UHasselt Computational Mathematics Preprint Series

Jochen Schütz and Vadym Aizinger

A hierarchical scale separation approach for the hybridized discontinuous Galerkin method

UHasselt Computational Mathematics Preprint Nr. UP-16-01

January 1, 2016

A hierarchical scale separation approach for the hybridized discontinuous Galerkin method

Jochen Schütz, Vadym Aizinger

November 26, 2015

In this work, the hierarchical scale separation (HSS) method developed for linear systems resulting from discontinuous Galerkin (DG) discretizations has been extended to hybridized discontinuous Galerkin (HDG) schemes. The HSS method is related to p-multigrid techniques for DG systems but is conceptually much simpler. Our extension of the HSS scheme to the HDG method tested using a convection-diffusion equation for a range of benchmark problems demonstrated excellent performance on a par with that of the HSS method for a non-hybridized DG approximation. In the limiting case of a pure convection equation, the measured convergence rate of the HSS scheme was significantly better than the rates demonstrated in the non-hybridized case.

1 Introduction

Discontinuous Galerkin (DG) methods [2] combine the most attractive features of the finite volume (local conservation, robustness for advection-dominated problems, shock-capturing features, etc.) and the finite element (high order discretizations, Galerkin formulation, etc.) methods. However, the price is a computationally more expensive scheme, partly due to a significantly greater number of degrees of freedom than a finite element discretization of equal order, partly caused by more expensive evaluations of element and boundary integrals. This performance disadvantage has even more impact if a linear equation system resulting from a DG discretization must be solved as a part of the solution algorithm – e.g., in an implicit time-stepping scheme.

A number of recent publications, see, e.g., [6, 7, 8, 11, 13, 12, 15], introduced the hybridizable discontinuous Galerkin (HDG) methods aiming to address this drawback. The main idea of the HDG method lies in the introduction of an approximation space for traces of primary unknowns on element boundaries, relying on classical ideas in the context of hybrid mixed methods, see, e.g., [4, 5]. This approximation space on the mesh 'skeleton' constitutes a globally connected problem, whereas the original system unknowns are computed in a postprocessing step by solving element-local problems that utilize those traces. This approach substantially reduces the number of global degrees of freedom, lends itself readily to an efficient parallelization, and even sometimes results in better convergence rates than its non-hybridizable counterparts [6].

In order to speed up the linear system solves for DG discretizations, a hierarchical scale separation (HSS) approach has been introduced in [1] for a nonsymmetric interior penalty Galerkin (NIPG) method. The main idea of the HSS technique is related to the p-multigrid scheme proposed for DG in [9] that, in its turn, was motivated by a similar approach introduced for the spectral method in [14]. Contrary to the classical h-multigrid method that utilizes a mesh hierarchy to suppress the low wave number errors present in the finest-mesh solution, the p-multigrid method uses in the same role a hierarchy of approximation spaces on a fixed mesh (usually corresponding to DG spaces of different polynomial order). The hierarchical scale

separation (HSS) method goes even further and resorts to a two-space technique resulting in a much simpler algorithm and a more complete separation of fine- and coarse-scale solutions than the p-multigrid method. The coarse-scale problem has the computational structure of a cell-centered finite volume method and is solved globally. The fine-scale problems are computed as local corrections to the coarse-scale solution; thus producing a numerical algorithm highly suitable for an efficient parallel implementation. In [1], the HSS method demonstrated performance on a par or exceeding that of the p-multigrid method. The main promise of this type of method lies in a much reduced parallel communication overhead as compared to traditional linear solvers; this advantage is expected to become even more significant for massively parallel applications.

In the present paper, we extend the HSS paradigm to an HDG discretization of a convection-diffusion equation and demonstrate the performance of the method on a range of benchmark problems. The remainder of this paper is organized as follows. In the next section, we formulate the boundary value problem for a generic convection-diffusion equation and discretize it using an HDG method. Sec. 3 details the HSS algorithm. Some numerical examples illustrating the performance of the proposed method are presented in Sec. 4. Conclusions are offered in Sec. 5.

2 The hybridized discontinuous Galerkin method

In this work, we consider the two-dimensional linear convection-diffusion equation on a bounded domain Ω,

$$-\varepsilon\Delta u + \nabla\cdot(\mathbf{c}u) = f, \quad \forall \mathbf{x}\in\Omega, \qquad u = g, \quad \forall \mathbf{x}\in\partial\Omega_D \tag{1}$$

for a constant vector $\mathbf{c}\in\mathbb{R}^2$, a constant scalar $\varepsilon\in\mathbb{R}_+$ and functions $f\in L^2(\Omega)$, $g\in L^2(\partial\Omega)$. $\partial\Omega_D$ denotes the Dirichlet boundary which encompasses the whole domain boundary in the convection-diffusion case, i.e. $\partial\Omega_D=\partial\Omega$ if $\varepsilon>0$, or is limited to the inflow part of the external boundary $\partial\Omega_D=\{\mathbf{x}\in\partial\Omega \mid \mathbf{c}\cdot\mathbf{n}\le 0\}$ for the pure convection $\varepsilon=0$, where $\mathbf{n}$ denotes an an exterior unit normal to $\partial\Omega$.

