


ASC Report No. 27/2015

# High order exactly divergence-free Hybrid Discontinuous Galerkin Methods for unsteady incompressible flows

C. Lehrenfeld, J. Schöberl



Institute for Analysis and Scientific Computing —  
Vienna University of Technology — TU Wien  
[www.asc.tuwien.ac.at](http://www.asc.tuwien.ac.at) ISBN 978-3-902627-05-6

## Most recent ASC Reports

- 26/2015 *A. Jüngel, and S. Schuchnigg*  
Entropy-dissipating semi-discrete Runge-Kutta schemes for nonlinear diffusion equations
- 25/2015 *W. Auzinger, H. Hofstätter, D. Ketcheson, O. Koch*  
Practical splitting methods for the adaptive integration of nonlinear evolution equations.  
Part I: Construction of optimized schemes and pairs of schemes
- 24/2015 *J. Burkotová, M. Hubner, I. Rachunková, and E.B. Weinmüller*  
Asymptotic properties of Kneser solutions to nonlinear second order ODEs with regularly varying coefficients
- 23/2015 *C. Abert, G. Hrkac, D. Praetorius, M. Ruggeri, D. Suess*  
Coupling of dynamical micromagnetism and a stationary spin drift-diffusion equation: A step towards a fully implicit spintronics frame
- 22/2015 *F. Achleitner, A. Arnold and D. Stürzer*  
Large-time behavior in non-symmetric Fokker-Planck equations
- 21/2015 *J. Burkotová, I. Rachunková, and E.B. Weinmüller*  
On singular BVPs with unsmooth data.  
Part 2: Convergence of the collocation schemes
- 20/2015 *J. Burkotová, I. Rachunková, and E.B. Weinmüller*  
On singular BVPs with unsmooth data.  
Part 1: Analysis of the linear case with variable coefficient matrix
- 19/2015 *M. Miletic, D. Stürzer, A. Arnold, A. Kugi*  
Stability of an Euler-Bernoulli beam with a nonlinear dynamic feedback system
- 18/2015 *M. Langer, H. Woracek*  
Direct and inverse spectral theorems for a class of canonical systems with two singular endpoints
- 17/2015 *M. Halla*  
A new proof of convergence for radial perfectly matched layer discretizations of Helmholtz scattering and resonance problems

Institute for Analysis and Scientific Computing  
Vienna University of Technology  
Wiedner Hauptstraße 8–10  
1040 Wien, Austria

**E-Mail:** [admin@asc.tuwien.ac.at](mailto:admin@asc.tuwien.ac.at)  
**WWW:** <http://www.asc.tuwien.ac.at>  
**FAX:** +43-1-58801-10196

ISBN 978-3-902627-05-6

© Alle Rechte vorbehalten. Nachdruck nur mit Genehmigung des Autors.



# High order exactly divergence-free Hybrid Discontinuous Galerkin Methods for unsteady incompressible flows

Christoph Lehrenfeld<sup>1\*</sup> and Joachim Schöberl<sup>1</sup>

<sup>1</sup> *Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstrasse 8-10, 1040 Wien, Austria*

## SUMMARY

In this paper we present an efficient discretization method for the solution of the unsteady incompressible Navier-Stokes equations based on a high order (Hybrid) Discontinuous Galerkin formulation. The crucial component for the efficiency of the discretization method is the distinction between stiff linear parts and less stiff non-linear parts with respect to their *temporal and spatial* treatment.

Exploiting the flexibility of *operator-splitting time integration schemes* we combine two spatial discretizations which are tailored for two simpler sub-problems: a corresponding hyperbolic transport problem and an unsteady Stokes problem.

For the hyperbolic transport problem a spatial discretization with an Upwind Discontinuous Galerkin method and an explicit treatment in the time integration scheme is rather natural and allows for an efficient implementation. The treatment of the Stokes part involves the solution of linear systems. In this case a discretization with Hybrid Discontinuous Galerkin methods is better suited. We consider such a discretization for the Stokes part with two important features:  $H(\text{div})$ -conforming finite elements to guarantee exactly divergence-free velocity solutions and a projection operator which reduces the number of globally coupled unknowns. We present the method, discuss implementational aspects and demonstrate the performance on two and three dimensional benchmark problems. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

**KEY WORDS:** Navier-Stokes equations, Hybrid Discontinuous Galerkin Methods,  $H(\text{div})$ -conforming Finite Elements, exactly divergence-free, Operator-Splitting, reduced stabilization

## 1. INTRODUCTION

### 1.1. Problem statement

We consider the numerical solution of the unsteady incompressible Navier Stokes equations in a velocity-pressure formulation:

$$\begin{cases} \frac{\partial}{\partial t} u + \text{div}(-\nu \nabla u + u \otimes u + pI) = f & \text{in } \Omega \\ \text{div } u = 0 & \text{in } \Omega \end{cases} \quad (1)$$

with boundary conditions  $u = u_D$  on  $\Gamma_D \subset \partial\Omega$  and  $(\nu \nabla u - pI) \cdot n = 0$  on  $\Gamma_{out} = \partial\Omega \setminus \Gamma_D$ . Here,  $\nu = \text{const}$  is the kinematic viscosity,  $u$  the velocity,  $p$  the pressure, and  $f$  is an external body force. We consider a discretization to (1) with high order (Hybrid) Discontinuous Galerkin (DG) finite element methods for complex geometries with underlying meshes consisting of possibly curved tetrahedra, hexahedra, prisms and pyramids.

---

\*Correspondence to: Christoph Lehrenfeld, Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstrasse 8-10, 1040 Wien, Austria. E-mail: christoph.lehrenfeld@gmail.com

## 1.2. Literature

Since its introduction in the paper of Reed and Hill [1], DG methods have been developed and used for hyperbolic problems [2, 3, 4, 5, 6] and later extended to second-order elliptic (and parabolic) problems [7, 8, 9]. DG methods are specifically popular for flow problems [10, 11]. For incompressible flows different finite element methods have been discussed in the literature [12, 13, 14]. Among these methods several DG formulations have been considered, e.g. [15, 16, 17, 18, 19].

DG methods provide a flexibility which can be utilized for different purposes. In our case, the motivation to consider this type of discretizations is twofold. First, relevant flow problems that can be modeled with the incompressible Navier-Stokes equations are often convection dominated. Using the Upwind mechanism DG methods offer a natural way to devise stable discretizations of (dominating) convection. Secondly, abandoning  $H^1$ -conforming finite element spaces, DG methods allow to consider (only)  $H(\text{div})$ -conforming finite elements (see [20]) for (Navier-)Stokes problems. This is attractive as it facilitates the design of a discretization with exactly solenoidal solution which in turn implies energy-stability of the discretization for the Navier-Stokes problem, cf. [18, 19, 21].

Compared to Continuous Galerkin (CG) methods the number of degrees of freedom of Discontinuous Galerkin methods increases significantly. This drawback is often outweighed by the advantages of the method. However, when it comes to solving linear systems Discontinuous Galerkin methods suffer most from drastically increased globally coupled degrees of freedom. An approach to compensate for this is the concept of Hybridization where additional unknowns on element interfaces are introduced. This increases the number of degrees of freedom, but introduces two advantages. First, the global couplings are reduced and secondly, the structure of the couplings allows to apply static condensation for the element unknowns. The concept has originally been introduced in the context of mixed finite element methods, cf. [20]. In the last decade Hybridization has also been applied to Discontinuous Galerkin method for a variety of problems, cf. [22, 23, 24, 25, 26]. We also mention, that recently similar concepts are also known in the literature under the name “Hybridized Weak Galerkin” methods [27] where a slightly different framework is used to derive the methods. With the emphasis to treat general polyhedral meshes “Hybrid High-Order” (HHO) methods [28] have recently been introduced where a combination of Hybrid (or hybridized) methods and higher order spaces is considered.

Concerning HDG discretizations for incompressible flows, we also mention the papers [29, 25, 30]. Further, different approaches to implement exact incompressible finite element solutions to the Stokes problem using Hybridization have been investigated in [31, 32, 33]. In [34] the Stokes-Brinkman problem, the Stokes problem with an additional zero order term, has been discretized using a hybridized  $H(\text{div})$ -conforming finite element formulation.

An interesting and often praised aspect of Hybrid mixed finite element methods is the fact that a post-processing step can be used to reconstruct interior unknowns of an increased higher order accuracy for elliptic problems [20]. The same is also possible for HDG methods in mixed formulation which is another advantage over conventional DG methods, see e.g. [24]. For elliptic problems an accuracy of order  $k + 2$  in the volume can be achieved if polynomials of order  $k$  are used on the element interfaces. One main contribution in this paper is the introduction of a projection operator which achieves the same effect. In [21] we already discussed this operator which has recently also been addressed under the name “reduced stabilization” in [35, 36, 37, 38]. In these papers the construction of the corresponding operator is only possible in two dimensions using special integration rules. We explain how the operator can be implemented in a more general setting.

The efficiency and practicability of HDG methods is rarely addressed in the literature. Interesting exception are [39, 40] where the computational cost of the method is compared to CG and DG methods.

The nature of diffusive (viscous) and convective terms appearing in convection-diffusion-type problems have a substantially different character. Discretizations of problems involving only one of the two mechanisms would typically lead to different *spatial and temporal* discretizations. It is therefore often desirable to consider operator-splitting time discretization schemes which allow for a

separate treatment of the different operators as in [41, 42, 43, 44, 45, 46]. In these papers the spatial discretization for the stiff (diffusion/Stokes) and the non-stiff (convection) operators is typically the same. We consider a different treatment of the operators with respect to their *temporal and spatial* discretization.

### 1.3. The concept: Efficiency through operator splitting

A crucial ingredient in the considered discretization is the fact that the convection and the Stokes problem are separated by means of an operator-splitting method. The operator-splitting method is chosen such that only operator evaluations of the convection operator are required so that the time integration scheme is explicit in terms of the convection operator. This is often affordable as the time step restrictions following from the convection operator are the least restrictive ones in the Navier-Stokes problem. Moreover, the convection operator is non-linear, s.t. the set up of linear systems of equations and corresponding preconditioners or solvers would have to take place every time step which renders implicit approaches for the convection very expensive. The Stokes operator is dealt with implicitly, i.e. it appears in linear systems in every time step. Due to the differential-algebraic structure of the Navier-Stokes equations this is necessary w.r.t. the pressure and the incompressibility constraint. As completely explicit handling of the viscosity terms would introduce severe time step restrictions it is also advisable to treat viscous forces implicit. Note that the Stokes operator is time-independent such that the setup of linear systems and preconditioners or solvers can be done once and re-used in every time step.

The spatial discretizations for the different operators are designed differently as the treatment of the convection term can be optimized for operator evaluations while the discretization of the Stokes operator has to provide an efficient handling of implicit solution steps, i.e. the solution of corresponding linear systems. This different treatment reflects in the use of two different finite element spaces which are used: An  $H(\text{div})$ -conforming Hybrid DG space for the velocity-pressure-pair for solving Stokes problems and a DG space for handling convection.

