High-order arbitrary Lagrangian–Eulerian discontinuous Galerkin methods for the incompressible Navier–Stokes equations

Niklas Fehn*, Johannes Heinz, Wolfgang A. Wall, Martin Kronbichler

Institute for Computational Mechanics, Technical University of Munich,
Boltzmannstr. 15, 85748 Garching, Germany

Abstract

This paper presents robust discontinuous Galerkin methods for the incompressible Navier–Stokes equations on moving meshes. High-order accurate arbitrary Lagrangian–Eulerian formulations are proposed in a unified framework for both monolithic as well as projection or splitting-type Navier–Stokes solvers. The framework is flexible, allows implicit and explicit formulations of the convective term, and adaptive time-stepping. The Navier–Stokes equations with ALE transport term are solved on the deformed geometry storing one instance of the mesh that is updated from one time step to the next. Discretization in space is applied to the time discrete equations so that all weak forms and mass matrices are evaluated at the end of the current time step. This design ensures that the proposed formulations fulfill the geometric conservation law automatically, as is shown theoretically and demonstrated numerically by the example of the free-stream preservation test. We discuss the peculiarities related to the imposition of boundary conditions in intermediate steps of projection-type methods and the ingredients needed to preserve high-order accuracy. We show numerically that the formulations proposed in this work maintain the formal order of accuracy of the Navier–Stokes solvers. Moreover, we demonstrate robustness and accuracy for under-resolved turbulent flows.

Keywords: arbitrary Lagrangian–Eulerian (ALE), incompressible Navier–Stokes, discontinuous Galerkin, matrix-free methods, projection methods

1. Introduction

The arbitrary Lagrangian–Eulerian (ALE) continuum mechanics description is the basis of many methods to capture flow problems on deforming domains. A very prominent class of applications are fluid–structure interaction problems with moderate deformations of the structure, where moderate means that a mesh moving or mesh smoothing algorithm is able to handle the mesh deformation of the fluid mesh following the deformations imposed at the fluid–structure interface, as opposed to very large deformations and topological changes that require other, geometrically more flexible techniques. ALE methods have a long tradition and have first been developed for finite difference methods [1], see also [2] for a review on the early development of this methodology, as well as [3] for a survey of ALE methods. It has later been developed for finite element discretizations of the compressible Navier–Stokes equations in [4, 2], of the incompressible Navier–Stokes equations using linear elements in [5] and spectral element discretizations in [6], and also for finite volume discretizations, see for example [7]. In the context of discontinuous Galerkin (DG) discretizations, this technique has first been applied to the compressible Navier–Stokes equations being solved on the deforming domain [8, 9, 10], or solving transformed equations on a reference domain [11, 12]. For the incompressible

---

Navier–Stokes equations, the development of ALE-DG methods lagged somewhat behind as detailed below. An ALE method satisfying the geometric conservation law (GCL) [13] is able to preserve a constant flow state on moving meshes. The GCL has extensively been discussed in the context of finite volume discretizations of the compressible Navier–Stokes equations, see for example the review article [14] and references therein. Here, we especially refer to two works [15, 16] addressing the solution of incompressible flow problems and being particularly relevant for the present work. Following the design described in [15] one can easily construct incompressible flow solvers that automatically fulfill the geometric conservation law, and we therefore do not discuss this topic at length in the present work.

Before discussing ALE-DG methods for incompressible flows, let us first summarize the key issue of incompressible Navier–Stokes DG solvers in the Eulerian case with static meshes. The main problem originates from the nonlinearity of the convective term in case the numerical velocity field is not pointwise divergence-free like with standard L^2-conforming spaces. In this circumstance, the energy stability derived e.g. for an upwind flux with linear transport terms of constant speed is lost. Special DG discretizations that are exactly mass conserving and energy stable have been discussed in early mathematical literature on DG methods for the incompressible Navier–Stokes equations, see [17, 18, 19]. However, the importance of this aspect and its relevance for practical problems have long not been realized in application-oriented DG literature, where no attempts have been made to fulfill these properties [20, 21, 22, 23, 24, 25]. It was later found in [26, 27, 28, 29] that compliance with (i) the divergence-free constraint and (ii) inter-element continuity of the normal velocity is crucial in obtaining robust solvers for engineering applications where the solution is under-resolved, such as turbulent flows. Methods explicitly addressing these two requirements can be categorized into two groups, those fulfilling them exactly for example by choosing appropriate function spaces such as $H(\mathrm{div})$-conforming spaces where the divergence of the velocity space lies in the pressure space [17, 18, 19, 30, 31, 32, 33], and those fulfilling them weakly by appropriate stabilization terms [27, 28, 29, 34, 35]. These constraints on the velocity solution can be imposed in form of a postprocessing step or in an inbuilt/monolithic way, independently of the two categories a method belongs to. A comparative study of the two categories has been shown recently in [36] in the context of under-resolved turbulent flows.

ALE formulations for the incompressible Navier–Stokes equations using DG discretizations have been presented in [37, 38], with the mesh motion being restricted to rigid body rotations without mesh deformation in [37]. Both methods are based on the dual splitting projection scheme proposed by [39, 40] and use a discontinuous Galerkin discretization of the velocity–pressure coupling terms without integration by parts, originating from the method proposed in [22], that has been shown to be unstable for small time step sizes by independent studies, see [41, 42]. Both works [37, 38] use equal-order poylnomials for velocity and pressure, but it was shown in [41] that the dual splitting scheme per se is not inf–sup stable as is sometimes believed. Morever, the works [37, 38] enforce the important aspect of $H(\mathrm{div})$-conformity discussed above neither exactly nor weakly, so that the robustness of these methods remains questionable. These works also do not comment on the fulfillment of the geometric conservation law theoretically or by numerical experiments. Second-order convergence in time is shown in [37] for a rotating, non-deforming mesh with fixed boundaries, hence, not being representative of a deforming fluid domain or a fluid–structure interaction problem. A temporal convergence test is shown in [38] for a full FSI problem by comparing the error against the solution for the smallest time step size and second-order accuracy is shown. It remains unclear from these works whether third order accuracy can be achieved for the dual splitting scheme in the ALE case, which has for example been shown in [28] for a DG method solving the Eulerian form of the equations. Open questions remain therefore from previous works [37, 38] on ALE-DG methods for the incompressible Navier–Stokes equations regarding how these methods are implemented exactly and questions about their numerical properties in terms of stability (small time steps, inf–sup problem, and under-resolved turbulence or energy stability), fulfillment of the geometric conservation law, and temporal convergence rates.

More sophisticated convergence tests are presented in [43] for a space–time HDG method on deforming domains, where it is argued that the reasons for choosing the space–time approach are automatically satifying the geometric conservation law and achieving arbitrarily high order in space and time. An energy-stable version of such a space–time HDG approach has been proposed recently in [44] achieved through a velocity field that is $H(\mathrm{div})$-conforming and exactly divergence-free. As shown in the present work, fulfillment

of the GCL and discretizations that formally exhibit arbitrarily high order of accuracy in space can also be achieved with a classical method-of-lines approach. Achieving arbitrarily high order of accuracy in time with projection-type Navier–Stokes solvers is non-trivial, irrespective of the grid motion. However, our experience is that second or third-order time integration schemes are sufficient in terms of accuracy for practical problems, especially if the time step size is restricted according to the CFL condition when treating the convective term explicitly in time, which is the state-of-the-art solution technique used by some of the most sophisticated and computationally efficient high-order CFD solvers, such as Nektar++[45] and Nek5000 [46]. We emphasize that formal orders of accuracy describe the optimal behavior observable only in the asymptotic regime for sufficiently smooth solution. In this context, it should be mentioned that high order of convergence in space is rarely observed for practical problems. The dissipation/dispersion properties and the associated improved resolution capabilities of high-order methods rather than theoretical rates of convergence motivate the use of high-order methods for application-relevant, turbulent flows [47, 48, 49]. When it comes to the aspect of computational costs, it has not yet been demonstrated that space–time approaches can keep up with the fast solution techniques for incompressible flows mentioned above. A similar argument holds for the computational efficiency of matrix-based HDG solvers, which significantly trail behind fast matrix-free DG implementations on modern CPU hardware, as shown by a recent study [50].

For these reasons, our goal is to develop ALE-DG methods based on the method-of-lines approach that combine computationally efficient matrix-free DG implementations [51] and fast Navier–Stokes solution algorithms [41, 29] with desirable discretization properties in terms of optimal convergence rates in time and space, the geometric conservation law, and robustness for turbulent flows. To develop algorithms as simple as possible, we make the following design choices: We solve the ALE equations on the deformed geometry, storing one instance of the mesh that is updated from one time step to the next by updating the coordinates of all nodal points. We introduce the ALE equations on the level of differential equations, subsequently discretized in time and space. This way, it is straight-forward to satisfy the geometric conservation law automatically [15], i.e., independently of the mesh motion and how the mesh velocity is computed numerically. We present a unified framework for both monolithic solvers and widely used projection-type solvers. The formulation is flexible regarding implicit versus explicit formulations of the convective term, and the framework naturally includes the option for adaptive time-stepping. Although we consider analytical mesh motions in the present work, the methods are formulated with fluid–structure interaction problems in mind, i.e., the ALE formulations are implemented in a way that they only require knowledge about the coordinates of all grid nodes at discrete instances of time, and the grid velocity is computed from these grid coordinates in a way that the formal order of accuracy of the time integration schemes is maintained on moving meshes.

The outline of this article is as follows. We derive the ALE form of the incompressible Navier–Stokes equations in Section 2. Aspects related to the temporal discretization are discussed in Section 3, and the spatial discretization is subject of Section 4. Here, our focus lies on the aspects relevant to ALE, with the goal to provide a comprehensive and simple formulation that can easily be included in existing flow solvers. Numerical results are presented in Section 5, and we summarize our results in Section 6.

2. Incompressible Navier–Stokes equations in arbitrary Lagrangian–Eulerian formulation

We consider the incompressible Navier–Stokes equations in a domain $\Omega \subset \mathbb{R}^d$

$$\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\boldsymbol{X}} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}) + \nabla p = \boldsymbol{f} \, , \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \, , \tag{2}$$

where $\boldsymbol{u} = (u_1, ..., u_d)^\mathsf{T}$ is the velocity vector and p the kinematic pressure. The body force vector is denoted by $\boldsymbol{f} = (f_1, ..., f_d)^\mathsf{T}$. Spatial derivatives are defined w.r.t. the Eulerian coordinates $\boldsymbol{x} = (x_1, ..., x_d)^\mathsf{T}$, $\nabla \phi = \frac{\partial \phi}{\partial \boldsymbol{x}}$, and $|_{\boldsymbol{X}}$ denotes the material time derivative, i.e., the total time derivative along the trajectory of a material point $\boldsymbol{X}$ of the fluid. The viscous term is written in Laplace formulation $\boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}) = \nu \nabla \boldsymbol{u}$ with the constant kinematic viscosity ν.

To obtain the ALE form of the above equations, the material time derivative (or the Eulerian time derivative

3

when considering the incompressible Navier–Stokes equations written in Eulerian coordinates) is replaced by a time derivative with respect to a fixed point of the mesh (denoted as ALE time derivative in the following), which gives rise to an additional transport term with transport by the grid velocity $\boldsymbol{u}_{\mathrm{G}}$. However, this transport term has the same structure as the convective term in Eulerian description, so that the same implementation with transport velocity $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{u}_{\mathrm{G}}$ instead of the fluid velocity $\boldsymbol{u}$ can be used. The motivation behind is to apply the time integration scheme with an update of the solution vectors just as in the Eulerian case, thereby automatically obtaining the solution coefficients on the new mesh. By this technique, expensive transformations of the solution vector (containing the degrees of freedom of the finite element expansion) from one mesh at a previous time instant onto another one at the current time instant is avoided. Apart from the Eulerian coordinates $\boldsymbol{x}$ describing an (arbitrary) point in Euclidean space and the material coordinates $\boldsymbol{X}$ describing a fixed fluid element, we introduce the mesh coordinates $\boldsymbol{\chi}$ describing a fixed point of the mesh. For transient problems, $\boldsymbol{x}(\boldsymbol{X}, t)$ and $\boldsymbol{x}(\boldsymbol{\chi}, t)$ describe the trajectories of a fixed fluid element or a fixed point of the mesh in the Eulerian coordinates $\boldsymbol{x}$ as a function of time. The fluid and mesh velocity are therefore given as

$$\boldsymbol{u} = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{X}} , \ \boldsymbol{u}_{\mathrm{G}} = \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{\chi}} .$$

The material time derivative of an arbitrary quantity ϕ is given as

$$\left.\frac{\partial \phi\left(\boldsymbol{x}(\boldsymbol{X}, t), t\right)}{\partial t}\right|_{\boldsymbol{X}} = \left.\frac{\partial \phi}{\partial t}\right|_{\boldsymbol{x}} + \frac{\partial \phi}{\partial \boldsymbol{x}} \cdot \underbrace{\left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{X}}}_{=u} = \left.\frac{\partial \phi}{\partial t}\right|_{\boldsymbol{x}} + (u \cdot \nabla) \phi . \tag{3}$$

The same relation can be stated for the mesh reference frame $\boldsymbol{\chi}$ to obtain the desired relation between the Eulerian and ALE time derivatives

$$\left.\frac{\partial \phi\left(\boldsymbol{x}(\boldsymbol{\chi}, t), t\right)}{\partial t}\right|_{\boldsymbol{\chi}} = \left.\frac{\partial \phi}{\partial t}\right|_{\boldsymbol{x}} + \frac{\partial \phi}{\partial \boldsymbol{x}} \cdot \underbrace{\left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{\chi}}}_{=\boldsymbol{u}_{\mathrm{G}}} = \left.\frac{\partial \phi}{\partial t}\right|_{\boldsymbol{x}} + (\boldsymbol{u}_{\mathrm{G}} \cdot \nabla) \phi . \tag{4}$$

Inserting equations (3) and (4) into equation (1), we arrive at the incompressible Navier–Stokes equations in ALE formulation

$$\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\boldsymbol{\chi}} + ((\boldsymbol{u} - \boldsymbol{u}_{\mathrm{G}}) \cdot \nabla) \boldsymbol{u} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}) + \nabla p = \boldsymbol{f} \ \text{in } \Omega(t) , \tag{5}$$

$$\nabla \cdot \boldsymbol{u} = 0 \ \text{in } \Omega(t) . \tag{6}$$

Remark Note that the formulation chosen as a starting point for discretization in time and space has important implications regarding compliance with the geometric conservation law. Using the above differential formulation with the convective term written in non-conservative form to derive temporal and spatial discretization allows to satisfy the GCL automatically [15]. Alternative conservative formulations with time derivative in front of the integral over a temporally changing domain contain an additional term in which the divergence of the mesh velocity occurs, and fulfilling the GCL is more complicated in this case [15, 16]. We mention that the study [16] is inconclusive in the sense that the non-conservative formulation is not restricted to first-order accuracy in time as implied in that work. As demonstrated in [15] and in the present work, high-order accuracy in time can be achieved with the non-conservative formulation.

The motion of the domain $\Omega(t)$ is described by a function $\boldsymbol{f}_{\mathrm{G}} = \boldsymbol{f}_{\mathrm{G}}(\boldsymbol{\chi}, t)$

$$\boldsymbol{f}_{\mathrm{G}} : \begin{cases} \Omega_0 \times [0, T] \to \Omega(t), \ \Omega_0, \Omega(t) \subset \mathbb{R}^d , \\ (\boldsymbol{\chi}, t) \mapsto \boldsymbol{x}(\boldsymbol{\chi}, t) . \end{cases}$$
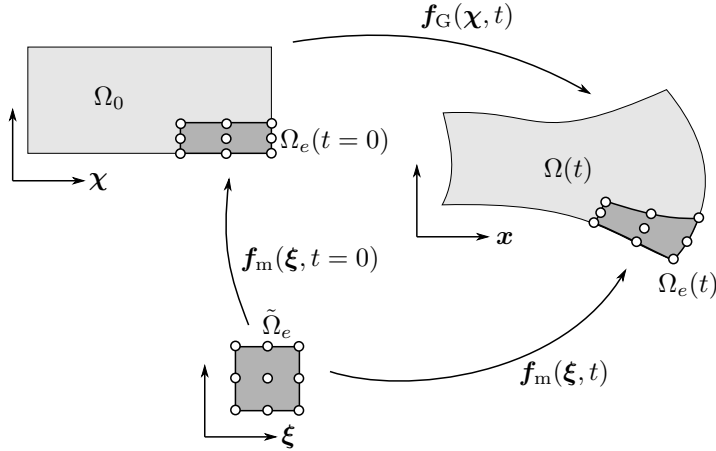
Figure 1: Illustration of coordinate systems $\boldsymbol{\chi}$ and $\boldsymbol{x}$ with mesh deformation $\boldsymbol{f}_{\mathrm{G}}$, and reference coordinates $\boldsymbol{\xi}$ with finite element mapping $\boldsymbol{f}_{\mathrm{m}}$ of polynomial degree $k_{\mathrm{m}} = 2$.

With respect to the argument $\boldsymbol{\chi}$, the map $\boldsymbol{f}_{\mathrm{G}}$ is a homeomorphism for all times, and the argument t describes a continuous deformation over time. In the context of this work and for the numerical results shown below, $\boldsymbol{f}_{\mathrm{G}}$ will be an analytically defined, smooth function in space and time. An illustration is shown in Figure 1. Without loss of generality, we assume $\boldsymbol{x}\,(\boldsymbol{\chi}, t = 0) = \boldsymbol{\chi}$, and therefore $\Omega(t = 0) = \Omega_0$. To demonstrate high order of accuracy of the multistep BDF time integration schemes that are used in this work and that require a starting procedure to demonstrate the formal order of accuracy, it is essential that the mesh motion is continuously differentiable in time. In the context of fluid–structure interaction, the mesh motion is defined by the deformation of the fluid–structure interface according to the structural displacements and a mesh smoothing algorithm calculating the mesh deformation in the interior of the fluid domain.

The incompressible Navier–Stokes equations (5) and (6) are subject to the initial condition

$$\boldsymbol{u}(\boldsymbol{x}, t = 0) = \boldsymbol{u}_0(\boldsymbol{x}) \ \text{ in } \Omega_0 \ ,$$

where $\boldsymbol{u}_0(\boldsymbol{x})$ is divergence-free and fulfills the velocity Dirichlet boundary condition shown below. On the boundary $\Gamma = \partial\Omega = \Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}}$ with $\Gamma^{\mathrm{D}} \cap \Gamma^{\mathrm{N}} = \emptyset$, Dirichlet and Neumann boundary conditions are prescribed

$$\boldsymbol{u} = \boldsymbol{g}_u \ \text{ on } \Gamma^{\mathrm{D}}(t) \ , \tag{7}$$

$$(\boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}) - p\boldsymbol{I}) \cdot \boldsymbol{n} = \boldsymbol{h} \ \text{ on } \Gamma^{\mathrm{N}}(t) \ , \tag{8}$$

where $\boldsymbol{n}$ is the outward pointing unit normal vector and $\boldsymbol{I}$ the identity matrix. As explained in [41], the Neumann boundary condition is split into a viscous part $\boldsymbol{h}_u$ and a pressure part g_p in case of projection-type Navier–Stokes solvers, i.e., $\boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}) \cdot \boldsymbol{n} = \boldsymbol{h}_u$ and $p = g_p$ on $\Gamma^{\mathrm{N}}(t)$ with $\boldsymbol{h} = \boldsymbol{h}_u - g_p\boldsymbol{n}$. Another characteristic of the incompressible Navier–Stokes equations is that the pressure is only defined up to an additive constant in case of pure Dirichlet boundary conditions, $\Gamma = \Gamma^{\mathrm{D}}$. A unique pressure solution is obtained from the constraint $\int_\Omega p \ \mathrm{d}\Omega = 0$. In that special case the velocity Dirichlet boundary condition has to fulfill the constraint $\int_{\Gamma^{\mathrm{D}}} \boldsymbol{g}_u \cdot \boldsymbol{n} \ \mathrm{d}\Gamma = 0$, which is the integral version of the continuity equation (6), transformed into a surface integral via Gauss' divergence theorem.

3. Temporal discretization

A multitude of solution strategies have been proposed over the last decades to solve the incompressible Navier–Stokes equations. While monolithic approaches are straight-forward in terms of time integration and

in achieving high-order accuracy, projection methods with a splitting of velocity and pressure unknowns are particularly interesting from the point of view of computational costs, as these techniques decompose the problem into easier-to-solve equations such as simple Poisson or Helmholtz-like problems. The literature on projection methods is vast, see [52, 53] for an overview. Here, we focus on those methods that we believe are most widely used and that also cover different aspects related to implicit versus mixed explicit–implicit formulations and the availability of high-order formulations, e.g., through rotational formulations. As representatives of projection methods, we investigate (incremental) pressure-correction schemes in rotational form, see [54, 55, 56, 57, 58, 59] for the development of this approach, as well as velocity-correction schemes, using the high-order formulation proposed in [39, 40]. For these methods, the splitting of the incompressible Navier–Stokes equations is performed on the level of differential equations, as compared to algebraic splitting methods. In the following subsections 3.1, 3.2, and 3.3 we briefly summarize the different solution strategies considered in this work with a focus on the ALE relevant aspects and especially boundary conditions, while we refer to previous works [41, 29] for a more detailed description in the context of Eulerian formulations and high-order DG methods. In Section 3.4, the different Navier–Stokes solvers are discussed in terms of stability and achievable rates of convergence.