As is frequently done for DG-type methods, we rewrite this equation as a first order system,

$$\boldsymbol{\sigma} = \nabla u, \qquad \nabla\cdot(\mathbf{c}u - \varepsilon\boldsymbol{\sigma}) = f, \quad \forall\mathbf{x}\in\Omega.$$

In order to introduce the HDG method properly, we need to define the approximation spaces. V_h is the standard DG space, defined on a triangulation $\Omega=\bigcup_{k=1}^{\mathtt{ne}}\Omega_k$ as

$$V_h := \{f\in L^2(\Omega) \mid f_{|\Omega_k}\in\Pi^p(\Omega_k),\ k=1,\ldots,\mathtt{ne}\},$$

where Π^p is the space of polynomials of total degree at most p. Approximating the unknown u on the skeleton of the mesh necessitates the introduction of yet another space, the so-called hybrid ansatz space. So let Γ denote the set of all edges of the mesh, and let $\Gamma=\bigcup_{k=1}^{\mathtt{nf}}\Gamma_k$. (Every edge only occurs once, independent of whether it is a boundary edge or not.) Then, M_h is defined by

$$M_h := \{f\in L^2(\Gamma) \mid f_{|\Gamma_k}\in\Pi^p(\Gamma_k),\ k=1,\ldots,\mathtt{nf}\}.$$

For a point $\mathbf{x}\in\partial\Omega_k$, we define the one-sided values of a scalar quantity $w=w(\mathbf{x})$ by

$$w^-(\mathbf{x}) := \lim_{\varepsilon\to 0^+} w(\mathbf{x}-\varepsilon\,\mathbf{n}) \qquad \text{and} \qquad w^+(\mathbf{x}) := \lim_{\varepsilon\to 0^+} w(\mathbf{x}+\varepsilon\,\mathbf{n})\,,$$

respectively. (Obviously, along the physical boundary of the domain, only the first expression is defined.) The one-sided values of a vector-valued quantity $\mathbf{v}$ are defined analogously. Then using the standard DG notation, the *average* and the *jump* of w and $\mathbf{v}$ in $\mathbf{x}$ are given by

$$\{w\} := \frac{w^-+w^+}{2} \qquad \text{and} \qquad [\![w]\!] := w^-\mathbf{n} - w^+\mathbf{n}\,,$$
$$\{\mathbf{v}\} := \frac{\mathbf{v}^-+\mathbf{v}^+}{2} \qquad \text{and} \qquad [\![\mathbf{v}]\!] := \mathbf{v}^-\cdot\mathbf{n} - \mathbf{v}^+\cdot\mathbf{n}$$

respectively, where $\mathbf{n}$ denotes an exterior unit normal to Ω_k. Note that $[\![w]\!]$ is a vector-valued quantity, and $[\![\mathbf{v}]\!]$ is a scalar.

The distinctive feature of the HDG method is that only degrees of freedom from M_h occur in the globally connected system [6]. In order to clarify this, we introduce the so-called non-homogeneous local solves as follows: Find $(\boldsymbol{\sigma}_{nh}, u_{nh}) \in V_h^2 \times V_h$, such that

$$\begin{aligned} (\boldsymbol{\sigma}_{nh}, \boldsymbol{\tau}_h)_{\Omega_k} + (u_{nh}, \nabla \cdot \boldsymbol{\tau}_h)_{\Omega_k} &= 0, && \forall \boldsymbol{\tau}_h \in V_h^2; \\ -(\mathbf{c}u_{nh} - \varepsilon\boldsymbol{\sigma}_{nh}, \nabla\varphi_h)_{\Omega_k} + \left\langle \alpha u_{nh}^- - \varepsilon\boldsymbol{\sigma}_{nh}^- \cdot \mathbf{n}, \varphi_h \right\rangle_{\partial\Omega_k} &= (f, \varphi_h)_{\Omega_k}, && \forall \varphi_h \in V_h. \end{aligned}$$

Note that this constitutes a discretization of the equation (1) on a cell Ω_k with homogeneous boundary conditions on $\partial\Omega_k$. In contrast, the homogeneous local solves depend on a function $\lambda_h \in M_h$ and are defined by the following: For a given $\lambda_h \in M_h$, find $(\boldsymbol{\sigma}(\lambda_h), u(\lambda_h)) \in V_h^2 \times V_h$, such that

$$\begin{aligned} (\boldsymbol{\sigma}(\lambda_h), \boldsymbol{\tau}_h)_{\Omega_k} + (u(\lambda_h), \nabla \cdot \boldsymbol{\tau}_h)_{\Omega_k} - \langle \lambda_h, \boldsymbol{\tau}_h \cdot \mathbf{n} \rangle_{\partial\Omega_k} &= 0, \quad \forall \boldsymbol{\tau}_h \in V_h^2; \\ -(\mathbf{c}u(\lambda_h) - \varepsilon\boldsymbol{\sigma}(\lambda_h), \nabla\varphi_h)_{\Omega_k} + \left\langle \left(\widehat{\mathbf{c}u(\lambda_h) - \varepsilon\boldsymbol{\sigma}}(\lambda_h) \right) \cdot \mathbf{n}, \varphi_h \right\rangle_{\partial\Omega_k} &= 0, \quad \forall \varphi_h \in V_h. \end{aligned}$$

The flux $\widehat{\mathbf{c}u - \varepsilon\boldsymbol{\sigma}}$ is defined by

$$\widehat{\mathbf{c}u - \varepsilon\boldsymbol{\sigma}} := \mathbf{c}\lambda_h - \alpha(\lambda_h - u^-)\mathbf{n} - \varepsilon\boldsymbol{\sigma}_h^-,$$

where α is a stabilization parameter. Note that, for given λ_h and Ω_k, this is a *locally*, not globally coupled system. (And so is the non-homogeneous local solve.) Thus, it allows for a very easy parallelism. Boundary conditions are set via λ_h.