### 1.4. Main contributions and structure of this paper

In this paper we introduce a new discretization method for the Navier-Stokes equations. The discretization is based on a decomposition of the problem into the (unsteady) Stokes problem and a hyperbolic transport problem. The decomposition is exploited in the spatial as well as in temporal discretization. The main contributions of this paper are:

1. Introduction of a new  $H(\text{div})$ -conforming high order accurate HDG method with a *projected jumps* formulation (also known as *reduced stabilization* or *reduced-order* HDG) for the solution of Stokes-type problems.
2. Presentation of a combined spatial discretization for the Navier-Stokes equations based on a standard Upwind DG formulation for the hyperbolic transport problem with the new HDG method for Stokes-Brinkman problems.
3. Discussion of operator-splitting time integration schemes which restrict solutions of linear systems to Stokes-Brinkman problems.

In section 2 we discuss the discretization of spatial operators. The discussion is divided into two parts, the discretization of the Stokes-Brinkman problem, the problem which involves all relevant spatial operators except for the convection and the discretization of the convection operator. For the former part we consider an HDG discretization, for the latter part a standard DG discretization. In section 3 operator-splitting time integration schemes are discussed which are tailored for such a situation. Finally, in section 4 we give numerical examples which demonstrate the performance of the method.

### 1.5. Preliminaries

Before describing the methods, we introduce some basic notation and assumptions:  $\Omega$  is an open bounded domain in  $\mathbb{R}^d$  with a Lipschitz boundary  $\Gamma$ . It is decomposed into a shape regular partition

$\mathcal{T}_h$  of  $\Omega$  consisting of elements  $T$  which are (curved) simplices, quadrilaterals, hexahedrons, prisms or pyramids. For ease of presentation we assume that all elements are not curved and consider only homogeneous Dirichlet boundary conditions. The element interfaces and element boundaries coinciding with the domain boundary are called *facets*. The set of those facets  $F$  is denoted by  $\mathcal{F}_h$  and there holds  $\bigcup_{T \in \mathcal{T}_h} \partial T = \bigcup_{F \in \mathcal{F}_h} F$ .

In the sequel we distinguish functions with support only on facets indicated by a subscript  $F$  and those with support also on the volume elements which is indicated by a subscript  $T$ . Compositions of both types are used for the HDG discretization of the velocity which is denoted by  $u = (u_T, u_F)$ .

For vector-valued functions the superscripts  $t$  denotes the application of the tangential projection:  $v^t = v - (v \cdot n) \cdot n \in \mathbb{R}^d$ . The index  $k$  which describes the polynomial degree of the finite element approximation at many places through out the paper is an arbitrary but fixed positive integer number.

We identify finite element functions  $u \in X_h$  with their representation in terms of coefficient vectors  $\mathbf{u} \in \mathbb{R}^{N_X}$ , s.t.  $u = \sum_{i=1}^{N_X} \mathbf{u}_i \varphi_i$  for a corresponding basis  $\{\varphi_i\}$  of  $X_h$  and  $N_X = \dim(X_h)$ . A (generic) bilinear form  $\mathcal{G}_h : X_h \times Y_h \rightarrow \mathbb{R}$  is identified with the matrix  $\mathbf{G} \in \mathbb{R}^{N_Y \times N_X}$ , s.t.  $\mathbf{G}_{i,j} = \mathcal{G}_h(\varphi_j^X, \varphi_i^Y)$ .

## 2. DG/HDG SPATIAL DISCRETIZATION

In this section we introduce the spatial discretization for the Stokes operator, the convection operator and transfer operations between both. First, we introduce the  $H(\text{div})$ -conforming Hybrid DG discretization of the Stokes-Brinkman problem, i.e. the stationary Navier-Stokes problem without convection and an additional zero order term in section 2.1. This discretization is improved significantly. Further on, a modification of the discretization using the idea of *projected jumps* (also known as *reduced stabilization* or *reduced-order HDG*) is presented in section 2.2 including an explanation of how the projection operator can be realized. For the convection part of the Navier-Stokes problem we consider a DG discretization using standard approaches. This is discussed in section 2.3. As the discretization spaces for the Stokes part and the convection part are different, we present transfer operations between the spaces in section 2.4 which allow us to finally formulate the semi-discrete problem in section 2.5.

### 2.1. $H(\text{div})$ -conforming HDG formulation of the Stokes-Brinkman problem

In this part we consider the discretization of the Stokes part of the Navier-Stokes problem. For simplicity we restrict the discussion to homogeneous Dirichlet boundary conditions. We present the method and elaborate on important properties. For an error analysis of the method we refer to [21]. The reaction term  $\tau^{-1}$  corresponding to an inertia term stemming from an implicit time integration scheme is further incorporated. The resulting problem is known as the (stationary) Stokes-Brinkman problem:

$$\begin{cases} \tau^{-1}u + \text{div}(-\nu \nabla u + pI) &= f & \text{in } \Omega \\ \text{div } u &= 0 & \text{in } \Omega \end{cases} \quad (2)$$

We first introduce the finite element spaces, followed by the definition of the bilinear forms corresponding to the involved operators.

**2.1.1.  $H(\text{div})$ -conforming Finite Elements for Stokes.** Following [18] a DG formulation for the incompressible Navier Stokes equations which is locally conservative and energy-stable at the same time has to provide discrete solutions which are exactly divergence-free. This can be achieved with a suitable pair of finite element spaces. We consider the use of  $H(\text{div})$ -conforming Finite Element spaces for the velocity  $u$ .  $H(\text{div})$ -conformity requires that every discrete function  $u_h$  is in

$$H(\text{div}, \Omega) = \{u \in [L_2(\Omega)]^d : \text{div } u \in L_2(\Omega)\}$$

We use piecewise polynomials, so that on each element the functions are in  $H(\text{div}, T)$ . For global conformity, continuity of the normal component is necessary, resulting in

$$W_h := \{u_T \in \prod_{T \in \mathcal{T}_h} [\mathcal{P}^k(T)]^d, \llbracket u_T \cdot n \rrbracket_F = 0 \forall F \in \mathcal{F}_h\} \subset H(\text{div}, \Omega), \quad N_W := \dim(W_h), \quad (3)$$

with  $\llbracket \cdot \rrbracket_F$  the usual jump operator and  $\mathcal{P}^k$  the space of polynomials up to degree  $k$ . We refer to [20] for details on the construction of  $H(\text{div})$ -conforming Finite Elements.

The appropriate Finite Element space for the pressure is the space of piecewise polynomials which are discontinuous and of one degree less:

$$Q_h := \prod_{T \in \mathcal{T}_h} \mathcal{P}^{k-1}(T), \quad N_Q := \dim(Q_h). \quad (4)$$

This velocity-pressure pair  $W_h/Q_h$  fulfills

$$\text{div}(W_h) = Q_h. \quad (5)$$

The crucial point of this choice of the velocity-pressure pair is the property:

If a velocity  $u_T \in W_h$  is weakly incompressible, it is also strongly incompressible:

$$\int_{\Omega} \text{div}(u_T) q \, dx = 0 \forall q_h \in Q_h \quad \Leftrightarrow \quad \text{div}(u_T) = 0 \text{ in } \Omega. \quad (6)$$

The benefit of (6) is twofold: First, it allows to show energy-stability for the incompressible Navier-Stokes equations, cf. section 2.5. Secondly, error estimates for the velocity error can be derived which are independent of the pressure field, we comment on this in remark 5 below.

As solutions of the incompressible Navier Stokes equations are  $[H^1(\Omega)]^d \times L_2(\Omega)$ -regular and tangential continuity is not imposed as an essential condition on the finite element space the discrete formulation has to incorporate the tangential continuity weakly. We do this with a corresponding (Hybrid) DG formulation for the tangential components across element interfaces

*Remark 1* (Reduced spaces)

Due to (6) solutions of the discretized (Navier)-Stokes problem will be exactly divergence-free velocity fields. This a priori knowledge can be exploited. The basis for the space  $W_h$  can be constructed in such a way such that we can discard certain higher order basis functions (with non-zero divergence) that will have no contribution. A corresponding basis is introduced in [47] in the context of a set of higher order basis functions which fulfill an exact sequence property. The resulting space  $W_h^{\text{red}}$  has  $\text{div}(W_h^{\text{red}}) = Q_h^{\text{red}} := \{p|_T = \text{const} : T \in \mathcal{T}_h\}$  such that also most of the degrees of freedom of the pressure space can be discarded. The reduction of basis functions for  $W_h$  and  $Q_h$  is explained in more detail in [21, Chapter 2.2].

**2.1.2. The HDG space for the velocity.** To (weakly) enforce continuity we apply a discontinuous Galerkin (DG) formulation such as the Interior Penalty method [48]. However, to avoid the full coupling of degrees of freedom of neighboring elements, we introduce additional unknowns on the skeleton, the *facet unknowns*, which represent an approximation of the tangential trace of the solution. The DG formulation is then replaced with a corresponding HDG formulation, s.t. degrees of freedom of neighboring elements couple only through the facet unknowns.

As normal continuity is already implemented in  $W_h$  we only need a DG enforcement of continuity in the tangential direction and hence only introduce the facet unknown for the tangential direction of the trace:

$$F_h := \{u_F \in \prod_{F \in \mathcal{F}_h} [\mathcal{P}^k(F)]^d, u_F \cdot n = 0\}, \quad N_F := \dim(F_h). \quad (7)$$

Functions in  $F_h$  have normal component zero. For the discretization of the velocity field we use the composite space

$$U_h := W_h \times F_h, \quad N_U := \dim(U_h). \quad (8)$$

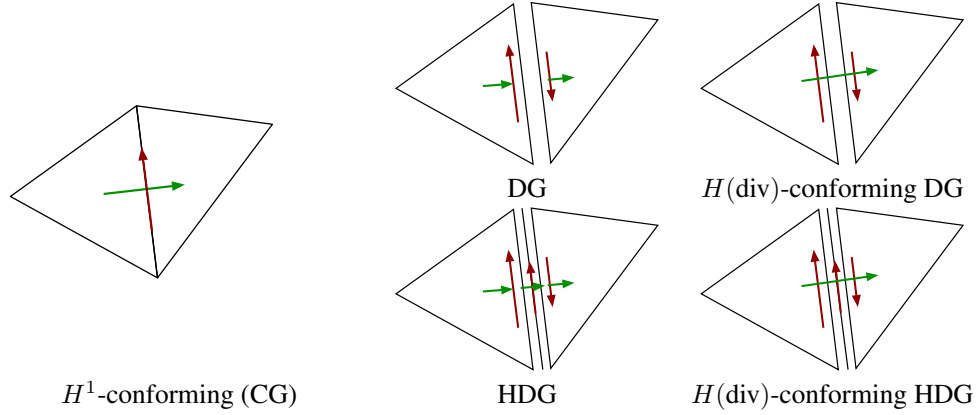


Figure 1. Tangential and normal continuity for different finite element methods

*Remark 2* (The role of the facet space)

We note that the space  $F_h$  is only introduced to allow for a more *efficient handling of linear systems*. In section 4.1 we consider a numerical example where a vector-valued Poisson problem is considered for the HDG space  $U_h$  in comparison to other ( $H(\text{div})$ -conforming) DG spaces to illustrate the impact of Hybridization. In case that only explicit applications of discrete operators are used, the introduction of the space  $F_h$  entails no advantages. In this paper, facet variables appear only in the discretization of the viscous forces. Although the facet variable has no contribution in the discretization of inertia and pressure forces or the incompressibility constraint we define the corresponding operations for  $u \in U_h$  instead of  $u_T \in W_h$  to simplify the presentation.

