

# A Hybridized Discontinuous Galerkin Method for Unsteady Flows with Shock-Capturing

Alexander Jaust<sup>\*</sup>, Jochen Schütz<sup>†</sup> and Michael Woopen<sup>‡</sup>

Preprint No. 399

July 2014

Key words: hybridized discontinuous Galerkin,  
instationary convection-diffusion equations,  
shock-capturing, mesh adaptation

AMS Subject Classifications: 65L06, 65M60, 35M33, 35L67

**Institut für Geometrie und Praktische Mathematik  
RWTH Aachen**

**Templergraben 55, D-52056 Aachen, (Germany)**

---

<sup>\*</sup> PhD Student, IGPM, RWTH Aachen University, Aachen, 52062, Germany

<sup>†</sup> Postdoctoral Researcher, IGPM, RWTH Aachen University, Aachen, 52062, Germany

<sup>‡</sup> PhD Student, AICES, RWTH Aachen University, Aachen, 52062, Germany

# A Hybridized Discontinuous Galerkin Method for Unsteady Flows with Shock-Capturing

Alexander Jaust\*, Jochen Schütz† and Michael Woopen‡

*RWTH Aachen University, Aachen, 52062, Germany*

We present a hybridized discontinuous Galerkin (HDG) solver for the time-dependent compressible Euler and Navier-Stokes equations. In contrast to discontinuous Galerkin (DG) methods, the number of globally coupled degrees of freedom is usually tremendously smaller for HDG methods, as these methods can rely on hybridization. However, applying the method to a time-dependent problem amounts to solving a differential-algebraic nonlinear system of equations (DAE), rendering the problem extremely stiff. This implies that *implicit* time discretization has to be used. Suited methods for the treatment of these DAEs are, e.g., diagonally implicit Runge-Kutta (DIRK) methods, or the backward differentiation formulas (BDF). In order to solve a wide range of problems in an efficient manner, we employ adaptive time stepping using an embedded error estimator. Additionally, we investigate the use of artificial viscosity for shock-capturing in this setting, and we propose a new strategy of coarsening the mesh using non-standard polygonal elements.

## I. Introduction

Recently, high-order methods have gained lots of attention for problems arising in computational fluid dynamics (CFD). Arguably, one of the most popular class of these methods is constituted by the discontinuous Galerkin (DG) methods.<sup>1,2,3,4,5,6,7,8</sup> Based on a triangulation of the computational domain, the approximate solution within the DG method is represented by polynomials defined locally on each of the elements. This allows for a flexible description by varying the degree  $p$  of the polynomials or the number of elements  $N$  to attain a desired accuracy, while keeping a high degree of locality of the scheme. One potential drawback of DG methods is that they suffer from a high number of globally coupled degrees of freedom (DOF). This number scales like  $\mathcal{O}((p+1)^d N)$  where  $d$  denotes spatial dimension. In order to reduce the number of globally coupled degrees of freedom, *hybridized* discontinuous Galerkin methods have been introduced.<sup>9,10,11,12,13,14,15</sup> The underlying idea is analogue to the idea of *hybrid mixed* methods:<sup>16,17</sup> An additional variable, usually called  $\lambda_h$ , is introduced that has support on the skeleton of the mesh only. At first sight, this obviously leads to an increase in the total number of unknowns. However, this proceeding allows for static condensation of the resulting system of equations. Consequently, the nonlinear system to be solved has typically fewer degrees of freedom, as its size behaves like  $\mathcal{O}((p+1)^{d-1} \hat{N})$  for hybrid methods, with  $\hat{N}$  being the number of edges of the elements.

Applying the HDG method to an unsteady convection-diffusion equation such as the Navier-Stokes equations leads to a differential algebraic equation (DAE) of index one. Such an equation is usually highly stiff, and it is therefore necessary to use implicit time integration methods. Integrators such as backward differentiation formulas (BDF) and diagonally implicit Runge-Kutta (DIRK) methods spring to mind, and indeed, the HDG method has been successfully applied to different time-dependent problems using those integrators.<sup>18,12,19,20</sup> It has been shown that the overall method is stable and of high-order.

Using *implicit* instead of *explicit* time integrators results in higher cost per time step, as one has to solve a (usually nonlinear) system of equations. However, implicit methods are in general not bound by a CFL condition (which can be rather severe especially for parabolic problems), and so it is possible to adapt the size of the time step to keep the computational effort as low as possible. We show that this can be achieved

---

\*PhD Student, IGPM, RWTH Aachen University, Aachen, 52062, Germany

†Postdoctoral Researcher, IGPM, RWTH Aachen University, Aachen, 52062, Germany

‡PhD Student, AICES, RWTH Aachen University, Aachen, 52062, Germany

by using embedded DIRK methods.<sup>21,22,23</sup> These schemes for solving ODEs produce two approximations to the solution, whose difference can be used for error estimation and, subsequently, for time step adaptation.

In the context of computational fluid dynamics, discontinuities in the solution occur frequently, usually for transonic or supersonic flows with Mach number  $\text{Ma} > 1$ . Especially for high order schemes, this can pose severe stability problems. In order to have a robust discretization that can handle a wide range of problems it is therefore necessary to introduce some sort of stabilization. Within finite volume and discontinuous Galerkin methods, limiting<sup>24,25,26</sup> is a popular way of stabilizing the method. However, applying a straightforward limiting were to destroy the possibility of using static condensation, as one has to rely on inter-element quantities of the approximate solution  $w_h$ . A common approach to stabilize numerical schemes are artificial viscosity models,<sup>27,28,29,30</sup> mimicking the stabilization used in streamline diffusion finite elements<sup>31</sup> or in upwind-type finite differences,<sup>32</sup> by adding viscosity in an explicit fashion. Again, as in the case of limiting, not every artificial viscosity model is suited for HDG methods, as some of those present can destroy the locality of these methods and ergo the possibility of using hybridization. This is especially true when using, e.g., jumps of certain quantities as indicators. A model specifically designed for HDG has been introduced by Nguyen and Peraire.<sup>30</sup> It uses the dilatation as a shock sensor and adds artificial viscosity scaled with dilatation. Their focus is on stationary problems, and we have observed that in our test cases, this sensor adds overly much diffusion. Therefore, in this work we combine the HDG method with a shock-capturing method introduced by Persson and Peraire.<sup>33</sup> Their capturing scheme utilizes the fact that for smooth solutions, the higher-order basis coefficients corresponding to a hierarchical Legendre basis tend to zero rather fast. Therefore, the base coefficients corresponding to the highest order basis functions can serve as a discontinuity detector.

It is obvious that temporal adaptation is not enough to obtain an overall efficient algorithm. To this end, we also rely on an unsteady mesh adaptation procedure. Refinement of marked elements is done using a straightforward refinement strategy. However, to make the coarsening more flexible, we generate non-standard polygonal elements by simply collocating existing triangles. This way, we do not have to keep track of existing refinement hierarchies, but the coarsening step is straightforward. For an analysis of a discontinuous Galerkin finite element method on polygonal elements, we refer to Cangiani et al.<sup>34</sup>

The outline of this work is as follows: We briefly introduce Euler and Navier-Stokes equations, and subsequently their discretization using the HDG method. Then, we describe the use of embedded diagonally implicit Runge-Kutta (DIRK) methods for time integration, including the time step adaptation strategy. Afterward the artificial viscosity model for the HDG method is presented, and we comment on using mesh adaptation. Finally, we show results for the time adaptive case and the shock-capturing case.

## II. The Hybridized Discontinuous Galerkin Method for Unsteady Flows

### II.A. Underlying Equations

We consider unsteady convection-diffusion problems on two-dimensional domains  $\Omega \subset \mathbb{R}^2$ . Those problems can be written in form

$$w_t + \nabla \cdot (f(w) - f_v(w, \nabla w)) = 0 \quad \forall (x, t) \in \Omega \times [0, T] \quad (1)$$

$$w(x, 0) = w_0(x) \quad \forall x \in \Omega \quad (2)$$

with convective flux  $f = (f_1, f_2)$  and viscous flux  $f_v = (f_{v,1}, f_{v,2})$ . We are interested in the solution at some prescribed final time  $T > 0$ . A particular example are the Euler and Navier-Stokes equations, where the vector of unknowns  $w = (\rho, \rho u_1, \rho u_2, E)^T$  contains the density  $\rho$ , momentum in  $x_1$ -direction  $\rho u_1$ , respectively  $\rho u_2$  in  $x_2$ -direction and total energy  $E$ . The convective flux of the Euler and Navier-Stokes equations and the viscous flux of the Navier-Stokes equations are given by:

$$f_1 = (\rho u_1, P + \rho u_1^2, \rho u_1 u_2, u_1(E + P))^T, \quad f_2 = (\rho u_2, \rho u_1 u_2, P + \rho u_2^2, u_2(E + P))^T, \quad (3)$$

$$f_{v,1} = (0, S_{11}, S_{21}, S_{11}u_1 + S_{12}u_2 + \kappa\theta_{x_1})^T, \quad f_{v,2} = (0, S_{12}, S_{22}, S_{21}u_1 + S_{22}u_2 + \kappa\theta_{x_2})^T. \quad (4)$$

We denote the stress tensor by  $S$ , the pressure by  $P$ , the temperature by  $\theta$  and the thermal conductivity by  $\kappa$ . The pressure  $P$  is connected to the other quantities through the ideal gas law

$$P = (\gamma - 1) \left( E - \frac{\rho(u_1^2 + u_2^2)}{2} \right) \quad (5)$$

where  $\gamma$  (assumed to be 1.4) is the ratio of specific heats. To obtain a set of partial differential equations of first order we rewrite equation (1) as

$$\sigma = \nabla w \quad \forall (x, t) \in \Omega \times [0, T] \quad (6)$$

$$w_t + \nabla \cdot (f(w) - f_v(w, \sigma)) = 0 \quad \forall (x, t) \in \Omega \times [0, T] \quad (7)$$

$$w(x, 0) = w_0(x) \quad \forall x \in \Omega \quad (8)$$

The system (6)-(8) has to be equipped with boundary conditions. As they are different for different test cases, we comment on them in the numerical results section.