The global problem that has to be solved is to seek for $\lambda_h \in M_h$, such that

$$a_h(\lambda_h, \mu_h) := \left\langle [\![\widehat{\mathbf{c}u(\lambda_h) - \varepsilon\boldsymbol{\sigma}}(\lambda_h)]\!], \mu_h \right\rangle_\Gamma = -\langle [\![\alpha u_{nh}\, \mathbf{n} - \varepsilon\boldsymbol{\sigma}_{nh}]\!], \mu_h \rangle_\Gamma =: b(\mu_h), \qquad \forall \mu_h \in M_h. \tag{2}$$

After obtaining the solution to the global problem λ_h the original unknowns u_h, $\boldsymbol{\sigma}_h$ can be easily constructed as

$$\begin{aligned} u_h &= u_{nh} + u(\lambda_h), \\ \boldsymbol{\sigma}_h &= \boldsymbol{\sigma}_{nh} + \boldsymbol{\sigma}(\lambda_h). \end{aligned}$$

Remark 1. *Note that we did not split the stability constant α into a convective and a viscous part. Usually, this constant is assumed to be $\mathcal{O}(1)$.*

Remark 2. *It is well-known [6] that the HDG method presented here is equivalent to a DG method*

$$\begin{aligned} (\boldsymbol{\sigma}_h, \tau_h)_{\Omega_k} + (u_h, \nabla \cdot \tau_h)_{\Omega_k} - \langle \widehat{u}_h, \tau_h \cdot \mathbf{n} \rangle_{\partial\Omega_k} &= 0 \\ -(\mathbf{c}u_h - \varepsilon\boldsymbol{\sigma}_h, \nabla\varphi_h)_{\Omega_k} + \left\langle \left(\widehat{\mathbf{c}u_h - \varepsilon\boldsymbol{\sigma}}_h \right) \cdot \mathbf{n}, \varphi \right\rangle_{\partial\Omega_k} &= (f, \varphi_h)_{\Omega_k} \end{aligned}$$

with numerical fluxes

$$\widehat{u}_h := \{u_h\} + \frac{\varepsilon}{2\alpha}[\![\boldsymbol{\sigma}_h]\!], \qquad \widehat{\mathbf{c}u_h - \varepsilon\boldsymbol{\sigma}}_h := \mathbf{c}\widehat{u} - \alpha\left(\widehat{u} - u_h^-\right)\mathbf{n} - \varepsilon\boldsymbol{\sigma}_h^-.$$

A straightforward computation reveals that the fluxes are indeed conservative.

3 Hierarchical scale separation

In the spirit of [1], we assume that M_h is equipped with a hierarchical basis, i.e., every $\lambda_h \in M_h$ allows a decoupling into *coarse* and *fine* scales $\bar{\lambda}_h + \lambda'_h$. This could be a Taylor basis as in [1], or a Legendre basis, which is what we use in this publication. More precisely, coarse scales belong to piecewise constants in this work. Then again, (2) yields a linear system of equations that, in the spirit of coarse and fine scales, can be written as

$$\begin{pmatrix} \bar{A} & \bar{B} \\ B' & A' \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{\lambda}} \\ \boldsymbol{\lambda}' \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{b}} \\ \mathbf{b}' \end{pmatrix}.$$

The key idea of the hierarchical scale separation is to start from an initial guess for $\boldsymbol{\lambda}$, and only solve the linear system of equation associated to $\bar{A}$, and the block-diagonal version of A'. More precisely, let A' be split into a block-diagonal matrix $\tilde{A}'$ (the blocks corresponding to degrees of freedom on only one edge) and the remainder $(A' - \tilde{A}')$. Then the algorithm is given in Alg. 1 (see again [1]).

$\boldsymbol{\lambda}^{(0)} = (\bar{\boldsymbol{\lambda}}^{(0)}, \boldsymbol{\lambda}'^{(0)})^T$ – initial guess; $k = 0$;
while $\boldsymbol{\lambda}^{(k)}$ *not converged* **do**
  $\bar{A}\,\bar{\boldsymbol{\lambda}}^{(k+1)} = \bar{\mathbf{b}} - \bar{B}\boldsymbol{\lambda}'^{(k)}$;
  $\tilde{A}'\boldsymbol{\lambda}'^{(k+1)} = \mathbf{b}' - (A' - \tilde{A}')\boldsymbol{\lambda}'^{(k)} - B'\bar{\boldsymbol{\lambda}}^{(k+1)}$;
  $\boldsymbol{\lambda}^{(k+1)} = (\bar{\boldsymbol{\lambda}}^{(k+1)}, \boldsymbol{\lambda}'^{(k+1)})^T$;
  $k = k + 1$;
end
$\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(k)}$;

Algorithm 1: HSS algorithm

Note that this algorithm consists of a 'coarse grid' part on the space of piecewise polynomials, and a block-Jacobi smoother thereafter. Both, the choice of the coarse space and of the smoother can be subject to further investigation and might be tuned to improve the performance of the method. However, this is not the focus of the present work and will be left for future investigation.

4 Numerical results

In this section, we present numerical results demonstrating the performance of the hierarchical scale separation coupled to the HDG method. All linear systems of equations are solved with a GMRES method with a default tolerance of 10^{-10}. Local solves are computed using a direct solver, which is standard as these matrices are small but dense. If not stated otherwise, we choose the stabilization coefficient α to be one. Based on our experience with the HDG method, the algorithm should not be too sensitive with respect to this parameter. However, we discuss cases where HSS does not converge for some values of alpha, mostly for the non-hybridized DG method.