Notation Backward differentiation formula (BDF) time integration is used in this work. The time interval $[0, T]$ is divided into N time steps of variable size. With $n = 0, ..., N - 1$ denoting the time step number, the equations are advanced from time t_n to $t_{n+1} = t_n + \Delta t_n$ in time step n, leading to the time grid $\{t_i\}_{i=0}^{N} = \{t_0 + \sum_{j=0}^{i-1} \Delta t_j\}_{i=0}^{N}$. BDF schemes of order $J = 1, 2, 3$ are considered. Although A-stability is only achieved for time integration schemes of order $J = 1, 2$, third-order accurate schemes have been found to be useful for practical problems as well and will be investigated in this work. Coefficients γ_0 and α_i of the BDF time integration scheme as well as coefficients β_i of the extrapolation scheme used to extrapolate explicit terms are listed in Table 1 for the case of a constant time step size $\Delta t = T/N$, see also [40]. An extension to variable time step sizes is straight-forward, where the coefficients γ_0^n, α_i^n, and β_i^n vary from one time step to the next and can be expressed as simple rational functions of the time step sizes $\Delta t_n, ..., \Delta t_{n-J+1}$, see also [60]. The time integration constant for adaptive time-stepping are summarized in Appendix A.

Table 1: Coefficients of BDF time integration scheme and extrapolation scheme for constant time step size, see [40].

Order	γ_0	α_0	α_1	α_2	β_0	β_1	β_2
1	1	1	-	-	1	-	-
2	3/2	2	$-1/2$	-	2	-1	-
3	11/6	3	$-3/2$	1/3	3	-3	1

3.1. Coupled solution approach

Applying the BDF scheme to equations (5) and (6) and using a fully implicit formulation (including the convective term), we obtain

$$\frac{\gamma_0^n \boldsymbol{u}^{n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}^{n-i}}{\Delta t_n}\bigg|_{\boldsymbol{\chi}} + \left(\left(\boldsymbol{u}^{n+1} - \boldsymbol{u}_{\mathrm{G}}^{n+1}\right) \cdot \nabla\right) \boldsymbol{u}^{n+1} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}^{n+1}) + \nabla p^{n+1} = \boldsymbol{f}\left(t_{n+1}\right), \quad (9)$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0, \quad (10)$$

where $\big|_{\boldsymbol{\chi}}$ means that all terms of the BDF sum are evaluated at constant $\boldsymbol{\chi}$, i.e., an ALE-type time derivative has to be considered. The boundary conditions are

$$\boldsymbol{u}^{n+1} = \boldsymbol{g}_u^{n+1} \;\; \text{on } \Gamma^{\mathrm{D}},$$

$$\left(\boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}^{n+1}) - p^{n+1}\boldsymbol{I}\right) \cdot \boldsymbol{n} = \boldsymbol{h}^{n+1} \;\; \text{on } \Gamma^{\mathrm{N}}.$$

6

As an alternative formulation, we also study an explicit formulation of convective term, discretized in time via an extrapolation scheme of order J

$$\left.\frac{\gamma_0^n \boldsymbol{u}^{n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}^{n-i}}{\Delta t_n}\right|_{\boldsymbol{\chi}} + \sum_{i=0}^{J-1} \beta_i^n \left(\left(\boldsymbol{u}^{n-i} - \boldsymbol{u}_{\mathrm{G}}^{n+1}\right) \cdot \nabla\right) \boldsymbol{u}^{n-i} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}^{n+1}) + \nabla p^{n+1} = \boldsymbol{f}\left(t_{n+1}\right) ,$$

$$(11)$$

where the following boundary condition will be used in the convective term

$$\boldsymbol{u}^{n-i} = \boldsymbol{g}_u^{n+1} \ \text{ on } \Gamma^{\mathrm{D}} . \tag{12}$$

Remark While the boundary condition $\boldsymbol{u}^{n-i} = \boldsymbol{g}_u^{n-i}$ might be considered a natural formulation as well, it is interesting to study whether equation (12) also preserves optimal rates of convergence. The advantage of this formulation is that only one version of the boundary condition is required at any one time of the solution of the transient problem, therefore easing implementation without the need to store and keep track of previous versions of the boundary condition for fluid–structure interaction problems. For the projection methods discussed below, we will see that we can not fully maintain this goal as these methods are more involved regarding the formulation of boundary conditions.

3.2. High-order dual splitting scheme

The high-order dual splitting scheme [40] consists of the following four sub-steps to be solved in each time step

$$\left.\frac{\gamma_0^n \hat{\boldsymbol{u}} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}^{n-i}}{\Delta t_n}\right|_{\boldsymbol{\chi}} = -\sum_{i=0}^{J-1} \beta_i^n \left(\left(\boldsymbol{u}^{n-i} - \boldsymbol{u}_{\mathrm{G}}^{n+1}\right) \cdot \nabla\right) \boldsymbol{u}^{n-i} + \boldsymbol{f}\left(t_{n+1}\right),$$
$$\boldsymbol{u}^{n-i} = \boldsymbol{g}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},$$

$$(13)$$

$$-\nabla^2 p^{n+1} = -\frac{\gamma_0^n}{\Delta t_n} \nabla \cdot \hat{\boldsymbol{u}},$$
$$\nabla p^{n+1} \cdot \boldsymbol{n} = h_p^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},$$
$$p^{n+1} = g_p^{n+1} \quad \text{on } \Gamma^{\mathrm{N}},$$
$$\hat{\boldsymbol{u}} = \boldsymbol{g}_{\hat{u}}^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},$$

$$(14)$$

$$\hat{\hat{\boldsymbol{u}}} = \hat{\boldsymbol{u}} - \frac{\Delta t_n}{\gamma_0^n} \nabla p^{n+1},$$
$$p^{n+1} = g_p^{n+1} \quad \text{on } \Gamma^{\mathrm{N}},$$

$$(15)$$

$$\frac{\gamma_0^n}{\Delta t_n} \boldsymbol{u}^{n+1} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}}\left(\boldsymbol{u}^{n+1}\right) = \frac{\gamma_0^n}{\Delta t_n} \hat{\hat{\boldsymbol{u}}},$$
$$\boldsymbol{u}^{n+1} = \boldsymbol{g}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},$$
$$\boldsymbol{F}_{\mathrm{v}}(\boldsymbol{u}^{n+1}) \cdot \boldsymbol{n} = \boldsymbol{h}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{N}}.$$

$$(16)$$

The convective term, the pressure term, and the viscous term are taken into account in different sub-steps, and the pressure Poisson equation (14) is obtained from equation (15) by requiring $\nabla \cdot \hat{\hat{\boldsymbol{u}}} = 0$. The viscous term is formulated implicitly in time, while the convective term is formulated explicitly for this scheme. The authors are not aware of variants of this splitting scheme that allow an implicit treatment of the convective term and – at the same time – achieve higher-order accuracy in time.

The pressure Neumann boundary condition h_p and the Dirichlet boundary condition $\boldsymbol{g}_{\hat{u}}$ for the intermediate velocity need to be discussed in more detail. A consistent Neumann boundary condition for the pressure

is derived by multiplying the momentum equation of the incompressible Navier–Stokes equations by the normal vector $\boldsymbol{n}$ [39, 40, 53], which yields the following result in case of the ALE form of the incompressible Navier–Stokes equations

$$
\begin{aligned}
h_p\left(t_{n+1}\right) = -&\left[\left.\frac{\gamma_0^n \boldsymbol{g}_u^{n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{g}_u^{n-i}}{\Delta t_n}\right|_{\boldsymbol{\chi}} - \boldsymbol{f}\left(t_{n+1}\right)\right] \cdot \boldsymbol{n}^{n+1} \\
-&\left[\sum_{i=0}^{J_p-1} \beta_i^n\left(\left(\left(\boldsymbol{u}^{n-i} - \boldsymbol{u}_{\mathrm{G}}^{n+1}\right) \cdot \nabla\right) \boldsymbol{u}^{n-i} + \nu \nabla \times \boldsymbol{\omega}^{n-i}\right)\right] \cdot \boldsymbol{n}^{n+1} .
\end{aligned}
\tag{17}
$$

As already noted in our previous work [41] dealing with the Eulerian case, the time derivative term and body force term appear on the right-hand side as well in the general case of time dependent boundary conditions and right-hand side vectors $\boldsymbol{f} \neq \boldsymbol{0}$, compared to the original formulation in [40] and a later work [6] where these terms are dropped. Compared to [41] where the exact derivative $\partial \boldsymbol{g}_u / \partial t$ is used by exploiting that this term is known analytically in the Eulerian case on Γ^{D}, the acceleration term has to be replaced by a discrete BDF time derivative in the ALE or fluid–structure interaction case where the boundary condition is only known at discrete times. Hence, one has to record the history of Dirichlet boundary values $\boldsymbol{g}_u$ in case of the dual splitting scheme. Since the time derivative is of ALE-type at constant $\boldsymbol{\chi}$, the convective term needs to be formulated in ALE form as well. The velocity $\boldsymbol{u}_h^{n+1}$ at time t_{n+1} is unknown at this point of the algorithm, so that the convective term and the viscous term need to be formulated explicitly. For the viscous term we use the well-known rotational formulation $\nabla \times \boldsymbol{\omega}$ (with vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$) which is obtained from the identity $\nabla^2 \boldsymbol{u} = \nabla(\nabla \cdot \boldsymbol{u}) - \nabla \times (\nabla \times \boldsymbol{u}) = -\nabla \times (\nabla \times \boldsymbol{u}) = -\nabla \times \boldsymbol{\omega}$. This formulation proposed by [39, 40] makes use of the incompressibility constraint $\nabla \cdot \boldsymbol{u} = 0$, and it is well understood that the rotational formulation significantly improves accuracy as compared to the Laplace formulation [53, 52]; nevertheless the latter formulation $\nabla^2 \boldsymbol{u}$ is also used sometimes [61, 62, 38].

The velocity Dirichlet boundary condition for the intermediate velocity $\hat{\boldsymbol{u}}$ is derived in a similar fashion from equation (13), see also [41] where this boundary condition has been proposed for the Eulerian case

$$
\boldsymbol{g}_{\hat{u}}\left(\boldsymbol{\chi}, t_{n+1}\right) = \sum_{i=0}^{J-1} \frac{\alpha_i^n}{\gamma_0^n} \boldsymbol{g}_u\left(\boldsymbol{\chi}, t_{n-i}\right) - \frac{\Delta t_n}{\gamma_0^n} \sum_{i=0}^{J-1} \beta_i^n\left(\left(\boldsymbol{u}^{n-i} - \boldsymbol{u}_{\mathrm{G}}^{n+1}\right) \cdot \nabla\right) \boldsymbol{u}^{n-i} + \frac{\Delta t_n}{\gamma_0^n} \boldsymbol{f}\left(t_{n+1}\right) ,
\tag{18}
$$

As indicated in the above equation, the history of the boundary condition $\boldsymbol{g}_u$ is evaluated in grid coordinates $\boldsymbol{\chi}$ following the moving mesh from one time instant to the next, so that the ALE form of the convective term is required in this boundary condition similar to the pressure Neumann boundary condition (17). The convective term again needs to be extrapolated.

Remark One might raise the question why the terms in equation (17) coming from the acceleration term evaluate the prescribed boundary data $\boldsymbol{g}_u$ instead of simply evaluating the numerical solution $\boldsymbol{u}$ coming from the interior of the domain, since this is also done for the convective and viscous terms in the boundary conditions. From the formulations in [22, 24, 63, 62, 38] it is unclear whether the acceleration term in equation (17) should directly evaluate the numerical solution $\boldsymbol{u}$ if the time derivative is not known analytically. The same holds for the BDF time derivative terms in equation (18). We found that using $\boldsymbol{u}$ instead of $\boldsymbol{g}_u$ for the time derivative terms in equations (17) and (18) leads to instabilities that occur for small time step sizes. We therefore recommend to use the available boundary data $\boldsymbol{g}_u$ in the above boundary conditions whenever possible. We further note that the works [22, 24] do not use a boundary condition like equation (18) since these works use a different DG discretization of the velocity divergence term in equation (14) compared to the present work, namely a formulation without integration by parts and, hence, without imposition of boundary conditions, see Section 4 for details. However, it was shown in [41] that integration by parts of this term in the context of discontinuous Galerkin methods is essential to obtain stability for small time steps and that a consistent Dirichlet boundary condition according to equation (18) is necessary to obtain high-order accuracy in time.

3.3. Pressure-correction scheme

Pressure-correction schemes are another class of projection methods, treating both convective and viscous terms in the same sub-step, see [52] for detailed information. Extending the formulation of pressure-correction schemes for non-moving meshes shown in [41, 29] to the ALE formulation of the incompressible Navier–Stokes equations, the pressure-correction schemes can be summarized as follows

$$
\left. \frac{\gamma_0^n \hat{\boldsymbol{u}} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}^{n-i}}{\Delta t_n} \right|_{\boldsymbol{\chi}} + \left(\left(\hat{\boldsymbol{u}} - \boldsymbol{u}_{\mathrm{G}}^{n+1} \right) \cdot \nabla \right) \hat{\boldsymbol{u}} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}} \left(\hat{\boldsymbol{u}} \right) =
$$

$$
- \sum_{i=0}^{J_p-1} \beta_i^n \nabla p^{n-i} + \boldsymbol{f} \left(t_{n+1} \right),
$$

$$
\hat{\boldsymbol{u}} = \boldsymbol{g}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},
$$
$$
\boldsymbol{F}_{\mathrm{v}}(\hat{\boldsymbol{u}}) \cdot \boldsymbol{n} = \boldsymbol{h}_u^{n+1}, \quad \text{on } \Gamma^{\mathrm{N}},
$$
$$
p^{n-i} = g_p^{n-i} \quad \text{on } \Gamma^{\mathrm{N}},
$$
(19)

$$
-\nabla^2 \phi^{n+1} = -\frac{\gamma_0^n}{\Delta t_n} \nabla \cdot \hat{\boldsymbol{u}},
$$
$$
\nabla \phi^{n+1} \cdot \boldsymbol{n} = h_\phi^{n+1} = 0 \quad \text{on } \Gamma^{\mathrm{D}},
$$
$$
\hat{\boldsymbol{u}} = \boldsymbol{g}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},
$$
$$
\phi^{n+1} = g_\phi^{n+1} \quad \text{on } \Gamma^{\mathrm{N}},
$$
(20)

$$
p^{n+1} = \phi^{n+1} + \sum_{i=0}^{J_p-1} \beta_i^n p^{n-i} - \zeta \nu \nabla \cdot \hat{\boldsymbol{u}},
$$
$$
\hat{\boldsymbol{u}} = \boldsymbol{g}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{D}},
$$
(21)

$$
\boldsymbol{u}^{n+1} = \hat{\boldsymbol{u}} - \frac{\Delta t_n}{\gamma_0^n} \nabla \phi^{n+1},
$$
$$
\phi^{n+1} = g_\phi^{n+1} \quad \text{on } \Gamma^{\mathrm{N}}.
$$
(22)

Here, J_p is the order of extrapolation of the pressure gradient term. Schemes with $J_p = 0$ are called non-incremental, and schemes with $J_p \geq 1$ incremental pressure-correction schemes. The standard formulation is obtained for $\zeta = 0$, while the rotational formulation corresponds to $\zeta = 1$. We exclusively study the rotational version in the present work due to improved convergence rates and accuracy [52]. In the above equations, the convective term is formulated implicitly. As for the monolithic solver, we consider an alternative formulation with explicit treatment of the convective term, resulting in the following momentum equation in the first sub-step

$$
\left. \frac{\gamma_0^n \hat{\boldsymbol{u}} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}^{n-i}}{\Delta t_n} \right|_{\boldsymbol{\chi}} - \nabla \cdot \boldsymbol{F}_{\mathrm{v}} \left(\hat{\boldsymbol{u}} \right) = -\sum_{i=0}^{J-1} \beta_i^n \left(\left(\boldsymbol{u}^{n-i} - \boldsymbol{u}_{\mathrm{G}}^{n+1} \right) \cdot \nabla \right) \boldsymbol{u}^{n-i} - \sum_{i=0}^{J_p-1} \beta_i^n \nabla p^{n-i} + \boldsymbol{f} \left(t_{n+1} \right),
$$
(23)

where the following boundary condition is imposed for the convective term

$$
\boldsymbol{u}^{n-i} = \boldsymbol{g}_u^{n+1} \quad \text{on } \Gamma^{\mathrm{D}} .
$$
(24)

The pressure Poisson equation (20) is subject to the pressure Dirichlet boundary condition

$$
g_\phi(\boldsymbol{\chi}, t_{n+1}) = g_p \left(\boldsymbol{\chi}, t_{n+1} \right) - \sum_{i=0}^{J_p-1} \beta_i g_p \left(\boldsymbol{\chi}, t_{n-i} \right) .
$$
(25)

9

Remark As mentioned in [41] for the Eulerian version of the present solver, the boundary condition (25) is an extension of the boundary condition (10.3) in [52] towards time-dependent pressure boundary conditions on Γ^N. Note that we prescribe $p^{n-i} = g_p^{n-i}$ in equation (19) to be consistent with boundary condition (25). We otherwise observe suboptimal rates of convergence when prescribing $p^{n-i} = g_p^{n+1}$. Hence, a history of pressure Dirichlet boundary values on Γ^N has to be stored for the pressure-correction scheme for higher-order schemes with $J_p \geq 1$, which can be seen as a consequence of the operator splitting as compared to the monolithic solver described in Section 3.1.

3.4. Discussion of incompressible Navier–Stokes solvers

Let us briefly summarize and discuss the different Navier–Stokes solver strategies. Already when considering the unsteady Stokes equations without convection, projection methods are in general only conditionally stable for higher extrapolation order J_p. According to [64], the dual splitting scheme is unconditionally stable for $J_p \leq 2$ independent of the order $1 \leq J \leq 4$ of the BDF scheme, but only conditionally stable for $J_p > 2$. Hence, the scheme with parameters $J = 3$ and $J_p = 2$ provides the highest order of accuracy, namely Δt^3, among the schemes that are unconditionally stable. Pressure-correction schemes are only unconditionally stable for $J_p \leq 1$, while they are conditionally stable for $J_p \geq 2$, see [52]. This is also in agreement with our numerical results where we observe instabilities for $J_p = 2$. Hence, the highest accuracy combined with unconditional stability is achieved for $J = 2, J_p = 1$, resulting in a second-order accurate scheme, Δt^2. Let us mention that the scheme $J = 3, J_p = 2$ shows indeed third-order accuracy, but is only conditionally stable and not suited for practical problems. From these considerations, we derive the parameters $J_p = \min(2, J), J \leq 3$ for the dual splitting scheme, and $J_p = \min(2, J) - 1, J \leq 2$ for the pressure-correction scheme as an optimal choice. The monolithic solution approach and the dual splitting scheme have the advantage that the third-order schemes are stable. The fact that the pressure-correction scheme allows both implicit and explicit formulations of the convective term can be seen as an advantage over the dual splitting scheme, especially when one wants to avoid restrictions of the time step size according to the CFL condition and when high-order of accuracy is not the primary target. Finally, the simpler structure of algebraic equations is often considered an advantage of projection methods over the monolithic approach. We therefore analyze these different methods in the present work since we believe they cover different facets of incompressible Navier–Stokes solvers.

4. Spatial discretization

Notation The computational domain $\Omega_h = \bigcup_{e=1}^{N_{\text{el}}} \Omega_e \in \mathbb{R}^d$ is composed of N_{el} finite elements, which are non-overlapping and of quadrilateral/hexahedral shape in the context of this work. The boundary $\Gamma_h = \partial \Omega_h$ approximates Γ, and it holds $\Gamma_h = \Gamma_h^D \cup \Gamma_h^N$ with $\Gamma_h^D \cap \Gamma_h^N = \emptyset$ as in the spatially continuous case. Approximations to velocity $\boldsymbol{u}(\boldsymbol{x}, t)$ and pressure $p(\boldsymbol{x}, t)$ are denoted by $\boldsymbol{u}_h(\boldsymbol{x}, t) \in \mathcal{V}_h^u$ and $p_h(\boldsymbol{x}, t) \in \mathcal{V}_h^p$, where the discontinuous Galerkin finite element spaces of test and trial functions are defined as

$$\mathcal{V}_h^u = \left\{ \boldsymbol{u}_h \in [L_2(\Omega_h)]^d \; : \; \boldsymbol{u}_h\left(\boldsymbol{x}^e(\boldsymbol{\xi}, t)\right)|_{\Omega_e} = \tilde{\boldsymbol{u}}_h^e(\boldsymbol{\xi})|_{\tilde{\Omega}_e} \in \mathcal{V}_{h,e}^u = [\mathcal{Q}_{k_u}(\tilde{\Omega}_e)]^d , \;\; \forall e = 1, \ldots, N_{\text{el}} \right\} \; ,$$

$$\mathcal{V}_h^p = \left\{ p_h \in L_2(\Omega_h) \; : \; p_h\left(\boldsymbol{x}^e(\boldsymbol{\xi}, t)\right)|_{\Omega_e} = \tilde{p}_h^e(\boldsymbol{\xi})|_{\tilde{\Omega}_e} \in \mathcal{V}_{h,e}^p = \mathcal{Q}_{k_p}(\tilde{\Omega}_e) , \;\; \forall e = 1, \ldots, N_{\text{el}} \right\} \; .$$

The polynomial space $\mathcal{Q}_k(\tilde{\Omega}_e)$ of tensor degree $\leq k$ is defined on the reference element $\tilde{\Omega}_e = [0, 1]^d$ with reference coordinates $\boldsymbol{\xi} = (\xi_1, ..., \xi_d)^\mathsf{T}$. We approximate velocity and pressure on element e by nodal Lagrange polynomials

$$\tilde{\boldsymbol{u}}_h^e(\boldsymbol{\xi}, t) = \sum_{i_1, \ldots, i_d = 0}^{k_u} l_{i_1 \ldots i_d}^{k_u}(\boldsymbol{\xi}) \boldsymbol{u}_{i_1 \ldots i_d}^e(t) \;\; , \;\; \tilde{p}_h^e(\boldsymbol{\xi}, t) = \sum_{i_1, \ldots, i_d = 0}^{k_p} l_{i_1 \ldots i_d}^{k_p}(\boldsymbol{\xi}) p_{i_1 \ldots i_d}^e(t) \; , \tag{26}$$

where $\boldsymbol{u}_{i_1 \ldots i_d}^e$ and $p_{i_1 \ldots i_d}^e$ denote the nodal degrees of freedom of the velocity and pressure solution on element e, respectively. The multidimensional shape functions $l_{i_1 \ldots i_d}^k$ are given as the tensor product of

10

one-dimensional shape functions, $l^k_{i_1 \dots i_d}(\boldsymbol{\xi}) = \prod_{n=1}^{d} l^{k,1D}_{i_n}(\xi_n)$, where $l^{k,1D}_i(\xi)$ are the Lagrange polynomials of degree k based on the Legendre–Gauss–Lobatto nodes.