**2.1.3. Viscous forces.** For ease of presentation we consider a hybridized version of the Interior Penalty method for the discretization of the viscous term. In remark 3 we comment on alternatives. In the hybridized version of the Interior Penalty method, the usual jump across element interfaces of the Interior Penalty method is replaced with jumps between element interior and facet unknown (in tangential direction)  $\llbracket u^t \rrbracket = u_T^t - u_F$  from both sides of a facet. The bilinear form corresponding to the HDG discretization of viscous forces is

$$\begin{aligned} \mathcal{A}_h(u, v) := & \sum_{T \in \mathcal{T}_h} \int_T \nu \nabla u_T : \nabla v_T \, dx - \int_{\partial T} \nu \frac{\partial u_T}{\partial n} \llbracket v^t \rrbracket \, ds \\ & - \int_{\partial T} \nu \frac{\partial v_T}{\partial n} \llbracket u^t \rrbracket \, ds + \int_{\partial T} \nu \frac{\alpha}{h} \llbracket u^t \rrbracket \llbracket v^t \rrbracket \, ds, \quad u = (u_T, u_F), v = (v_T, v_F) \in U_h \end{aligned} \quad (9)$$

In this bilinear form the four terms have different functions. While the first two terms ensure consistency (in the sense of Galerkin orthogonality) the third and fourth term are tailored to ensure symmetry (adjoint consistency) and stability, respectively. Due to continuity of the solution ( $\llbracket \cdot^t \rrbracket = 0$ ) the latter terms also preserve consistency. For a more detailed introduction we refer to [21, section 2.3.1]. With respect to the discrete norm

$$\|u\|_1^2 := \sum_{T \in \mathcal{T}_h} \left\{ \|\nabla u_T\|_T^2 + \frac{1}{h} \|\llbracket u^t \rrbracket\|_{\partial T}^2 + h \left\| \frac{\partial u_T}{\partial n} \right\|_{\partial T}^2 \right\}$$

which is an appropriately modified version of the discrete norm typically used in the analysis of Standard Interior Penalty methods, the bilinearform  $\mathcal{A}_h$  is (for a sufficiently large  $\alpha$ ) consistent, bounded and coercive.

The coupling through facet unknowns enforces the same kind of continuity as in the Interior Penalty DG formulation while preserving the following structure in the sparsity pattern: element unknowns are only coupled with unknowns associated with the same element or unknowns associated with aligned facets.



$k$	4	8	16	32
quadrilateral	0.305	0.313	0.315	0.315
triangle	0.167	0.190	0.201	0.205

Table I. LBB constant in dependence of  $k$  for a single element.

*Remark 3* (Interior Penalty and alternatives)

A drawback of the (Hybrid) Interior Penalty method is the fact that the stabilization parameter  $\alpha$  depends on the shape regularity of the mesh. Often  $\alpha$  is chosen on the safe side, but as was pointed out in [49] the condition number of arising linear systems increases with  $\alpha$ . In the same paper a hybridized variant of the Bassi-Rebay stabilization method (cf. [4, 5, 50]) for a scalar Poisson equation has been proposed, see also remark 11. Such a variant is used in the numerical examples, but as it does not have any further consequences for the remainder of this work, we stick to the well-known (Hybrid) Interior Penalty method for ease of presentation.

*2.1.4. Mass bilinear form.* The HDG mass matrix is defined as

$$\mathcal{M}_h^U(u, v) := \int_{\Omega} u_T v_T \, dx, \quad u = (u_T, u_F), v = (v_T, v_F) \in U_h. \quad (10)$$

Note that we defined the mass matrix for  $u \in U_h$  although  $u_F$  has no contribution, cf. remark 2.

*2.1.5. Pressure force and incompressibility constraint.* For the pressure force and the incompressibility constraint we define the bilinearform

$$\mathcal{D}_h(u, p) := \sum_{T \in \mathcal{T}_h} - \int_T p \operatorname{div} u_T \, dx \quad \text{for } u = (u_T, u_F) \in U_h, p \in Q_h \quad (11)$$

for which the *LBB-condition*

$$\sup_{u \in U_h} \frac{\mathcal{D}_h(u, p)}{\|u\|_1} \geq c_{LBB} \|p\|_{L^2}, \quad \forall p \in Q_h \quad (12)$$

holds true for a  $c_{LBB}$  independent of the mesh size  $h$ , cf. [21, Proposition 2.3.5].

*Remark 4* (Robustness in polynomial degree  $k$ )

In numerical experiments we observed that the LBB constant  $c_{LBB}$  in (12) is also robust in  $k$ . To this end we computed the LBB constant of one element with Dirichlet boundary conditions and the condition  $\int_{\Omega} p \, dx = 0$  on the pressure. The results are shown in Table I for varying the polynomial degree  $k$ . From the boundedness of the LBB constant on one element and the  $h$ -robustness in (12), the robustness in  $h$  and  $k$  on the global spaces follows. This will be in the forthcoming master's thesis of Philip Lederer.

*2.1.6. Discretization of the Brinkman-Stokes problem* With the introduced discretizations of the spatial operators the discrete Brinkman-Stokes problem can be written as:

Find  $u \in U_h$  and  $p \in Q_h$ , s.t.

$$\begin{cases} \tau^{-1} \mathcal{M}_h(u, v) + \mathcal{A}_h(u, v) + \mathcal{D}_h(v, p) &= \langle f, v \rangle \quad \forall v \in U_h, \\ \mathcal{D}_h(u, q) &= 0 \quad \forall q \in Q_h. \end{cases} \quad (13)$$

Due to coercivity of  $\mathcal{A}_h$  ( respectively  $\tau^{-1} \mathcal{M}_h + \mathcal{A}_h$  ), the LBB-condition of  $\mathcal{D}_h$  and consistency and continuity of all bilinear forms Brezzi's famous theorem (see [20]) can be applied to obtain optimal order a priori error estimates.

*Remark 5* (Pressure-independence of velocity error)

Classical error estimates for mixed problems result in error estimates which are formulated in the

compound norm of the velocity and pressure space. This has the disadvantage that the discretization error in the velocity depends on the approximation error in the pressure. As was pointed out in [51], ideally this should not be the case. Due to the fact that discrete solutions to (13) are exactly divergence-free an error estimate for the velocity field which does not involve the pressure can be derived, cf. [21, Lemma 2.3.13].

## 2.2. Projected jumps: An enhancement of the HDG Stokes discretization

The proposed HDG formulation for viscous forces in section 2.1.3 can also be derived as a hybrid mixed method with a modified flux. This is done in detail in [24] (see also [21, section 1.2.2] or [30]). In that setting the unknowns for the primal variable ( $u_T$ ) are approximated with the same polynomial degree  $k$  as the facet unknowns ( $u_F$ ). Afterwards, in a postprocessing step approximations for the primal unknown, of one degree higher,  $k + 1$ , are reconstructed in an element-by-element fashion. This approach ends up with a higher order approximation than the previously introduced HDG method considering the use of the same polynomial degree  $k$  for the facet unknowns. This sub-optimality can be overcome by means of a projection operator which leads to the *projected jumps* formulation. We first introduce the method and explain how the method can be implemented afterwards.

**2.2.1. Projected jumps: The method.** The idea of the *projected jumps* formulation is to reduce the polynomial degree of the facet unknowns in  $F_h$  to  $k - 1$ ,

$$F_h \rightarrow \bar{F}_h := \{u_F \in \prod_{F \in \mathcal{F}_h} [\mathcal{P}^{k-1}(F)]^d, u_F \cdot n = 0, \forall F \in \mathcal{F}_h\}, \quad (14)$$

while keeping the polynomial degree  $k$  in  $W_h$ . In order to do this in a consistent fashion we have to modify the bilinearform  $\mathcal{A}_h$ . First, we introduce the  $L_2$  projection  $\Pi$  for a fixed facet  $F \in \mathcal{F}_h$ :

$$\Pi : [\mathcal{P}^k(F)]^d \rightarrow [\mathcal{P}^{k-1}(F)]^d, \quad \int_F (\Pi f) v \, dx = \int_F f v \, dx \quad \forall v \in [\mathcal{P}^{k-1}(F)]^d \quad (15)$$

Due to the fact that  $\frac{\partial u_T}{\partial n} \in \mathcal{P}^{k-1}(F)$  on affine linearly mapped elements, the test functions  $v_T$  and  $v_F$  in the first integral on the element boundary in (9) can be replaced by their  $L_2(F)$  projections  $\Pi v_T$  and  $\Pi v_F$ . If we additionally use  $\Pi[u^t]$  instead of  $\llbracket u^t \rrbracket$  to symmetrize and stabilize the formulation in (9) we end up with a modified version of the Hybrid Interior Penalty method:

$$\begin{aligned} \mathcal{A}_h^r(u, v) := & \sum_{T \in \mathcal{T}_h} \int_T \nu \nabla u_T : \nabla v_T \, dx - \int_{\partial T} \nu \frac{\partial u_T}{\partial n} \Pi[v^t] \, ds \\ & - \int_{\partial T} \nu \frac{\partial v_T}{\partial n} \Pi[u^t] \, ds + \int_{\partial T} \nu \frac{\alpha}{h} \Pi[u^t] \Pi[v^t] \, ds, \quad u, v \in W_h \times \bar{F}_h \end{aligned} \quad (16)$$

Note that the first two boundary integrals are just reformulated while only the last integral is really changed. This modification preserves the important properties of  $\mathcal{A}_h$ : It is still consistent and bounded. Further, a deeper look into the coercivity proof (see [21]) reveals that coercivity can easily be shown in the modified (weaker) norm

$$\|u\|_{1,*}^2 := \sum_{T \in \mathcal{T}_h} \left\{ \|\nabla u_T\|_T^2 + \frac{1}{h} \|\Pi u^t\|_{\partial T}^2 + h \left\| \frac{\partial u_T}{\partial n} \right\|_{\partial T}^2 \right\}. \quad (17)$$

Notice that, as we do not modify  $W_h$ , the normal component of  $u_T \in W_h$  is still a polynomial of degree  $k$ .

The idea of applying such a *reduced stabilization* is also discussed and analyzed in [36, 37]. However, only for the two-dimensional case the realization of the projection  $\Pi$  is discussed (by means of Gauss quadrature, cf. [36, section 3.4]). In the next section a simple way to implement this projection operator is presented.

*Remark 6* (Interplay with operator-splitting)

If a Hybrid DG formulation for a problem involving diffusion (viscosity) and convection is applied, such a reduction of the facet unknowns is only appropriate if diffusion is dominating. One premise of the operator-splitting in this paper is that convection is not involved in implicit solution steps. Hence, the facet variable will never be involved in the discretization of convection, s.t. even if the physical problem is convection dominated, the *projected jumps* modification can be applied without loss of accuracy.

**2.2.2. Projected jumps: Realization.** The following way to implement *projected jumps* needed for  $\mathcal{A}_h^r$  in (16) relies on an  $L_2$ -orthogonal basis for the facet functions in  $F_h$ . To obtain an  $L_2$ -orthogonal basis, we take a local coordinate system on each facet (spanned by  $d - 1$  aligned edges) and an  $L^2$ -orthogonal Dubiner basis (see [52]) for each vector component. We consider the three dimensional case, and denote the vectors of the local coordinate system by  $e_1$  and  $e_2$ . Translation to the two dimensional case is then obvious.