4.1 Simple case

We start with a simple test case, choosing $\mathbf{c} = (1, 2)^T$ and $\varepsilon \in \{1, 10^{-1}, 10^{-2}\}$. The equation (1) is equipped with homogeneous boundary conditions, and the source term f is chosen in such a way that the solution is given as

$$u(x, y) = \sin(2\pi x)\sin(2\pi y). \tag{3}$$

Domain Ω is the unit square. In Figs. 1-3, we report the convergence history of the HSS algorithm in combination with the HDG method for different viscosity coefficients. Polynomial orders are chosen as $p = 1$ (left), $p = 4$ (middle) and $p = 7$ (right) to demonstrate the different behavior for varying polynomial degrees. The following observation can be made from these findings:

- The number of iterations to reach the stopping criterion seems to be, at least approximately for $\varepsilon = 10^{-2}$, mesh-independent.
- Rather surprisingly, the number of iterations depends very little on the value of ε. In the case of the non-hybridized HSS-realization [1], this dependence was quite pronounced. It is also known that multigrid iterations in general tend to be sensitive to the ratio of the diffusion and convection coefficients [3, 10]. We believe that the effect observed here is due to the fact that the local solves already take away much of the difficulties associated to the vanishing viscosity.
- Not surprising at all is the fact that the higher the polynomial degree, the more iterations are necessary.
- As already stated, we choose $\alpha = 1$, independent of the mesh resolution, without observing any convergence problems for the HSS iteration.

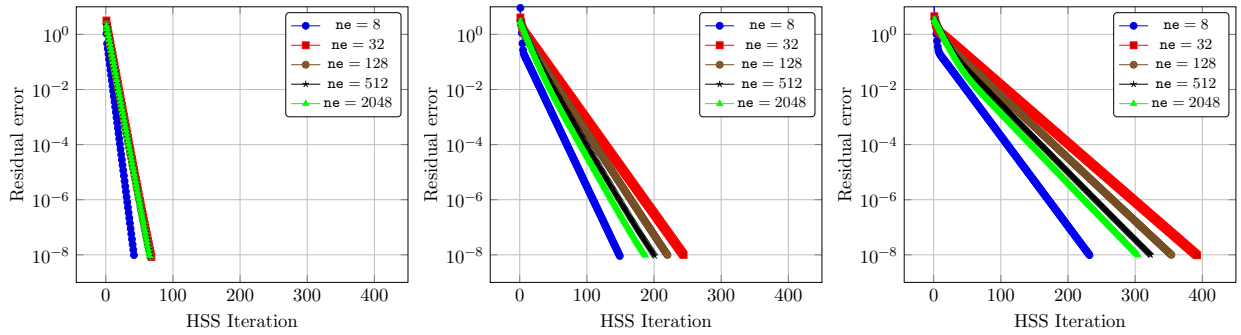

Figure 1: Simple test case with solution (3). $\mathbf{c} = (1, 2)^T$ and $\varepsilon = 1.0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

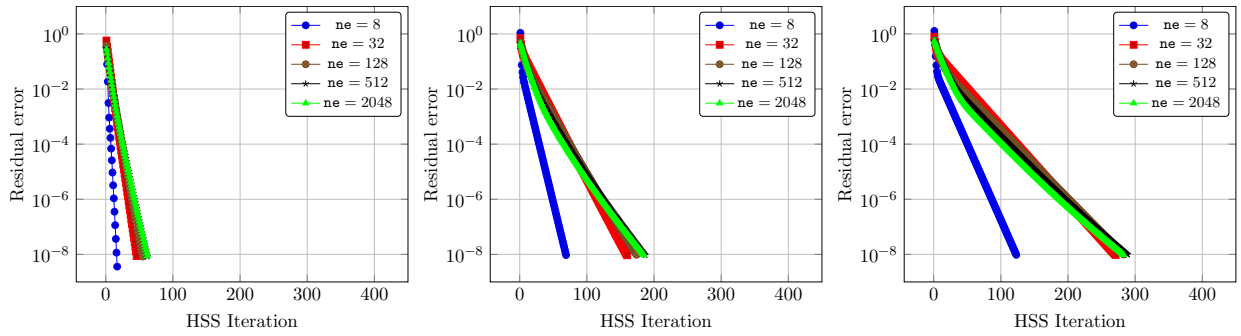

Figure 2: Simple test case with solution (3). $\mathbf{c} = (1, 2)^T$ and $\varepsilon = 0.1$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

In particular, the last part is very intriguing since this is not the case for a non-hybridized DG method, where choosing $\alpha = \mathcal{O}\left(\frac{1}{\Delta x}\right)$ was necessary in order to get a convergent HSS algorithm. This, of course, adds more numerical diffusion than necessary to the problem. Note that this problem has already been