## II.B. Semi-Discrete Formulation

We assume that an appropriate triangulation of the domain  $\Omega$  into  $N$  elements is given by

$$\Omega = \bigcup_{k=1}^N \Omega_k. \quad (9)$$

For the discretization, one needs the edges  $e_k$  of the elements. The set  $\Gamma$  contains all edges of two intersecting elements and elements intersecting the physical boundary  $\partial\Omega$ . The number of edges in  $\Gamma$  is given by  $\hat{N} := |\Gamma|$ .

Now, the idea of the hybridized DG method is to introduce an additional unknown  $\lambda$  that exists only on the edges  $e_k \in \Gamma$  such that  $\lambda := w|_{\Gamma}$ . Then, the vector of unknowns is denoted by

$$\mathbf{w} := (\sigma, w, \lambda). \quad (10)$$

The total number of unknowns is now larger than for the initial problem (6)-(8) where only  $\sigma$  and  $w$  need to be determined. However, after discretization, the resulting system of equations allows for the elimination of  $\sigma_h$  and  $w_h$  such that the system is only globally coupled in  $\lambda_h$ .

For the representation of the approximate solution  $\mathbf{w}_h := (\sigma_h, w_h, \lambda_h)$  we introduce the following function spaces

$$H_h := \{f \in L^2(\Omega) \mid f|_{\Omega_k} \in \Pi^p(\Omega_k) \forall k = 1, \dots, N\}^{2m} \quad (11)$$

$$V_h := \{f \in L^2(\Omega) \mid f|_{\Omega_k} \in \Pi^p(\Omega_k) \forall k = 1, \dots, N\}^m \quad (12)$$

$$M_h := \{f \in L^2(\Omega) \mid f|_{e_k} \in \Pi^p(e_k) \forall k = 1, \dots, \hat{N}, e_k \in \Gamma\}^m \quad (13)$$

with abbreviation  $\mathbb{X}_h := H_h \times V_h \times M_h$ . The space of polynomials of degree  $p$  on a certain domain is  $\Pi^p$  and  $m$  is the dimension of the problem. In our case  $m = 4$  as there are four unknowns in the vector  $w$ . To keep the notation short we denote integrals on elements and edges by

$$(h_1, h_2) := \sum_{k=1}^N \int_{\Omega_k} h_1 \cdot h_2 \, dx, \quad \langle h_1, h_2 \rangle_{\Gamma} := \sum_{k=1}^{\hat{N}} \int_{\Gamma_k} h_1 \cdot h_2 \, d\sigma, \quad \langle h_1, h_2 \rangle_{\partial\Omega_k} := \sum_{k=1}^{\hat{N}} \int_{\partial\Omega_k} h_1 \cdot h_2 \, d\sigma.$$

With these preliminaries, we can state the semi-discrete system obtained using the HDG discretization on the spatial part of the PDE: Find a  $\mathbf{w}_h(\cdot, t) \in \mathbb{X}_h$  such that for all  $t \in (0, T)$

$$(\sigma_h - \nabla w_h, \tau_h) - \langle \lambda_h - w_h^-, \tau_h^- \cdot n \rangle_{\partial\Omega_k} = 0 \quad \forall \tau_h \in H_h \quad (14)$$

$$((w_h)_t, \varphi_h) - (f(w_h) - f_v(w_h, \sigma_h), \nabla \varphi_h) + \langle (\hat{f} - \hat{f}_v) \cdot n, \varphi_h^- \rangle_{\partial\Omega_k} = 0 \quad \forall \varphi_h \in V_h \quad (15)$$

$$\langle \llbracket \hat{f} - \hat{f}_v \rrbracket \cdot n, \mu_h \rangle_{\Gamma} = 0 \quad \forall \mu_h \in M_h \quad (16)$$

is fulfilled with numerical fluxes

$$\hat{f} := f(\lambda_h) - \alpha(\lambda_h - w_h^-)n, \quad \hat{f}_v := f_v(\lambda_h, \sigma_h^-) + \beta(\lambda_h, w_h^-)n. \quad (17)$$

Here,  $\alpha$  and  $\beta$  are positive real parameters that are problem dependent. For the limiting cases  $f \equiv 0$ , there holds  $\alpha \equiv 0$ , while for  $f_v \equiv 0$ , there holds  $\beta \equiv 0$ .  $\alpha$  can be chosen in a similar fashion as the viscosity constant in a local Lax-Friedrichs / Rusanov flux. For simplicity however, we usually chose it to be constant. Note that the treatment of boundary conditions is 'hidden' in the definition of the numerical fluxes. We do not want to go into detail here, we choose the boundary treatment in such a way that, for the stationary case, it is adjoint consistent. For more details, refer to Schütz and May<sup>35</sup> or, for a more recent publication, to Woopen et al.<sup>36</sup>

Note that when we compute solutions to Euler's equations ( $f_v(w, \sigma) \equiv 0$ ), there is no need to approximate  $\sigma$ , and we neglect (14).

## II.C. Applying Time Integration

We now focus on how to apply the time integration to the semi-discrete formulation. For this we refer to the set of equations (14)-(16) as

$$\mathcal{T}((w_h)_t, \varphi_h) + \mathcal{N}(\mathbf{w}_h; \mathbf{x}_h) = 0, \quad \forall \mathbf{x}_h \in \mathbb{X}_h \quad (18)$$

where  $\mathbf{x}_h := (\tau_h, \varphi_h, \mu_h)$  represents the test functions,  $\mathcal{T}((w_h)_t, \varphi_h) = (0, ((w_h)_t, \varphi_h), 0)^T$  is a vector containing the time derivative of  $w_h$  and  $\mathcal{N}(\mathbf{w}_h; \mathbf{x}_h)$  represents the remaining terms. It is inherent to the problem that there does neither occur a time derivative of  $\sigma_h$  nor of  $\lambda_h$ . Therefore, the set of equations (18) constitutes a differential algebraic equation (DAE). Because of that, it is not possible to apply explicit time integration methods, but some implicit methods can be used. Common choices for HDG methods are backward differentiation formulas (BDF) and implicit Runge-Kutta methods, especially diagonally implicit Runge-Kutta (DIRK) methods.<sup>37</sup> DIRK methods offer better stability properties than BDF for orders of accuracy higher than two. The first to combine HDG methods with BDF and DIRK time integrators were Nguyen et al.<sup>18,12</sup>

DIRK schemes offer yet another benefit, as it is possible to use them in embedded form to adapt the time step. For this, a time integrator of lower order can be embedded into a DIRK method for error estimation. Using such a scheme, we compute the solution at a sequence of times  $0 = t^0 < t^1 < \dots < t^M = T$  using the time step  $\Delta t^n := t^{n+1} - t^n$ . We use  $n$  as the time index where  $n+1$  is the unknown state that has to be determined. The solution at  $t^{n+1}$  can be determined using a DIRK scheme by solving

$$\mathcal{T}(w_h^{n+1} - w_h^n, \varphi_h) + \Delta t^n \sum_{i=1}^r b_i \mathcal{N}(\mathbf{w}_h^{n,i}; \mathbf{x}_h) = 0, \quad \forall \mathbf{x}_h \in \mathbb{X}_h. \quad (19)$$

The number of stages of the DIRK method is denoted by  $r$ , while the intermediate solution of each stage  $i = 1, \dots, r$  is denoted by  $\mathbf{w}_h^{n,i}$ . It has to be determined by solving the following equation

$$\mathcal{T}(w_h^{n,i} - w_h^n, \varphi_h) + \Delta t^n \sum_{j=1}^i a_{ij} \mathcal{N}(\mathbf{w}_h^{n,j}; \mathbf{x}_h) = 0, \quad \forall \mathbf{x}_h \in \mathbb{X}_h \quad (20)$$

for each stage  $i$ . Once the intermediate solutions  $\mathbf{w}_h^{n,i}$  are known, computing  $\mathbf{w}_h^{n+1}$  from equation (19) is an explicit operation. The real coefficients  $b_i$  and  $a_{ij}$  depend on the DIRK scheme used. A common way to present these is in a Butcher tableau as given in Table 1. Besides  $b_i$ , embedded DIRK methods have an

$c_1$	$a_{11}$			
$c_2$	$a_{21}$	$a_{22}$		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$c_r$	$a_{r1}$	$\dots$	$\dots$	$a_{rr}$
	$b_1$	$\dots$	$\dots$	$b_r$
	$\hat{b}_1$	$\dots$	$\dots$	$\hat{b}_r$

Table 1. Butcher tableau of an embedded DIRK method.

additional row  $\hat{b}_i$  that is used to compute a second solution  $\hat{w}_h^{n+1}$  by replacing  $b_i$  in equation (19) by  $\hat{b}_i$ .