For reasons of inf–sup stability, the polynomial degree for the pressure is $k_p = k_u - 1$, see [41]. For ease of notation, we simply write $k_u = k$ in the following. In the above equations, $\boldsymbol{x}^e(\boldsymbol{\xi},t) : \tilde{\Omega}_e \to \Omega_e(t)$ denotes the mapping from reference space to physical space

$$\boldsymbol{f}^e_{\mathrm{m}} : \begin{cases} \tilde{\Omega}_e \times [0,T] \to \Omega_e(t), \ \tilde{\Omega}_e = [0,1]^d, \Omega_e(t) \subset \mathbb{R}^d, \\ (\boldsymbol{\xi},t) \mapsto \boldsymbol{x}^e(\boldsymbol{\xi},t) . \end{cases}$$

For the mapping, the same ansatz is used as for approximating the solution, but with polynomial degree k_{m}

$$\boldsymbol{x}^e(\boldsymbol{\xi},t) = \sum_{i_1,\dots,i_d=0}^{k_{\mathrm{m}}} l^{k_{\mathrm{m}}}_{i_1 \dots i_d}(\boldsymbol{\xi}) \boldsymbol{x}^e_{i_1 \dots i_d}(t) .$$

The mapping can be seen in analogy to the function $\boldsymbol{f}_{\mathrm{G}}$ describing the topological changes of the domain Ω. While $\boldsymbol{f}_{\mathrm{G}}$ is defined globally for the whole domain and in a spatially continuous way, the finite element mapping describes the mesh motion for each element of the mesh in the discrete setting and is of finite dimension. For the following derivations, it is important to realize that a point with constant $\boldsymbol{\chi}$ can be thought of as a point with fixed $\boldsymbol{\xi}$ coordinates within one element, i.e., there exists a bijective map between $\boldsymbol{\chi}$ and $\boldsymbol{\xi}$ for each element. An illustration is given in Figure 1. In the context of this work, we restrict ourselves to problems for which the topology of the mesh does not change, i.e., no remeshing. Hence, the data structures do not have to be adjusted dynamically when moving the mesh. Updating the mesh means updating the $d(k_{\mathrm{m}} + 1)^d$ mapping degrees of freedom per element, i.e.,

$$\boldsymbol{x}^e_{i_1 \dots i_d}(t^{n+1}) = \boldsymbol{f}_{\mathrm{G}}(\boldsymbol{x}^e_{i_1 \dots i_d}(t = 0), t^{n+1}) .$$

Moderately large deformations are possible as long as $\boldsymbol{f}_{\mathrm{G}}$ remains invertible, and invalid elements with invalid mapping or Jacobian will otherwise occur in the discrete setting. The numerical examples shown in this work use a high-order, isoparametric mapping with $k_{\mathrm{m}} = k_u$.

In DG methods, integrals have to be computed over the interface $f_{e^-/e^+} = \partial\Omega_{e^-} \cap \partial\Omega_{e^+}$ of two adjacent elements Ω_{e^-} and Ω_{e^+}, where the outward pointing normal vectors are $\boldsymbol{n}^-$ for Ω_{e^-} and $\boldsymbol{n}^+$ for Ω_{e^+}. Furthermore, let u_h^- and u_h^+ denote the solution u_h on f_{e^-,e^+} evaluated from the interior of element e^- and element e^+, respectively. By Γ_h^{int} we denote the set of all interior faces. In the following, we make use of the average operator $\{\{u\}\} = (u^- + u^+)/2$, the jump operator $[\![u]\!] = u^- \otimes \boldsymbol{n}^- + u^+ \otimes \boldsymbol{n}^+$, and the oriented jump operator $[u] = u^- - u^+$. Moreover, an element-by-element formulation is used where volume integrals are performed over the current element Ω_e and face integrals over the boundary $\partial\Omega_e$ of element e. By definition, we denote interior information on the current element Ω_e by the superscript $(\cdot)^-$ and exterior information from neighboring elements by the superscript $(\cdot)^+$. In this context, the normal vector $\boldsymbol{n}$ equals $\boldsymbol{n}^-$, while $\boldsymbol{n}^+ = -\boldsymbol{n}^- = -\boldsymbol{n}$. Finally, we introduce the abbreviations $(v,u)_{\Omega_e} = \int_{\Omega_e} v \odot u \, \mathrm{d}\Omega$ and $(v,u)_{\partial\Omega_e} = \int_{\partial\Omega_e} v \odot u \, \mathrm{d}\Gamma$, where the operator $\odot$ symbolizes inner products and will become clear from the context. An integral over the computational domain is to be understood as $(v,u)_{\Omega_h} = \sum_{e=1}^{N_{\mathrm{el}}} (v,u)_{\Omega_e}$, and similarly for integrals over all interior faces, e.g., $(\{\{v\}\}, u^*)_{\Gamma_h^{\mathrm{int}}} = \sum_{e=1}^{N_{\mathrm{el}}} \left(\frac{1}{2}v, u^*\right)_{\partial\Omega_e \backslash \Gamma_h}$ if u^* is single-valued.

Due to the ansatz (26) with a separation of space and time, applying the temporal discretization to the spatially discretized equations in ALE form becomes trivial in the sense that the structure of the equations is equivalent to the Eulerian case. Consider the time derivative term in equation (5) multiplied by test

functions $\boldsymbol{v}_h$, integrated over element $\Omega_e(t = t_{n+1})$, and to be discretized in time

$$
\begin{aligned}
\left(\boldsymbol{v}_h, \left.\frac{\partial \boldsymbol{u}_h(t)}{\partial t}\right|_{\boldsymbol{\chi}}\right)_{\Omega_e^{n+1}} &= \left(\boldsymbol{v}_h, \left.\frac{\partial \sum_{i_1,\ldots,i_d=0}^{k_u} l_{i_1\ldots i_d}^{k_u}(\boldsymbol{\xi})\boldsymbol{u}_{i_1\ldots i_d}^e(t)}{\partial t}\right|_{\boldsymbol{\xi}}\right)_{\Omega_e^{n+1}} \\
&= \left(\boldsymbol{v}_h, \sum_{i_1,\ldots,i_d=0}^{k_u} l_{i_1\ldots i_d}^{k_u}(\boldsymbol{\xi})\frac{\partial \boldsymbol{u}_{i_1\ldots i_d}^e(t)}{\partial t}\right)_{\Omega_e^{n+1}} \\
&\approx \left(\boldsymbol{v}_h, \sum_{i_1,\ldots,i_d=0}^{k_u} l_{i_1\ldots i_d}^{k_u}(\boldsymbol{\xi})\frac{\gamma_0^n \boldsymbol{u}_{i_1\ldots i_d}^{e,n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}_{i_1\ldots i_d}^{e,n-i}}{\Delta t_n}\right)_{\Omega_e^{n+1}} \\
&= \left(\boldsymbol{v}_h, \left.\frac{\gamma_0^n \boldsymbol{u}_h^{n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}_h^{n-i}}{\Delta t_n}\right|_{\boldsymbol{\xi}}\right)_{\Omega_e^{n+1}} .
\end{aligned}
$$

The BDF rule introduced in the third row of the above equation approximates the acceleration at time t_{n+1} consistently with the integral over Ω_e taken at the same instant of time. In the following, we skip the label $|_{\boldsymbol{\xi}}$ for simplicity, as it is clear from the above derivation that the BDF rule is simply applied to the global solution vector containing the unknown degrees of freedom and that all terms of the BDF sum use the same mass matrix at time t_{n+1}. As explained in more detail in Section 4.4, using the same mass matrix for all solution vectors is important in order to satisfy the geometric conservation law [15]. The above equation highlights that discretization in space and time commutate, meaning that the last term of the above equation would have also been obtained by discretizing equation (9) in space. However, the projection-type solution methods considered in this work are already formulated in a time-discrete manner, since the splitting is performed on the level of differential operators. For this reason, the derivation of DG formulations shown in the following starts from the time-discrete problems stated in Section 3.

Following [15], the grid velocity is computed in the same way via a BDF time derivative of the nodal grid coordinates $\boldsymbol{x}_{i_1\ldots i_d}^e$ in the time discrete case in order to achieve high-order temporal convergence on moving meshes

$$
\boldsymbol{u}_{\mathrm{G},h}^{n+1} = \left.\frac{\partial \boldsymbol{x}_h}{\partial t}(t_{n+1})\right|_{\boldsymbol{\chi}} \approx \left.\frac{\gamma_0^n \boldsymbol{x}_h^{n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{x}_h^{n-i}}{\Delta t_n}\right|_{\boldsymbol{\xi}} . \tag{27}
$$

This procedure is different from [5, 6] where the grid coordinates are updated by integrating the mesh velocity forward in time. In the following, we summarize the variational formulation of the different ALE Navier–Stokes solvers.

Table 2: Weak imposition of boundary conditions: choice of exterior values $(\cdot)^+$ on domain boundaries as a function of interior values $(\cdot)^-$ and prescribed boundary data for velocity and pressure in order to weakly impose boundary conditions according to the mirror principle. The procedure is equivalent to the Eulerian case [41].

	Γ_h^{D}	Γ_h^{N}
velocity	$\boldsymbol{u}_h^+ = -\boldsymbol{u}_h^- + 2\boldsymbol{g}_u$	$\boldsymbol{u}_h^+ = \boldsymbol{u}_h^-$
	$\nabla \boldsymbol{u}_h^+ \cdot \boldsymbol{n} = \nabla \boldsymbol{u}_h^- \cdot \boldsymbol{n}$	$\nabla \boldsymbol{u}_h^+ \cdot \boldsymbol{n} = -\nabla \boldsymbol{u}_h^- \cdot \boldsymbol{n} + \frac{2\boldsymbol{h}_u}{\nu}$
pressure	$p_h^+ = p_h^-$	$p_h^+ = -p_h^- + 2g_p$
	$\nabla p_h^+ \cdot \boldsymbol{n} = -\nabla p_h^- \cdot \boldsymbol{n} + 2h_p$	$\nabla p_h^+ \cdot \boldsymbol{n} = \nabla p_h^- \cdot \boldsymbol{n}$

4.1. Monolithic solution approach

Beginning with the monolithic solution approach, the weak discontinuous Galerkin formulation of the fully discrete problem with implicit formulation of the convective term can be summarized as follows: Find $\boldsymbol{u}_h^{n+1} \in \mathcal{V}_h^u$, $p_h^{n+1} \in \mathcal{V}_h^p$ such that

$$
\left(\boldsymbol{v}_h, \frac{\gamma_0^n \boldsymbol{u}_h^{n+1} - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}_h^{n-i}}{\Delta t_n} \right)_{\Omega_e^{n+1}} + c_h^{e,n+1} \left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}, \boldsymbol{u}_{\mathrm{G},h}^{n+1}; \boldsymbol{g}_u^{n+1} \right)
$$
$$
+ v_h^{e,n+1} \left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}; \boldsymbol{g}_u^{n+1}, \boldsymbol{h}_u^{n+1} \right) + g_h^{e,n+1} \left(\boldsymbol{v}_h, p_h^{n+1}; g_p^{n+1} \right)
$$
$$
+ a_{\mathrm{D},h}^{e,n+1}(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}) + a_{\mathrm{C},h}^{e,n+1}(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}; \boldsymbol{g}_u^{n+1}) - (\boldsymbol{v}_h, \boldsymbol{f}(t_{n+1}))_{\Omega_e^{n+1}} = 0 , \tag{28}
$$
$$
- d_h^{e,n+1}(q_h, \boldsymbol{u}_h^{n+1}; \boldsymbol{g}_u^{n+1}) = 0 , \tag{29}
$$

for all $(\boldsymbol{v}_h, q_h) \in \mathcal{V}_{h,e}^u \times \mathcal{V}_{h,e}^p$ and for all elements $e = 1, ..., N_{\mathrm{el}}$. The time label $n+1$, e.g. in $c_h^{e,n+1}$, indicates that the integral is evaluated on the domain Ω_e^{n+1}. When formulating the convective term explicitly, the convective term in the discretized momentum equation (28) is replaced by

$$
c_h^{e,n+1} \left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}, \boldsymbol{u}_{\mathrm{G},h}^{n+1}; \boldsymbol{g}_u^{n+1} \right) \to \sum_{i=0}^{J-1} \beta_i^n c_h^{e,n+1} \left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n-i}, \boldsymbol{u}_{\mathrm{G},h}^{n+1}; \boldsymbol{g}_u^{n+1} \right) .
$$

As proposed in [49], a computational efficient variant of the monolithic system of equations, equations (28) and (29), is to apply the divergence and continuity penalty terms in a postprocessing step

$$
\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1} \right)_{\Omega_e^{n+1}} + a_{\mathrm{D},h}^{e,n+1} \left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1} \right) \Delta t_n + a_{\mathrm{C},h}^{e,n+1} \left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}; \boldsymbol{g}_u^{n+1} \right) \Delta t_n = (\boldsymbol{v}_h, \hat{\boldsymbol{u}}_h)_{\Omega_e^{n+1}} , \tag{30}
$$

where $\hat{\boldsymbol{u}}_h$ is an intermediate velocity obtained as the solution of the coupled system of equations without penalty terms. For the numerical results studied in this work, the formulation shown in equations (28) and (29) is used with penalty terms added to the momentum equation.

We next present the discontinuous Galerkin formulation of the individual terms of the incompressible Navier–Stokes equations, see also [41, 29] for more detailed derivations. Boundary conditions are imposed according to the mirror principle as summarized in Table 2. Central flux functions are used for the velocity divergence term

$$
d_{h,\mathrm{weak}}^e (q_h, \boldsymbol{u}_h; \boldsymbol{g}_u) = - (\nabla q_h, \boldsymbol{u}_h)_{\Omega_e} + (q_h, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n})_{\partial\Omega_e}
$$
$$
= - (\nabla q_h, \boldsymbol{u}_h)_{\Omega_e} + (q_h, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n})_{\partial\Omega_e \backslash \Gamma_h} + (q_h, \boldsymbol{u}_h \cdot \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{N}}} + (q_h, \boldsymbol{g}_u \cdot \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}} , \tag{31}
$$

and for the pressure gradient term

$$
g_{h,\mathrm{weak}}^e (\boldsymbol{v}_h, p_h; g_p) = - (\nabla \cdot \boldsymbol{v}_h, p_h)_{\Omega_e} + (\boldsymbol{v}_h, \{\!\{p_h\}\!\}\boldsymbol{n})_{\partial\Omega_e}
$$
$$
= - (\nabla \cdot \boldsymbol{v}_h, p_h)_{\Omega_e} + (\boldsymbol{v}_h, \{\!\{p_h\}\!\}\boldsymbol{n})_{\partial\Omega_e \backslash \Gamma_h} + (\boldsymbol{v}_h, p_h \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}} + (\boldsymbol{v}_h, g_p \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{N}}} . \tag{32}
$$

As an alternative to the above weak forms, we consider the so-called strong formulations by performing integration-by-parts once again

$$
d_{h,\mathrm{strong}}^e (q_h, \boldsymbol{u}_h) = (q_h, \nabla \cdot \boldsymbol{u}_h)_{\Omega_e} - \left(q_h, \frac{1}{2} [\boldsymbol{u}_h] \cdot \boldsymbol{n} \right)_{\partial\Omega_e} , \tag{33}
$$

$$
g_{h,\mathrm{strong}}^e (\boldsymbol{v}_h, p_h) = (\boldsymbol{v}_h, \nabla p_h)_{\Omega_e} - \left(\boldsymbol{v}_h, \frac{1}{2} [p_h] \boldsymbol{n} \right)_{\partial\Omega_e} . \tag{34}
$$

The weak and strong formulations are equivalent as long as integrals are evaluated exactly, which does not hold in general for the quadrature rules typically used, see Section 4.6. For this reason, the weak

and strong formulations behave differently regarding the fulfillment of the geometric conservation law as discussed in Section 4.4. The discretization of the viscous term is based on the symmetric interior penalty Galerkin (SIPG) method [65]

$$v_h^e(\boldsymbol{v}_h, \boldsymbol{u}_h; \boldsymbol{g}_u, \boldsymbol{h}_u) = (\nabla \boldsymbol{v}_h, \nu \nabla \boldsymbol{u}_h)_{\Omega_e} - \left(\nabla \boldsymbol{v}_h, \frac{\nu}{2}[\![\boldsymbol{u}_h]\!]\right)_{\partial\Omega_e \setminus \Gamma_h} - (\boldsymbol{v}_h, \nu\{\!\{\nabla \boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n})_{\partial\Omega_e \setminus \Gamma_h}$$
$$+ (\boldsymbol{v}_h, \nu\tau[\![\boldsymbol{u}_h]\!] \cdot \boldsymbol{n})_{\partial\Omega_e \setminus \Gamma_h} \ . \tag{35}$$

Inserting the boundary conditions acccording to Table 2, the viscous operator $v_h^e = v_{h,\mathrm{hom}}^e + v_{h,\mathrm{inhom}}^e$ can be split into homogeneous contributions

$$v_{h,\mathrm{hom}}^e(\boldsymbol{v}_h, \boldsymbol{u}_h) = (\nabla \boldsymbol{v}_h, \nu \nabla \boldsymbol{u}_h)_{\Omega_e} - \left(\nabla \boldsymbol{v}_h, \frac{\nu}{2}[\![\boldsymbol{u}_h]\!]\right)_{\partial\Omega_e \setminus \Gamma_h} - (\nabla \boldsymbol{v}_h, \nu \boldsymbol{u}_h \otimes \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}}$$
$$- (\boldsymbol{v}_h, \nu\{\!\{\nabla \boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n})_{\partial\Omega_e \setminus \Gamma_h} - (\boldsymbol{v}_h, \nu \nabla \boldsymbol{u}_h \cdot \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}}$$
$$+ (\boldsymbol{v}_h, \nu\tau[\![\boldsymbol{u}_h]\!] \cdot \boldsymbol{n})_{\partial\Omega_e \setminus \Gamma_h} + (\boldsymbol{v}_h, 2\nu\tau\boldsymbol{u}_h)_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}} \ ,$$

and inhomogeneous contributions

$$v_{h,\mathrm{inhom}}^e(\boldsymbol{v}_h; \boldsymbol{g}_u, \boldsymbol{h}_u) = (\nabla \boldsymbol{v}_h, \nu\, \boldsymbol{g}_u \otimes \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}} - (\boldsymbol{v}_h, \boldsymbol{h}_u)_{\partial\Omega_e \cap \Gamma_h^{\mathrm{N}}} - (\boldsymbol{v}_h, 2\nu\tau\boldsymbol{g}_u)_{\partial\Omega_e \cap \Gamma_h^{\mathrm{D}}} \ .$$

The SIPG penalty parameter τ depends on the polynomial degree k and a characteristic element length h. It has to be large enough to ensure coercivity of the bilinear form. For quadrilateral/hexahedral elements used in the present work, bounds for the penalty parameter have been derived in [66]. Following this work, the penalty parameter τ_e of element e is calculated as

$$\tau_e = (k+1)^2 \frac{A\left(\partial\Omega_e \setminus \Gamma_h\right)/2 + A\left(\partial\Omega_e \cap \Gamma_h\right)}{V\left(\Omega_e\right)} \ , \tag{36}$$

where $V\left(\Omega_e\right) = \int_{\Omega_e} \mathrm{d}\Omega$ and $A(f) = \int_{f \subset \partial\Omega_e} \mathrm{d}\Gamma$ are the element volume and surface area, respectively. The maximum value from both sides is chosen on interior faces, $\tau = \max\left(\tau_{e^-}, \tau_{e^+}\right)$ if face $f \subseteq \partial\Omega_e \setminus \Gamma_h$, while $\tau = \tau_e$ is used on boundary faces $f \subseteq \partial\Omega_e \cap \Gamma_h$.