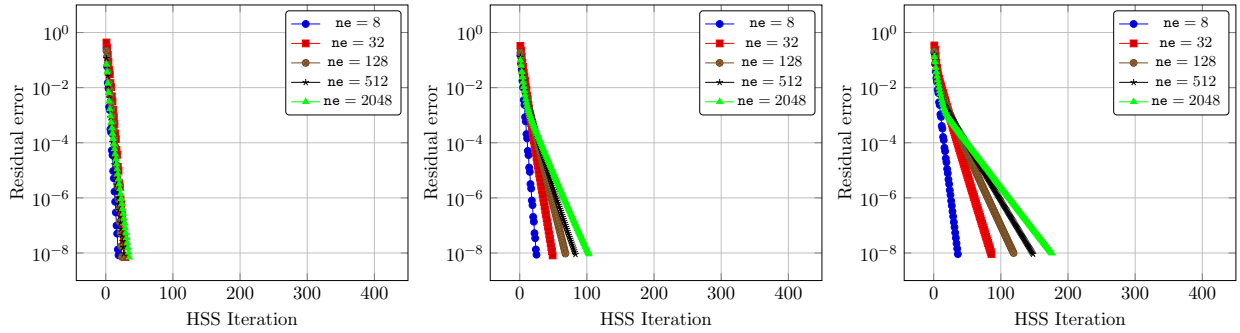

Figure 3: Simple test case with solution (3). $\mathbf{c} = (1, 2)^T$ and $\varepsilon = 0.01$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

reported in [1]. Furthermore, the $p = 0$ subsystem that one has to solve in step one of the HSS algorithm has in general a worse condition than its corresponding HDG counterpart, being expressed in the fact that GMRES takes much longer to converge.

4.2 Boundary layer

A somewhat more challenging test case is presented in this section. We show results for $\mathbf{c} = (1, 1)^T$ and varying ε. Domain Ω is again given as the unit square $\Omega = [0, 1] \times [0, 1]$. The source term f is chosen in such a way that the exact solution u is given as the boundary layer solution

$$u(x, y) = \left(x + \frac{e^{\frac{x}{\varepsilon}} - 1}{1 - e^{\frac{1}{\varepsilon}}} \right) \left(y + \frac{e^{\frac{y}{\varepsilon}} - 1}{1 - e^{\frac{1}{\varepsilon}}} \right), \tag{4}$$

see also [7]. An image of this solution can be seen in Fig. 4. It is clearly visible that there is indeed a boundary layer that steepens as $\varepsilon \to 0$ in the upper right corner. We present numerical results for polynomial degrees $p = 1$ (always left image), $p = 4$ (middle image) and $p = 7$ to investigate the performance of the HSS algorithm. Convergence of the HSS iterations can be seen in Figs. 5-7. Again, we can observe the findings already given in Sec. 4.1, showing that these findings seem to be independent of the complexity of the test case.

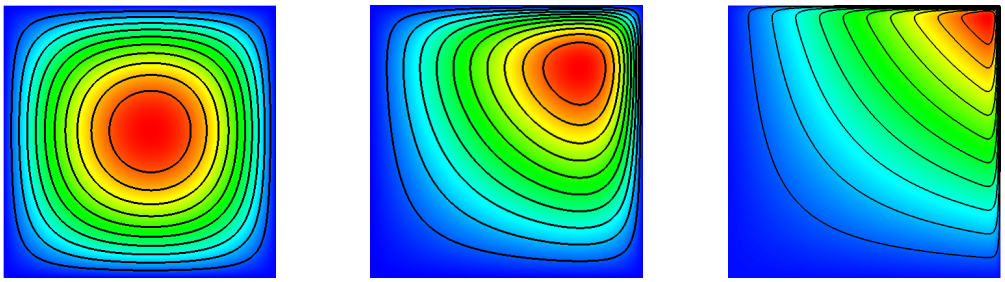

Figure 4: Boundary layer test case: Solution for $\varepsilon = 1$, 0.1 and 0.01 (left to right).

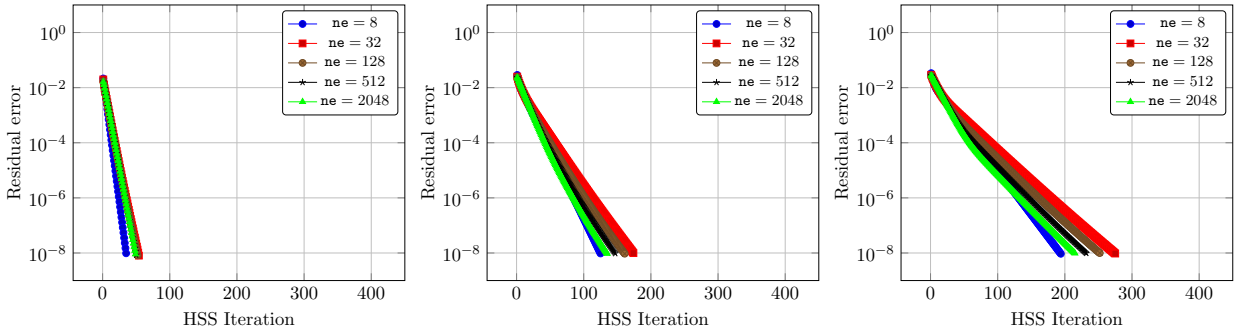

Figure 5: Boundary layer test case with solution (4). $\mathbf{c} = (1, 1)^T$ and $\varepsilon = 1.0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