On each facet  $F \in \mathcal{F}_h$ , the space of (vector-valued) polynomials up to degree  $k$ ,  $\text{span}(e_1, e_2) \cdot \mathcal{P}^k(F)$ , can be split into the orthogonal subspaces

$$\mathcal{V}_{k-1} := \text{span}\{e_1, e_2\} \cdot \mathcal{P}^{k-1}(F) \quad \text{and} \quad \mathcal{V}_{k-1}^\perp := \text{span}\{e_1, e_2\} \cdot [\mathcal{P}^k(F) \cap (\mathcal{P}^{k-1}(F))^\perp]. \quad (18)$$

On each facet  $F$  the facet unknown  $u_F$  can thus be written as  $u_F = \bar{u}_F + \lambda$  with unique  $\bar{u}_F \in \mathcal{V}_{k-1}$ ,  $\lambda \in \mathcal{V}_{k-1}^\perp$ . We now replace the highest order function  $\lambda$  with two copies  $\lambda_T$  and  $\lambda_{T'}$ , each of these functions is associated with one of the two neighbouring elements  $T$  and  $T'$ , the functions  $\lambda_T$  are only defined element-local.  $\lambda_T$  can thus be eliminated after the computation of the element matrix and finally, only  $u_T \in W_h$  and  $\bar{u}_F \in \bar{F}_h$  appear in the global system.

$\lambda_T$  is only responsible for implicitly realising the projection operator. To see this we now consider one facet  $F \in \mathcal{F}_h$  and one of the neighboring elements  $T \in \mathcal{T}_h$ . We use the decomposition corresponding to (18) for trial and test functions

$$u_F = \bar{u}_F + \lambda_T \quad \text{and} \quad v_F = \bar{v}_F + \mu_T, \quad (19)$$

with  $\bar{u}_F, \bar{v}_F \in \mathcal{V}_{k-1}$  and  $\lambda_T, \mu_T \in \mathcal{V}_{k-1}^\perp$  where  $\lambda_T, \mu_T$  are only supported on  $T$ . In (13) we choose the test function  $v$  such that  $v_T = \bar{v}_F = 0$ ,  $\mu_T \in \mathcal{V}_{k-1}^\perp$  on  $T$  and  $\mu_{T'} = 0$  on every other element  $T'$ . This yields

$$\int_F \nu \frac{\partial u_T}{\partial n} \mu_T \, ds + \int_F \nu \frac{\alpha}{h} (u_T^t - \bar{u}_F - \lambda_T) \mu_T \, ds = \int_F \nu \frac{\alpha}{h} (u_T^t - \lambda_T) \mu_T \, ds = 0, \quad \forall \mu_T \in \mathcal{V}_{k-1}^\perp. \quad (20)$$

Hence, there holds  $\lambda_T = (I - \Pi)u_T^t$  and we have

$$[u^t] = u_T^t - \bar{u}_F - \lambda_T = \Pi(u_T^t) - \bar{u}_F = \Pi(u_T^t - \bar{u}_F) = \Pi[u^t] \quad (21)$$

We conclude that it is sufficient to consider the HDG bilinear form  $\mathcal{A}_h$  as before, where the local element matrices are computed according to  $F_h$ . For each element matrix the degrees of freedom corresponding to  $\mathcal{V}_{k-1}^\perp$  are then eliminated forming a corresponding Schur complement. This yields a final stiffness matrix which is only set up with respect to space unknowns of  $W_h \times \bar{F}_h$ .

### 2.3. DG formulation for the convection

For the discretization of the convection part we consider a standard DG finite element space:

$$V_h := \{u : u \in [\mathcal{P}^k(T)]^d \, \forall T \in \mathcal{T}_h\}, \quad N_V := \dim(V_h). \quad (22)$$

We define the mass bilinear form

$$\mathcal{M}_h^V(w, z) := \int_\Omega w z \, dx, \quad w, z \in V_h. \quad (23)$$

Using an  $L^2$ -orthogonal basis on each element renders the associated mass matrix  $\mathbf{M}^V : \mathbb{R}^{N_V} \rightarrow \mathbb{R}^{N_V}$  diagonal. A stable spatial discretization of (1) with respect to the convection part is achieved using a Standard Upwind DG trilinearform. We assume that the convection velocity  $u_T$  is exactly divergence-free.

$$\mathcal{C}_h(u_T; w, z) := \sum_T - \int_T w \otimes u_T : \nabla z \, dx + \int_{\partial T} u_T \cdot n \, \hat{w} \, z \, ds, \quad u_T \in W_h, \quad w, z \in V_h. \quad (24)$$

where  $\hat{w}$  denotes the upwind value  $\hat{w} = \lim_{\varepsilon \searrow 0} w(x - \varepsilon u_T(x))$ . Standard Upwind DG formulations are stable in the sense that with  $\partial\Omega_{in} := \{x \in \partial\Omega, u_T(x) \cdot n < 0\}$  there holds

$$\frac{1}{2} \int_{\partial\Omega_{in}} |u_T \cdot n| w^2 \, ds + \mathcal{C}_h(u_T; w, w) \geq 0, \quad \forall w \in V_h, \quad u \in U_h, \quad \operatorname{div}(u_T) = 0. \quad (25)$$

#### 2.4. Transfer operations and embeddings

The discrete convection and Stokes operators have been defined on different spaces. In order to combine both we introduce transfer operations between the (finite element) spaces to make both discretizations compatible. We restrict ourselves to two types of transfer operations which are based on embeddings:

- I:** With  $W_h \subset V_h$  we have a canonical embedding of  $U_h$  in  $V_h$  with the embedding operator  $\mathcal{I} : (u_T, u_F) \in U_h \rightarrow u_T \in V_h$ . Note that the corresponding operation in terms of coefficient vectors, denoted by  $\mathbf{I} : \mathbb{R}^{N_U} \rightarrow \mathbb{R}^{N_V}$ , is not an identity.
- $\mathbf{I}^T$ :** The embedding operator  $\mathcal{I}$  implies the canonical embedding  $\mathcal{I}' : V'_h \rightarrow U'_h$ , which maps functionals  $\mathcal{I}' : f \in V'_h \rightarrow [(u_T, u_F) \in U_h \rightarrow f(u_T)] \in U'_h$ . The corresponding matrix representation is  $\mathbf{I}^T : \mathbb{R}^{N_V} \rightarrow \mathbb{R}^{N_U}$ .

To realize the operator  $\mathbf{I}$  we consider the equivalent  $L^2$  problem for  $u \in U_h$ .

$$\int_{\Omega} (\mathcal{I}u) v \, dx = \int_{\Omega} u_T v \, dx \quad \forall v \in V_h. \quad (26)$$

In terms of coefficient vectors this reads as

$$\mathbf{M}^V(\mathbf{I}\mathbf{u}) = \mathbf{M}^{UV}\mathbf{u}, \quad \forall \mathbf{u} \in \mathbb{R}^{N_U} \quad \implies \quad \mathbf{I} = (\mathbf{M}^V)^{-1}\mathbf{M}^{UV} \quad (27)$$

with the mixed mass matrix

$$\mathbf{M}_{i,j}^{UV} = \int_{\Omega} \varphi_j^W \varphi_i^V \, dx, \quad i = 1, \dots, N_V, j = 1, \dots, N_W \quad \text{and} \quad \mathbf{M}_{i,j}^{UV} = 0, j > N_W.$$

Note that  $\mathbf{M}^V$  is diagonal (for affine linear transformations) such that  $(\mathbf{M}^V)^{-1}$  can be evaluated very efficiently. The overall cost of the transfer operator  $\mathbf{I}$  is essentially that of one sparse matrix multiplication.

Let  $\mathbf{C}^V : \mathbb{R}^{N_U} \times \mathbb{R}^{N_V} \rightarrow \mathbb{R}^{N_V}$  denote the discrete convection operator corresponding to the trilinearform  $\mathcal{C}_h$  in (25).

With the operator  $\mathbf{I}$  we can formulate applications of the convection and the mass operations  $\mathbf{C}^V(\mathbf{u}), \mathbf{M}^V : \mathbb{R}^{N_V} \rightarrow \mathbb{R}^{N_V}$  with respect to functions in the HDG space  $U_h$  and denote the corresponding operators by  $\mathbf{C}^U(\mathbf{u}), \mathbf{M}^U : \mathbb{R}^{N_U} \rightarrow \mathbb{R}^{N_U}$ ,

$$\mathbf{C}^U(\mathbf{u}) := \mathbf{I}^T \mathbf{C}^V(\mathbf{I}\mathbf{u}), \quad \mathbf{M}^U := \mathbf{I}^T \mathbf{M}^V \mathbf{I}. \quad (28)$$

*Remark 7* (Restriction on time integration scheme)

Note that the restriction to these two transfer operations implies that we do not allow to apply any part of the Stokes operator to a function in  $V_h$  and that no functional on  $U_h$  can be used in solution steps involving the convection operator. This is a restriction on the time integration scheme.

*Remark 8 (Curved elements)*

If curved elements are considered we no longer have  $W_h \subset V_h$  due to the Piola transform usually applied to construct  $H(\text{div})$ -conforming finite elements. In this case  $\mathcal{I} : U_h \rightarrow V_h$  is not an embedding, but the  $L^2$  projection. Nevertheless,  $\mathbf{M}^V$  is block diagonal with blocks which are not diagonal matrices only on curved elements. Hence, applications of  $(\mathbf{M}^V)^{-1}$  are still cheap.

*2.5. The semidiscrete formulation*

With the definitions of the bilinear forms in the previous sections we arrive at the following spatially discrete DAE problem: Find  $u(t) \in U_h$  and  $p(t) \in Q_h$ , such that

$$\begin{cases} \mathcal{M}_h^U(\frac{\partial}{\partial t}u, v) + \mathcal{A}_h(u, v) + \mathcal{C}_h(u_T; u, v) + \mathcal{D}_h(v, p) = \langle f, v \rangle & \forall v \in U_h, \quad t \in [0, T], \\ \mathcal{D}_h(u, q) = 0 & \forall q \in Q_h, \quad t \in [0, T], \\ \mathcal{M}_h(u, v) = \mathcal{M}_h(u_0, v) & \forall v \in U_h, \quad t = 0. \end{cases} \quad (29)$$

Here, we implicitly used the embedding  $\mathcal{I}$  to define  $\mathcal{C}_h(u_T; \cdot, \cdot)$  on  $U_h$ . Due to (25) we have the stability of the kinetic energy:

$$\frac{d}{dt} \|u\|_{L_2} \leq \frac{1}{2} \|f(t)\|_{L_2} \quad (30)$$

In the next section we discuss operator splitting time integration methods to solve (29) efficiently.