The convective term is particularly relevant in the ALE context and is written in non-conservative form with grid velocity $\boldsymbol{u}_{\mathrm{G},h}$. We perform integration by parts twice (strong formulation), since we observed sub-optimal rates of convergence for the weak formulation in case of even polynomial degrees. Then, an upwind flux is used as numerical flux function to obtain

$$c_h^e\left(\boldsymbol{v}_h, \boldsymbol{u}_h, \boldsymbol{u}_{\mathrm{G},h}; \boldsymbol{g}_u\right) = (\boldsymbol{v}_h, (\nabla \boldsymbol{u}_h) \cdot (\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}))_{\Omega_e} - (\boldsymbol{v}_h, ((\{\!\{\boldsymbol{u}_h\}\!\} - \boldsymbol{u}_{\mathrm{G},h}) \cdot \boldsymbol{n})\,\boldsymbol{u}_h)_{\partial\Omega_e}$$
$$+ \left(\boldsymbol{v}_h, \underbrace{((\{\!\{\boldsymbol{u}_h\}\!\} - \boldsymbol{u}_{\mathrm{G},h}) \cdot \boldsymbol{n})\{\!\{\boldsymbol{u}_h\}\!\} + \frac{1}{2}|(\{\!\{\boldsymbol{u}_h\}\!\} - \boldsymbol{u}_{\mathrm{G},h}) \cdot \boldsymbol{n}|\,[\![\boldsymbol{u}_h]\!]}_{\text{upwind flux}}\right)_{\partial\Omega_e} \ . \tag{37}$$

To keep the formulation compact, we do not explicitly highlight the dependency of the variational form c_h^e on the boundary condition $\boldsymbol{g}_u$ as it is clear that the boundary condition enters the formulation through the choice of exterior values on domain boundaries according to Table 2, i.e., $\boldsymbol{u}_h^+ = -\boldsymbol{u}_h^- + 2\boldsymbol{g}_u$ on Γ_h^{D}. Since the convective term is nonlinear and the residual forms the right-hand side of the linear solver in case of a Newton–Krylov approach, there is no need to split the convective operator into homogeneous and inhomogeneous contributions.

Finally, the divergence penalty term $a_{\mathrm{D},h}^e$ and continuity penalty term $a_{\mathrm{C},h}^e$ have to be defined. These terms can be interpreted as a weak enforcement of $H(\mathrm{div})$-conformity (normal continuous velocity) along with suitable function spaces for velocity and pressure that lead to an exactly (pointwise) divergence-free velocity (such as Raviart–Thomas) [29, 36, 35]. These penalty terms are mandatory to obtain a robust discretization for under-resolved problems such as turbulent flows when using standard L^2-conforming spaces. The penalty

terms are defined as [29]

$$a_{\mathrm{D},h}^e(\boldsymbol{v}_h,\boldsymbol{u}_h) = (\nabla \cdot \boldsymbol{v}_h, \tau_{\mathrm{D}} \nabla \cdot \boldsymbol{u}_h)_{\Omega_e}\,, \quad \tau_{\mathrm{D},e} = \zeta_{\mathrm{D}}\, \overline{\|\boldsymbol{u}_h^{n+1,\mathrm{ex}}\|}\, \frac{h_e}{k_u+1}\,,$$

$$a_{\mathrm{C},h}^e(\boldsymbol{v}_h,\boldsymbol{u}_h;\boldsymbol{g}_u) = (\boldsymbol{v}_h \cdot \boldsymbol{n}, \tau_{\mathrm{C}}\,[\![\boldsymbol{u}_h]\!] \cdot \boldsymbol{n})_{\partial\Omega_e}\,, \; \tau_{\mathrm{C},e} = \zeta_{\mathrm{C}}\, \overline{\|\boldsymbol{u}_h^{n+1,\mathrm{ex}}\|}\,,$$

where $\boldsymbol{u}_h^{n+1,\mathrm{ex}} = \sum_{i=0}^{J-1}\beta_i^n \boldsymbol{u}_h^{n-i}$ is an extrapolation of the velocity field of order J, $\overline{(\cdot)}$ an elementwise volume-averaged quantity, and $h_e = V_e^{1/3}$ a characteristic element length with V_e the volume of the element. A minor modification compared to [29] is that the continuity penalty term is applied not only on interior faces, but also on boundary faces with exterior values according to Table 2. While mainly numerical examples with periodic boundary conditions have been studied [29] where this does not make a difference, we observed that it is advantageous to apply this penalty term on all faces for general boundary conditions. The continuity penalty parameter is $\tau_{\mathrm{C}} = \{\!\{\tau_{\mathrm{C},e}\}\!\}$ on interior faces and $\tau_{\mathrm{C}} = \tau_{\mathrm{C},e}$ on boundary faces. As shown above for other operators, the continuity penalty term can then be split into homogeneous and inhomogeneous contributions, $a_{\mathrm{C},h}^e(\boldsymbol{v}_h,\boldsymbol{u}_h;\boldsymbol{g}_u) = a_{\mathrm{C},h,\mathrm{hom}}^e(\boldsymbol{v}_h,\boldsymbol{u}_h) + a_{\mathrm{C},h,\mathrm{inhom}}^e(\boldsymbol{v}_h;\boldsymbol{g}_u)$.

Remark Consistency of the above variational formulation, equations (28) and (29), immediately follows from the fact that the weak form is derived using integration by parts, using consistent numerical flux functions, and consistent boundary conditions according to Table 2. The additional penalty terms are consistent as well, since these terms contain the divergence of the velocity or the jump of the velocity over interior faces.

4.2. High-order dual splitting scheme

For the dual splitting projection scheme, the variational formulation can be summarized as follows: Find $\hat{\boldsymbol{u}}_h, \hat{\hat{\boldsymbol{u}}}_h, \hat{\hat{\hat{\boldsymbol{u}}}}_h, \boldsymbol{u}_h^{n+1} \in \mathcal{V}_h^u$ and $p_h^{n+1} \in \mathcal{V}_h^p$ such that for all $\boldsymbol{v}_h \in \mathcal{V}_{h,e}^u$, $q_h \in \mathcal{V}_{h,e}^p$ and for all elements $e = 1,...,N_{\mathrm{el}}$

$$\left(\boldsymbol{v}_h, \frac{\gamma_0^n \hat{\boldsymbol{u}}_h - \sum_{i=0}^{J-1}\alpha_i^n \boldsymbol{u}_h^{n-i}}{\Delta t_n}\right)_{\Omega_e^{n+1}} = -\sum_{i=0}^{J-1}\beta_i^n c_h^{e,n+1}\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n-i}, \boldsymbol{u}_{\mathrm{G},h}^{n+1}; \boldsymbol{g}_u^{n+1}\right) + (\boldsymbol{v}_h, \boldsymbol{f}(t_{n+1}))_{\Omega_e^{n+1}}\,, \quad (38)$$

$$l_{h,\mathrm{hom}}^{e,n+1}\left(q_h, p_h^{n+1}\right) = -\frac{\gamma_0^n}{\Delta t_n} d_h^{e,n+1}\left(q_h, \hat{\boldsymbol{u}}_h; \boldsymbol{g}_{\hat{u}}^{n+1}\right) - l_{h,\mathrm{inhom}}^{e,n+1}\left(q_h; g_p^{n+1}, h_p^{n+1}\right)\,, \quad (39)$$

$$\left(\boldsymbol{v}_h, \hat{\hat{\boldsymbol{u}}}_h\right)_{\Omega_e^{n+1}} = (\boldsymbol{v}_h, \hat{\boldsymbol{u}}_h)_{\Omega_e^{n+1}} - \frac{\Delta t_n}{\gamma_0^n} g_h^{e,n+1}\left(\boldsymbol{v}_h, p_h^{n+1}; g_p^{n+1}\right)\,, \quad (40)$$

$$\left(\boldsymbol{v}_h, \frac{\gamma_0^n}{\Delta t_n}\hat{\hat{\hat{\boldsymbol{u}}}}_h\right)_{\Omega_e^{n+1}} + v_{h,\mathrm{hom}}^{e,n+1}\left(\boldsymbol{v}_h, \hat{\hat{\hat{\boldsymbol{u}}}}_h\right) = \left(\boldsymbol{v}_h, \frac{\gamma_0^n}{\Delta t_n}\hat{\hat{\boldsymbol{u}}}_h\right)_{\Omega_e^{n+1}} - v_{h,\mathrm{inhom}}^{e,n+1}(\boldsymbol{v}_h; \boldsymbol{g}_u^{n+1}, \boldsymbol{h}_u^{n+1})\,, \quad (41)$$

$$\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}\right)_{\Omega_e^{n+1}} + a_{\mathrm{D},h}^{e,n+1}\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}\right)\Delta t_n + a_{\mathrm{C},h,\mathrm{hom}}^{e,n+1}\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}\right)\Delta t_n =$$
$$\left(\boldsymbol{v}_h, \hat{\hat{\hat{\boldsymbol{u}}}}_h\right)_{\Omega_e^{n+1}} - a_{\mathrm{C},h,\mathrm{inhom}}^{e,n+1}\left(\boldsymbol{v}_h; \boldsymbol{g}_u^{n+1}\right)\Delta t_n\,. \quad (42)$$

The Laplace operator l_h^e appearing in the pressure Poisson equation is discretized using the SIPG method

$$l_h^e\left(q_h, p_h; g_p^{n+1}, h_p^{n+1}\right) = (\nabla q_h, \nabla p_h)_{\Omega_e} - \left(\nabla q_h, \frac{1}{2}[\![p_h]\!]\right)_{\partial\Omega_e} - (q_h, \{\!\{\nabla p_h\}\!\} \cdot \boldsymbol{n})_{\partial\Omega_e}$$
$$+ (q_h, \tau[\![p_h]\!] \cdot \boldsymbol{n})_{\partial\Omega_e}\,, \quad (43)$$

and is again split into homogeneous contributions

$$l_{h,\mathrm{hom}}^e\left(q_h, p_h\right) = (\nabla q_h, \nabla p_h)_{\Omega_e} - \left(\nabla q_h, \frac{1}{2}[\![p_h]\!]\right)_{\partial\Omega_e \setminus \Gamma_h} - (\nabla q_h, p_h \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{N}}}$$
$$- (q_h, \{\!\{\nabla p_h\}\!\} \cdot \boldsymbol{n})_{\partial\Omega_e \setminus \Gamma_h} - (q_h, \nabla p_h \cdot \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\mathrm{N}}} \quad (44)$$
$$+ (q_h, \tau[\![p_h]\!] \cdot \boldsymbol{n})_{\partial\Omega_e \setminus \Gamma_h} + (q_h, 2\tau p_h)_{\partial\Omega_e \cap \Gamma_h^{\mathrm{N}}}\,.$$

and inhomogeneous contributions

$$l_{h,\text{inhom}}^e\left(q_h; g_p^{n+1}, h_p^{n+1}\right) = (\nabla q_h, g_p \boldsymbol{n})_{\partial\Omega_e \cap \Gamma_h^{\text{N}}} - (q_h, h_p)_{\partial\Omega_e \cap \Gamma_h^{\text{D}}} - (q_h, 2\tau g_p)_{\partial\Omega_e \cap \Gamma_h^{\text{N}}} . \tag{45}$$

In our previous work [49], the penalty terms have been applied in the projection equation (40). However, since we evaluate the continuity penalty operator also on boundary faces in the present work, the penalty terms are evaluated in a postprocessing step, equation (42). Adding the penalty terms to the projection equation (40) would prevent to achieve high-order temporal accuracy since prescribing the boundary condition $\boldsymbol{g}_u$ for the intermediate velocity $\hat{\hat{\boldsymbol{u}}}_h$ would be inconsistent.

Some comments are in order regarding the evaluation of the boundary conditions $\boldsymbol{g}_{\hat{u}}(t_{n+1})$ and $h_p(t_{n+1})$ on the right-hand side of equation (39). The convective and viscous terms have to be evaluated on $\partial\Omega_e$ using the finite element expansion of the velocity solution $\boldsymbol{u}_h$ on element e. In the discrete case, the convective term is simply calculated as $(\nabla \boldsymbol{u}_h) \cdot (\boldsymbol{u}_h - \boldsymbol{u}_{\text{G},h})$ by taking the derivative of the shape functions. Since the viscous term involves second derivatives, we calculate it in a two-step process, computing the vorticity $\boldsymbol{\omega}_h \in \mathcal{V}_h^u$ in a first step by a local L^2-projection, see [28, 41]

$$(\boldsymbol{v}_h, \boldsymbol{\omega}_h)_{\Omega_e} = (\boldsymbol{v}_h, \nabla \times \boldsymbol{u}_h)_{\Omega_e} .$$

In the second step, the contribution of the viscous term to the pressure Neumann boundary condition is obtained by calculating the curl of the vorticity $\boldsymbol{\omega}_h$ according to equation (17).

4.3. Pressure-correction scheme

Finally, the variational formulation is stated for the class of pressure-correction methods: Find $\hat{\boldsymbol{u}}_h, \hat{\hat{\boldsymbol{u}}}_h, \boldsymbol{u}_h^{n+1} \in \mathcal{V}_h^u$ and $\phi_h^{n+1}, p_h^{n+1} \in \mathcal{V}_h^p$ such that for all $\boldsymbol{v}_h \in \mathcal{V}_{h,e}^u$, $q_h \in \mathcal{V}_{h,e}^p$ and for all elements $e = 1, ..., N_{\text{el}}$

$$\left(\boldsymbol{v}_h, \frac{\gamma_0^n \hat{\boldsymbol{u}}_h - \sum_{i=0}^{J-1} \alpha_i^n \boldsymbol{u}_h^{n-i}}{\Delta t_n}\right)_{\Omega_e^{n+1}} + c_h^{e,n+1}\left(\boldsymbol{v}_h, \hat{\boldsymbol{u}}_h, \boldsymbol{u}_{\text{G},h}^{n+1}; \boldsymbol{g}_u^{n+1}\right) + v_h^{e,n+1}\left(\boldsymbol{v}_h, \hat{\boldsymbol{u}}_h; \boldsymbol{g}_u^{n+1}, \boldsymbol{h}_u^{n+1}\right)$$
$$= -\sum_{i=0}^{J_p-1} \beta_i^n g_h^e\left(\boldsymbol{v}_h, p_h^{n-i}; g_p^{n-i}\right) + (\boldsymbol{v}_h, \boldsymbol{f}(t_{n+1}))_{\Omega_e^{n+1}} , \tag{46}$$

$$l_{h,\text{hom}}^{e,n+1}\left(q_h, \phi_h^{n+1}\right) = -\frac{\gamma_0^n}{\Delta t_n} d_h^{e,n+1}\left(q_h, \hat{\boldsymbol{u}}_h; \boldsymbol{g}_u^{n+1}\right) - l_{h,\text{inhom}}^{e,n+1}\left(q_h; g_\phi^{n+1}, h_\phi^{n+1}\right), \tag{47}$$

$$\left(q_h, p_h^{n+1}\right)_{\Omega_e^{n+1}} = \left(q_h, \phi_h^{n+1} + \sum_{i=0}^{J_p-1}\left(\beta_i p_h^{n-i}\right)\right)_{\Omega_e^{n+1}} - \zeta\nu\, d_h^{e,n+1}\left(q_h, \hat{\boldsymbol{u}}_h; \boldsymbol{g}_u^{n+1}\right), \tag{48}$$

$$\left(\boldsymbol{v}_h, \hat{\hat{\boldsymbol{u}}}_h\right)_{\Omega_e^{n+1}} = (\boldsymbol{v}_h, \hat{\boldsymbol{u}}_h)_{\Omega_e^{n+1}} - \frac{\Delta t_n}{\gamma_0^n} g_h^{e,n+1}\left(\boldsymbol{v}_h, \phi_h^{n+1}; g_\phi^{n+1}\right), \tag{49}$$

$$\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1}\right)_{\Omega_e^{n+1}} + a_{\text{D},h}^{e,n+1}(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1})\Delta t_n + a_{\text{C},h,\text{hom}}^{e,n+1}(\boldsymbol{v}_h, \boldsymbol{u}_h^{n+1})\Delta t_n =$$
$$\left(\boldsymbol{v}_h, \hat{\hat{\boldsymbol{u}}}_h\right)_{\Omega_e^{n+1}} - a_{\text{C},h,\text{inhom}}^{e,n+1}(\boldsymbol{v}_h, \boldsymbol{g}_u^{n+1})\Delta t_n. \tag{50}$$

Similar to the monolithic solver, we consider an alternative formulation that formulates the convective term explicitly, replacing

$$c_h^{e,n+1}\left(\boldsymbol{v}_h, \hat{\boldsymbol{u}}_h, \boldsymbol{u}_{\text{G},h}^{n+1}; \boldsymbol{g}_u^{n+1}\right) \to \sum_{i=0}^{J-1} \beta_i^n c_h^{e,n+1}\left(\boldsymbol{v}_h, \boldsymbol{u}_h^{n-i}, \boldsymbol{u}_{\text{G},h}^{n+1}; \boldsymbol{g}_u^{n+1}\right) .$$

4.4. Geometric conservation law

It can be proven that the fully discrete ALE-DG methods derived above satisfy the geometric conservation law, i.e., they are able to preserve a constant flow field [13]. In other words, the constant solution $\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}_0$, $p(\boldsymbol{x}, t) = p_0$ is a solution of the fully discrete formulations for $\boldsymbol{f} = \boldsymbol{0}$.

Theorem 4.1. *Assume the solution at time instant t_n is given as $\boldsymbol{u}_h^n = \boldsymbol{u}_0$, $p_h^n = p_0$ where $u_{i,0} = C_i$, $i = 1, ..., d$, and $p_0 = C$ (and similarly for previous time instants in case of high-order schemes), and further assume $\boldsymbol{f} = \boldsymbol{0}$ and exact numerical integration of the velocity divergence term and pressure gradient term. Then, the fully discrete ALE-DG incompressible Navier–Stokes solvers introduced in Sections 4.1, 4.2, and 4.3 preserve a constant solution and yield $\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_0$, $p_h^{n+1} = p_0$ at time $t_{n+1} = t_n + \Delta t_n$, independently of the order of the time integration and extrapolation schemes, and for arbitrary mesh velocities.*

Proof For the monolithic solver, equations (28) and (29), it is obvious that the time derivative term becomes zero since it holds $\gamma_0^n = \sum_{i=0}^{J-1} \alpha_i^n$. As explained in [15], this originates from the fact that we discretize the differential form of the ALE equations with time derivative applied to the velocity only, as opposed to formulations that apply the time derivative to an integral quantity. Hence, to complete the proof one needs to show that the weak forms of all individual terms evaluate to zero. This is trivial for the divergence and continuity penalty terms, since these evaluate the divergence inside the element or the jump over interior faces, both vanishing for a constant solution. The volume term of the convective operator contains the gradient, and the face terms vanish due to the conservativity and consistency of the numerical flux. Note that it is enough to consider interior faces in this context, as one realizes that boundary faces behave as interior faces when evaluating the boundary values according to Table 2 for a constant solution. The SIPG discretization of the viscous term also vanishes, as each form either contains gradients or jumps of the solution. The strong formulation of the velocity divergence term, equation (33), and pressure gradient term, equation (34), vanish for constant solutions, since the volume integrals contain derivatives of the solution and the face integrals contain jumps. The strong formulation therefore always satisfies the (discrete) geometric conservation law. Since strong and weak formulation are only equivalent under the assumption of exact integration, these terms do not vanish exactly for the weak formulation of the velocity–pressure coupling terms, equation (31) and (32), when applying standard Gaussian quadrature rules and considering arbitrarily deformed elements. Since the divergence and continuity penalty terms vanish for a constant solution, the GCL is also fulfilled when applying these terms in a postprocessing step, equation (30).

Regarding the dual splitting scheme, it follows from the above argumentation that the first sub-step in equation (38) yields $\hat{\boldsymbol{u}}_h = \boldsymbol{u}_0$. Particular attention has to be paid to the boundary conditions h_p in equation (17) and $\boldsymbol{g}_{\hat{u}}$ in equation (18). The pressure Neumann boundary condition vanishes for $\boldsymbol{f} = \boldsymbol{0}$, as it contains derivatives in either space or time, and we obtain $\boldsymbol{g}_{\hat{u}} = \boldsymbol{g}_u$ in the case of a constant solution. Hence, the divergence operator on the right-hand side of the pressure Poisson equation (39) vanishes, and the pressure Poisson equation is satisfied for a constant solution p_0. It immediately follows from the argumentation for the monolithic solver that the remaining sub-steps, equations (40), (41), and (42), yield $\hat{\hat{\boldsymbol{u}}}_h = \boldsymbol{u}_0$, $\hat{\hat{\hat{\boldsymbol{u}}}}_h = \boldsymbol{u}_0$, and $\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_0$, thus preserving a constant flow state.

Turning to the pressure-correction scheme, we first notice that the momentum equation (46) results in $\hat{\boldsymbol{u}}_h = \boldsymbol{u}_0$ for reasons explained above. The pressure boundary condition g_ϕ becomes zero, as $\sum_{i=0}^{J_p-1} \beta_i^n = 1$, so that $\phi_h^{n+1} = 0$ is a solution of the pressure Poisson equation (47). The pressure update equation (48) results in $p_h^{n+1} = p_0$, since the divergence term on the right-hand side becomes zero. The same holds for the pressure gradient term in equation (49), since its arguments are $\phi_h^{n+1} = 0$, $g_\phi^{n+1} = 0$, so that the constant solution $\hat{\hat{\boldsymbol{u}}}_h = \boldsymbol{u}_0$ and $\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_0$ is recovered in the last sub-steps. $\qquad\square$

Remark It is worth noting that no assumption has been made regarding the mesh velocity or how it is computed numerically to show compliance with the GCL. This is also a consequence of the fact that we discretize – as suggested in [15] – the differential form of the ALE equations in convective formulation, equation (5). In Section 5, we demonstrate numerically that the geometric conservation is fulfilled exactly for the strong formulations, and that it is not fulfilled exactly down to rounding errors for the weak formulation of the velocity–pressure coupling terms in general. However, since the temporal discretization and spatial discretization are designed to satisfy the GCL, we expect that no relevant difference is observed for practical problems (where the solution is non-constant) due to this variational crime, since those problems always suffer from non-exact integration, and variational crimes have to be accepted in several respects. For example, fulfilling discrete energy stability exactly with respect to the velocity–pressure coupling terms, see Section 4.5, requires that one term is formulated in weak form, and the other one in strong form, so that the

formulation becomes symmetric independently of integration errors. We also emphasize that previous works have shown that fulfilling the GCL is neither a necessary nor a sufficient condition for the time integrator to preserve its high-order accuracy on moving meshes [16, 67]. We will therefore carefully investigate the temporal convergence behavior of the present ALE schemes and demonstrate that the high-order accuracy of the Navier–Stokes solvers on fixed meshes is preserved on moving meshes when using definition (27) to calculate the mesh velocity.