Using these two different solutions  $w_h^{n+1}$  and  $\hat{w}_h^{n+1}$ , the error done in the time step can be estimated by

$$e_h^n := \|w_h^{n+1} - \hat{w}_h^{n+1}\|_{L^2}. \quad (21)$$

The solution  $w_h^{n+1}$  is accepted only if the relation

$$e_h^n < \text{tol} \cdot \Delta t^n \quad (22)$$

holds true. This guarantees that the error at the end of the simulation is limited to  $\sum_{i=0}^{M-1} e_h^i < \text{tol} \cdot T$ . The tolerance  $\text{tol}$  has to be specified by the user. Independently of whether the solution  $w_h^{n+1}$  is accepted or not, the new time step size  $\Delta t$  is computed using the following formula proposed by Hairer and Wanner:<sup>23</sup>

$$\Delta t^{n+1} = \Delta t^n \cdot 0.9 \frac{2n_{\text{it,max}} + 1}{2n_{\text{it,max}} + n_{\text{it}}} \cdot \left( \frac{e_h^n}{\text{tol} \cdot \Delta t^n} \right)^{-\frac{1}{q-1}}. \quad (23)$$

The error estimation includes a safety factor depending on the number of Newton steps  $n_{\text{it}}$  needed when solving the system of equations and the maximum number of allowed Newton steps  $n_{\text{it,max}}$ . The last term ensures that the time step is decreased if the estimated error grows too large. Additionally, it controls the time step such that it is as large as possible without reaching the error bound (22). This minimizes the total number of time steps needed to reach  $T$ .  $q$  denotes the temporal accuracy of the DIRK scheme.

We currently use three different DIRK methods by Cash<sup>21</sup> ( $q = 3, r = 3$ ), Al-Rabeh<sup>22</sup> ( $q = 4, r = 4$ ) and Hairer and Wanner<sup>23</sup> ( $q = 4, r = 5$ ). All of these DIRK methods have a lower order method embedded and are  $A$ -stable. Therefore, the schemes are appropriate to solve stiff problems.

#### II.D. Shock-Capturing for Inviscid Flows Using an Artificial Viscosity Model

In order to apply the hybridized DG method to a wide range of problems, the method has to be capable of handling shocks. Let us briefly motivate the use of artificial viscosity: The Euler equations are given by

$$w_t + \nabla \cdot (f(w)) = 0. \quad (24)$$

The idea of shock-capturing is to add a viscous flux  $g$  to the equation,

$$w_t + \nabla \cdot (f(w) - g(\varepsilon, w, \nabla w)) = 0. \quad (25)$$

The simplest possible case (for a discussion of other cases, see the publication by Persson and Peraire<sup>33</sup>) is to set this viscous flux to

$$g := \varepsilon \nabla w \quad (26)$$

with an artificial viscosity function  $\varepsilon$  that is supposed to measure smoothness of the underlying solution  $w$ . For the definition of  $\varepsilon$ , we use the model introduced by Persson and Peraire.<sup>33</sup> In order to stay 'as consistent as possible' with the original problem the viscosity shall only be added near discontinuities. Since we use a finite element method, we can express the solution on cell  $\Omega_k$  as

$$w|_{\Omega_k}(x, t) = \sum_{i=1}^{N(p)} w_i^k(t) \varphi_i^k(x) \quad (27)$$

with orthogonal basis functions  $\varphi_i^k(x)$  and coefficients  $w_i^k(t)$ . The number of basis functions needed to have a complete approximation of order  $p$  is denoted by  $N(p)$ . (For example, for triangles in two dimensions, there holds  $N(p) = \frac{(p+1)(p+2)}{2}$ .) Additionally, the reconstructed solution using only the basis functions corresponding to the highest polynomial degree  $p$  is given by

$$\bar{w}|_{\Omega_k}(x, t) = \sum_{i=N(p-1)+1}^{N(p)} w_i^k(t) \varphi_i^k(x). \quad (28)$$

The idea behind this is that the polynomial approximation behaves similar to a Fourier expansion in such a way that  $\bar{w}|_{\Omega_k}$  will decay rapidly for smooth solutions compared to  $w$ . Therefore, the smoothness indicator<sup>33</sup> is defined as

$$S_k := \frac{(\bar{w}|_{\Omega_k}, \bar{w}|_{\Omega_k})_{L_2(\Omega_k)}}{(w|_{\Omega_k}, w|_{\Omega_k})_{L_2(\Omega_k)}} \quad (29)$$

where  $(\cdot, \cdot)_{L_2(\Omega_k)}$  is the standard  $L_2(\Omega_k)$  inner product. The actual amount of viscosity is then deduced from the following (smooth) function

$$\varepsilon_k = \begin{cases} 0 & , s_k < s_0 - \kappa_{\text{av}} \\ \frac{\varepsilon_0}{2} \left( 1 + \sin \left( \frac{\pi(s_k - s_0)}{2\kappa_{\text{av}}} \right) \right) & , s_0 - \kappa_{\text{av}} \leq s_k \leq s_0 + \kappa_{\text{av}} \\ \varepsilon_0 & , s_k > s_0 + \kappa_{\text{av}} \end{cases} \quad (30)$$

with  $s_k := \log_{10} S_k$ . The problem dependent parameters  $\varepsilon_0 \sim \frac{h}{p}$ ,  $s_0 \sim \log(p)$  and  $\kappa_{\text{av}}$  have to be chosen such that the shock is accurately approximated.<sup>33,38</sup>

Incorporating this artificial viscosity model is done by adding the term  $(\varepsilon \nabla w_h, \nabla \varphi_h)$  to (15), where  $\varepsilon$  denotes the collection of the  $\varepsilon_k$ . Obviously, this is not a consistent discretization of (25). However, it is not our intention to approximate (25) accurately, but only to obtain a stabilizing term. A similar choice, with a different artificial viscosity model, has been done by Hartmann.<sup>39</sup>

In this paper we use the representation of the density  $\rho$  as smoothness indicator.

## II.E. Mesh Adaptation

The shock-capturing procedure allows to approximate the shock within a few elements.<sup>33</sup> However, depending on the size of the element this may still lead to a large shock width. To lower the width in order to obtain a sharper shock profile, and therefore to obtain a more accurate solution, we adaptively refine the mesh at locations where a discontinuity in the density is present. (More precisely, we refine whenever our viscosity coefficient  $\varepsilon_k$  is positive.) Obviously, mesh refinement is more involved for unsteady simulations in comparison to steady ones, as the discontinuities are likely to move. Without coarsening, this would lead to a mesh that is refined every time the shock approaches coarse cells. Therefore, one usually obtains meshes for which the size is steadily increasing. In order to reduce the computational effort it is beneficial to coarsen the mesh once a discontinuity (or another problematic flow feature) has moved out of a cell.

For this, we allow to form polygons from triangles of the underlying mesh. By agglomerating cells when possible, and splitting them in the presence of a shock, we obtain a highly flexible algorithm. It has the potential of reducing the amount of work that has to be done for organizing the mesh compared to other techniques because we do not need to save the hierarchy of refined elements during the simulation. To find acceptable partitions when refining the polygons we use the graph partitioning tool Metis.<sup>40</sup> This may lead to strangely looking triangulations, but does minimize the number of edges contained in the mesh. The HDG method benefits from this as the number of globally coupled unknowns is linked to the number of edges. More precisely, we use Metis as follows: Based on a set of triangular elements  $\{\Omega_i\}$  and a number of partitions  $n_p$ , Metis returns a map  $i \mapsto m(i)$ ,  $m(i) \leq n_p$ , and we can generate a coarser mesh using the collocation

$$\Omega_j^H := \bigcup_{i:m(i)=j} \Omega_i. \quad (31)$$

This can yield a considerably coarser mesh at the price of having to deal with non-standard element shapes. An example mesh can be seen in Figure 1.

Since we focus on non-smooth problems, we use the applied artificial viscosity in each cell as indicator for refinement and coarsening procedures. Once a cell has artificial viscosity applied to it, it is marked for refinement. Cells without artificial viscosity are marked for coarsening with neighboring cells that may also be coarsened. Note that this is a rather heuristic criterion, and ongoing research investigates more suited adaptation criteria.

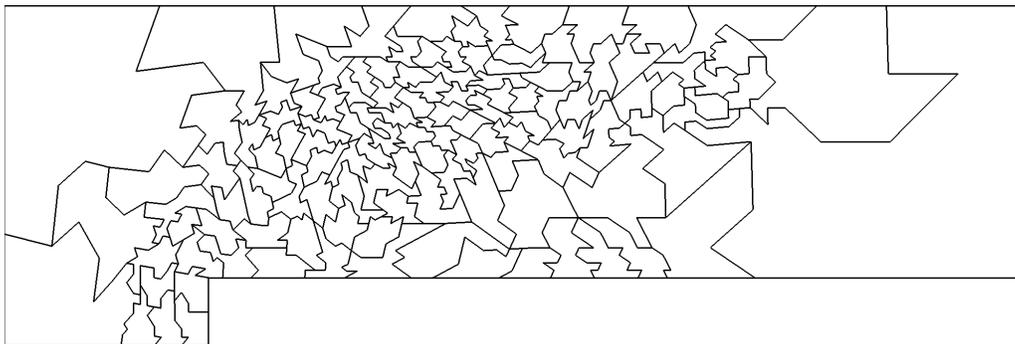


Figure 1. A mesh that was constructed by agglomerating elements using Metis. Note the varying geometrical shapes.

### III. Numerical Results

In this section, we present recent results of our simulations. We focus on two different classes of problems, namely problems with smooth and non-smooth solutions. For test cases with smooth solutions we are particularly interested in the usefulness of the time step adaptation. Preliminary work on that particular problem has been done for simple test cases,<sup>20</sup> however, we focus on more complex problems in this work. For inviscid problems with shocks we are both interested in the behavior of the shock capturing algorithm and again the time step adaptation.

#### III.A. NACA0012 Airfoil Tandem

This example has been motivated by case C2.1 of both first and second *International Workshop on High-Order CFD Methods*.<sup>41</sup> Two NACA0012 airfoils are positioned behind each other. The first airfoil is translated and rotated slightly such that for the given flow conditions –  $\text{Re} = 10^4$ ,  $\text{Ma} = 0.2$ ,  $\text{Pr} = 0.72$  and  $\gamma = 1.4$  – vortices begin to shed from it. These vortices travel towards the second airfoil where they dissipate. The computational domain is a circle with radius of 1000 chords centered at the mid chord of the first airfoil. We apply characteristics-based boundary conditions in the far field, and no-slip wall boundary conditions at the surface of the airfoils. In Figure 2(a) we show an excerpt of the mesh. It consists of  $N = 3\,335$  elements in total and has been refined around the airfoils such that the size of each element in the depicted area is smaller than 6% of the chord length. We use Legendre polynomials of order  $p = 2$  as basis functions. The solutions have been computed with an adaptive time step with bounds  $5 \cdot 10^{-3} \leq \Delta t^n \leq 1.0$ , tolerance  $\text{tol} = 5 \cdot 10^{-2}$  and  $n_{\text{it,max}} = 10$ .