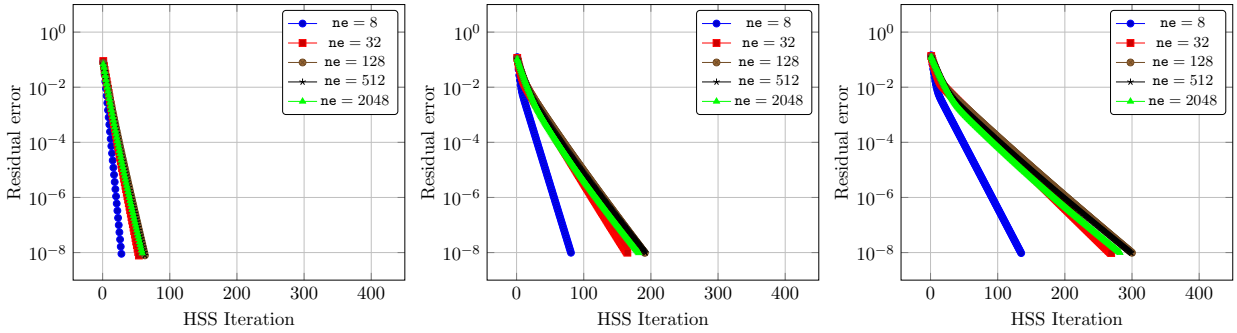

Figure 6: Boundary layer test case with solution (4). $\mathbf{c} = (1, 1)^T$ and $\varepsilon = 0.1$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

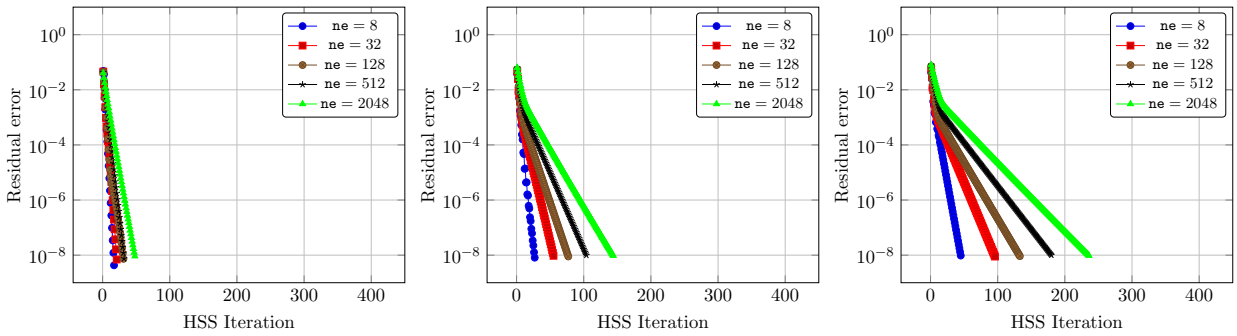

Figure 7: Boundary layer test case with solution (4). $\mathbf{c} = (1, 1)^T$ and $\varepsilon = 0.01$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

4.3 Unstructured mesh

In order to test whether the results obtained thus far depend on the domain geometry and mesh structure, we use a more involved test case here. The right-hand side is chosen to be $f \equiv 0$, boundary conditions are

$u(x,y) = \sin(2\pi(x-y))$. We choose $\mathbf{c} = (1,1)^T$, so with $\varepsilon \to 0$, the solution obviously converges toward $u(x,y) = \sin(2\pi(x-y))$.

A picture of both solution and geometry can be seen in Fig. 8. For obvious reasons, we refer to this geometry as *pacman-like* geometry. We observe that the conclusions drawn from the previous test case also hold here (see Figs. 9-11), so the domain geometry and the mesh do not appear to have a significant influence on the quality of the HSS algorithm.

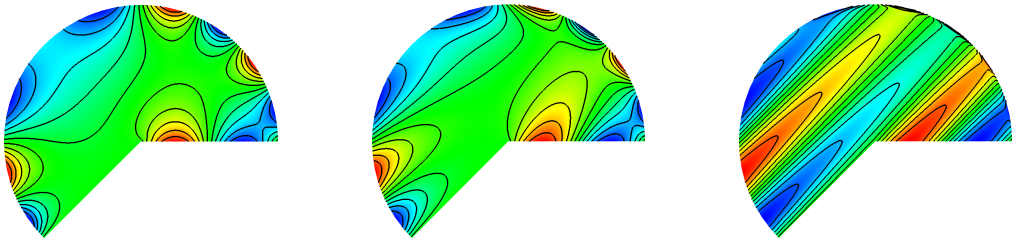

Figure 8: Test case on unstructured grid: Solution for $\varepsilon = 1$, 0.1 and 0.01 (left to right).

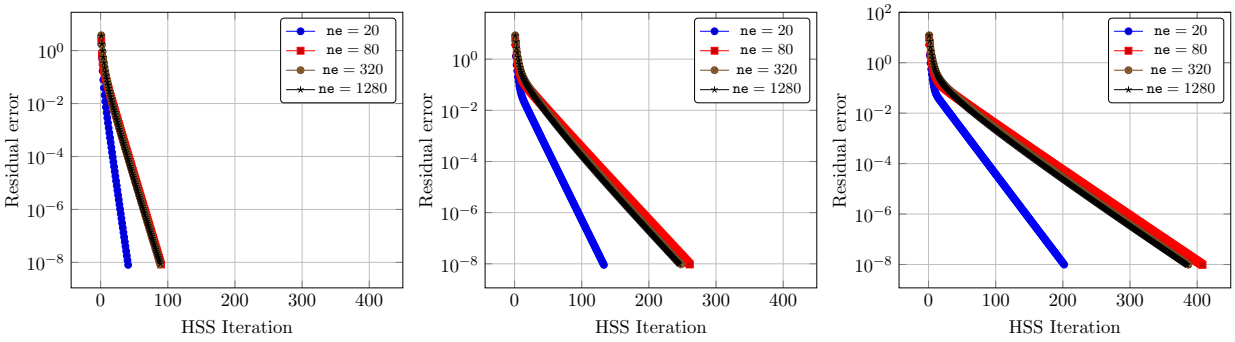

Figure 9: *Pacman* test case. $\mathbf{c} = (1,1)^T$ and $\varepsilon = 1.0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