**3. OPERATOR-SPLITTING TIME INTEGRATION**

In this section we are faced with the problem of solving the semi-discrete Navier-Stokes problem with a proper time integration scheme. For ease of presentation we neglect external forces ( $\mathbf{f} = 0$ ) in the following and consider the problem in terms of discrete operators corresponding to the bilinear (trilinear) forms introduced in the previous section: Find  $\mathbf{u}(t) \in \mathbb{R}^{N_U}$  and  $\mathbf{p}(t) \in \mathbb{R}^{N_Q}$ , such that

$$\begin{cases} \mathbf{M}^U \frac{\partial \mathbf{u}}{\partial t} + \mathbf{A} \mathbf{u} + \mathbf{C}^U(\mathbf{u}) \mathbf{u} \Delta t \mathbf{D} \mathbf{p} = \mathbf{f} & \text{in } [0, T], \\ \mathbf{D}^T \mathbf{u} = 0 & \text{in } [0, T], \\ \mathbf{u}(t=0) = \mathbf{u}_0. \end{cases} \quad (31)$$

In the time integration scheme we want to explicitly exploit the properties of the spatial discretization, i.e. the convection operator  $\mathbf{C}$  should only be involved explicitly in terms of operator evaluation. Due to the DAE-structure we require time integration schemes which are *stiffly accurate*. Hence, the solution of a Stokes-Brinkman problem should conclude every time step so that the incompressibility constraint is ensured. For operator splittings of this kind different approaches exist. We briefly discuss three approaches. In section 3.1 we discuss *additive* decomposition methods, like the famous class of IMplicit EXplicit (IMEX) schemes. *Product* decomposition methods, sometimes also called *exponential factor* splittings, are discussed in section 3.2 in the framework of operator-integration-factor splittings introduced in [45]. We discuss the advantages and disadvantages of both approaches and motivate the consideration of a different approach. An operator-splitting modification of the famous fractional step method, cf. [53], which eliminates the most important disadvantages of the additive and multiplicative decomposition methods is then introduced and discussed in section 3.3. We want to stress that the considered operator splitting approaches are of *convection-diffusion* type and should not be confused with projection methods like the Chorin splitting [54].

*3.1. Additive decomposition using IMEX schemes*

Additive decomposition methods distinguish spatial operators that are treated implicitly and those that are treated explicitly. The decomposition is *additive* in the sense that every solution (sub-)step involves both operators. This is used by IMplicit EXplicit (IMEX) schemes (see [41, 42, 43]). The simplest of these schemes is the *semi-implicit Euler* method for which one time step of size  $\Delta t$

reads as:

$$\begin{cases} (\mathbf{M}^U + \Delta t \mathbf{A}) \mathbf{u}^{n+1} + \Delta t \mathbf{D} \mathbf{p}^{n+1} = \mathbf{M}^U \mathbf{u}^n - \Delta t \mathbf{C}^U(\mathbf{u}^n) \mathbf{u}^n \\ \mathbf{D}^T \mathbf{u}^{n+1} = 0 \end{cases} \quad (32)$$

Convection only appears on the r.h.s. so that only operator evaluations occur for the convection operator, while the remainder appears fully implicit. This scheme is obviously only first order accurate and only conditionally stable. For higher order methods of this decomposition type (e.g. using Multistep or partitioned Runge-Kutta schemes) we refer to [41, 42, 43].

The major drawback of this type of operator splitting methods is the fact that the time step size of the explicit and the implicit part of the decomposition have to coincide. Thereby the stability restriction caused by the explicit treatment of the convection part dictates not only the number of explicit evaluations but also - which is typically more expensive - the number of solution steps for the implicit part. The decomposition methods considered in the sections 3.2 and 3.3 overcome this issue.

### 3.2. Product decomposition with operator-integration-factor splitting

The disadvantage of IMEX methods can be avoided with product decomposition methods, where sequences of separated problems are solved successively. The separated problems then only involve the Stokes or the convection operator at the same time. The major benefit of this is the fact, that the numerical solution of the sub-problems (e.g. the size of the time steps) can be chosen completely different. The derivation of the so called *operator-integration-factor splitting* approach has been derived in [45]. We sketch the idea here. The DAE (31) can *formally* be rewritten as

$$\begin{cases} \frac{\partial}{\partial t}(Q^{t \rightarrow t^*} \mathbf{u}) + Q^{t \rightarrow t^*} \mathbf{M}^{-1}(\mathbf{A} \mathbf{u} + \mathbf{D} \mathbf{p}) = 0 & \text{in } [0, T], \\ \mathbf{D}^T \mathbf{u} = 0 & \text{in } [0, T], \end{cases} \quad (33)$$

for some arbitrary  $t^* \in \mathbb{R}$  with  $Q^{t \rightarrow t^*}$  the *propagation operator*, specified later in section 3.2.2. Be aware that  $\mathbf{M}^{-1}$  is to be understood only formally for now. We are able to apply any suitable (i.e. implicit and stiffly accurate) time integration method for (33). We do this for a first order method here to explain the procedure and refer to the literature [45] for higher order variants.

**3.2.1. Implicit Euler.** We employ the implicit Euler method on (33) and arrive at

$$\begin{cases} \frac{1}{\Delta t}(Q^{t^{n+1} \rightarrow t^*} \mathbf{u}^{n+1} - Q^{t^n \rightarrow t^*} \mathbf{u}^n) + Q^{t^{n+1} \rightarrow t^*} \mathbf{M}^{-1}(\mathbf{A} \mathbf{u}^{n+1} + \mathbf{D} \mathbf{p}^{n+1}) = 0, \\ \mathbf{D}^T \mathbf{u}^{n+1} = 0. \end{cases} \quad (34)$$

After setting  $t^* = t^{n+1}$  and multiplication with  $\mathbf{M}$  this simplifies to

$$\begin{cases} (\mathbf{M} + \Delta t \mathbf{A}) \mathbf{u}^{n+1} + \Delta t \mathbf{D} \mathbf{p}^{n+1} = \mathbf{M}^{V,U} Q^{t^n \rightarrow t^{n+1}} \mathbf{I} \mathbf{u}^n, \\ \mathbf{D}^T \mathbf{u}^{n+1} = 0, \end{cases} \quad (35)$$

which is the solution of a Stokes-Brinkman problem as in (32), but with a different right hand side.

**3.2.2. The propagation operator.** The propagation operator  $Q^{t^1 \rightarrow t^2}$  is defined as  $Q^{t^1 \rightarrow t^2} \mathbf{w} = \mathbf{v}(t_2)$  where  $\mathbf{v}(s)$  solves

$$\frac{\partial \mathbf{v}}{\partial s} + (\mathbf{M}^V)^{-1} \mathbf{C}^V(s) \mathbf{v}(s) = 0, \quad \forall s \in (t^1, t^2], \quad \mathbf{v}(t_1) = \mathbf{w}. \quad (36)$$

Here  $\mathbf{C}^V(s) := \mathbf{C}^V(\hat{\mathbf{u}}(s))$  with  $\hat{\mathbf{u}}(s)$  an *extrapolation* of divergence-free solutions of previous time steps. Note, that the extrapolation of convection velocities renders the problem (36) *linear* hyperbolic and ensures stability in the sense of (25). After replacing  $Q$  by a numerical time integrator  $Q_{\Delta t}$  for (36), the time integration method is completely specified. The order of accuracy of the extrapolation  $\hat{\mathbf{u}}$  and the time integrator  $Q_{\Delta t}$  should coincide with the order of the time integration method applied on (33).

**3.2.3. Properties of product decompositions.** At this point the advantage of product decomposition methods over IMEX schemes is evident: As the time integration of (36) is independent of the one for (33) the (not necessarily) explicit time integrator can deal with stability restrictions (typically using multiple time steps) without influencing the time step for (33). This separation of the two problems also introduces the biggest disadvantage of product decomposition methods: an additional consistency error. Even if the DAE (31) has a stable stationary solution, product decomposition methods may not reach it. This is not the case for monolithic or additive decomposition approaches. Nevertheless, this *splitting error* is controlled by the time discretization error.

*Remark 9* ((Marchuk-)Yanenko splitting)

If the implicit Euler discretization as in (35) is combined with an Euler method for (36), the famous Yanenko splitting method is recovered.

### 3.3. A modified fractional-step- $\theta$ -scheme

In this section we introduce an approach to circumvent severe CFL-restrictions and splitting errors at the same time. We no longer ask for sub-problems that only involve the Stokes or the convection operator (as in section 3.2), but ask for sub-problems which *only involve one of both implicitly*. The method is based on [46] where the well-known fractional-step- $\theta$ -scheme, cf. [53, Chapter II, section 10], is modified. The resulting method is an additive decomposition method without the time step restrictions of IMEX schemes.

We start with *formally* writing down an operator-splitting version of the fractional-step- $\theta$ -scheme. The scheme is divided into three steps, the first and the last step treat the Stokes part implicitly and the convection part explicitly as in (32) while the second step treats *convection implicitly* and viscosity forces explicitly.

$$\text{Step 1 } (t^{n_0} \rightarrow t^{n_1}) : \begin{cases} (\mathbf{M} + \theta\Delta t \mathbf{A})\mathbf{u}^{n_1} + \theta\Delta t \mathbf{D} \mathbf{p}^{n_1} = \mathbf{M}\mathbf{u}^{n_0} - \theta\Delta t \mathbf{C}\mathbf{u}^{n_0} \\ \mathbf{D}^T \mathbf{u}^{n_1} = 0 \end{cases} \quad (37a)$$

$$\text{Step 2 } (t^{n_1} \rightarrow t^{n_2}) : \begin{cases} (\mathbf{M} + \theta^* \Delta t \mathbf{C})\mathbf{u}^{n_2} = \mathbf{M}\mathbf{u}^{n_1} - \theta^* \Delta t (\mathbf{A}\mathbf{u}^{n_1} + \mathbf{D}\mathbf{p}^{n_1}) \end{cases} \quad (37b)$$

$$\text{Step 3 } (t^{n_2} \rightarrow t^{n_3}) : \begin{cases} (\mathbf{M} + \theta\Delta t \mathbf{A})\mathbf{u}^{n_3} + \theta\Delta t \mathbf{D} \mathbf{p}^{n_3} = \mathbf{M}\mathbf{u}^{n_2} - \theta\Delta t \mathbf{C}\mathbf{u}^{n_2} \\ \mathbf{D}^T \mathbf{u}^{n_3} = 0 \end{cases} \quad (37c)$$

The time stages are labeled by the superscripts  $n_0, n_1, n_2, n_3$ , respectively, where  $n_0$  denotes initial data and  $n_3$  the final time stage. The time steps size for the first and the last step is  $\theta\Delta t$  and  $\theta^*\Delta t$  in the middle step with  $\theta^* = 1 - 2\theta$ . Here  $\theta = 1 - 1/\sqrt{2}$ . We note that specifications for  $\mathbf{M}$  and  $\mathbf{C}$  with respect to the considered spaces and the convection velocity for  $\mathbf{C}$  are still missing at this points. The scheme is second order accurate. In contrast to the unsplit fractional-step- $\theta$ -scheme the stability analysis of this time integration scheme is an open problem. In our experience, however, the stability restrictions are much less restrictive than those of a comparable IMEX schemes.

**3.3.1. Sub-steps in different spaces.** Initially, we stated that we want to avoid solving linear systems involving convection, for efficiency reasons. This seems to be contradictory to what is formulated in (37b). Nevertheless, as only convection is involved implicitly an efficient numerical solution is still possible. We apply a simple iterative scheme which only involves explicit operator evaluations of the convection to do so. We explain this in section 3.3.2. To do this efficiently, we want Step 2 to be formulated in the space  $V_h$  while Step 1 and 3 are to be formulated in the space  $(U_h, Q_h)$ . This poses problems the solution of which we discuss in this section.