4.5. Energy stability

As detailed in the introduction, energy stability is a crucial ingredient to obtain a flow solver that is robust in the under-resolved regime. In this section, we analyze the energy balance of the present ALE-DG methods. For this analysis, we assume vanishing body forces, $\boldsymbol{f} = \boldsymbol{0}$, and consider the inviscid limit, $\nu = 0$. This is reasonable since the viscous term has a dissipative character both physically and numerically, where the numerically dissipative character immediately follows from the fact that the SIPG discretization of the viscous term, equation (35), is positive definite and symmetric. Therefore, the critical case is to investigate whether a numerical method is energy stable in the absence of viscous dissipation. For simplicity, we also assume periodic boundaries (including periodicity of the grid velocity). Under these assumptions and for smooth solutions, the ALE incompressible Navier–Stokes equations (5) and (6) are energy-conserving in the following sense

$$\frac{\partial}{\partial t} \int_{\Omega_0} \frac{1}{2} \boldsymbol{u} \cdot \boldsymbol{u} \det \boldsymbol{J} \, \mathrm{d}\Omega \bigg|_{\boldsymbol{\chi}} = \int_{\Omega(t)} \frac{\partial \frac{1}{2} \boldsymbol{u} \cdot \boldsymbol{u}}{\partial t} \bigg|_{\boldsymbol{\chi}} \mathrm{d}\Omega + \int_{\Omega(t)} \frac{1}{2} \left(\boldsymbol{u} \cdot \boldsymbol{u} \right) \nabla \cdot \boldsymbol{u}_{\mathrm{G}} \, \mathrm{d}\Omega = 0 \,, \tag{51}$$

where $\boldsymbol{J} = \partial \boldsymbol{x} / \partial \boldsymbol{\chi}$ is the Jacobian. A derivation of this relation is shown in Appendix B. A spatially discretized ALE incompressible Navier–Stokes solver

$$\left(\boldsymbol{v}_h, \frac{\partial \boldsymbol{u}_h}{\partial t} \bigg|_{\boldsymbol{\chi}} \right)_{\Omega_e} + c_h^e \left(\boldsymbol{v}_h, \boldsymbol{u}_h, \boldsymbol{u}_{\mathrm{G},h} \right) + g_h^e \left(\boldsymbol{v}_h, p_h \right) + a_{\mathrm{D},h}^e (\boldsymbol{v}_h, \boldsymbol{u}_h) + a_{\mathrm{C},h}^e (\boldsymbol{v}_h, \boldsymbol{u}_h) = \boldsymbol{0} \,, \tag{52}$$

$$-d_h^e(q_h, \boldsymbol{u}_h) = 0 \,, \tag{53}$$

is called energy-stable if it fulfills the following discrete analogy

$$\int_{\Omega_h(t)} \frac{\partial \frac{1}{2} \boldsymbol{u}_h \cdot \boldsymbol{u}_h}{\partial t} \bigg|_{\boldsymbol{\chi}} \mathrm{d}\Omega + \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_e(t)} \frac{1}{2} \left(\boldsymbol{u}_h \cdot \boldsymbol{u}_h \right) \nabla \cdot \boldsymbol{u}_{\mathrm{G},h} \mathrm{d}\Omega \leq 0 \,. \tag{54}$$

We begin with reformulating the left term in equation (54) so that the semi-discrete momentum equation (52) can be inserted

$$\int_{\Omega_h(t)} \frac{\partial \frac{1}{2} \boldsymbol{u}_h \cdot \boldsymbol{u}_h}{\partial t} \bigg|_{\boldsymbol{\chi}} \mathrm{d}\Omega = \int_{\Omega_h(t)} \boldsymbol{u}_h \cdot \frac{\partial \boldsymbol{u}_h}{\partial t} \bigg|_{\boldsymbol{\chi}} \mathrm{d}\Omega = \sum_{e=1}^{N_{\mathrm{el}}} \left(\boldsymbol{u}_h, \frac{\partial \boldsymbol{u}_h}{\partial t} \bigg|_{\boldsymbol{\chi}} \right)_{\Omega_e}$$

$$= -\sum_{e=1}^{N_{\mathrm{el}}} \left(c_h^e \left(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{u}_{\mathrm{G},h} \right) + g_h^e \left(\boldsymbol{u}_h, p_h \right) + a_{\mathrm{D},h}^e (\boldsymbol{u}_h, \boldsymbol{u}_h) + a_{\mathrm{C},h}^e (\boldsymbol{u}_h, \boldsymbol{u}_h) \right) \,. \tag{55}$$

Regarding the pressure gradient term, we first observe that the following relation holds

$$\sum_{e=1}^{N_{\mathrm{el}}} g_{h,\mathrm{strong}}^e \left(\boldsymbol{u}_h, p_h \right) = \left(\boldsymbol{u}_h, \nabla p_h \right)_{\Omega_h} - \left(\{\{\boldsymbol{u}_h\}\}, [\![p_h]\!] \right)_{\Gamma_h^{\mathrm{int}}} = -\sum_{e=1}^{N_{\mathrm{el}}} d_{h,\mathrm{weak}}^e \left(p_h, \boldsymbol{u}_h \right) = 0 \,, \tag{56}$$

and similarly between $g_{h,\mathrm{weak}}^e$ and $d_{h,\mathrm{strong}}^e$. For these combinations of the velocity–pressure coupling terms, the formulation is symmetric independently of integration errors, and the pressure gradient term will not contribute to the energy evolution due to the discrete continuity equation (53).

18

The divergence and continuity penalty terms have a dissipative character as these are positive semi-definite by definition. Hence, it remains to consider the convective term. Inserting equation (37) and reformulating yields

$$
\sum_{e=1}^{N_{\mathrm{el}}} c_h^e \left(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{u}_{\mathrm{G},h} \right) = + \sum_{e=1}^{N_{\mathrm{el}}} \left(\left(\boldsymbol{u}_h, \left(\nabla \boldsymbol{u}_h \right) \cdot \left(\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h} \right) \right)_{\Omega_e} - \left(\boldsymbol{u}_h, \left(\left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} \right) \boldsymbol{u}_h \right)_{\partial \Omega_e} \right)
$$

$$
+ \sum_{e=1}^{N_{\mathrm{el}}} \left(\boldsymbol{u}_h, \left(\left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} \right) \{\!\{ \boldsymbol{u}_h \}\!\} + \frac{1}{2} | \left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} | \, [\![\boldsymbol{u}_h]\!] \right)_{\partial \Omega_e}
$$

$$
= + \sum_{e=1}^{N_{\mathrm{el}}} \left(\left(\boldsymbol{u}_h, \left(\nabla \boldsymbol{u}_h \right) \cdot \left(\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h} \right) \right)_{\Omega_e} - \left(\boldsymbol{u}_h, \left(\left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} \right) \frac{1}{2} [\![\boldsymbol{u}_h]\!] \right)_{\partial \Omega_e} \right)
$$

$$
+ \left([\![\boldsymbol{u}_h]\!], \frac{1}{2} | \left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} | \, [\![\boldsymbol{u}_h]\!] \right)_{\Gamma_h^{\mathrm{int}}} .
$$

$$(57)$$

In Appendix C, we show that the first row on the right-hand side of the above equation can be reformulated as follows by algebraic manipulations

$$
\sum_{e=1}^{N_{\mathrm{el}}} \left(\left(\boldsymbol{u}_h, \left(\nabla \boldsymbol{u}_h \right) \cdot \left(\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h} \right) \right)_{\Omega_e} - \left(\boldsymbol{u}_h, \left(\left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} \right) \frac{1}{2} [\![\boldsymbol{u}_h]\!] \right)_{\partial \Omega_e} \right) =
$$

$$
= - \frac{1}{2} \left(\nabla \cdot \left(\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h} \right), \boldsymbol{u}_h \cdot \boldsymbol{u}_h \right)_{\Omega_h} + \frac{1}{2} \left([\![\boldsymbol{u}_h]\!] \cdot \boldsymbol{n}, \{\!\{ \boldsymbol{u}_h \cdot \boldsymbol{u}_h \}\!\} \right)_{\Gamma_h^{\mathrm{int}}} ,
$$

$$(58)$$

which is easier to interpret in terms of energy stability since this formulation contains the divergence of the velocity, i.e., a residual of the incompressible Navier–Stokes equations, and the divergence of the grid velocity. It is worth emphasizing that the face integrals related to the moving mesh dropped out completely, since the ALE term is a linear transport term. The volume integral of the ALE transport term does not drop out since the grid velocity is not divergence-free. However, one can see that this is exactly the second term in (54). Inserting equations (56), (57), and (58) into equation (55) yields the following result

$$
\int_{\Omega_h(t)} \left. \frac{\partial \frac{1}{2} \boldsymbol{u}_h \cdot \boldsymbol{u}_h}{\partial t} \right|_{\boldsymbol{\chi}} \mathrm{d}\Omega + \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_e(t)} \frac{1}{2} \left(\boldsymbol{u}_h \cdot \boldsymbol{u}_h \right) \nabla \cdot \boldsymbol{u}_{\mathrm{G},h} \mathrm{d}\Omega =
$$

$$
= + \frac{1}{2} \left(\nabla \cdot \boldsymbol{u}_h, \boldsymbol{u}_h \cdot \boldsymbol{u}_h \right)_{\Omega_h} - a_{\mathrm{D},h}(\boldsymbol{u}_h, \boldsymbol{u}_h)
$$

$$
- \frac{1}{2} \left([\![\boldsymbol{u}_h]\!] \cdot \boldsymbol{n}, \{\!\{ \boldsymbol{u}_h \cdot \boldsymbol{u}_h \}\!\} \right)_{\Gamma_h^{\mathrm{int}}} - a_{\mathrm{C},h}(\boldsymbol{u}_h, \boldsymbol{u}_h)
$$

$$
- \left([\![\boldsymbol{u}_h]\!], \frac{1}{2} | \left(\{\!\{ \boldsymbol{u}_h \}\!\} - \boldsymbol{u}_{\mathrm{G},h} \right) \cdot \boldsymbol{n} | \, [\![\boldsymbol{u}_h]\!] \right)_{\Gamma_h^{\mathrm{int}}} .
$$

$$(59)$$

It is interesting to realize that the result derived in [29] for the conservative formulation of the convective term in the Eulerian case is very similar to the convective formulation in ALE form considered here, where terms with exactly the same structure occur. In particular, one observes that the ALE formulation does not introduce new terms on the right-hand side as compared to the Eulerian case (a consequence of the fact that the additional ALE term is a linear transport term). One might therefore argue that energy stability for the Eulerian case translates into energy stability for the ALE case with moving meshes. The third row on the right-hand side is the upwind stabilization term of the convective term and always exhibits a dissipative behavior. However, it is well-known that this term is not able to render the nonlinear convective term energy-stable. As argued in [29, 36], the consistent divergence and continuity penalty terms are positive semi-definite and control the non-vanishing divergence and non-vanishing jumps of the velocity in normal direction. By the use of consistent penalty terms, energy stability of the DG discretization is enforced

19

weakly; there is currently no proof that the kinetic energy is strictly non-increasing at all times. The sign-indefinite terms of the above equation are exactly zero for an $H(\text{div})$-conforming (normal continuous) velocity space together with a pressure space that ensures that the velocity is pointwise divergence-free. For these reasons, the stabilized approach with divergence and continuity penalty terms can also be denoted as $H(\text{div})$-stabilization.

4.6. Numerical integration and implementation

Integrals in the variational form are evaluated numerically by means of Gaussian quadrature, where we choose the number of one-dimensional quadrature points to ensure exact integration on affine element geometries with constant Jacobian. The velocity mass matrix term, the body force term, the viscous term, the velocity divergence term, the pressure gradient term, and the two penalty terms are integrated with $n_{\mathrm{q}} = k_u + 1$ quadrature points. To avoid aliasing effects, we use $n_{\mathrm{q}} = \lfloor \frac{3k_u}{2} \rfloor + 1$ quadrature points for the convective term containing quadratic nonlinearities. The Laplace operator in the pressure Poisson equation and the pressure mass matrix operator are integrated with $n_{\mathrm{q}} = k_p + 1$ quadrature points. The present ALE-DG methods are implemented in `C++` using the `deal.II` finite element library [68], and especially the matrix-free evaluation techniques developed in [51] for the evaluation of volume and face integrals of discretized DG operators. State-of-the-art iterative solvers are used to solve the (non-)linear systems of equations of the fully discrete problem.

4.7. CFL condition

For the formulations with explicit treatment of the convective term, the time step size is restricted according to the CFL condition. Since the transport velocity is $\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}$ in the ALE case, the CFL condition used on static meshes has to be adjusted accordingly using the relative velocity between fluid and grid motion. Here, we distinguish between two types of CFL condition. The global CFL condition [21, 29] applied to the ALE case

$$\Delta t = \frac{\mathrm{Cr}}{k_u^{1.5}} \frac{h_{\min}}{\|\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}\|_{\max}} \ , \tag{60}$$

with global estimates of the minimum element length $h_{\min}$ and maximum velocity $\|\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}\|_{\max}$ is used for time stepping with constant Δt. In the above equation, Cr denotes the Courant number and the term $k_u^{1.5}$ was found to describe well the relation between critical time step size and polynomial degree for the present DG discretization [29]. Since the minimum element length and the maximum velocity are difficult to estimate a priori and since the maximum velocity does not necessarily occur in the smallest element, a local CFL condition is used in case of adaptive time-stepping [69]

$$\Delta t = \min_{e=1,\ldots,N_{\mathrm{el}}} \Delta t_e, \ \Delta t_e = \min_{q=1,\ldots,N_{q,e}} \frac{\mathrm{Cr}}{k_u^{1.5}} \left. \frac{h}{\|\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}\|} \right|_{q,e} \ , \tag{61}$$

with the local velocity-to-mesh-size ratio $\left. \frac{\|\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}\|}{h} \right|_{q,e} = \|\boldsymbol{J}^{-\mathsf{T}}(\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h})\|_{q,e}$ evaluated at quadrature point q of element e. This CFL condition ensures that the time step size does not exceed the critical one in any element in any quadrature point.

5. Numerical results

The aim of this section is to display the numerical discretization properties of the proposed ALE-DG incompressible Navier–Stokes solvers. We select a set of academic test cases that address different aspects of ALE solvers, with the goal to obtain a picture as complete as possible. In detail, we study the geometric conservation property by the example of the free stream preservation test in Section 5.1. The convergence behavior in terms of temporal as spatial convergence rates are investigated in Section 5.2 by the example of a two-dimensional vortex problem with moving Dirichlet and Neumann boundaries. In this section,

we also test the robustness of the different incompressible Navier–Stokes solvers in the limit of large grid deformations. Finally, the robustness of the proposed discretization methods for under-resolved turbulent flows is studied in Section 5.3 by considering the three-dimensional Taylor–Green vortex problem for viscous flows at Re = 1600, and also in the very challenging inviscid limit.

Convergence rates and relative L^2-errors for problems with known analytical solution are computed as defined in [41]. Solver tolerances are selected to not spoil accuracy, e.g., by choosing relative solver tolerances of 10^{-6} and absolute solver tolerances of 10^{-12}. In case of pure Dirichlet boundary conditions, the pressure level is undefined and can be fixed, e.g., by setting the mean value of the pressure DoF vector to zero.

5.1. Geometric conservation law – free stream preservation test

We study the free stream preservation test to investigate whether the ALE formulations derived above fulfill the geometric conservation law. Satisfying the geometric conservation law means that a constant flow field is not disturbed by a moving mesh, i.e., the solver is able to preserve the free stream flow conditions exactly and independently of the mesh deformation. The analytical solution of the free stream preservation test is therefore the constant flow state

$$\boldsymbol{u}(\boldsymbol{x},t) = (1,\ldots,1)^{\mathsf{T}} , \ p(\boldsymbol{x},t) = 1 ,$$

where we prescribe pure Dirichlet boundary conditions $\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{g}_u(\boldsymbol{x},t)$ on $\Gamma_h^{\mathrm{D}}(t) = \Gamma_h(t)$. The computational domain at initial time is $\Omega_0 = \Omega(t=0) = [-L/2, L/2]^2$ with $L = 1$. The simulation is run over a time interval of $0 \le t \le T = 10$. The following analytical mesh movement with sine functions in both time and space is prescribed in two space dimensions

$$\boldsymbol{x}(\boldsymbol{\chi},t) = \boldsymbol{\chi} + A\sin\left(2\pi\frac{t}{T_{\mathrm{G}}}\right)\begin{pmatrix}\sin\left(2\pi\frac{\chi_2+L/2}{L}\right)\\ \sin\left(2\pi\frac{\chi_1+L/2}{L}\right)\end{pmatrix} , \tag{62}$$

and in three space dimensions

$$\boldsymbol{x}(\boldsymbol{\chi},t) = \boldsymbol{\chi} + A\sin\left(2\pi\frac{t}{T_{\mathrm{G}}}\right)\begin{pmatrix}\sin\left(2\pi\frac{\chi_2+L/2}{L}\right)\sin\left(2\pi\frac{\chi_3+L/2}{L}\right)\\ \sin\left(2\pi\frac{\chi_1+L/2}{L}\right)\sin\left(2\pi\frac{\chi_3+L/2}{L}\right)\\ \sin\left(2\pi\frac{\chi_1+L/2}{L}\right)\sin\left(2\pi\frac{\chi_2+L/2}{L}\right)\end{pmatrix} , \tag{63}$$

where the amplitude is set to $A = 0.08$ resulting in a strongly deformed mesh. The period length of the grid motion is set to $T_{\mathrm{G}} = T/10$ and the wavenumber in space is chosen such that the length and height of the domain are exactly one period. In Figure 2(b), the mesh deformation is illustrated for $d = 2$, where the initial, undeformed mesh is a uniform Cartesian grid. The mesh reaches its maximum deformation at times $t = T_{\mathrm{G}}/4 + i\, T_{\mathrm{G}}/2, i = 0, 1, 2, \ldots$. The viscosity is set to $\nu = 0.025$. Adaptive time-stepping is used where the time step size is adjusted dynamically according to the CFL condition (61) using Cr = 0.25. In the following, we use a mesh with 8^d elements as in Figure 2 and consider a polynomial degree of $k = 3$. Both absolute and relative solver tolerances are set to a small value of 10^{-14}. Table 3 reports relative errors for velocity and pressure for BDF schemes of order 1 to 3 and the three different incompressible Navier–Stokes solvers considered in this work. For the dual splitting scheme, $J_p = \min(2, J)$ is used but the simulations have also been stable for the choice $J_p = J$ for the high-order BDF scheme $J = 3$. For the pressure-correction scheme, $J_p = \min(2, J) - 1$ is used for all J and the rotational formulation with $\chi = 1$. Here, the choice $J_p = J - 1$ lead to instabilities for the high-order scheme $J = 3$ in agreement with theory.

The results in Table 3 reveal that all schemes fulfill the geometric conservation law for $d = 2$, and in particular also for the weak formulation of the velocity–pressure coupling terms. For $d = 3$, the geometric conservation law is fulfilled exactly only when using the strong formulations $d_{h,\mathrm{strong}}^e$ and $g_{h,\mathrm{strong}}^e$ as expected theoretically, see Section 4.4. Errors larger than the solver tolerances are observed for the weak formulations for $d = 3$. For all variants studied here, similar results are obtained when using an implicit formulation of

21

(a) undeformed mesh   (b) deformed mesh ($A/L = 0.08$)   (c) theoretical limit $A_{\max}/L = \frac{1}{2\pi}$
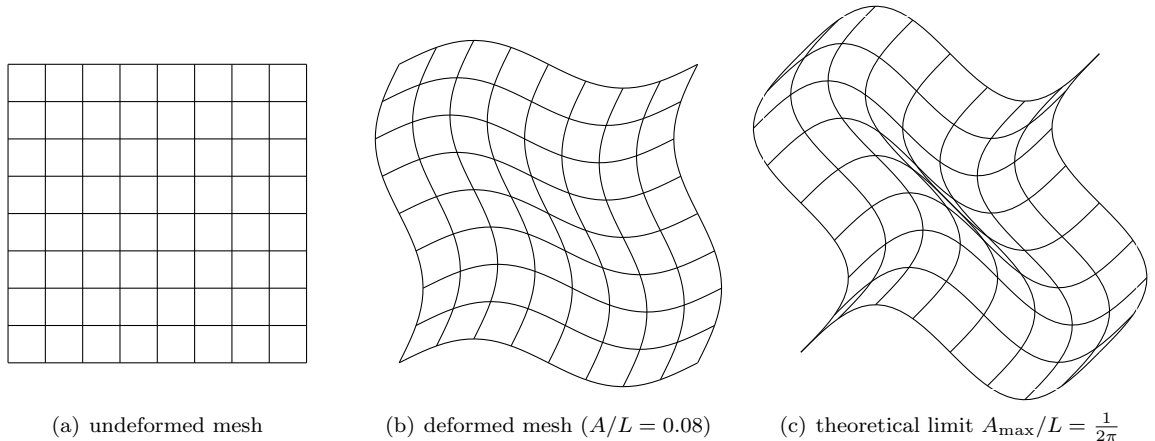
Figure 2: Illustration of sine-like mesh motion in two space dimensions according to equation (62).

the convective term for the monolithic solver and the pressure-correction scheme, where we had to slightly relax the solver tolerances to 10^{-12} to ensure convergence of the Newton solver. Hence, it remains to explain why the weak formulation of velocity–pressure coupling terms fulfills the GCL exactly for $d = 2$. As the critical aspect in this context is the exact evaluation of integrals, see Section 4.4, it can be conjectured that integrals of the velocity–pressure coupling terms are indeed evaluated exactly for $d = 2$ for constant solutions on deformed elements. However, this is a special case that only occurs for the free stream preservation test due to a solution of lowest polynomial degree, and this does not hold for general non-constant solutions and arbitrarily deformed elements. Hence, we do not pay further attention to this point. In a similar direction, our results show that the weak formulations are very accurate as well, and will therefore be used for the following examples. The reason for this choice is that the conclusion is even stronger when being able to demonstrate optimal convergence behavior for a formulation that appears to be sub-optimal regarding the free stream preservation test.