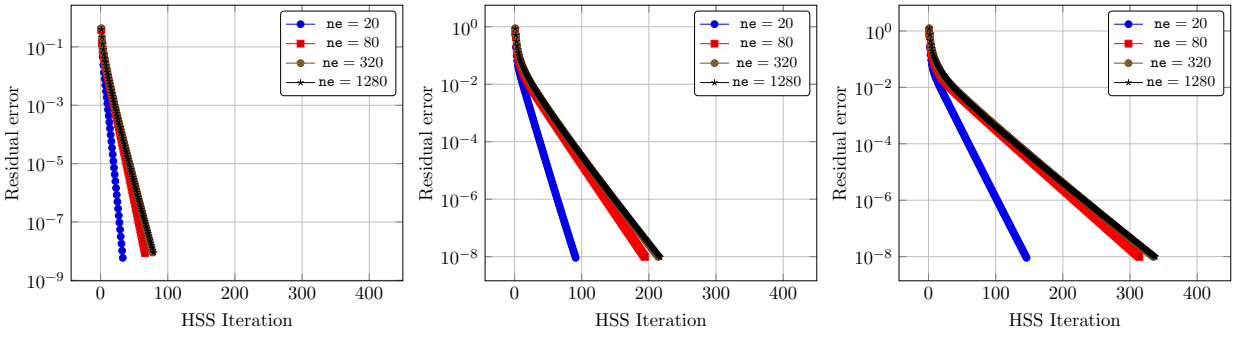

Figure 10: *Pacman* test case. $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0.1$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

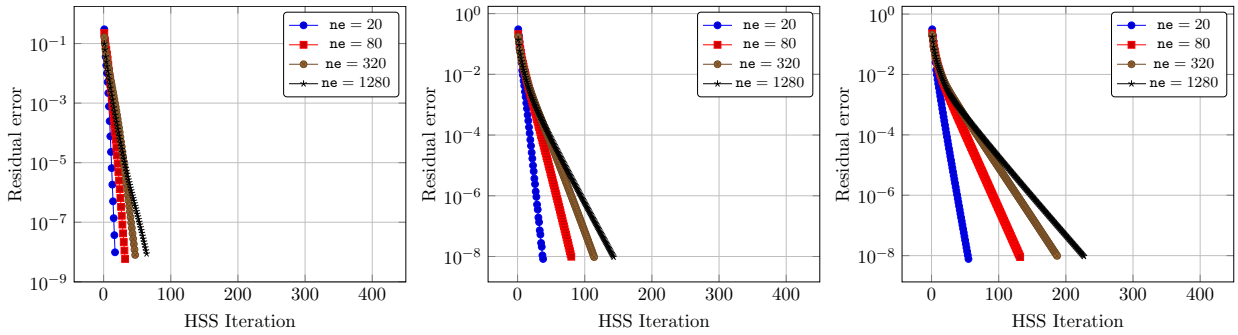

Figure 11: *Pacman* test case. $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0.01$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

4.4 Convection equation: Simple

Our main interest is in applying the methodology presented here to the compressible Euler or Navier-Stokes equations. It is well-known [10, 16] that straightforward extensions of multigrid algorithms for elliptic problems do not perform very well in this setting. In particular, also the hierarchical scale separation for non-hybridized DG does not perform as well as for non-zero diffusion – as observed in [1]. However, we show some interesting findings here. We choose $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$ in this section, again for Ω being the unit square. Obviously, this necessitates outflow boundary conditions where $\mathbf{c} \cdot \mathbf{n} > 0$, which we approximate by setting $\lambda_h = u_h^-$ on $\partial\Omega$. Inflow boundary conditions are set as $u(x,y) = \sin(2\pi(x-y))$, which yields the solution

$$u(x,y) = \sin(2\pi(x-y)). \tag{5}$$

This time, we compare HDG to its completely equivalent, i.e., $u_h^{DG} = u_h^{HDG}$, DG counterpart. Our numerical results are shown in Fig. 12 for the HDG and Fig. 13 for the DG method. The following observations can be made:

- Obviously, the number of steps needed until convergence is reached is not mesh-independent, neither for DG nor for HDG. This is in good agreement with well-known results from multigrid theory. Note that we do not aim at improving this convergence behavior by using an ordering of the degrees of freedom that depends on the linearity of the convection, because this is not directly applicable to Euler equations.
- For $p = 4$ and $p = 7$, the HDG method shows a rather odd convergence behavior. The residual error is increased up to a certain point from where convergence starts and is, in fact, very fast. We do not yet have an explanation for this phenomenon, nor do we know how to exploit this to obtain even faster convergence.
- We choose, as before $\alpha = 1$ for the HDG method. This time, this choice seems to be sensitive. Increasing α beyond four results in a divergence of the HSS algorithm. In contrast, $\alpha = 1$ is not possible for the DG method, where $\alpha = 2$ is needed.
- The convergence behavior of the DG method is monotonously decreasing, however very slow. This is in good agreement with findings in [1].
- In comparison, it seems that HDG is much more efficient than DG for this setting.

