–Stokes equations on bounded domains. *J. Math. Fluid Mech.*, **14**(3):471–483, 2012. http://dx.doi.org/10.1007/s00021-011-0076-8.