In Step 1 and Step 3 the adjustments are obvious: Step 1 only depends on initial data in  $U_h$ . The initial data for Step 3 is in  $V_h$  but appears only in terms of functionals  $(\mathbf{M}\mathbf{v}$  and  $\mathbf{C}\mathbf{v})$  such that the transfer operations are clear. Step 2 is more involved, cf. remark 7. Functionals in  $U'_h$  (such as  $\mathbf{A}\mathbf{u}$ ) are in general not functionals in  $V'_h$ . We use the first equation in (37a) to formally define a different representation of the functionals required in Step 2:

$$\theta\Delta t \mathbf{g}^{n_1} = -\theta\Delta t (\mathbf{A}\mathbf{u}^{n_1} + \mathbf{D}\mathbf{p}^{n_1}) = \mathbf{M}(\mathbf{u}^{n_1} - \mathbf{u}^{n_0}) + \theta\Delta t \mathbf{C}\mathbf{u}^{n_0} \quad (38)$$

Now we replace  $\mathbf{M}$  and  $\mathbf{C}$  with operations suitable for a setting in  $V_h$ :

$$\mathbb{R}^{N_V} \ni \mathbf{g}^{n_1} := \frac{1}{\theta \Delta t} \mathbf{M}^{U,V}(\mathbf{u}^{n_1} - \mathbf{u}^{n_0}) + \theta \Delta t \mathbf{C}^V(\mathbf{u}^{n_0}) \mathbf{I} \mathbf{u}^{n_0} \quad (39)$$

We arrive at the modified fractional-step- $\theta$ -scheme:

$$\text{Step 1 : } \begin{cases} (\mathbf{M}^U + \theta \Delta t \mathbf{A}) \mathbf{u}^{n_1} + \theta \Delta t \mathbf{D} \mathbf{p}^{n_1} = \mathbf{M}^U \mathbf{u}^{n_0} - \theta \Delta t \mathbf{I}^T \mathbf{C}^V(\mathbf{u}^{n_0}) \mathbf{I} \mathbf{u}^{n_0} \\ \mathbf{D}^T \mathbf{u}^{n_1} = 0 \end{cases} \quad (40a)$$

$$\text{Step 2 : } \begin{cases} (\mathbf{M}^V + \theta^* \Delta t \mathbf{C}^V(t^{n_2})) \mathbf{v}^{n_2} = \mathbf{M}^{U,V} \mathbf{u}^{n_1} - \theta^* \Delta t \mathbf{g}^{n_1} \end{cases} \quad (40b)$$

$$\text{Step 3 : } \begin{cases} (\mathbf{M}^U + \theta \Delta t \mathbf{A}) \mathbf{u}^{n_3} + \theta \Delta t \mathbf{D} \mathbf{p}^{n_3} = \mathbf{M}^{V,U} \mathbf{v}^{n_2} - \theta \Delta t \mathbf{I}^T \mathbf{C}^V(t^{n_2}) \mathbf{v}^{n_2} \\ \mathbf{D}^T \mathbf{u}^{n_3} = 0 \end{cases} \quad (40c)$$

where we replaced the generic operators  $\mathbf{M}$ ,  $\mathbf{C}$  with suitable ones. We recall the definition of the extrapolated convection operator  $\mathbf{C}^V(t^{n_2}) := \mathbf{C}^V(\dot{\mathbf{u}}(t^{n_2}))$  the convection velocity of which is (linearly) extrapolated from exactly divergence-free velocities.

**3.3.2. Iterative solution of the implicit convection problem.** In (40b) we need to solve a problem of the form

$$\mathbf{v} + \tau^* (\mathbf{M}^V)^{-1} \mathbf{C}^V \mathbf{v} = \mathbf{g}^* \quad (41)$$

for given  $\tau^*$ ,  $\mathbf{g}^*$  and constant convection  $\mathbf{C}^V$ . We do this by means of a pseudo time-stepping method, i.e. we formulate (41) as the stationary solution to

$$\frac{\partial}{\partial s} \mathbf{v}(s) + \mathbf{v}(s) + \tau^* \mathbf{C}^V \mathbf{v}(s) = \mathbf{g}^*, \quad s \in [0, \infty). \quad (42)$$

Note that a stationary solution exists as  $\mathbf{I} \mathbf{d} + \tau^* \mathbf{C}^V$  only has eigenvalues  $\lambda$ , with  $\text{Re}(\lambda) > 1$ . This stationary solution is approximated with a few explicit Euler time step with an artificial time step size  $\Delta s$ .

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta s (\mathbf{g}^* - \mathbf{v}_i + \tau^* (\mathbf{M}^V)^{-1} \mathbf{C}^V \mathbf{v}_i). \quad (43)$$

This procedure can be interpreted as a Richardson iteration applied to (41). With a time step size which is tailored for stability (as in the numerical solution to (36)) we iterate (43) until the initial residual is reduced by a prescribed factor. To solve the convection step, Step 2, we only require operator evaluations for the convection. The time step size  $\Delta s$  in this iteration is decoupled from the time step size  $\Delta t$  used of the overall scheme.

## 4. NUMERICAL EXAMPLES

In this section we consider three different problems. In section 4.1 a vector-valued Poisson problem is considered using the  $H(\text{div})$ -conforming HDG space  $U_h$ . The impact of hybridization in the context of linear systems is compared to other DG methods.

Within the DFG Priority Research Programm 'Flow Simulation on High Performance Computers' a set of challenging benchmark problems had been defined. In the sections 4.2 and 4.3 we consider two transient problems from this set which are formulated in [55]. In section 4.2 a two-dimensional benchmark problem is considered to demonstrate the accuracy of our method for a demanding test case and to compare the discussed time integration methods. Finally, in section 4.3 we discuss a three-dimensional, and hence computationally challenging, benchmark problem from this set. We compare accuracy and run-time performance to the data of the study in [56].

The methods discussed in this paper have been implemented in the add-on package `ngsflow` [57] for the high order finite element library `NGSolve` [58]. The computations in this section have also been carried out with this software. Throughout this section we only consider direct solvers and comment on linear solvers below, in remark 11.

For the computations of the benchmark problem in sections 4.2 and 4.3 we used the reduced  $H(\text{div})$ -conforming space  $W_h^{\text{red}}$ , cf. remark 1 and the *projected jumps* modification, cf. section 2.2.



#### 4.1. Linear systems - A comparison between DG and HDG methods

We consider the comparably simple vector-valued Poisson problem

$$-\Delta u = f \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega, \quad (44)$$

discretized with four different methods. The three-dimensional domain  $\Omega$  is the same as the one in section 4.3 with an unstructured tetrahedral mesh consisting of 3487 elements. The problem (44) leads to discretizations with symmetric positive definite matrices  $\mathbf{A}$ . Only the lower triangular part of the sparse matrix has to be stored and a sparse Cholesky factorization algorithm can be used. In this comparison, we used the sparse direct solver `PARDISO`, cf. [59, 60]. We are only concerned with the sparsity pattern of different methods before and after static condensation and do not compare accuracy or conditioning of the discretizations. Moreover, for the sake of simplicity we do *not* apply the reduction of the space  $W_h$  as discussed in remark 1. The following quantities of interest for the linear systems arising from discretizations of (44) are displayed in Table II:

#dof[K] : number of unknowns (in thousands)

#cdof[K] : number of unknowns after static condensation (in thousands)

#nzeA[K] : number of non-zero entries in the system matrix  $\mathbf{A}$  (lower triangle) (in thousands)

#nzeL[K] : number of non-zero entries in the Cholesky factor  $\mathbf{L}$  (in thousands)

Four different methods are considered on  $W_h$  or  $U_h$  with varying polynomial degree between  $k = 1$  and  $k = 6$ :

1. HDG: The HDG method proposed in section 2 without *projected jumps*.
2. PHDG: The HDG method with *projected jumps*, cf. section 2.2.
3. Std.DG.: A standard DG method using the space  $W_h$  where the basis is constructed such that all degrees of freedom from one element couple with *all* degrees of freedom from adjacent elements.
4. N.DG: A nodal DG method using the space  $W_h$  where basis functions are assumed to be constructed such that degrees of freedom associated to one element couple only with degrees of freedom from adjacent elements which have support on the shared facet. At the same time we assume that basis functions are constructed such that the number of basis functions with support on a facets is minimized, cf. [10]. In terms of the sparsity pattern this nodal DG method represents the best case for a DG method without hybridization.

	Std.DG	N.DG	HDG	PHDG	Std.DG	N.DG	HDG	PHDG
	$k = 1$				$k = 2$			
#dof[K]	23	23	69	38	67	67	158	112
#cdof[K]	23	23	69	38	67	67	137	91
#nzeA[K]	732	676	2 037	637	5 073	4 177	8 103	3 628
#nzeL[K]	10 113	10 208	17 768	5 569	64 463	69 831	70 138	31 415
	$k = 3$				$k = 4$			
#dof[K]	146	146	298	237	271	271	500	423
#cdof[K]	146	146	229	168	271	261	343	267
#nzeA[K]	21 686	16 051	22 443	12 124	69 525	46 912	50 412	30 588
#nzeL[K]	261 977	260 416	194 524	104 496	814 168	731 253	435 129	264 183
	$k = 5$				$k = 6$			
#dof[K]	453	453	773	681	702	702	1 128	1 022
#cdof[K]	453	411	480	389	702	597	640	533
#nzeA[K]	184 035	101 473	98 696	64 816	424 764	200 813	175 321	121 944
#nzeL[K]	2 099 690	1 741 072	847 913	557 798	4 752 072	3 519 061	1 502 558	1 045 444

Table II. Comparison of different DG methods for the vector-valued reaction Poisson problem (44).

In Table II we observe that for small polynomial degree  $k$  the amount of additional unknowns required for the HDG formulation is quite large as are the nonzero entries in the system matrix.

Nevertheless except for  $k = 1$ , the number of nonzero entries in the Cholesky factor are comparable ( $k = 2$ ) to the Standard DG methods or less ( $k > 2$ ). For high order, i.e.  $k \geq 4$  the HDG method performs significantly better as it has less nonzero entries in  $\mathbf{A}$  and  $\mathbf{L}$ . The HDG space with the *projected jumps* modification improves the situation dramatically. It essentially compensates the overhead of the HDG method for small  $k$ . But even for  $k = 6$  the effect is still significant. We note, that the difference between the first three methods increases for increasing polynomial degree  $k$  whereas the difference between the last two method, PHDG and HDG, decreases. In all cases the HDG method with *projected jumps* outperforms all alternatives.

#### 4.2. A two-dimensional benchmark problem

In this section we consider the benchmark problem denoted as “2D-2Z” in [55] where a laminar flow around a circle-shaped obstacle is considered. The Reynolds number is moderately high ( $Re = 100$ ) and results in a periodic vortex street behind the obstacle. We briefly introduce the problem (for more details we refer to [55]) and the numerical setup to investigate spatial and temporal discretization errors. Finally, we discuss the obtained results.

**4.2.1. Geometrical setup and boundary conditions.** The domain is a rectangular channel without an almost vertically centered circular obstacle, cf. Figure 2,

$$\Omega := [0, 2.2] \times [0, 0.41] \setminus \{\|x - (0.2, 0.2)\|_2 \leq 0.05\}. \quad (45)$$

The boundary is decomposed into  $\Gamma_{in} := \{x = 0\}$ , the inflow boundary,  $\Gamma_{out} := \{x = 2.2\}$ , the outflow boundary and  $\Gamma_W := \partial\Omega \setminus (\Gamma_{in} \cup \Gamma_{out})$ , the wall boundary. On  $\Gamma_{out}$  we prescribe natural boundary conditions  $(-\nu \nabla u + pI) \cdot n = 0$ , on  $\Gamma_W$  homogeneous Dirichlet boundary conditions for the velocity and on  $\Gamma_{in}$  the inflow Dirichlet boundary conditions

$$u(0, y, t) = u_D = (3/2 \cdot \bar{u}) \cdot 4 \cdot y(d_y - y)/d_y^2 \cdot (1, 0, 0).$$

Here,  $\bar{u} = 1$  and the viscosity is fixed to  $\nu = 10^{-3}$  which results in a Reynolds number  $Re = 100$ .