In Figures 2 and 3 we show Mach number plots and the evolution of the time step. We note that for Figure 2, we use Cash’s DIRK scheme. The time interval has been chosen such that an incoming vortex at the second airfoil forces a reduced time step size. The first Figure 2(b) shows the vortex approaching which induces a smaller time step size. However, it is not reduced fast enough such that time steps are rejected when the vortex hits the second airfoil (Figure 2(e)). The time step size while the vortex is dissipating (Figure 2(d)) is very close to the lower bound. Shortly after, the time step size is increased until the next vortex reaches the second airfoil.

In the second case, the DIRK method of Al-Rabeh has been used as time integration method. While a vortex comes close to the second airfoil (Figure 3(a) and 3(b)) the time step is only slightly reduced (Figure 3(d)). Therefore, the time step size is way too large when the vortex reaches the second airfoil. In order to get to an acceptable time step size seven computed time steps have to be rejected. After that the time step size reaches the lower bound such that the eighth time step must be accepted. After the vortex is dissipated the time step only increases slowly because there are already new vortices approaching the second airfoil. However, when no vortex is near the second airfoil the time step size is much larger than it is for Cash’s DIRK method (Figure 2(e) and 3(d)).

Overall, the time step adaptation for smooth flows works nicely. The DIRK methods of Cash and Al-Rabeh are able to detect the approaching vortex. While both tend to neglect time steps when a vortex starts dissipating Al-Rabeh’s DIRK method uses larger time steps at times where no vortex dissipates.

#### III.B. Unsteady Flows with Shocks

##### III.B.1. Sod’s Shock Tube

For flows with shocks the solver is tested using Sod’s shock tube problem.<sup>42</sup> This Euler problem is actually a one-dimensional Riemann problem on  $[0, 1]$ , however, we solve it in the two-dimensional domain  $[0, 1] \times [0, \frac{2}{N}]$  with  $N$  being the number of elements of the mesh (see Figure 4 for initial mesh). The initial states  $w_l$  for  $x_1 < 0.5$  and  $w_r$  for  $x_1 \geq 0.5$  are shown in Figure 5(a). From this a flow develops that can be divided into five regions for the density  $\rho$  as shown in Figure 5(b). In region I the initial state  $w_l$  is still present. From this starts an expansion wave (II) that is followed by a region of constant density III. Region III and IV are separated by a contact discontinuity leading to a jump in the density and energy while the velocity and pressure stay constant. Finally between region IV and V a shock occurs.

The solution is determined on a rectangular domain with slip wall boundary conditions everywhere. In Figure 6 results for a mesh containing 100 elements and polynomials of degree  $p = 2$  are presented. For time integration the BDF2 and DIRK methods have been applied with constant time step size  $\Delta t = 5 \cdot 10^{-4}$  for all methods. The parameters of the shock-capturing method are set to  $\varepsilon_0 = 0.45 \frac{h}{p}$ ,  $s_0 = -14 \log(p)$  and

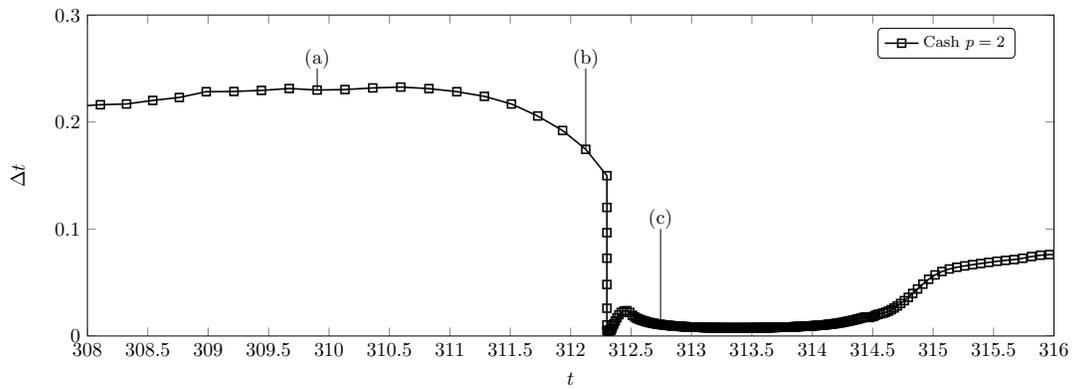
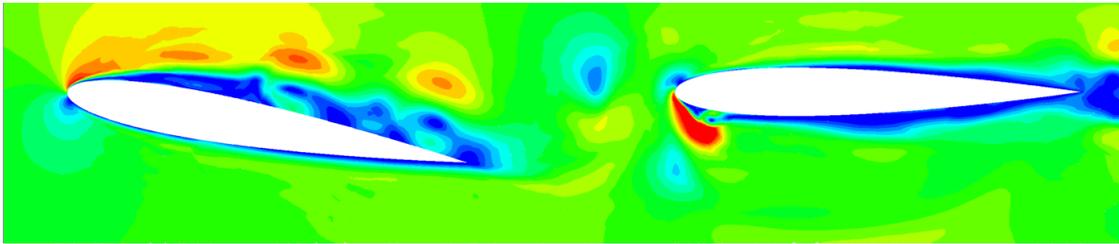
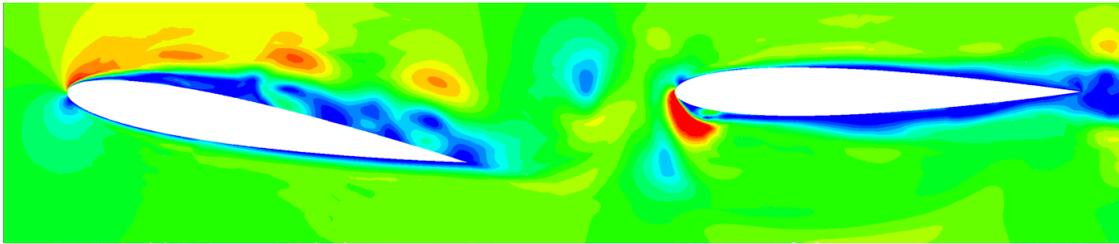
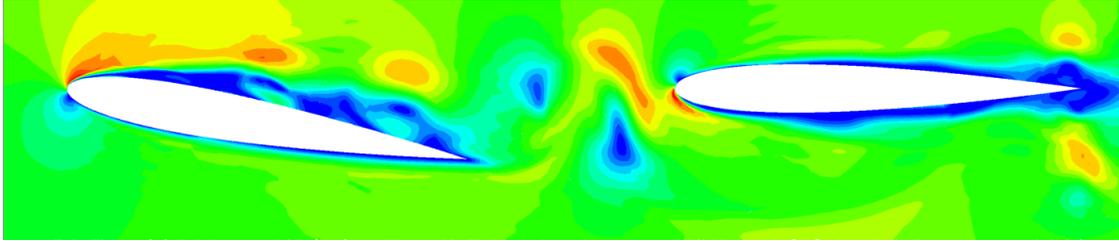
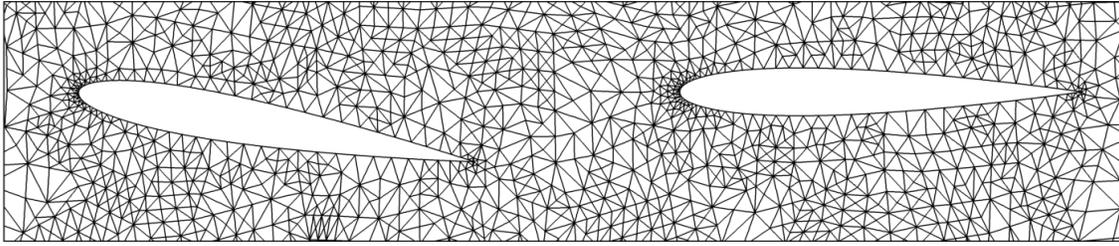
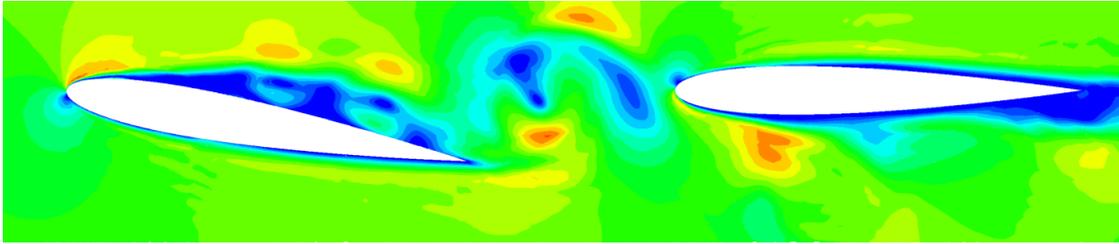
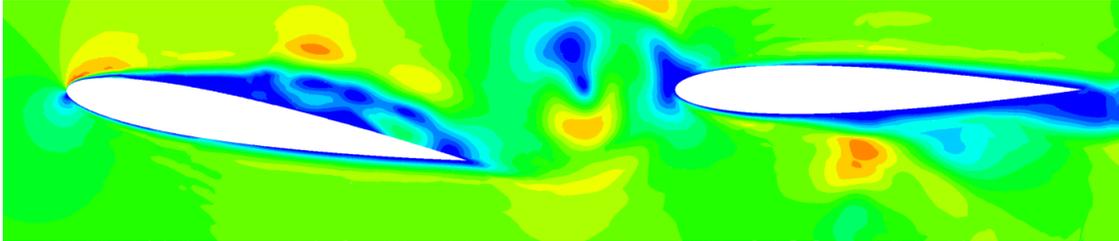


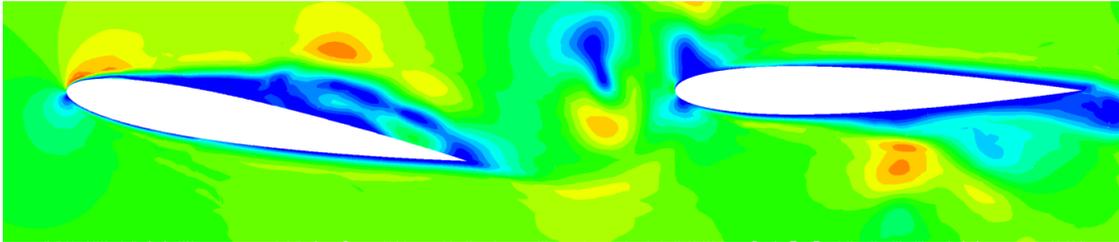
Figure 2. NACA0012 airfoil tandem: The Mach number distribution of the flow and the time step evolution using Cash's DIRK method.