5.2. Temporal and spatial convergence behavior

Next, we analyze the convergence behavior of the ALE-DG methods and study whether optimal rates of convergence observed in the Eulerian case carry over to moving meshes. For this purpose, we select the two-dimensional vortex problem from [22], which is an analytical solution of the two-dimensional incompressible Navier–Stokes equations in the absence of body forces, $\boldsymbol{f} = \boldsymbol{0}$, namely

$$
\begin{aligned}
\boldsymbol{u}(\boldsymbol{x}, t) &= \begin{pmatrix} -\sin(2\pi x_2) \\ +\sin(2\pi x_1) \end{pmatrix} \exp\left(-4\nu\pi^2 t\right) , \\
p(\boldsymbol{x}, t) &= -\cos(2\pi x_1)\cos(2\pi x_2)\exp\left(-8\nu\pi^2 t\right) ,
\end{aligned}
\tag{64}
$$

with viscosity set to $\nu = 0.025$, and simulated over the time interval $0 \leq t \leq T = 1$. The computational domain at start time is $\Omega_0 = [-L/2, L/2]^2$ with length $L = 1$, and is deformed according to the two-dimensional mesh movement function (62) with parameters $A = 0.08$ and $T_{\mathrm{G}} = 4T$ (maximum deformation reached at end time $t = T$) unless specified otherwise. Again, a mesh as depicted in Figure 2(b) with refinement level l is used. Sine-like mesh deformations are commonly used to verify high-order ALE-DG implementations, see for example [9, 10, 12]. In these works, however, the sine functions are defined in a way that the boundaries are not moving. Instead, we intentionally choose a setup for which the boundaries are moving since our goal is to test all parts of the algorithm relevant for FSI. The verification of boundary conditions is particularly relevant for the splitting-type solvers and some effects might not be visible if the boundaries are fixed. For example, if the boundaries are non-moving, the normal vector in equation (17) would not change over time and the ALE transport term would simply drop out since $\boldsymbol{u}_{\mathrm{G}} = \boldsymbol{0}$ on the

22

Table 3: Numerical results for free stream preservation test for both strong and weak formulations of velocity–pressure coupling terms and two- and three-dimensional problems: $J_p = \min(2, J)$ is used for the dual splitting scheme, and $J_p = \min(2, J) - 1$ for the pressure-correction scheme in incremental formulation. An explicit formulation of the convective term is used for all three solvers.

(a) Two-dimensional problem ($d = 2$), weak formulations $d^e_{h,\text{weak}}$ and $g^e_{h,\text{weak}}$

	relative L^2-error $\boldsymbol{u}_h$			relative L^2-error p_h		
	BDF1	BDF2	BDF3	BDF1	BDF2	BDF3
monolithic	1.8E–16	1.8E–15	2.1E–15	1.3E–16	6.0E–13	2.9E–13
dual splitting	1.2E–15	1.6E–15	1.8E–15	2.5E–13	7.5E–14	6.1E–13
pressure-correction	1.6E–15	2.7E–15	3.0E–15	1.8E–13	9.5E–14	4.6E–13

(b) Two-dimensional problem ($d = 2$), strong formulations $d^e_{h,\text{strong}}$ and $g^e_{h,\text{strong}}$

	relative L^2-error $\boldsymbol{u}_h$			relative L^2-error p_h		
	BDF1	BDF2	BDF3	BDF1	BDF2	BDF3
monolithic	1.8E–16	1.1E–15	1.8E–15	1.3E–16	2.1E–13	2.4E–13
dual splitting	1.0E–15	1.3E–15	1.4E–15	1.5E–13	3.8E–13	6.5E–13
pressure-correction	1.0E–15	1.6E–15	1.8E–15	2.3E–13	2.3E–13	4.7E–13

(c) Three-dimensional problem ($d = 3$), weak formulations $d^e_{h,\text{weak}}$ and $g^e_{h,\text{weak}}$

	relative L^2-error $\boldsymbol{u}_h$			relative L^2-error p_h		
	BDF1	BDF2	BDF3	BDF1	BDF2	BDF3
monolithic	1.3E–08	1.3E–08	1.3E–08	8.7E–09	8.2E–09	8.2E–09
dual splitting	1.1E–08	1.2E–08	1.2E–08	6.4E–09	6.6E–09	6.8E–09
pressure-correction	1.2E–08	1.3E–08	1.3E–08	7.0E–09	7.1E–08	6.1E–08

(d) Three-dimensional problem ($d = 3$), strong formulations $d^e_{h,\text{strong}}$ and $g^e_{h,\text{strong}}$

	relative L^2-error $\boldsymbol{u}_h$			relative L^2-error p_h		
	BDF1	BDF2	BDF3	BDF1	BDF2	BDF3
monolithic	3.5E–16	1.4E–14	1.2E–14	1.7E–16	8.5E–13	8.4E–13
dual splitting	3.5E–16	1.8E–15	1.1E–15	1.4E–13	7.3E–13	1.3E–12
pressure-correction	1.9E–15	1.9E–15	2.3E–15	3.3E–13	7.8E–13	1.1E–12

boundary. According to the setup in [22], each of the four sides of the square is split into a Dirichlet boundary and a Neumann boundary according to the inflow and outflow sections, respectively. Note that the chosen mesh deformation is in compliance with these boundary conditions.

Figure 3 shows a visualization of the solution at the time of maximal mesh deformation $t = T$ using polynomial shape functions of degree $k = 3$ and considering the two lowest refinement levels of $l = 1, 2$ (the grid has to consist of at least 2^2 elements due to the type of boundary conditions prescribed with each face of the rectangular domain cut into a Dirichlet part and a Neumann part). While the velocity field is already well resolved on the coarsest mesh with $l = 1$, the pressure field of polynomial degree 2 is approximated poorly for refinement level $l = 1$ with distinct discontinuities between the elements. For $l = 2$, the pressure solution appears to be visually converged with only minor differences as compared to the solution on even finer meshes. In the following, we study the convergence quantitatively in terms of errors against the analytical solution as well as convergence rates measured in space and time.

In a first set of experiments, we test the temporal convergence behavior for both constant and adaptive

(a) $p(\boldsymbol{x}, T)$ ($l = 1$, $k = 3$)   (b) $p(\boldsymbol{x}, T)$ ($l = 2$, $k = 3$)   (c) $\|\boldsymbol{u}(\boldsymbol{x}, T)\|$ ($l = 1$, $k = 3$)   (d) $\|\boldsymbol{u}(\boldsymbol{x}, T)\|$ ($l = 2$, $k = 3$)
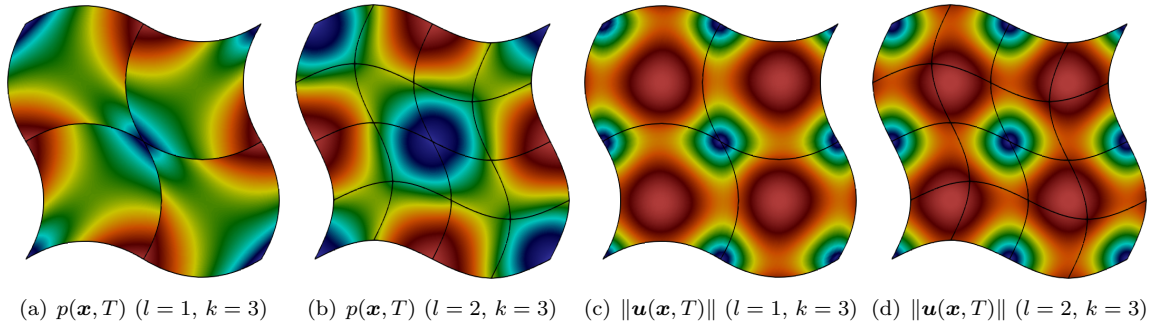
Figure 3: Vortex problem: visualization of solution at final time $t = T$ for two different mesh resolutions, $l = 1$ and $l = 2$, with polynomial degree $k = 3$ for the velocity and 2 for the pressure (red indicates high values and blue low values). The amplitude of the mesh deformation is $A = 0.08$.

time step sizes. The chosen spatial resolution is fine, $l = 3$ and $k = 8$, to make sure that errors are dominated by temporal discretization errors. The second aspect why this test case is interesting is the fact that the CFL condition does often not show up for this particular test case for moderate Reynolds numbers, probably due to the fact that the vortex is not moving. This allows to measure temporal convergence rates of an incompressible Navier–Stokes solver with explicit formulation of the convective term, which would be difficult otherwise because in the regular case with $\mathrm{Cr} < \mathrm{Cr}_{\mathrm{crit}}$ one is often operating in a regime where temporal discretization errors are already negligible as compared to spatial errors for higher-order time integration schemes.

Figure 4 shows results of a temporal convergence study for BDF schemes of order $J = 1, 2, 3$ using constant and adaptive time step sizes. All types of solvers converge with optimal rates of convergence on the moving mesh. Compared to additional simulations performed for a static, Cartesian mesh, the errors are almost the same and only slightly larger. The lowest errors are obtained for the monolithic solver with implicit formulation of the convective term. The dual splitting scheme and the monolithic solver with explicit convective term yield similar errors, and the errors are again slightly larger for both explicit and implicit pressure-correction formulations. For very small time step sizes and the BDF3 scheme, the spatial error becomes dominant at some point. Note that BDF3 schemes are not considered for the pressure-correction scheme since $J_p = 1$ (required for stability) limits convergence rates to second-order in that case. For the dual splitting scheme, large errors occur for Courant numbers $\mathrm{Cr} > 1$. This effect does not show up for the non-moving mesh and we conjecture that this effect originates from the CFL condition. For a regular Navier–Stokes problem with explicit formulation of the convective term, no stability can be expected in this range of Courant numbers. While a sharp CFL bound is not visible for the chosen parameters, all solvers become unstable for $\mathrm{Cr} > 1$ (for $J = 2$) when using a smaller viscosity of $\nu = 10^{-3}$ (higher Reynolds number), showing the usual sharp CFL bound.

In a second set of experiments, we study the spatial convergence behavior for polynomial degrees $k = 2, 3, 4, 5$ by a mesh refinement study, considering refine levels $l = 1, ..., 6$. The BDF2 time integration scheme with a small, constant time step of $\Delta t = 5 \cdot 10^{-5}$ is used to obtain small temporal errors. Figure 5 shows results for the three different solver types with an explicit formulation of the convective term. Since this is a spatial convergence test, the results would be indistinguishable when using an implicit formulation of the convective term and are therefore not shown explicitly. For comparison, we also show results for a non-moving, Cartesian mesh for the monolithic solution approach. Overall, all variants converge will optimal rates of convergence for all polynomial degrees until the temporal discretization error is reached. Compared to the static mesh, the errors are slightly larger on the moving mesh, and the gap between moving and static meshes increases for increasing polynomial degree.

Finally, we test the robustness by increasing the amplitude A of the mesh movement function to its theoretical limit, i.e., the value at which the aspect ratio tends to infinity or at which inverted elements occur, in order to characterize the point where the proposed ALE-DG methods will break down. This
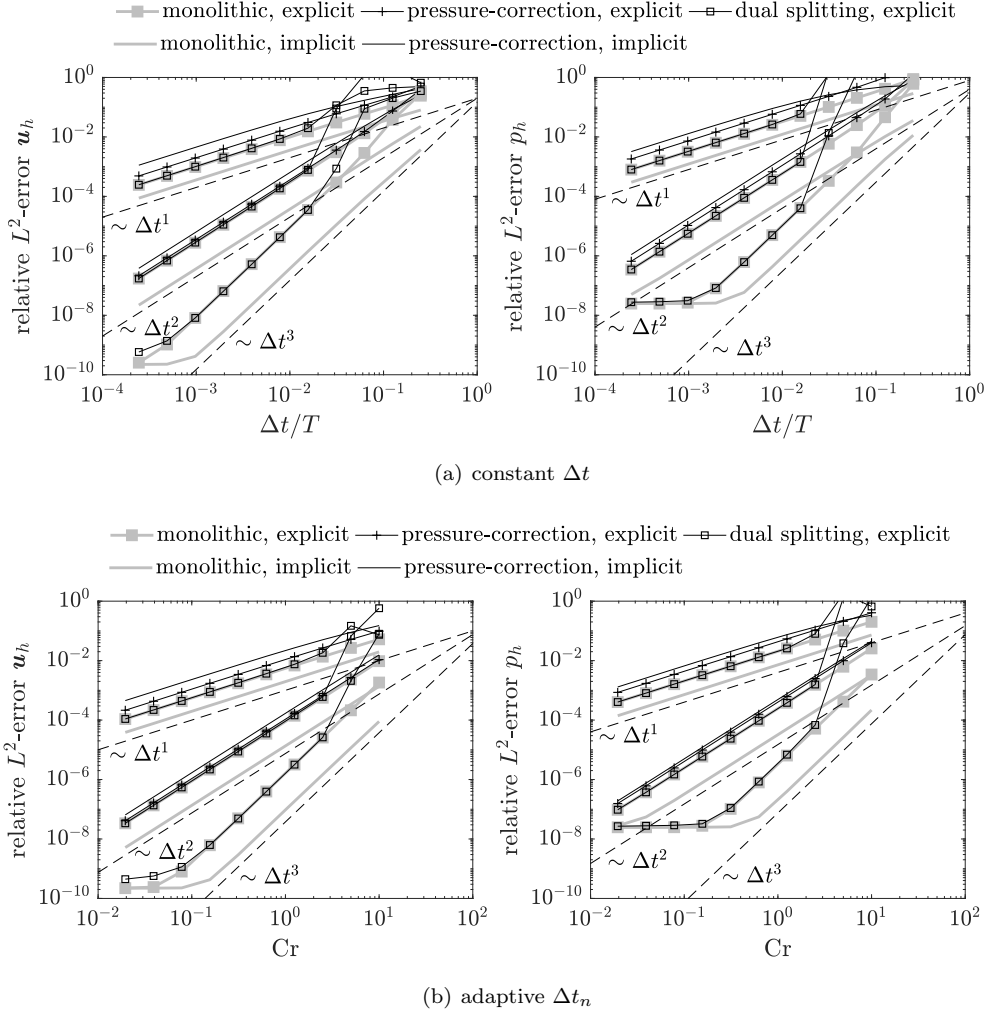
24

(a) constant Δt

(b) adaptive Δt_n

Figure 4: Vortex problem: temporal convergence tests for ALE incompressible Navier–Stokes solvers for BDF schemes of order $J = 1, ..., 3$ with $J_p = \min(2, J)$ for the dual splitting scheme and $J_p = \min(2, J) - 1, J \leq 2$ for the pressure-correction scheme.

theoretical limit is reached when the lower left corner of the domain becomes an arbitrarily thin needle, see Figure 2(c). Mathematically, this limit is reached when the slope of the lower domain boundary reaches a value of 1 in the lower left corner at the time of maximum deformation, i.e.,

$$\frac{\partial x_2(\chi_1, \chi_2 = -L/2, t = T_{\mathrm{G}}/4)}{\partial \chi_1}\bigg|_{\chi_1 = -L/2} = A \cos\left(2\pi \frac{\chi_1 + L/2}{L}\right)\bigg|_{\chi_1 = -L/2} \frac{2\pi}{L} \overset{!}{=} 1 \rightsquigarrow \frac{A_{\max}}{L} = \frac{1}{2\pi} \ .$$

In Figure 6, we plot the relative errors of velocity and pressure for a coarse mesh with 4^2 elements and polynomial degree $k = 3$ as a function of $A/A_{\max}$. Adaptive time-stepping, equation (61), with $\mathrm{Cr} = 0.2$ is used. The error increases moderately for small amplitudes of the mesh deformation, and a rapid increase in errors can be observed around 85% of the theoretically maximum amplitude for this problem.
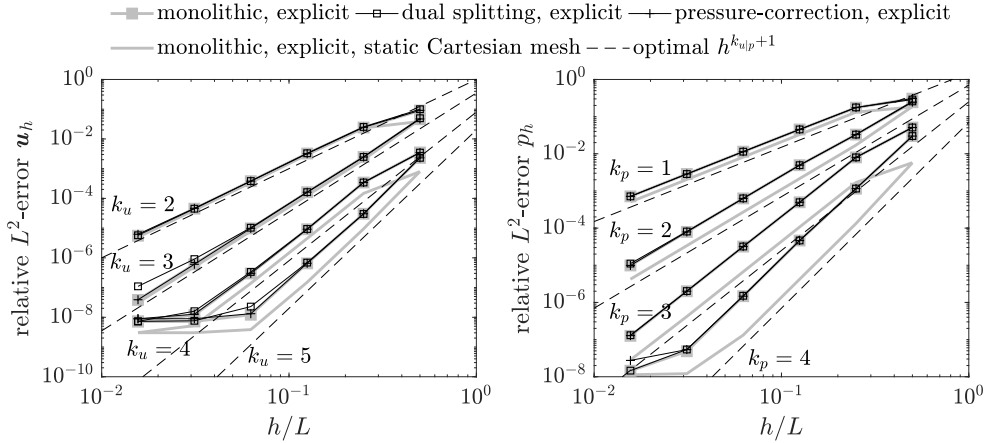
25

Figure 5: Vortex problem: spatial convergence tests for ALE incompressible Navier–Stokes solvers for polynomial degrees $k = 2, 3, 4, 5$ and comparison to static Cartesian mesh.
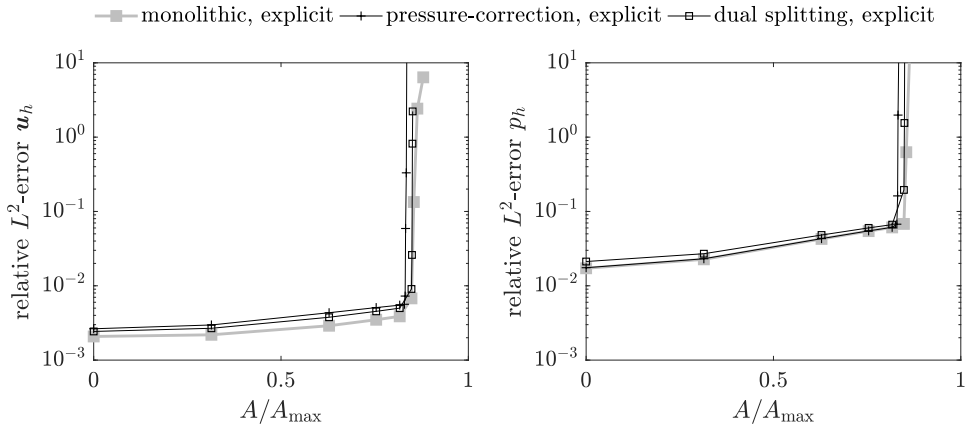

Figure 6: Vortex problem: robustness test of ALE incompressible Navier–Stokes solvers on a mesh with 4^2 elements and polynomial degree $k = 3$.

5.3. Robustness for under-resolved turbulent flows

We study the applicability of the present ALE-DG solvers to transitional and turbulent flows by the example of the three-dimensinal Taylor–Green vortex (TGV) problem [70]. Starting from the initial state

$$u_1(\boldsymbol{x}, t = 0) = + \sin(x_1) \cos(x_2) \cos(x_3) \ ,$$
$$u_2(\boldsymbol{x}, t = 0) = - \cos(x_1) \sin(x_2) \cos(x_3) \ ,$$
$$u_3(\boldsymbol{x}, t = 0) = 0 \ ,$$
$$p(\boldsymbol{x}, t = 0) = \frac{1}{16} \left(\cos(2x_1) + \cos(2x_2) \right) \left(\cos(2x_3) + 2 \right) \ ,$$

the flow transitions to turbulence in the absence of body forces, $\boldsymbol{f} = \boldsymbol{0}$. The Reynolds number is $\mathrm{Re} = 1/\nu$, and we consider both the standard setting $\mathrm{Re} = 1600$ and the inviscid limit $\mathrm{Re} \to \infty$ in the present work. The simulated time interval is $0 \le t \le T = 20$. At start time, the domain $\Omega_0 = [-L/2, L/2]^3 = [-\pi, \pi]^3$ is a Cartesian box that deforms over time according to the mesh motion described in equation (63), with an amplitude of $A = \pi/6$ and varying mesh velocities characterized by period times decreasing from $T_\mathrm{G} = 20$

to $T_{\mathrm{G}} = 1$. Periodic boundaries are used in all coordinate directions. The domain boundaries are moving for the given mesh motion, but the mesh deformation is defined periodically in order to ensure consistency with the periodic boundary conditions. An illustration of the mesh deformation for the above parameters is given in Figure 7. We consider meshes that are originally Cartesian with $N_{\mathrm{el}} = (2^l)^3$ elements, where l denotes the level of refinement, and denote the number of velocity nodes $(2^l(k+1))^3$ as effective mesh resolution. A BDF2 time integration scheme along with an explicit treatment of the convective term is used for all solver types, with $J_p = 2$ for the dual splitting scheme and $J_p = 1$ for the pressure-correction scheme unless specified otherwise. Moreover, we use adaptive time-stepping, equation (61), with a Courant number of $\mathrm{Cr} = 0.2$.