**4.2.2. Drag and Lift.** The quantities of interest in this example are the (maximal and minimal) drag and lift forces  $c_D, c_L$  that act on the disc. These are defined as

$$c_D := \frac{1}{\bar{u}^2 r} \int_{\Gamma_o} \left( \nu \frac{\partial u}{\partial n} - pn \right) \cdot e_x \, ds, \quad c_L := \frac{1}{\bar{u}^2 r} \int_{\Gamma_o} \left( \nu \frac{\partial u}{\partial n} - pn \right) \cdot e_y \, ds.$$

Here  $e_x, e_y$  denote the unit vectors in  $x$  and  $y$  direction,  $r = 0.05$  is the radius of the obstacle,  $\bar{u}$  is the average inflow velocity ( $\bar{u} = 1$ ) and  $\Gamma_o$  denotes the surface of the obstacle.

**4.2.3. Numerical setup.** We use an unstructured triangular grid with an additional layer of quadrilaterals around the disk which is anisotropically refined towards the disk once. In Figure 2 the geometry, the mesh and a typical solution is depicted.

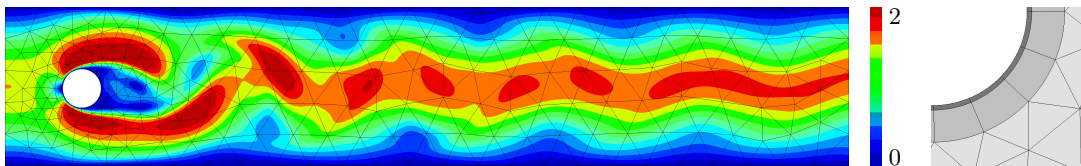


Figure 2. Sketch of the mesh and the solution (color coding corresponding to velocity magnitude  $\|u\|_2$ ) to the problem considered in section 4.2 at a fixed time  $t$  (left) and zoom-in on the boundary layer mesh (right).

In order to be able to neglect time discretization errors, when investigating the spatial accuracy, we consider the use of a well-known stiffly accurate second order Runge-Kutta-IMEX scheme, taken from [42, section 2.6], with an extremely small time step size  $3.125 \cdot 10^{-5}$  which means that roughly

10000 time steps are used to resolve one full period. The duration of one full cycle is roughly  $1/3s$ . We also fix the mesh and consider a pure  $p$ -refinement, i.e. variations of the polynomial degree  $k$ .

For the investigations of the temporal discretization error, we consider a fixed polynomial degree  $k = 6$ , with the same mesh. For the considered example the stability restriction of the second order IMEX scheme is severe. A time step of below  $10^{-3}$  has to be considered. The intention of the discussion of operator-splitting methods in section 3 has been to present strategies to circumvent or relax these severe conditions. We compare the performance of a product decomposition method, a second order operator-integration-factor splitting version of the BDF2 method, and the modified fractional-step- $\theta$ -scheme discussed in section 3.3. Note that these methods allows to consider much larger time steps than the IMEX scheme. As references we give the values from the literature [61] and from the IMEX scheme with  $\Delta t = 10^{-3}$  (stability limit) and the reference solution with  $\Delta t = 3.125 \cdot 10^{-5}$ .

**4.2.4. Numerical results: spatial discretization.** In Table III the quantities of interest are shown for varying polynomial degree  $k$ . As a reference we also show the result obtained by FEATFLOW [62] with a discretization using quadrilateral meshes and continuous second order finite elements for the velocity with a discontinuous piecewise linear pressure ( $Q_2/P_1^{\text{disc}}$ ). These results have been made accessible on [61]. We observe a rapid convergence for the  $p$ -refinement, i.e. the increase of the

	#dof	max $c_D$	min $c_D$	max $c_L$	min $c_L$
$k = 1$	2 211	2.52594	2.47871	0.65728	-0.81672
$k = 2$	4 148	3.22841	3.16260	1.00571	-1.03894
$k = 3$	6 558	3.23184	3.16842	0.98822	-1.02427
$k = 4$	9 441	3.22714	3.16401	0.98431	-1.01906
$k = 5$	12 797	3.22759	3.16432	0.98578	-1.02053
$k = 6$	16 626	3.22757	3.16430	0.98580	-1.02053
ref. [61]	167 232	3.22662	3.16351	0.98620	-1.02093
	667 264	3.22711	3.16426	0.98658	-1.02129

Table III. Accuracy of the spatial discretization: results for different polynomial degrees.

polynomial degree. Compared to the results from the literature [61] the same order of accuracy is achieved with a lot less degrees of freedoms.

**4.2.5. Numerical results: temporal discretization.** The results for the temporal discretization are shown in Table IV. First of all, we observe that the second order IMEX scheme is already very accurate at its stability limit. But the method does not allow to choose larger time steps. This is in contrast to the alternative methods discussed here. For these methods we can consider much large time steps and observe a second order convergence. In this example the product decomposition method is more accurate than the modified fractional step method by one (time) level. We note that one major concern with this method is however, that splitting errors appear also if a stationary to the flow problem exists. This is not the case for additive decomposition methods or the proposed modified fractional step method.

### 4.3. A three-dimensional benchmark problem

Finally, we consider a three dimensional benchmark problem, the problem “3D-3Z” in [55]. In contrast to the problem in section 4.2 the inflow velocity is varied over time and the observed time interval is fixed. The maximal Reynolds number in this configuration is also  $Re = 100$  as in the previous section. The focus in this section is on the study of performance in the sense of computational effort over accuracy.

**4.3.1. Geometrical setup and boundary conditions.** The geometrical setup is a generalization of the problem in the previous section. A cylindrical-shaped obstacle is places in a cuboid-shaped channel

	$1/\Delta t$	$\max c_D$	$\min c_D$	$\max c_L$	$\min c_L$
2nd order IMEX	<1 000		unstable		
	1 000	3.22754	3.16437	0.98486	-1.01957
	32 000	3.22757	3.16431	0.98580	-1.02053
operator-integration- factor splitting (BDF2)	125	3.38456	3.28279	1.16052	-1.19939
	250	3.25564	3.18643	1.01725	-1.05276
	500	3.23239	3.16836	0.98864	-1.02378
	1 000	3.22819	3.16394	0.98320	-1.02013
modified fractional- step- $\theta$ -scheme	125	3.38272	3.29673	1.07667	-1.12227
	250	3.28713	3.21483	1.00937	-1.04840
	500	3.25036	3.18127	0.99172	-1.02807
	1 000	3.23656	3.17094	0.98721	-1.02227

Table IV. Accuracy of operator-splitting time integration methods

slightly above the vertical center:

$$\Omega := [0, 2.5] \times [0, 0.41] \times [0, 0.41] \setminus \{ \|(x_1, x_2) - (0.5, 0.2)\|_2 \leq 0.05 \}. \quad (46)$$

Boundary conditions are chosen as in the previous example except for a change to unsteady inflow boundary conditions:

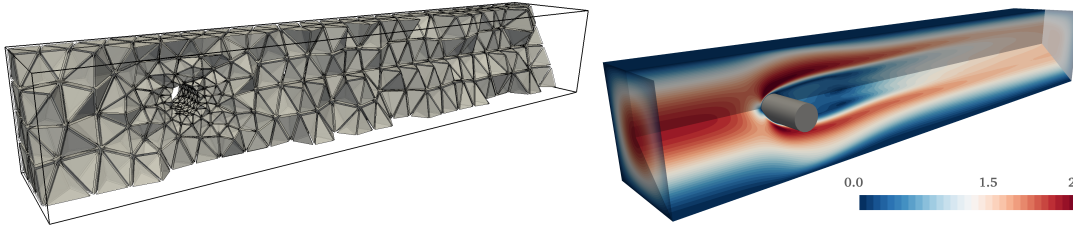
$$u(0, y, z, t) = u_D(t) = (9/4 \cdot \bar{u}(t)) \cdot 16 \cdot y(d_y - y)/d_y^2 \cdot z(d_z - z)/d_z^2 \cdot (1, 0, 0),$$

with  $\bar{u}(t)$  the average inflow velocity which is time-dependent,  $\bar{u}(t) = \sin(\pi t/8)$ . The considered time interval is  $[0, 8s]$ , the viscosity is set to  $\nu = 10^{-3}$  s.t. the Reynolds number varies between  $Re = 0$  and  $Re = 100$ .

**4.3.2. Drag and Lift.** Again, the quantities of interest in this example are the (maximal and minimal) drag and lift forces  $c_D, c_L$  that act on the disc. These are defined as

$$c_D := \frac{1}{\bar{u}_{\max}^2 r h} \int_{\Gamma_o} \left( \nu \frac{\partial u}{\partial n} - p n \right) \cdot e_x \, ds \quad c_L := \frac{1}{\bar{u}_{\max}^2 r h} \int_{\Gamma_o} \left( \nu \frac{\partial u}{\partial n} - p n \right) \cdot e_y \, ds$$

with  $\bar{u}_{\max} = \max_{t \in [0, 8]} \bar{u}(t) = 1$ ,  $r = 0.05$ ,  $h = 0.41$  and  $\Gamma_o$  the surface of the obstacle.

Figure 3. Used mesh (left) and solution at  $t = 0.4$  (right) for the benchmark problem in section 4.3.

**4.3.3. Numerical setup.** We use an unstructured tetrahedral mesh consisting of 5922 elements, cf. Figure 3. For the time discretization we use the modified fractional-step- $\theta$ -scheme discussed in section 3.3. To compensate for the increase in the spatial accuracy for increasing polynomial degree  $k$ , we adapt the number of time steps accordingly. The computations were carried out on a shared-memory computer with 24 cores. We comment on details of the computing times in remark 10.

	#ndof [K]	max $c_D$	max $c_L$	min $c_L$	$\Delta t$ [s]	comp. time [s]
$k = 2$	169	3.43046	0.00262	-0.016289	0.0080	$492 \times 24$
$k = 3$	343	3.29331	0.00277	-0.011099	0.0080	$964 \times 24$
$k = 4$	595	3.29853	0.00278	-0.010762	0.0040	$3\,087 \times 24$
$k = 5$	939	3.29798	0.00278	-0.011054	0.0040	$6\,670 \times 24$
ref. [56]	11 432	3.2963	0.0028	-0.010992	0.01	$35\,550 \times 24$
	89 760	3.2978	0.0028	-0.010999	0.005	$214\,473 \times 48$
ref. [63]	7 036	3.2968		-0.011		
ref. [55]		[3.2,3.3]	[0.002,0.004]			

Table V. Numerical results for the benchmark problem “3D3Z” in [55]. Results obtained with different polynomial degrees and reference values.

**4.3.4. Numerical results.** In Table V the results obtained are compared with the literature in terms of accuracy and computing time. We observe that we can achieve the same level of accuracy as the results in [56] (and [63]) with a computing time which is dramatically smaller. We note that the computing time per degree of freedom is actually worse than in [56]. Nevertheless, the same accuracy is achieved with much less degrees of freedoms using the high order method ( $k > 2$ ), s.t. our computations exceed the performance results in the literature. In the study [56] one of the conclusions is that their third order method ( $Q_2/P_1^{\text{disc}}$ ) is much more efficient compared to lower order methods. We extend this conclusion in the sense that the use of even higher order discretizations, i.e.  $k > 2$ , increases efficiency even further. Moreover, high order discretizations can be implemented efficiently. One important component for the efficient handling of the Navier-Stokes equations with our high order discretization is the time integration using operator-splitting.