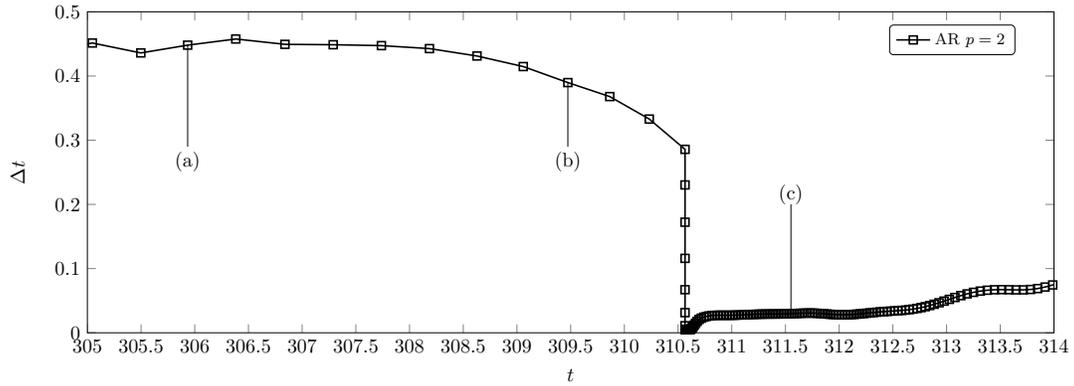
(a) Plot of Mach number with vortex right in front of the second airfoil.



(b) Plot of Mach number with vortex starting to dissipate at the second airfoil causing the time step size  $\Delta t$  to decrease.



(c) Plot of Mach number with vortex dissipating at the second airfoil.



(d) Evolution of the time step size  $\Delta t$  with the displayed scenes noted at the graph.

**Figure 3. NACA0012 airfoil tandem: The Mach number distribution of the flow and the time step evolution using Al-Rabeh's DIRK method.**



**Figure 4. Sod's shock tube: Mesh used for approximating Sod's shock tube problem.**

$\kappa_{av} = 0.4$ . The artificial viscosity  $\varepsilon$  is recomputed after each Newton step based on the updated solution. All results of Sod's shock tube problem have been extracted from the horizontal centerline of the mesh.

For all methods, we obtained reasonable results. There is a slight smoothing effect at the head and tail of the expansion wave. The largest amount of viscosity is applied at the shock location. This leads to a smooth, but still fairly sharp shock profile. We want to note that in our case the artificial viscosity has jumps, because we have extracted the data from the centerline. At the location of the contact discontinuity, there is also a small amount of viscosity added. It can be observed that the contact discontinuity gets smoothed much stronger than the shock itself.

The results for time integration methods are almost identical. This indicates that the accuracy in time of all methods is high enough to adequately discretize the problem and that the methods have minimal effect on the efficiency of the shock-capturing scheme.

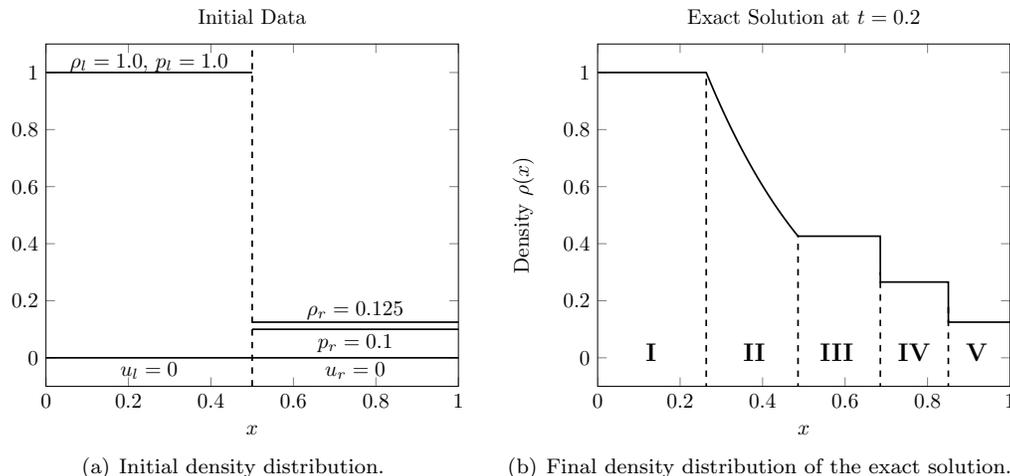


Figure 5. Initial data and solution of Sod's shock tube problem.

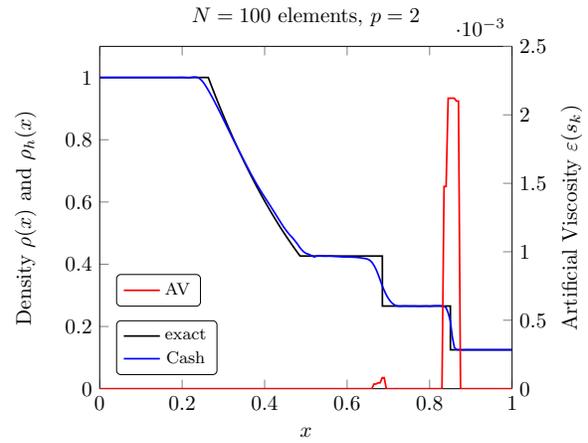
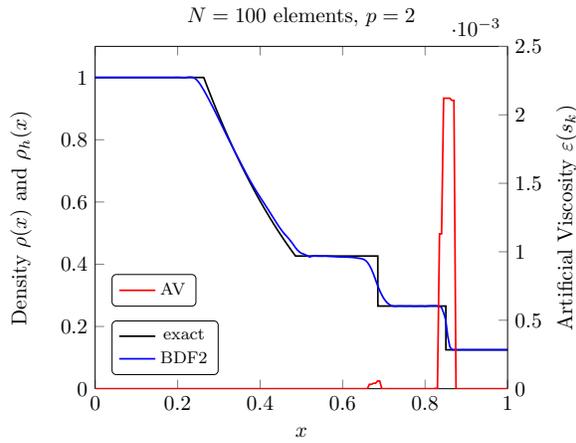
The time step adaptation of the DIRK methods can also be combined with the shock-capturing scheme to get a time adaptive scheme. Results are given in Figure 7. For this simulations the tolerance is set to  $\text{tol} = 10^{-1}$  and we specify the bounds  $1 \cdot 10^{-4} \leq \Delta t \leq 5 \cdot 10^{-3}$  for the time step size. However, these bounds are never reached, except for the first time step, because we set  $\Delta t^0 = 1 \cdot 10^{-4}$ .

Again, the differences in the solution of the different DIRK methods is minimal (Figure 7(a) - 7(c)). However, this time the amount of artificial viscosity applied differs significantly. For Cash's and Al-Rabeh's methods artificial viscosity is added near the contact discontinuity, but not for Hairer's and Wanner's method. This is in contrast to the previous case of equal time step sizes for all methods. Nonetheless, this does not seem to have a serious effect on the obtained solution.

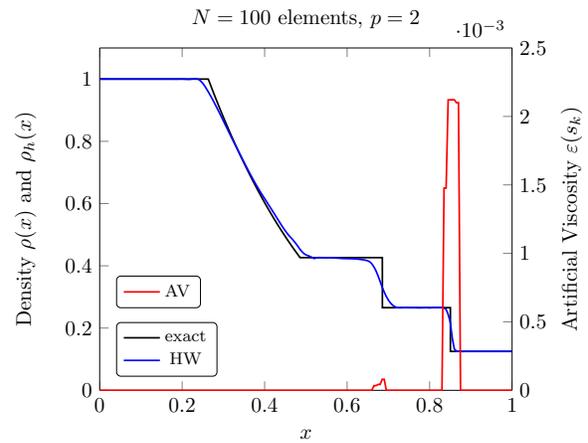
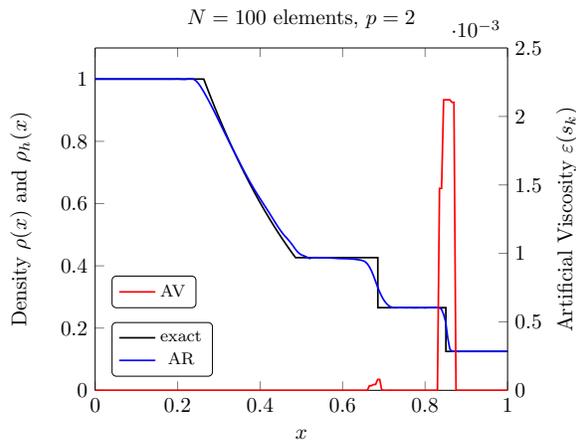
The evolution of the time step size in Figure 7(d) are fairly similar for all DIRK methods. Hairer's and Wanner's method tends to use the largest time steps and Cash's method tends to use the smallest time steps. This is most likely caused by the order of accuracy of the methods, as Hairer's and Wanner's method is of fourth order consistent in time while Cash's method is of third order consistent in time. Nevertheless, all methods struggle at the same points in time where the time step has to be reduced. For all methods this may lead to rejected time steps.