(a) $\|\boldsymbol{u}(\boldsymbol{x}, t = 0)\|$ (b) $\|\boldsymbol{u}(\boldsymbol{x}, t = 5)\|$ (c) $\|\boldsymbol{u}(\boldsymbol{x}, t = 10)\|$ (d) $\|\boldsymbol{u}(\boldsymbol{x}, t = 15)\|$ (e) $\|\boldsymbol{u}(\boldsymbol{x}, t = 20)\|$
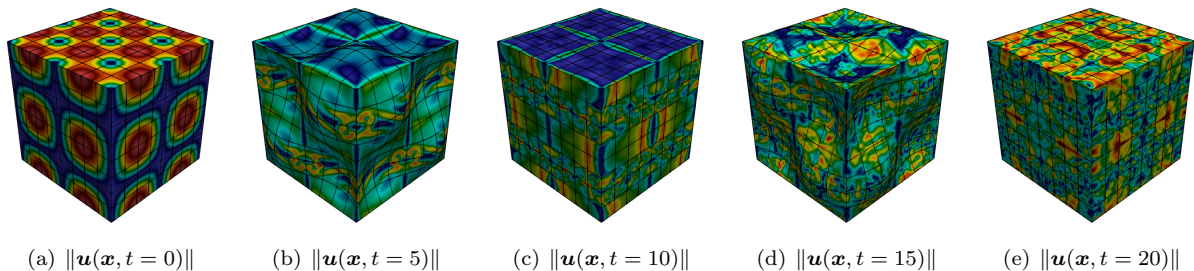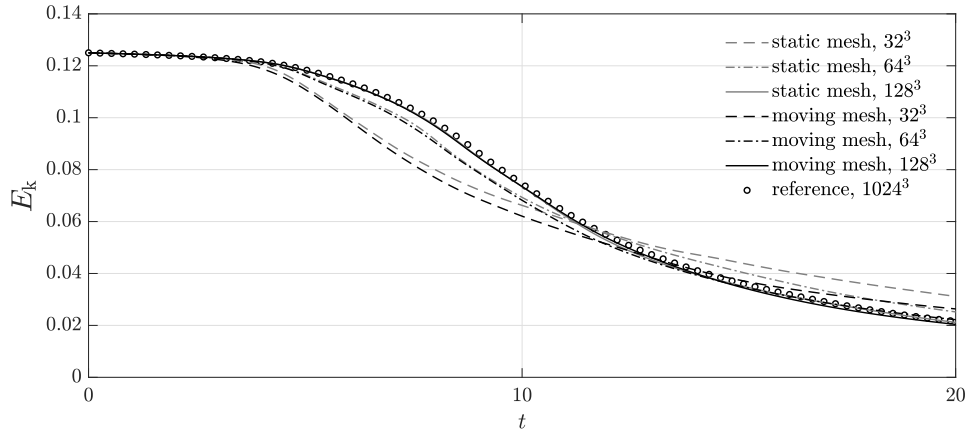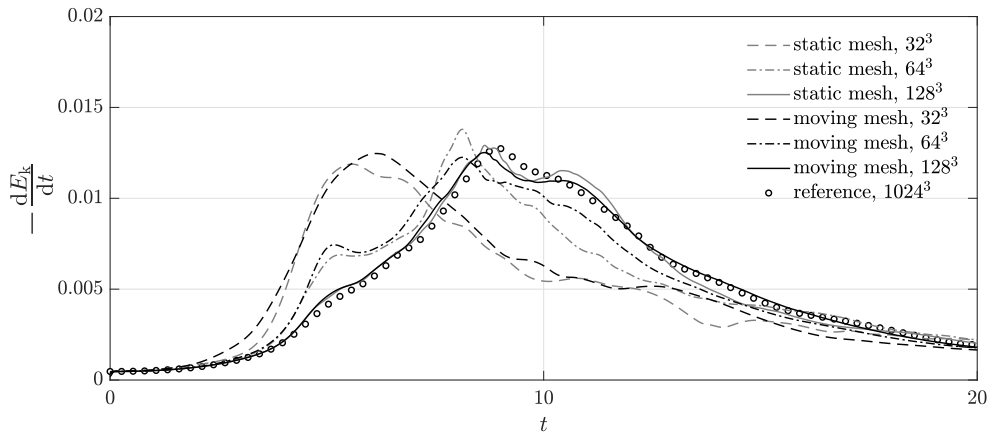
Figure 7: Taylor–Green vortex problem at Re = 1600: visualization of velocity magnitude at different times for a spatial resolution with $l = 3$ and polynomial degree $k = 7$. The parameters of the mesh deformation are $A = \pi/6$ and $T_{\mathrm{G}} = 20$. The color map has been rescaled for each time instant, where red indicates high velocity and blue low velocity. The results shown have been simulated with the dual splitting scheme.

In a first set of experiments, the viscous case at Re = 1600 is investigated. In Figure 8, results for the kinetic energy $E_k = \int_{\Omega_h} \frac{1}{2} \boldsymbol{u}_h \cdot \boldsymbol{u}_h \mathrm{d}\Omega / \int_{\Omega_h} \mathrm{d}\Omega$ and the dissipation rate of the kinetic energy obtained on the moving mesh are compared to results on a static Cartesian mesh. A mesh refinement study for degree $k = 3$ is performed using a rather slow mesh motion with a period time of $T_{\mathrm{G}} = T$. The results converge towards the accurate DNS reference solution under mesh refinement, and the solution quality is comparable for static and moving meshes. In Figure 9, we test the robustness w.r.t. the grid velocity for the 64^3 mesh resolution by decreasing the period of the mesh motion down to $T_{\mathrm{G}} = T/20 = 1$, resulting in a very fast mesh motion. With increasing mesh velocity, an oscillating behavior can be observed in the kinetic energy dissipation rate where the frequency of these oscillations follows the mesh motion. However, the temporal evolution of the kinetic energy is almost indistinguishable for the different mesh velocities. It can be observed that the oscillations in the dissipation rate are very small in the beginning of the simulation where the solution is smooth, while the oscillations grow once the transition to a turbulent state took place. In Section 4.5, it was noted that the ALE transport term does not contribute to the energy evolution apart from the upwind stabilization term. However, this only holds under the assumption of exact numerical integration, which is not fulfilled on generally deformed geometries. A possible explanation for the results in Figure 9 could therefore be that integration errors (which are larger for non-smooth solutions or under-resolved scenarios) are amplified if the period of the mesh motion T_{G} tends to zero and the mesh velocity tends to infinity. This is supported by the observation that the oscillications are larger on coarser meshes. Let us mention that this experiment is performed here to test the robustness of the solver and that such a scenario (increasing the mesh velocity for a fixed fluid velocity) is not representative of a fluid–structure interaction problem for which the fluid has to follow the motion of the fluid–structure interface due to no-slip conditions.

In a second set of experiments, the inviscid Taylor–Green vortex problem is studied, which is considered one of the most challenging benchmark examples to test the robustness of a flow solver for turbulent flows due to the absence of viscous dissipation, see for example [12]. Although the test case is academic, it can be expected that if a numerical method is robust for the inviscid Taylor–Green problem, it can also be successfully applied to practical, engineering problems. Table 4 shows results of robustness tests for the inviscid limit, where polynomial degrees of $k = 3, 5, 7, 11, 15$ are considered for several spatial refinement
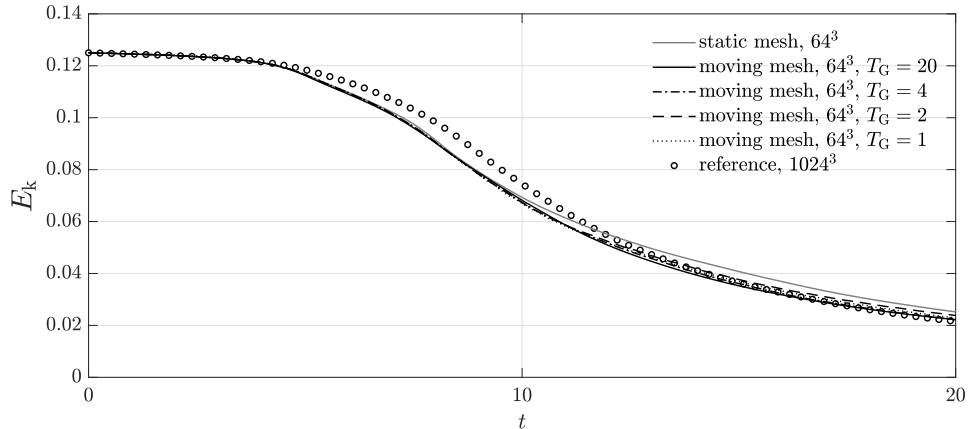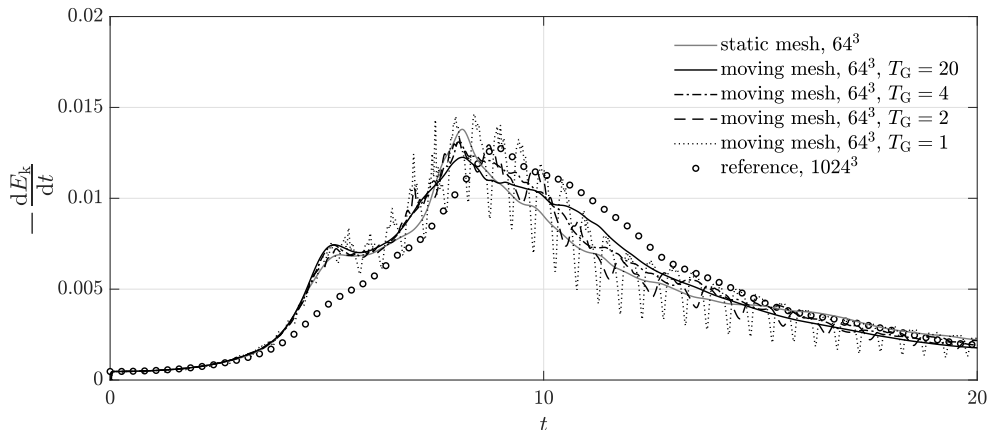
(a) Kinetic energy.

(b) Kinetic energy dissipation rate.

Figure 8: Taylor–Green vortex problem at Re = 1600: comparison between static and moving meshes for polynomial degree $k = 3$ for increasing mesh resolutions of 32^3, 64^3, and 128^3. The results shown have been simulated with the dual splitting scheme.

levels l. The highest mesh velocity studied above (corresponding to $T_{\mathrm{G}} = 1$) is used for the inviscid TGV simulations. Robustness is particularly critical for coarse spatial resolutions where the flow is severely under-resolved, see for example [29] for similar considerations on static meshes and a comparison of the present stabilized DG approach to a non-stabilized one, which is why we start our investigations with a mesh consisting of only one element, $l = 0$. With the standard penalty factor of $\zeta = 1$ for the divergence and continuity penalty terms, robustness is achieved for all spatial resolutions and for all solver types (monolithic, dual splitting, pressure-correction). There is one exception: For the pressure-correction scheme and the parameters described above, using the incremental formulation lead to instabilities for the coarsest possible spatial resolution, $l = 0$ and $k = 3$, while the simulation was again stable for the non-incremental formulation ($J_p = 0$). Overall, these results are encouraging in the sense that the stabilized DG approach proposed in [29] for static meshes is well designed, meaning that robustness carries over to moving meshes without having to adjust discretization parameters. At the same time, we emphasize again that it is unclear to which extent energy stability can be guaranteed theoretically for the present stabilized DG approach.

(a) Kinetic energy.

(b) Kinetic energy dissipation rate.

Figure 9: Taylor–Green vortex problem at Re = 1600: comparison between static and moving meshes for polynomial degree $k = 3$ for increasing mesh velocity (decreasing period times of $T_\mathrm{G} = 20, 4, 2, 1$). A mesh with polynomial degree $k = 3$ and effective resolution of 64^3 is considered. The results shown have been simulated with the dual splitting scheme.

6. Conclusion and outlook

We presented ALE-DG methods for the incompressible Navier–Stokes equations that are up to third-order accurate in time and arbitrarily high-order accurate in space for sufficiently smooth problems. Moving mesh formulations are derived for both monolithic and splitting approaches based on a method-of-lines approach, considering both implicit and explicit formulations of the convective term. The time integration framework relies on BDF and extrapolation schemes and extends naturally to adaptive time-stepping. Stable and high-order accurate boundary conditions are derived for the splitting-type approaches. The ALE methods are designed to automatically fulfill the geometric conservation law. An important aspect is that the proposed methods are simple to implement since the equations are solved on the deformed geometry, i.e., the generic finite element software takes care of the mapping and the geometry terms, and only one instance of the mesh is stored at a time. Fast matrix-free evaluation techniques are applied for all parts of the Navier–Stokes solvers as in the Eulerian case. A key feature is the use of consistent divergence and continuity penalty terms to stabilize the method for under-resolved turbulent flows. Numerical results demonstrate optimality in terms of convergence rates and the geometric conservation law, and robustness and accuracy

29

Table 4: Inviscid 3D Taylor–Green vortex problem: robustness of proposed ALE-DG methods is tested for refinement levels $l = 0, ..., 5$ and polynomial degrees $k = 3, 5, 7, 11, 15$. All simulations completed successfully as indicated by the symbol ✓. The sign '−' indicates that the specific spatial resolution is not considered since the effective resolution is limited to coarse discretizations for this robustness test.

	Polynomial degree k				
l	$k = 3$	$k = 5$	$k = 7$	$k = 11$	$k = 15$
0	✓	✓	✓	✓	✓
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	−
4	✓	✓	✓	−	−
5	✓	−	−	−	−

of the proposed methods has been investigated for under-resolved turbulent flows. In the future, application to fluid–structure interaction problems is planned.

Appendix A. Adaptive time-stepping

For variable time step sizes, the time integration constants can be derived from Lagrange interpolation polynomials. To obtain the BDF coefficients γ_0^n, α_i^n, the derivative of the Lagrange interpolation polynomials with support points at $t_{n+1}, t_n, ..., t_{n-J+1}$ is evaluated at time t_{n+1}. To obtain the extrapolation coefficients β_i^n, the Lagrange interpolation polynomials with support points at $t_n, ..., t_{n-J+1}$ are evaluated at time t_{n+1}. Table A.5 summarizes the time integration constants for variable time step sizes using the notation introduced in Section 3.

Table A.5: Coefficients of BDF time integration scheme and extrapolation scheme for adaptive time step sizes.

	BDF1	BDF2	BDF3
γ_0^n	1	$\frac{2\Delta t_n + \Delta t_{n-1}}{\Delta t_n + \Delta t_{n-1}}$	$1 + \frac{\Delta t_n}{\Delta t_n + \Delta t_{n-1}} + \frac{\Delta t_n}{\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2}}$
α_0^n	1	$\frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_{n-1}}$	$\frac{(\Delta t_n + \Delta t_{n-1})(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})}{\Delta t_{n-1}(\Delta t_{n-1} + \Delta t_{n-2})}$
α_1^n	-	$-\frac{\Delta t_n^2}{(\Delta t_n + \Delta t_{n-1})\Delta t_{n-1}}$	$-\frac{\Delta t_n^2(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})}{(\Delta t_n + \Delta t_{n-1})\Delta t_{n-1}\Delta t_{n-2}}$
α_2^n	-	-	$\frac{\Delta t_n^2(\Delta t_n + \Delta t_{n-1})}{(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})(\Delta t_{n-1} + \Delta t_{n-2})\Delta t_{n-2}}$
β_0^n	1	$\frac{\Delta t_n + \Delta t_{n-1}}{\Delta t_{n-1}}$	$\frac{(\Delta t_n + \Delta t_{n-1})(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})}{\Delta t_{n-1}(\Delta t_{n-1} + \Delta t_{n-2})}$
β_1^n	-	$-\frac{\Delta t_n}{\Delta t_{n-1}}$	$-\frac{\Delta t_n(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})}{\Delta t_{n-1}\Delta t_{n-2}}$
β_2^n	-	-	$\frac{\Delta t_n(\Delta t_n + \Delta t_{n-1})}{(\Delta t_{n-1} + \Delta t_{n-2})\Delta t_{n-2}}$

Appendix B. Energy conservation property on moving meshes

In this section, we derive an energy conservation property according to equation (51) for the ALE form of the incompressible Navier–Stokes equations under the assumption of $\nu = 0$, $\boldsymbol{f} = \boldsymbol{0}$, and periodic boundaries.

We begin with

$$\int_{\Omega(t)} \frac{\partial \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}}{\partial t}\bigg|_{\boldsymbol{\chi}}\,\mathrm{d}\Omega = \int_{\Omega(t)} \boldsymbol{u}\cdot\frac{\partial\boldsymbol{u}}{\partial t}\bigg|_{\boldsymbol{\chi}}\,\mathrm{d}\Omega$$
$$= -\int_{\Omega(t)} \boldsymbol{u}\cdot\big(((\boldsymbol{u}-\boldsymbol{u}_{\mathrm{G}})\cdot\nabla)\boldsymbol{u} + \nabla p\big)\,\mathrm{d}\Omega\,, \tag{B.1}$$

where the momentum equation in ALE form, equation (5), has been inserted in the second step. Next, the convective term is reformulated by making use of the identity $\boldsymbol{u}\cdot\nabla\boldsymbol{u}\cdot\boldsymbol{w} = -\frac{1}{2}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\nabla\cdot\boldsymbol{w} + \frac{1}{2}\nabla\cdot\left(\boldsymbol{w}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\right)$ with $\boldsymbol{w} = \boldsymbol{u} - \boldsymbol{u}_{\mathrm{G}}$, so that we obtain after applying Gauss' divergence theorem for the second term

$$\int_{\Omega(t)} \boldsymbol{u}\cdot\nabla\boldsymbol{u}\cdot(\boldsymbol{u}-\boldsymbol{u}_{\mathrm{G}})\,\mathrm{d}\Omega = -\frac{1}{2}\int_{\Omega(t)}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\nabla\cdot(\boldsymbol{u}-\boldsymbol{u}_{\mathrm{G}})\,\mathrm{d}\Omega + \frac{1}{2}\int_{\partial\Omega(t)}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)(\boldsymbol{u}-\boldsymbol{u}_{\mathrm{G}})\cdot\boldsymbol{n}\,\mathrm{d}\Gamma$$
$$= +\frac{1}{2}\int_{\Omega(t)}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\nabla\cdot\boldsymbol{u}_{\mathrm{G}}\,\mathrm{d}\Omega\,. \tag{B.2}$$

In the second step, we exploited that the surface integral vanishes due to the assumption of periodicity, and that $\nabla\cdot\boldsymbol{u} = 0$ holds (continuity equation). For the pressure gradient term, integration by parts is performed to obtain

$$\int_{\Omega(t)} \boldsymbol{u}\nabla p\,\mathrm{d}\Omega = -\int_{\Omega(t)} \underbrace{\nabla\cdot\boldsymbol{u}}_{=0}\,p\,\mathrm{d}\Omega + \int_{\partial\Omega(t)} p\,\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma = 0\,, \tag{B.3}$$

where the volume integral vanished due to $\nabla\cdot\boldsymbol{u} = 0$ and the surface integral due to periodic boundaries. Inserting equations (B.2) and (B.3) into equation (B.1), we arrive at the result

$$\int_{\Omega(t)} \frac{\partial \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}}{\partial t}\bigg|_{\boldsymbol{\chi}}\,\mathrm{d}\Omega + \int_{\Omega(t)} \frac{1}{2}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\nabla\cdot\boldsymbol{u}_{\mathrm{G}}\,\mathrm{d}\Omega = 0\,.$$

To explain why this equation describes conservation of kinetic energy, we transform the integral onto the reference domain Ω_0, using the Jacobian $\boldsymbol{J} = \partial\boldsymbol{x}/\partial\boldsymbol{\chi}$

$$\int_{\Omega_0} \frac{\partial \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}}{\partial t}\bigg|_{\boldsymbol{\chi}}\,\det\boldsymbol{J}\mathrm{d}\Omega + \int_{\Omega_0} \frac{1}{2}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\nabla\cdot\boldsymbol{u}_{\mathrm{G}}\,\det\boldsymbol{J}\,\mathrm{d}\Omega = 0\,.$$

Using the relation

$$\frac{\partial\det\boldsymbol{J}}{\partial t}\bigg|_{\boldsymbol{\chi}} = \nabla\cdot\boldsymbol{u}_{\mathrm{G}}\det\boldsymbol{J}\,,$$

which is frequently used in the context of ALE derivations, see for example [15], we can further simplify the integral

$$0 = \int_{\Omega_0} \frac{\partial \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}}{\partial t}\bigg|_{\boldsymbol{\chi}}\det\boldsymbol{J} + \frac{1}{2}\left(\boldsymbol{u}\cdot\boldsymbol{u}\right)\frac{\partial\det\boldsymbol{J}}{\partial t}\bigg|_{\boldsymbol{\chi}}\,\mathrm{d}\Omega = \int_{\Omega_0} \frac{\partial \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\det\boldsymbol{J}}{\partial t}\bigg|_{\boldsymbol{\chi}}\,\mathrm{d}\Omega =$$
$$= \frac{\partial}{\partial t}\int_{\Omega_0} \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\det\boldsymbol{J}\,\mathrm{d}\Omega\bigg|_{\boldsymbol{\chi}}\,.$$

In the last step, the time derivative has been pulled out of the integral since the integral is performed over Ω_0, which does not depend on t. From the above relation, it is obvious that energy is conserved under the given assumptions.

Appendix C. Energy balance of convective term

In this section, we show that equation (58) holds.