*Remark 10* (Computation times)

We remark on the computation for the case  $k = 5$ . In that computation approximately 65% of the computing time has been spend on the solution of linear systems for Stokes-type problems, 30% on convection operator evaluations and 5% on the setup and remaining operations. To solve the implicit convection problem (Step 2 in (40)) an average of 20 iterations has been applied.

*Remark 11* (Linear systems)

In the test cases in this section we only applied direct solvers which is possible due to the (comparably) small size of the arising linear systems. For problems with increasing complexity efficient linear solvers are mandatory. The development of suitable preconditioners of the Stokes problem is based on efficient preconditioning of the bilinear form  $\mathcal{A}_h$ . For the scalar problem we could show poly-logarithmic bounds in  $k$  for the condition number of standard p-version domain decomposition preconditioners, cf. [49]. We plan to investigate suitable generalizations of this preconditioner for the Stokes problem in the future.

## 5. CONCLUSION

We presented and discussed a combined DG/HDG discretization tailored for efficiency. We summarize the core components. We split the Navier-Stokes problem into linear Stokes-type problems and hyperbolic transport problems by means of operator-splitting time integration. For the Stokes-type problem we use an  $H(\text{div})$ -conforming Hybrid DG formulation with a new modification: the *projected jumps* formulation. The Hybrid DG formulation facilitates the efficient solution of linear systems compared to other DG methods. The *projected jumps* formulation improves its efficiency even more. For the hyperbolic transport problem we apply a standard DG formulation. In numerical test cases we demonstrated the performance of the method.

## ACKNOWLEDGEMENTS

The authors greatly appreciate the contribution of Philip Lederer related to the calculation of LBB-constants, cf. remark 4.

## REFERENCES

1. Reed WH, Hill T. Triangular mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479* 1973; .
2. Lesaint P, Raviart PA. On a finite element method for solving the neutron transport equation. *Mathematical aspects of finite elements in partial differential equations* 1974; **33**:89–123.
3. Johnson C, Pitkäranta J. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of computation* 1986; **46**(173):1–26.
4. Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of computational physics* 1997; **138**(2):251–285.
5. Bassi F, Rebay S, Mariotti G, Pedinotti S, Savini M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. *Proceedings of 2nd European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics*, Technologisch Instituut, Antwerpen, Belgium, 1997; 99–108.
6. Cockburn B, Shu CW. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics* 1998; **141**(2):199–224.
7. Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM journal on numerical analysis* 2002; **39**(5):1749–1779.
8. Houston P, Schwab C, Süli E. Discontinuous hp-finite element methods for advection-diffusion-reaction problems. *SIAM Journal on Numerical Analysis* 2002; **39**(6):2133–2163.
9. Rivière B. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. Society for Industrial and Applied Mathematics, 2008.
10. Hesthaven JS, Warburton T. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.
11. Karniadakis G, Sherwin S. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
12. Girault V, Raviart PA. *Finite element methods for Navier-Stokes equations: theory and algorithms*, vol. 5. Springer Science & Business Media, 2012.
13. Elman HC, Silvester DJ, Wathen AJ. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, 2014.
14. Donea J, Huerta A. *Finite element methods for flow problems*. John Wiley & Sons, 2003.
15. Toselli A. hp discontinuous Galerkin approximations for the Stokes problem. *Mathematical Models and Methods in Applied Sciences* 2002; **12**(11):1565–1597.
16. Schötzau D, Schwab C, Toselli A. Mixed hp-DGFEM for incompressible flows. *SIAM Journal on Numerical Analysis* 2002; **40**(6):2171–2194.
17. Girault V, Rivière B, Wheeler M. A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier-Stokes problems. *Mathematics of Computation* 2005; **74**(249):53–84.
18. Cockburn B, Kanschat G, Schötzau D. A locally conservative LDG method for the incompressible Navier-Stokes equations. *Mathematics of Computation* 2005; **74**(251):1067–1095.
19. Cockburn B, Kanschat G, Schötzau D. A note on discontinuous Galerkin divergence-free solutions of the Navier–Stokes equations. *Journal of Scientific Computing* 2007; **31**(1-2):61–73.
20. Brezzi F, Fortin M. *Mixed and hybrid finite element methods*, vol. 15. Springer Science & Business Media, 2012.
21. Lehrenfeld C. Hybrid discontinuous Galerkin methods for solving incompressible flow problems. *Rheinisch-Westfälischen Technischen Hochschule Aachen* 2010; .
22. Egger H, Schöberl J. A hybrid mixed discontinuous Galerkin method for convection–diffusion problems. *J. Numer. Anal* 2008; .
23. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. *Journal of Computational Physics* 2009; **228**(9):3232–3254.
24. Cockburn B, Gopalakrishnan J, Lazarov R. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis* 2009; **47**(2):1319–1365.
25. Cockburn B, Gopalakrishnan J, Nguyen N, Peraire J, Sayas FJ. Analysis of HDG methods for Stokes flow. *Mathematics of Computation* 2011; **80**(274):723–760.
26. Cesmelioglu A, Cockburn B, Nguyen NC, Peraire J. Analysis of HDG methods for Oseen equations. *Journal of Scientific Computing* 2013; **55**(2):392–431.
27. Zhai Q, Zhang R, Wang X. A hybridized weak Galerkin finite element scheme for the Stokes equations. *Science China Mathematics* 2015; :1–18.
28. Di Pietro D, Ern A, Lemaire S. A review of hybrid high-order methods: formulations, computational aspects, comparison with other methods. *Technical Report HAL-01163569* September 2015.
29. Egger H, Waluga C. hp analysis of a hybrid DG method for Stokes flow. *IMA Journal of Numerical Analysis* 2012; :drs018.
30. Nguyen N, Peraire J, Cockburn B. A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(9):582–597.
31. Carrero J, Cockburn B, Schötzau D. Hybridized globally divergence-free LDG methods. part i: The Stokes problem. *Mathematics of computation* 2006; **75**(254):533–563.

32. Cockburn B, Gopalakrishnan J. Incompressible finite elements via hybridization. part i: The Stokes system in two space dimensions. *SIAM Journal on Numerical Analysis* 2005; **43**(4):1627–1650.
33. Cockburn B, Gopalakrishnan J. Incompressible finite elements via hybridization. part ii: The Stokes system in three space dimensions. *SIAM Journal on Numerical Analysis* 2005; **43**(4):1651–1672.
34. Könnö J, Stenberg R. Numerical computations with H(div)-finite elements for the Brinkman problem. *Computational Geosciences* 2012; **16**(1):139–158.
35. Qiu W, Shi K. An HDG method for linear elasticity with strong symmetric stresses. *arXiv preprint arXiv:1312.1407* 2013; .
36. Oikawa I. A hybridized discontinuous Galerkin method with reduced stabilization. *Journal of Scientific Computing* 2014; :1–14.
37. Oikawa I. A reduced HDG method for the Stokes equations. *arXiv preprint arXiv:1502.01833* 2015; .
38. Qiu W, Shi K. A superconvergent HDG method for the incompressible Navier-Stokes equations on general polyhedral meshes. *CoRR* 2015; **abs/1506.07543**.
39. Kirby RM, Sherwin SJ, Cockburn B. To CG or to HDG: a comparative study. *Journal of Scientific Computing* 2012; **51**(1):183–212.
40. Huerta A, Angeloski A, Roca X, Peraire J. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering* 2013; **96**(9):529–560.
41. Ascher UM, Ruuth SJ, Wetton BT. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis* 1995; **32**(3):797–823.
42. Ascher UM, Ruuth SJ, Spiteri RJ. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics* 1997; **25**(2):151–167.
43. Kanevsky A, Carpenter MH, Gottlieb D, Hesthaven JS. Application of implicit–explicit high order Runge–Kutta methods to discontinuous-Galerkin schemes. *Journal of Computational Physics* 2007; **225**(2):1753–1781.
44. Strang G. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis* 1968; **5**(3):506–517.
45. Maday Y, Patera AT, Rønquist EM. An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow. *Journal of Scientific Computing* 1990; **5**(4):263–292.
46. Crispell J, Ervin V, Jenkins E. A fractional step  $\theta$ -method for convection–diffusion problems. *Journal of Mathematical Analysis and Applications* 2007; **333**(1):204–218.
47. Zaglmayr S. High order finite element methods for electromagnetic field computation. PhD Thesis, JKU Linz 2006.
48. Arnold DN. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis* 1982; **19**(4):742–760.
49. Schöberl J, Lehrenfeld C. Domain decomposition preconditioning for high order hybrid discontinuous Galerkin methods on tetrahedral meshes. *Advanced finite element methods and applications*. Springer, 2013; 27–56.
50. Brezzi F, Manzini G, Marini D, Pietra P, Russo A. Discontinuous finite elements for diffusion problems. *Atti Convegno in onore di F. Brioschi (Milano 1997), Istituto Lombardo, Accademia di Scienze e Lettere* 1999; :197–217.
51. Linke A. On the role of the Helmholtz decomposition in mixed methods for incompressible flows and a new variational crime. *Computer Methods in Applied Mechanics and Engineering* 2014; **268**:782–800.
52. Dubiner M. Spectral methods on triangles and other domains. *Journal of Scientific Computing* 1991; **6**(4):345–390.
53. Glowinski R. Finite element methods for incompressible viscous flow. *Handbook of numerical analysis* 2003; **9**:3–1176.
54. Chorin AJ. The numerical solution of the Navier-Stokes equations for an incompressible fluid. *Bulletin of the American Mathematical Society* 1967; **73**(6):928–931.
55. Schäfer M, Turek S, Durst F, Krause E, Rannacher R. Benchmark computations of laminar flow around a cylinder. *Flow simulation with high-performance computers II* 1996; :547–566.
56. Bayraktar E, Mierka O, Turek S. Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow. *International Journal of Computational Science and Engineering* 2012; **7**(3):253–266.
57. ngsflow: Flow solver package for NGSolve. [www.sourceforge.net/projects/ngsflow/](http://www.sourceforge.net/projects/ngsflow/).
58. Schöberl J. C++11 implementation of finite elements in NGSolve. *Technical Report ASC-2014-30*, Institute for Analysis and Scientific Computing September 2014. URL <http://www.asc.tuwien.ac.at/~schoeberl/wiki/publications/ngs-cpp11.pdf>.
59. PARDISO Solver project. [www.pardiso-project.org](http://www.pardiso-project.org).
60. Schenk O, Gärtner K. Solving unsymmetric sparse systems of linear equations with PARDISO. *J. of Future Generation Computer Systems* 2004; **20**:475–487.
61. FEATFLOW Finite element software for the incompressible Navier-Stokes equations. [www.featflow.de](http://www.featflow.de).
62. Turek S, Becker C. FEATFLOW: Finite element software for the incompressible Navier-Stokes equations, User Manual Release 1.1. *Preprint, Heidelberg* 1998; **4**.
63. John V. On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3d flow around a cylinder. *International Journal for Numerical Methods in Fluids* 2006; **50**(7):845–862.