As described in Section II.E we also want to utilize mesh adaptation in order to sustain a satisfactory approximation of the solution. By dynamically refining and coarsening the mesh it is possible to minimize the number of elements and thereby the number of degrees of freedoms. This is done using constructed polygons from triangles of an underlying – fine – mesh that also prescribes the finest level of refinement. Since we obtained reasonable results with a mesh consisting of  $N = 100$  elements, we start with this one. The polygons are now constructed by merging adjacent triangles.

We have investigated cases for which a polygon can consist of up to 5 (Figure 8) and up to 10 (Figure 9) triangles. The initial partitions are given in Figures 8(a) and 9(a), also with the underlying triangular mesh depicted in red in Figures 8(b) and 9(b). A polygon is split into four smaller polygons once artificial viscosity is applied to it. In the case a partition contains less than four elements, the polygon is split into the remaining triangles. A polygon is merged with its neighboring polygons if no artificial viscosity is applied

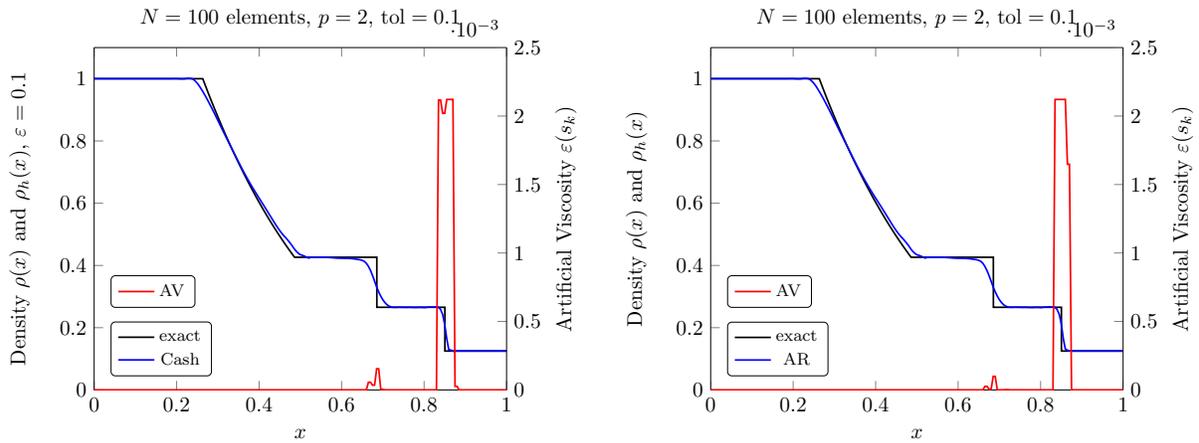


(a) Final density distribution using BDF2 for time integration. (b) Final density distribution using Cash's DIRK method for time integration.

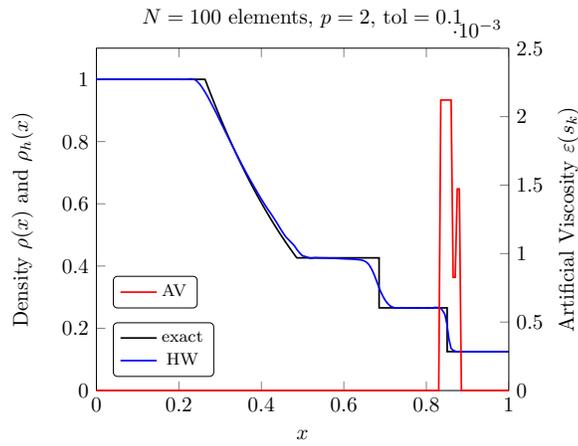


(c) Final density distribution using Al-Rabeh's DIRK method (d) Final density distribution using Hairer's and Wanner's DIRK method for time integration.

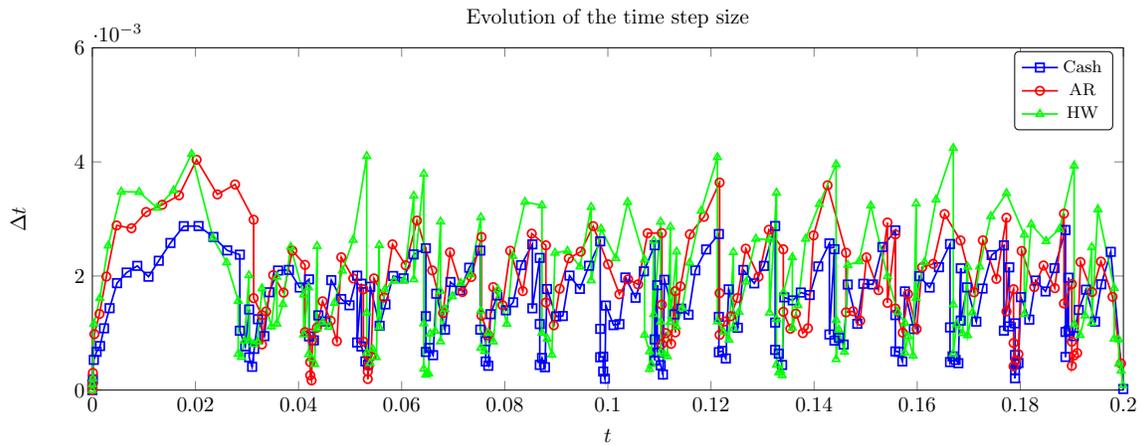
**Figure 6. Sod's shock tube: Comparison of the final density with the exact solution for different time integration methods and polynomial basis functions of order  $p = 2$ .**



(a) Final density distribution using Cash's DIRK method for time integration *with* time step adaptation. (b) Final density distribution using Al-Rabeh's DIRK method for time integration *with* time step adaptation.



(c) Final density distribution using Hairer's and Wanner's DIRK method for time integration *with* time step adaptation.



(d) Evolution of the time step size  $\Delta t$  of the DIRK methods.

Figure 7. Sod's shock tube: Comparison of the final density with the exact solution for different time integration methods and polynomial basis functions of order  $p = 2$ .

to the polygons and if the number of triangles contained by the new polygon is smaller than the initially specified bound, 5 or 10 in our case.

It can be observed in Figure 8(c) that the solution compares very well to the one with  $N = 100$ , though using considerably fewer elements (only 36 at  $t = 0.2$ ).

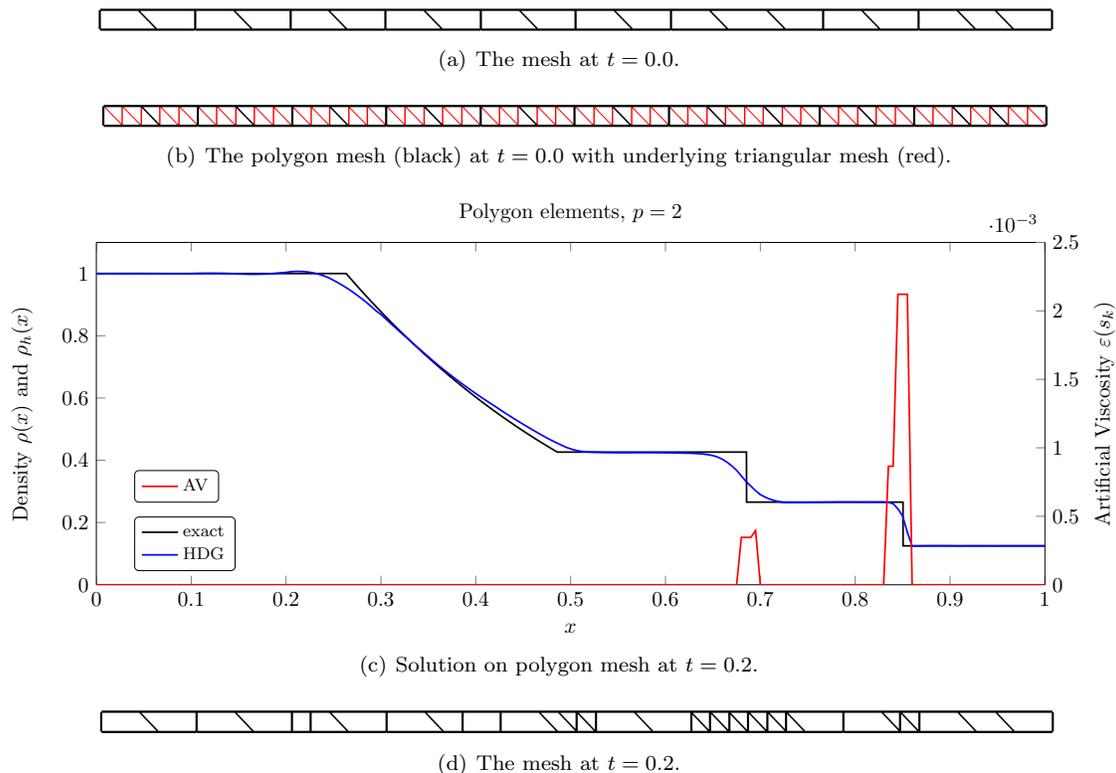


Figure 8. Sod's shock tube: Polygon mesh.