Proof The first goal is to achieve that the volume integral of the convective term contains the divergence of the velocity and the grid velocity. Similar to the continuous case in Appendix B, we make use of the identity

$$\boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h \cdot \boldsymbol{w}_h = -\frac{1}{2} \nabla \cdot \boldsymbol{w}_h \left(\boldsymbol{u}_h \cdot \boldsymbol{u}_h\right) + \frac{1}{2} \nabla \cdot \left(\boldsymbol{w}_h \left(\boldsymbol{u}_h \cdot \boldsymbol{u}_h\right)\right) ,$$

with $\boldsymbol{w}_h = \boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}$. Note that we cannot exploit that the divergence of the discrete velocity $\boldsymbol{u}_h$ is zero in the discrete case. Inserting the above identity into the left-hand side of equation (58) and applying Gauss' divergence theorem yields

$$
\begin{aligned}
&\sum_{e=1}^{N_{\mathrm{el}}} \left(\left(\boldsymbol{u}_h, (\nabla \boldsymbol{u}_h) \cdot (\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h})\right)_{\Omega_e} - \left(\boldsymbol{u}_h, ((\{\!\{\boldsymbol{u}_h\}\!\} - \boldsymbol{u}_{\mathrm{G},h}) \cdot \boldsymbol{n}) \frac{1}{2} [\boldsymbol{u}_h]\right)_{\partial \Omega_e} \right) \\
&= -\sum_{e=1}^{N_{\mathrm{el}}} \left(\frac{1}{2} \left(\nabla \cdot (\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}), \boldsymbol{u}_h \cdot \boldsymbol{u}_h\right)_{\Omega_e} \right) \\
&\quad + \sum_{e=1}^{N_{\mathrm{el}}} \left(\frac{1}{2} \left(\boldsymbol{u}_h \cdot \boldsymbol{u}_h, (\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}) \cdot \boldsymbol{n}\right)_{\partial \Omega_e} - \left(\boldsymbol{u}_h, ((\{\!\{\boldsymbol{u}_h\}\!\} - \boldsymbol{u}_{\mathrm{G},h}) \cdot \boldsymbol{n}) \frac{1}{2} [\boldsymbol{u}_h]\right)_{\partial \Omega_e} \right) \\
&= -\frac{1}{2} \left(\nabla \cdot (\boldsymbol{u}_h - \boldsymbol{u}_{\mathrm{G},h}), \boldsymbol{u}_h \cdot \boldsymbol{u}_h\right)_{\Omega_h} \\
&\quad \underbrace{-\frac{1}{2} \left([\boldsymbol{u}_h \cdot \boldsymbol{u}_h], \boldsymbol{u}_{\mathrm{G},h} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}} + \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \boldsymbol{u}_{\mathrm{G},h} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}}}_{=0 \text{ (linear transport)}} \\
&\quad + \frac{1}{2} \left(\boldsymbol{u}_h^- \cdot \boldsymbol{n}^-, \boldsymbol{u}_h^- \cdot \boldsymbol{u}_h^-\right)_{\Gamma_h^{\mathrm{int}}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{n}^+, \boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+\right)_{\Gamma_h^{\mathrm{int}}} - \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}} ,
\end{aligned}
\tag{C.1}
$$

where the summation over all elements has been performed in the second step. It can easily be verified that the surface integrals related to the grid velocity add up to zero, as expected for a linear transport term. The third row on the right-hand side of the above equation is more difficult to summarize, since none of the two slots is the same for both sides of the face (originating from the nonlinearity of the transport term), so we

need to insert a relation like $\boldsymbol{u}_h^+ = \boldsymbol{u}_h^- - [\boldsymbol{u}_h]$ in order to be able to combine these terms

$$
\begin{aligned}
&+ \frac{1}{2} \left(\boldsymbol{u}_h^- \cdot \boldsymbol{n}^-, \boldsymbol{u}_h^- \cdot \boldsymbol{u}_h^-\right)_{\Gamma_h^{\mathrm{int}}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{n}^+, \boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+\right)_{\Gamma_h^{\mathrm{int}}} - \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}} \\
&= + \underbrace{\frac{1}{2} \left(\boldsymbol{u}_h^- \cdot \boldsymbol{u}_h^-, \boldsymbol{u}_h^- \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} - \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^-, \boldsymbol{u}_h^- \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}}}_{\text{combine}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot [\boldsymbol{u}_h], \boldsymbol{u}_h^- \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} \\
&\quad + \underbrace{\frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^-, [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} - \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot [\boldsymbol{u}_h], [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}}}_{\text{combine}} - \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}} \\
&= + \underbrace{\frac{1}{2} \left([\boldsymbol{u}_h] \cdot \boldsymbol{u}_h^-, \boldsymbol{u}_h^- \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot [\boldsymbol{u}_h], \boldsymbol{u}_h^- \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}}}_{\text{combine}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+, [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} \\
&\quad - \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}} \\
&= + \underbrace{\left([\boldsymbol{u}_h] \cdot \frac{\boldsymbol{u}_h^- + \boldsymbol{u}_h^+}{2}, \boldsymbol{u}_h^- \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} - \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \{\!\{\boldsymbol{u}_h\}\!\} \cdot \boldsymbol{n}\right)_{\Gamma_h^{\mathrm{int}}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+, [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}}}_{\text{combine}} \\
&= + \underbrace{\left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\}, \left(\boldsymbol{u}_h^- - \{\!\{\boldsymbol{u}_h\}\!\}\right) \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} + \frac{1}{2} \left(\boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+, [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}}}_{\text{combine}} \\
&= + \frac{1}{2} \left([\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\} + \boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+, [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} \\
&= + \frac{1}{2} \left(\{\!\{\boldsymbol{u}_h \cdot \boldsymbol{u}_h\}\!\}, [\boldsymbol{u}_h] \cdot \boldsymbol{n}^-\right)_{\Gamma_h^{\mathrm{int}}} ,
\end{aligned}
\tag{C.2}
$$

where one can easily verify that $[\boldsymbol{u}_h] \cdot \{\!\{\boldsymbol{u}_h\}\!\} + \boldsymbol{u}_h^+ \cdot \boldsymbol{u}_h^+ = \{\!\{\boldsymbol{u}_h \cdot \boldsymbol{u}_h\}\!\}$. Inserting equation (C.2) into equation (C.1) completes the proof. $\qquad\square$

Acknowledgments

References

[1] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, Journal of Computational Physics 14 (3) (1974) 227 – 253. `doi:10.1016/0021-9991(74)90051-5`.

[2] J. Donea, S. Giuliani, J. Halleux, An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid-structure interactions, Computer Methods in Applied Mechanics and Engineering 33 (1) (1982) 689 – 723. `doi:10.1016/0045-7825(82)90128-1`.

[3] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodrguez-Ferran, Arbitrary Lagrangian–Eulerian Methods, American Cancer Society, 2017, pp. 1–23. `doi:10.1002/9781119176817.ecm2009`.

[4] J. Donea, P. Fasoli-Stella, S. Giuliani, Lagrangian and Eulerian finite element techniques for transient fluid-structure interaction problems, Therm and Fluid/Struct Dyn Anal.

[5] T. J. Hughes, W. K. Liu, T. K. Zimmermann, Lagrangian-Eulerian finite element formulation for incompressible viscous flows, Computer Methods in Applied Mechanics and Engineering 29 (3) (1981) 329 – 349. `doi:10.1016/0045-7825(81)90049-9`.

[6] A. Beskok, T. C. Warburton, An unstructured hp finite-element scheme for fluid flow and heat transfer in moving domains, Journal of Computational Physics 174 (2) (2001) 492 – 509. `doi:10.1006/jcph.2001.6885`.

[7] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, Computer Methods in Applied Mechanics and Engineering 134 (1) (1996) 71 – 90. `doi:10.1016/0045-7825(96)01028-6`.

[8] I. Lomtev, R. Kirby, G. Karniadakis, A discontinuous Galerkin ALE method for compressible viscous flows in moving domains, Journal of Computational Physics 155 (1) (1999) 128 – 159. doi:10.1006/jcph.1999.6331.

[9] V.-T. Nguyen, An arbitrary Lagrangian–Eulerian discontinuous Galerkin method for simulations of flows over variable geometries, Journal of Fluids and Structures 26 (2) (2010) 312 – 329. doi:10.1016/j.jfluidstructs.2009.11.002.

[10] D. J. Mavriplis, C. R. Nastase, On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes, Journal of Computational Physics 230 (11) (2011) 4285 – 4300, special issue High Order Methods for CFD Problems. doi:10.1016/j.jcp.2011.01.022.

[11] P.-O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains, Computer Methods in Applied Mechanics and Engineering 198 (2009) 1585–1595. doi:10.1016/j.cma.2009.01.012.

[12] G. Schnücke, N. Krais, T. Bolemann, G. J. Gassner, Entropy stable discontinuous Galerkin schemes on moving meshes with summation-by-parts property for hyperbolic conservation laws, arXiv preprint arXiv:1812.09093.

[13] P. D. Thomas, C. K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA Journal 17 (10) (1979) 1030–1037. doi:10.2514/3.61273.

[14] C. Farhat, P. Geuzaine, Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids, Computer Methods in Applied Mechanics and Engineering 193 (39) (2004) 4073 – 4095, the Arbitrary Lagrangian-Eulerian Formulation. doi:10.1016/j.cma.2003.09.027.

[15] C. Förster, W. A. Wall, E. Ramm, On the geometric conservation law in transient flow calculations on deforming domains, International Journal for Numerical Methods in Fluids 50 (12) (2006) 1369–1379. doi:10.1002/fld.1093.

[16] S. Étienne, A. Garon, D. Pelletier, Perspective on the geometric conservation law and finite element methods for ALE simulations of incompressible flow, Journal of Computational Physics 228 (7) (2009) 2313 – 2333. doi:10.1016/j.jcp.2008.11.032.

[17] B. Cockburn, G. Kanschat, D. Schötzau, A locally conservative LDG method for the incompressible Navier–Stokes equations, Mathematics of Computation 74 (251) (2005) 1067–1095.

[18] B. Cockburn, G. Kanschat, D. Schötzau, A note on discontinuous Galerkin divergence-free solutions of the Navier–Stokes equations, Journal of Scientific Computing 31 (1) (2007) 61–73. doi:10.1007/s10915-006-9107-7.

[19] B. Cockburn, G. Kanschat, D. Schötzau, An equal-order DG method for the incompressible Navier–Stokes equations, Journal of Scientific Computing 40 (1-3) (2009) 188–210.

[20] F. Bassi, A. Crivellini, D. D. Pietro, S. Rebay, An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations, Journal of Computational Physics 218 (2) (2006) 794 – 815. doi:10.1016/j.jcp.2006.03.006.

[21] K. Shahbazi, P. F. Fischer, C. R. Ethier, A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations, Journal of Computational Physics 222 (1) (2007) 391 – 407. doi:10.1016/j.jcp.2006.07.029.

[22] J. S. Hesthaven, T. Warburton, Nodal discontinuous Galerkin methods: Algorithms, analysis, and applications, Springer, 2007. doi:10.1007/978-0-387-72067-8.

[23] L. Botti, D. A. D. Pietro, A pressure-correction scheme for convection-dominated incompressible flows with discontinuous velocity and continuous pressure, Journal of Computational Physics 230 (3) (2011) 572 – 585. doi:10.1016/j.jcp.2010.10.004.

[24] E. Ferrer, R. H. J. Willden, A high order discontinuous Galerkin finite element solver for the incompressible Navier–Stokes equations, Computers & Fluids 46 (1) (2011) 224 – 230. doi:10.1016/j.compfluid.2010.10.018.

[25] B. Klein, F. Kummer, M. Oberlack, A SIMPLE based discontinuous Galerkin solver for steady incompressible flows, Journal of Computational Physics 237 (2013) 235 – 250. doi:10.1016/j.jcp.2012.11.051.

[26] D. T. Steinmoeller, M. Stastna, K. G. Lamb, A short note on the discontinuous Galerkin discretization of the pressure projection operator in incompressible flow, Journal of Computational Physics 251 (2013) 480 – 486. doi:10.1016/j.jcp.2013.05.036.

[27] S. M. Joshi, P. J. Diamessis, D. T. Steinmoeller, M. Stastna, G. N. Thomsen, A post-processing technique for stabilizing the discontinuous pressure projection operator in marginally-resolved incompressible inviscid flow, Computers & Fluids 139 (2016) 120 – 129, 13th USNCCM International Symposium of High-Order Methods for Computational Fluid Dynamics - A special issue dedicated to the 60th birthday of Professor David Kopriva. doi:10.1016/j.compfluid.2016.04.021.

[28] B. Krank, N. Fehn, W. A. Wall, M. Kronbichler, A high-order semi-explicit discontinuous Galerkin solver for 3D incompressible flow with application to DNS and LES of turbulent channel flow, Journal of Computational Physics 348 (2017) 634–659. doi:10.1016/j.jcp.2017.07.039.

[29] N. Fehn, W. A. Wall, M. Kronbichler, Robust and efficient discontinuous Galerkin methods for under-resolved turbulent incompressible flows, Journal of Computational Physics 372 (2018) 667–693. doi:10.1016/j.jcp.2018.06.037.

[30] C. Lehrenfeld, J. Schöberl, High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows, Computer Methods in Applied Mechanics and Engineering 307 (2016) 339 – 361. doi:10.1016/j.cma.2016.04.025.

[31] M. Piatkowski, S. Mthing, P. Bastian, A stable and high-order accurate discontinuous Galerkin based splitting method for the incompressible Navie–Stokes equations, Journal of Computational Physics 356 (2018) 220 – 239. doi:10.1016/j.jcp.2017.11.035.

[32] P. W. Schroeder, G. Lube, Divergence-free H(div)-FEM for time-dependent incompressible flows with applications to high Reynolds number vortex dynamics, Journal of Scientific Computing 75 (2) (2018) 830–858. doi:10.1007/s10915-017-0561-1.

[33] S. Rhebergen, G. N. Wells, A hybridizable discontinuous Galerkin method for the Navier–Stokes equations with pointwise divergence-free velocity field, Journal of Scientific Computing 76 (3) (2018) 1484–1501. doi:10.1007/s10915-018-0671-4.

[34] P. W. Schroeder, G. Lube, Stabilised dG-FEM for incompressible natural convection flows with boundary and moving

interior layers on non-adapted meshes, Journal of Computational Physics 335 (2017) 760 – 779. `doi:10.1016/j.jcp.2017.01.055`.

[35] M. Akbas, A. Linke, L. G. Rebholz, P. W. Schroeder, The analogue of grad–div stabilization in DG methods for incompressible flows: Limiting behavior and extension to tensor-product meshes, Computer Methods in Applied Mechanics and Engineering 341 (2018) 917 – 938. `doi:10.1016/j.cma.2018.07.019`.

[36] N. Fehn, M. Kronbichler, C. Lehrenfeld, G. Lube, P. W. Schroeder, High-order DG solvers for under-resolved turbulent incompressible flows: A comparison of L^2 and $H(\mathrm{div})$ methods, International Journal for Numerical Methods in Fluids in press (2019) n/a. `doi:10.1002/fld.4763`.

[37] E. Ferrer, R. H. J. Willden, A high order discontinuous Galerkin Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes, Journal of Computational Physics 231 (21) (2012) 7037 – 7056. `doi:10.1016/j.jcp.2012.04.039`.

[38] Y. Wang, A. Quaini, S. Čanić, A higher-order discontinuous Galerkin/arbitrary Lagrangian Eulerian partitioned approach to solving fluid–structure interaction problems with incompressible, viscous fluids and elastic structures, Journal of Scientific Computing 76 (1) (2018) 481–520. `doi:10.1007/s10915-017-0629-y`.

[39] S. A. Orszag, M. Israeli, M. O. Deville, Boundary conditions for incompressible flows, J. Sci. Comput. 1 (1) (1986) 75–111. `doi:10.1007/BF01061454`.

[40] G. E. Karniadakis, M. Israeli, S. A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, Journal of Computational Physics 97 (2) (1991) 414 – 443. `doi:10.1016/0021-9991(91)90007-8`.

[41] N. Fehn, W. A. Wall, M. Kronbichler, On the stability of projection methods for the incompressible Navier–Stokes equations based on high-order discontinuous Galerkin discretizations, Journal of Computational Physics 351 (2017) 392–421. `doi:10.1016/j.jcp.2017.09.031`.

[42] L. Xu, X. Xu, X. Ren, Y. Guo, Y. Feng, X. Yang, Stability evaluation of high-order splitting method for incompressible flow based on discontinuous velocity and continuous pressure, Advances in Mechanical Engineering 11 (10) (2019) 1687814019855586. `doi:10.1177/1687814019855586`.

[43] S. Rhebergen, B. Cockburn, A space–time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains, Journal of Computational Physics 231 (11) (2012) 4185 – 4204. `doi:10.1016/j.jcp.2012.02.011`.

[44] T. L. Horváth, S. Rhebergen, A locally conservative and energy-stable finite-element method for the Navier–Stokes problem on time-dependent domains, International Journal for Numerical Methods in Fluids 89 (12) (2019) 519–532. `doi:10.1002/fld.4707`.

[45] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. D. Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. Kirby, S. Sherwin, Nektar++: An open-source spectral/hp element framework, Computer Physics Communications 192 (2015) 205 – 219. `doi:10.1016/j.cpc.2015.02.008`.

[46] P. Fischer, S. Kerkemeier, A. Peplinski, D. Shaver, A. Tomboulides, M. Min, A. Obabko, E. Merzari, NEK5000 Web page, https://nek5000.mcs.anl.gov (2020).

[47] G. J. Gassner, A. D. Beck, On the accuracy of high-order discretizations for underresolved turbulence simulations, Theoretical and Computational Fluid Dynamics 27 (3) (2013) 221–237. `doi:10.1007/s00162-011-0253-7`.

[48] R. Moura, G. Mengaldo, J. Peiró, S. Sherwin, On the eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES / under-resolved DNS of Euler turbulence, Journal of Computational Physics 330 (2017) 615 – 623. `doi:10.1016/j.jcp.2016.10.056`.

[49] N. Fehn, W. A. Wall, M. Kronbichler, Efficiency of high-performance discontinuous Galerkin spectral element methods for under-resolved turbulent incompressible flows, International Journal for Numerical Methods in Fluids 88 (1) (2018) 32–54. `doi:10.1002/fld.4511`.

[50] M. Kronbichler, W. A. Wall, A Performance Comparison of Continuous and Discontinuous Galerkin Methods with Fast Multigrid Solvers, SIAM Journal on Scientific Computing 40 (5) (2018) A3423–A3448. `doi:10.1137/16M110455X`.

[51] M. Kronbichler, K. Kormann, Fast matrix-free evaluation of discontinuous Galerkin finite element operators, ACM Trans. Math. Softw. 45 (3) (2019) 29:1–29:40. `doi:10.1145/3325864`.

[52] J. L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Computer Methods in Applied Mechanics and Engineering 195 (4447) (2006) 6011 – 6045. `doi:10.1016/j.cma.2005.10.010`.

[53] G. E. Karniadakis, S. J. Sherwin, Spectral/hp element methods for computational fluid dynamics, Oxford University Press, 2013. `doi:10.1093/acprof:oso/9780198528692.001.0001`.

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

[55] C. Hirt, J. Cook, Calculating three-dimensional flows around structures and over rough terrain, Journal of Computational Physics 10 (2) (1972) 324–340.

[56] K. Goda, A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows, Journal of Computational Physics 30 (1) (1979) 76–95.

[57] J. Van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 870–891.

[58] L. Timmermans, P. Minev, F. Van De Vosse, An approximate projection scheme for incompressible flow using spectral elements, International Journal for Numerical Methods in Fluids 22 (7) (1996) 673–688.

[59] J. Guermond, J. Shen, On the error estimates for the rotational pressure-correction projection methods, Mathematics of Computation 73 (248) (2004) 1719–1737.

[60] D. Wang, S. J. Ruuth, Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations, Journal of Computational Mathematics (2008) 838–855.

[61] J. A. Escobar-Vargas, P. J. Diamessis, T. Sakai, A spectral quadrilateral multidomain penalty method model for high

Reynolds number incompressible stratified flows, International Journal for Numerical Methods in Fluids 75 (6) (2014) 403–425. `doi:10.1002/fld.3899`.

[62] P. Gao, J. Ouyang, P. Dai, W. Zhou, A coupled continuous and discontinuous finite element method for the incompressible flows, International Journal for Numerical Methods in Fluids 84 (8) (2017) 477–493. `doi:10.1002/fld.4358`.

[63] E. Ferrer, D. Moxey, R. H. J. Willden, S. J. Sherwin, Stability of projection methods for incompressible flows using high order pressure-velocity pairs of same degree: Continuous and discontinuous Galerkin formulations, Commun. Comput. Phys. 16 (2014) 817–840. `doi:10.4208/cicp.290114.170414a`.

[64] E. Leriche, E. Perchat, G. Labrosse, M. O. Deville, Numerical evaluation of the accuracy and stability properties of high-order direct Stokes solvers with or without temporal splitting, J. Sci. Comput. 26 (1) (2006) 25–43. `doi:10.1007/s10915-004-4798-0`.

[65] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM Journal on Numerical Analysis 39 (5) (2002) 1749–1779.

[66] K. Hillewaert, Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries, Ph.D. thesis, Univ. de Louvain (2013).

[67] P. Geuzaine, C. Grandmont, C. Farhat, Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations, Journal of Computational Physics 191 (1) (2003) 206 – 227. `doi:10.1016/S0021-9991(03)00311-5`.

[68] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, D. Wells, The `deal.II` library, version 9.0, J. Numer. Math. 26 (4) (2018) 173–184. `doi:10.1515/jnma-2018-0054`.
URL `www.dealii.org`

[69] G. E. Karniadakis, S. J. Sherwin, Spectral/hp element methods for computational fluid dynamics, 2nd Edition, Oxford University Press, 2005.

[70] G. Taylor, A. Green, Mechanism of the production of small eddies from large ones, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 158 (895) (1937) 499–521.