### III.B.2. Mach 3 Flow over Forward Facing Step

This test case originates from the work of Woodward and Colella<sup>43</sup> that compared different shock-capturing schemes. A channel with a forward facing step is initialized with a supersonic  $\text{Ma} = 3$  flow from which a curved shock develops. During the simulation the shock is reflected several times at the channel walls. In contrast to Sod's shock tube problem, the initial data is smooth. It is a very challenging test case, not only because of the evolving shock and its reflections, but also because the corner of the step is a singular point. In the original paper<sup>43</sup> a correction at the corner was suggested. We note that our simulation is performed without any correction.

In Figure 10 we present a solution obtained at time  $t = 0.5$  using polynomials of degree  $p = 2$ . The simulation was started with a coarse mesh containing only  $N = 120$  elements. Then, the elements where artificial viscosity is added are refined as long as their (and their neighbor's) size is larger than  $h = 0.04$ . This leads to a mesh containing  $N = 3\,654$  elements (Figure 10(a)). BDF2 is used for time-stepping with time step size  $\Delta t = 1 \cdot 10^{-4}$ . The parameter for the shock-capturing are set to  $\kappa_{\text{av}} = 0.4$ ,  $s_0 = -11.2 \cdot \log(p)$  and  $\varepsilon_0 = 0.5 \frac{h}{p}$ . No polygons and therefore no mesh coarsening is used in this test case.

The density  $\rho$  and the applied artificial viscosity in each element are displayed in Figure 10(b) and Figure 10(c). The shock forms an arc and is well-resolved with the current mesh. The width of the shock is about the size of one element. Few oscillations occur near the shock that may be diminished by a finer mesh. Viscosity is added not only at the shock, but also near the corner of the step which successfully stabilizes the method. However, the number of elements with artificial viscosity is very low compared to the total number of elements. This makes clear that an adaptive coarsening is advantageous in this test case.

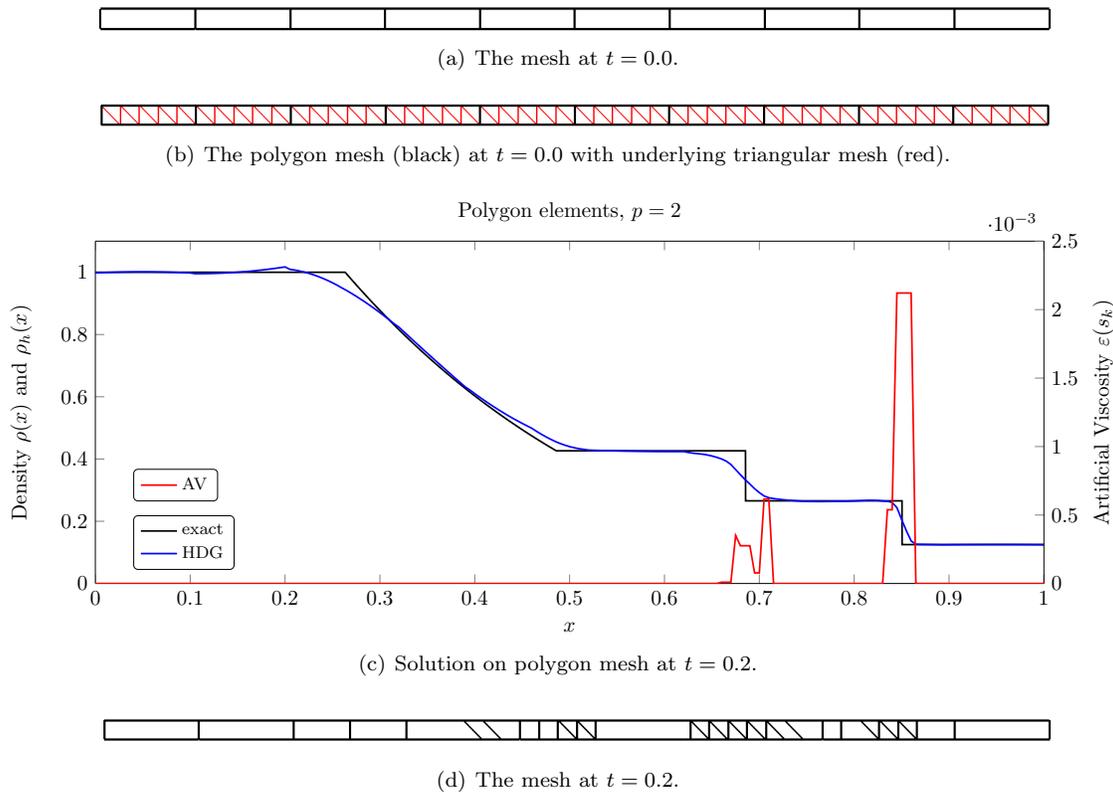


Figure 9. Sod's shock tube: Polygon mesh.

## IV. Conclusion and Outlook

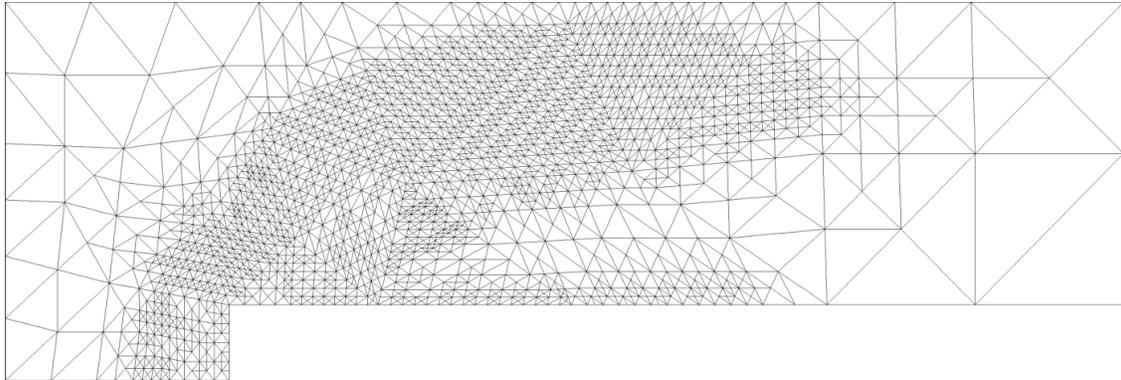
We have presented a hybridized DG method for compressible unsteady flows. It has been shown that an adaptive time discretization for complicated problems can be beneficial. This can be easily achieved using embedded DIRK methods. The investigated methods deliver very similar results such that it is not possible to identify a scheme that is superior.

The artificial viscosity model by Persson and Peraire operates very well with the HDG method. While the shock profiles remain sharp the method is still sufficiently stabilized for the considered problems. All time integration methods work with the viscosity model and even time step adaptation is still applicable.

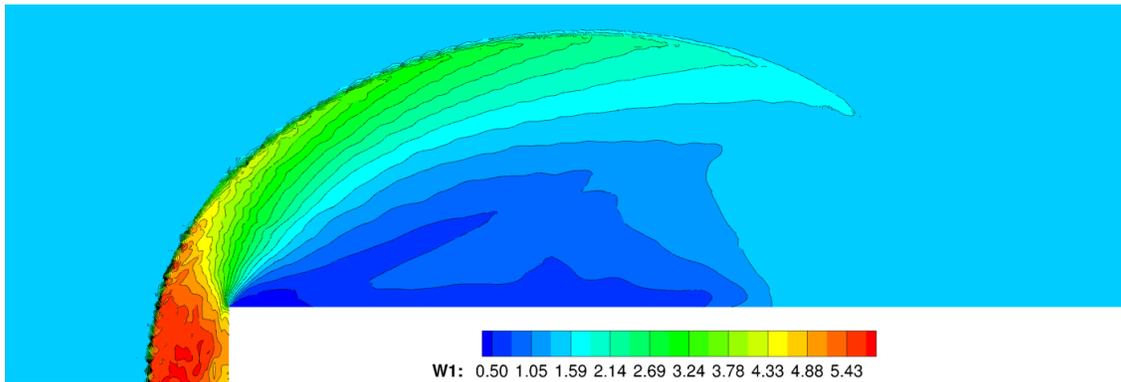
The obtained results underline the importance of an adaptive mesh refinement and coarsening in order to get sharp shock profiles while limiting the amount of computational work. For this, we plan to extend the approach using polygons also for more complicated test cases as the Mach 3 step and adaptive time integration. This includes efficient mapping functions to transfer the solution between different meshes and integration formulas, and is subject to ongoing work.

## References

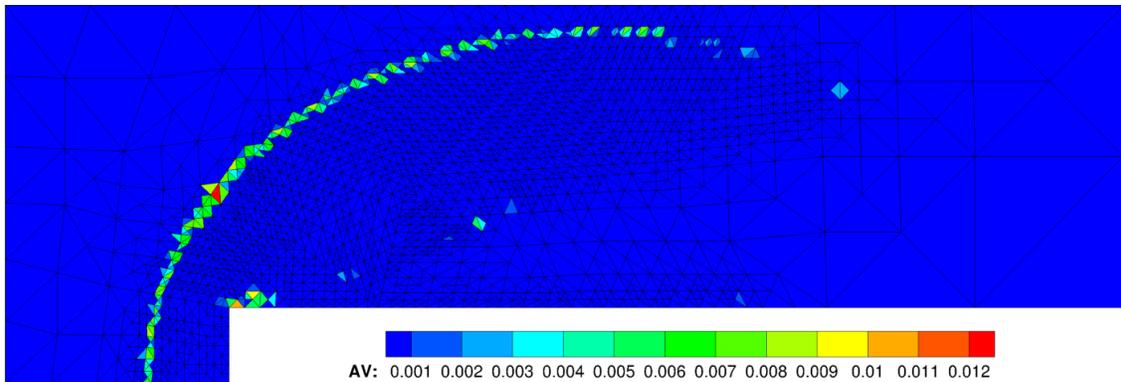
- <sup>1</sup>Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for elliptic problems," *SIAM Journal of Numerical Analysis*, Vol. 39, 2002, pp. 1749–1779.
- <sup>2</sup>Bassi, F. and Rebay, S., "A High-Order Accurate Discontinuous Finite-Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 131, 1997, pp. 267–279.
- <sup>3</sup>Houston, P. and Süli, E., "hp-Adaptive Discontinuous Galerkin Finite Element Methods for First Order Hyperbolic Problems," *SIAM Journal on Scientific Computing*, Vol. 23, 2001, pp. 1226–1252.
- <sup>4</sup>Hartmann, R. and Houston, P., "Adaptive Discontinuous Galerkin Finite Element Methods for the Compressible Euler Equations," *Journal of Computational Physics*, Vol. 183, 2002, pp. 508–532.
- <sup>5</sup>Hartmann, R. and Houston, P., "Symmetric Interior Penalty DG Methods for the Compressible Navier–Stokes Equations I: Method Formulation," *International Journal of Numerical Analysis and Modeling*, Vol. 3, No. 1, 2006, pp. 1–20.
- <sup>6</sup>Hartmann, R. and Houston, P., "Symmetric Interior Penalty DG Methods for the Compressible Navier–Stokes Equations



(a) Closeup of the used mesh.



(b) Contour plot of the density  $\rho$ .



(c) Plot denoting the amount of artificial viscosity applied to each element.

Figure 10. Mach 3 Step: Plots of the mesh, density  $\rho$  and artificial viscosity at  $t = 0.5$ .

II: Goal-Oriented A Posteriori Error Estimation,” *International Journal of Numerical Analysis and Modeling*, Vol. 3, No. 2, 2006, pp. 141–162.

<sup>7</sup>Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., “p-Multigrid solution of high-order Discontinuous Galerkin discretizations of the compressible Navier–Stokes equations,” *Journal of Computational Physics*, Vol. 207, 2005, pp. 92–113.

<sup>8</sup>Fidkowski, K., “Output Error Estimation Strategies for Discontinuous Galerkin Discretizations of Unsteady Convection-Dominated Flows,” *International Journal for Numerical Methods in Engineering*, Vol. 88, 2011, pp. 1297–1322.

<sup>9</sup>Nguyen, N. C., Peraire, J., and Cockburn, B., “An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations,” *Journal of Computational Physics*, Vol. 228, No. 9, 5 2009, pp. 3232–3254.

<sup>10</sup>Nguyen, N. C., Peraire, J., and Cockburn, B., “An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations,” *Journal of Computational Physics*, Vol. 228, No. 23, 12 2009, pp. 8841–8855.

<sup>11</sup>Peraire, J., Nguyen, N. C., and Cockburn, B., “A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations,” AIAA Paper 2010-362, 48th AIAA Aerospace Sciences Meeting and Exhibit, 2010.

<sup>12</sup>Nguyen, N. C. and Peraire, J., “Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics,” *Journal of Computational Physics*, Vol. 231, 2012, pp. 5955–5988.

<sup>13</sup>Cockburn, B., Gopalakrishnan, J., and Lazarov, R., “Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems,” *SIAM Journal on Numerical Analysis*, Vol. 47, No. 2, 01 2009, pp. 1319–1365.

<sup>14</sup>Egger, H. and Schöberl, J., “A hybrid mixed discontinuous Galerkin finite-element method for convection–diffusion problems,” *IMA Journal of Numerical Analysis*, Vol. 30, No. 4, 10 2010, pp. 1206–1234.

<sup>15</sup>Schütz, J. and May, G., “A Hybrid Mixed Method for the Compressible Navier-Stokes Equations,” *Journal of Computational Physics*, Vol. 240, 2013, pp. 58–75.

<sup>16</sup>D.N.Arnold and F.Brezzi, “Mixed and nonconforming Finite Element methods: Implementation, postprocessing and error estimates,” *RAIRO Model. Math. Anal. Numer.*, Vol. 19, 1985, pp. 7–32.

<sup>17</sup>Brezzi, F. and Fortin, M., “Mixed and Hybrid Finite Element Methods,” 1991.

<sup>18</sup>Nguyen, N. C., Peraire, J., and Cockburn, B., “High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics,” *Journal of Computational Physics*, Vol. 230, 2011, pp. 36953718.

<sup>19</sup>Schütz, J., Wopen, M., and May, G., “A Combined Hybridized Discontinuous Galerkin / Hybrid Mixed Method for Viscous Conservation Laws,” *Hyperbolic Problems: Theory, Numerics, Applications*, edited by F. Ancona, A. Bressan, P. Marcati, and A. Marson, American Institute of Mathematical Sciences, 2012, pp. 915–922.

<sup>20</sup>Jaust, A. and Schütz, J., “A temporally adaptive hybridized discontinuous Galerkin method for instationary compressible flows,” *Computers and Fluids*, Vol. 98, 2014, pp. 177–185.

<sup>21</sup>Cash, J. R., “Diagonally Implicit Runge-Kutta Formulae with Error Estimates,” *Journal of the Institute of Mathematics and its Applications*, Vol. 24, 1979, pp. 293–301.

<sup>22</sup>Al-Rabeh, A. H., “Embedded DIRK methods for the numerical integration of stiff systems of ODEs,” *International Journal for Computer Mathematicss*, Vol. 21:1, 1987, pp. 65–84.

<sup>23</sup>Hairer, E. and Wanner, G., *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin Heidelberg, 1991.

<sup>24</sup>van Leer, B., “Towards the ultimate conservative difference scheme I. The quest of monotonicity,” *Lecture Notes in Physics*, Vol. 18, Springer Berlin / Heidelberg, 1973, pp. 163–168.

<sup>25</sup>van Leer, B., “Towards the Ultimate Conservative Difference Scheme IV: A New Approach to Numerical Convection,” *Journal of Computational Physics*, Vol. 23, 1977, pp. 276–299.

<sup>26</sup>Sweby, P. K., “High resolution schemes using flux-limiters for hyperbolic conservation laws,” *SIAM Journal on Numerical Analysis*, Vol. 21, 1984, pp. 995–1011.

<sup>27</sup>Persson, P.-O. and Peraire, J., “Sub-Cell Shock Capturing for Discontinuous Galerkin Methods,” *AIAA Paper 2006-112*, 2006.

<sup>28</sup>Barter, G. and Darmofal, D. L., “Shock capturing with higher-order, PDE-based artificial viscosity,” *AIAA Paper 2007-3823*, 2007.

<sup>29</sup>Guermond, J.-L., Pasquetti, R., and Popov, B., “Entropy viscosity method for nonlinear Conservation Laws,” *Journal of Computational Physics*, Vol. 230, 2010, pp. 42484267.

<sup>30</sup>Nguyen, N. C. and Peraire, J., “An Adaptive Shock-Capturing HDG Method for Compressible Flows,” AIAA Paper 2011-3060, 20<sup>th</sup> AIAA Computational Fluid Dynamics Conference, 2011.

<sup>31</sup>Brooks, A. N. and Hughes, T. J. R., “Streamline Upwind/Petrov-Galerkin formulations for convection-dominated flows with particular emphasis on the incompressible Navier-Stokes equations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, 1982, pp. 199–259.

<sup>32</sup>Von Neumann, J. and Richtmyer, R., “A Method for the Numerical Calculation of Hydrodynamic Shocks,” *Journal of Applied Physics*, Vol. 21, 1950, pp. 232–237.

<sup>33</sup>Persson, P.-O. and Peraire, J., “Sub-Cell Shock Capturing for Discontinuous Galerkin Methods,” AIAA Paper 06-0112, American Institute of Aeronautics and Astronautics, 2006.

<sup>34</sup>Cangiani, A., Georgoulis, E. H., and Houston, P., “hp-version Discontinuous Galerkin methods on polygonal and polyhedral meshes,” *Mathematical Models and Methods in Applied Science*, 2014.

<sup>35</sup>Schütz, J. and May, G., “An Adjoint Consistency Analysis for a Class of Hybrid Mixed methods,” *IMA Journal of Numerical Analysis*, 2013, doi: 10.1093/imanum/drt036.

<sup>36</sup>Wopen, M., Balan, A., May, G., and Schütz, J., “A Comparison of Hybridized and Standard DG Methods for Target-Based hp-Adaptive Simulation of Compressible Flow,” *Computers and Fluids*, Vol. 98, 2014, pp. 3–16.

<sup>37</sup>Alexander, R., “Diagonally Implicit Runge-Kutta Methods for Stiff O.D.E.’s,” *SIAM Journal on Numerical Analysis*, Vol. 14, No. 14, 1977, pp. 1006–1021.

<sup>38</sup>Nguyen, N. C., Persson, P.-O., and Peraire, J., “RANS solutions using high order discontinuous Galerkin methods,” *AIAA Paper*, Vol. 914, 2007, pp. 2007.

<sup>39</sup>Hartmann, R., “Adaptive Discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations,” *International Journal for Numerical Methods in Fluids*, Vol. 51, 2006, pp. 1131–1156.

<sup>40</sup>Karypis, G. and Kumar, V., “A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs,” *SIAM Journal on Scientific Computing*, Vol. 20, 1999, pp. 359–392.

<sup>41</sup>Wang, Z. J., Fidkowski, K., Abrall, R., Bassi, F., Caraeni, D., Cary, A., Deconick, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., “High-Order CFD Methods: Current Status and Perspective,” *International Journal for Numerical Methods in Fluids*, Vol. 00, 2012, pp. 1–42.

<sup>42</sup>Sod, G. A., “A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws,” *Journal of Computational Physics*, Vol. 27, 1978, pp. 1–31.

<sup>43</sup>Woodward, P. and Colella, P., “The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks,” *Journal of Computational Physics*, Vol. 54, 1984, pp. 115–173.