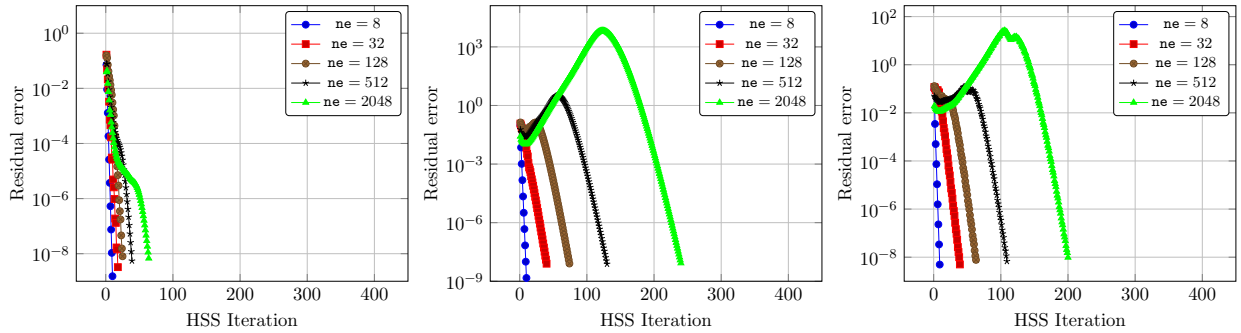

Figure 12: Convection equation with solution (5). $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is HDG with numerical stabilization $\alpha = 1$.

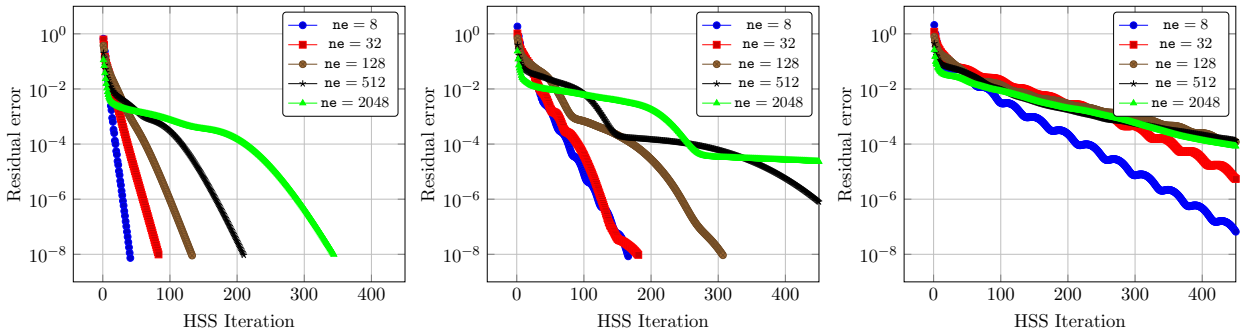

Figure 13: Convection equation with solution (5). $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is DG with numerical stabilization $\alpha = 2$.

4.5 Convection equation: Unstructured mesh.

To substantiate our findings for the pure convection case, we compute yet another example, again with a non-trivial domain, being in this case the ring-shaped geometry $\Omega = \{(x,y) \mid \frac{1}{2} \leq x^2 + y^2 \leq 1\}$. We choose $\mathbf{c} = (1,1)^T$, and inflow boundary conditions are set in such a way that the solution is $u(x,y) = e^{\cos(x-y)}$, see Fig. 14 for an illustration. Results are plotted in Fig. 15 for the HDG method, and Fig. 16 for the DG method. We can confirm our findings from the previous section, with the exception that the convergence behavior of the HDG method is less irregular than before. The reason behind this is certainly an interesting point of future research.

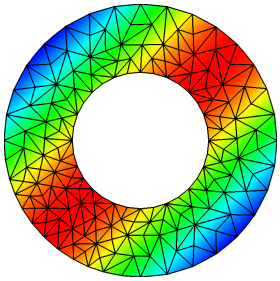

Figure 14: Convection on unstructured mesh. Plotted is medium mesh with 296 cells.

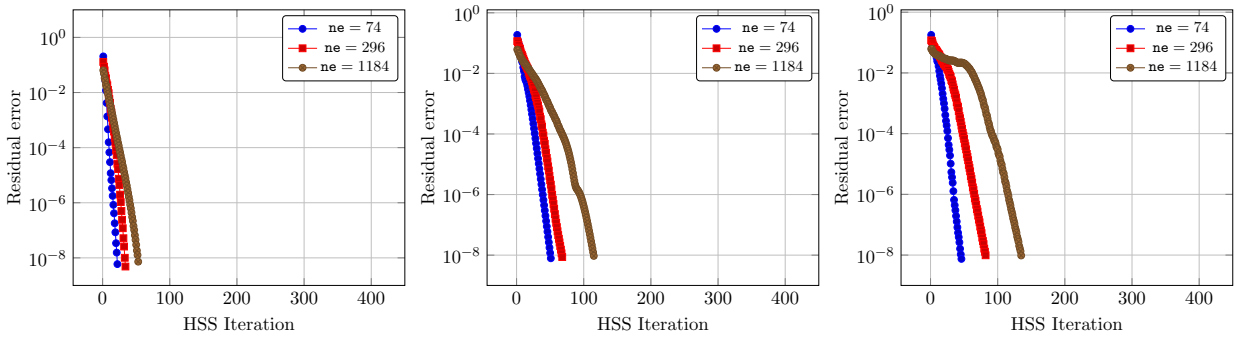

Figure 15: Convection equation with ring-shaped geometry. $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is HDG with numerical stabilization $\alpha = 1$.

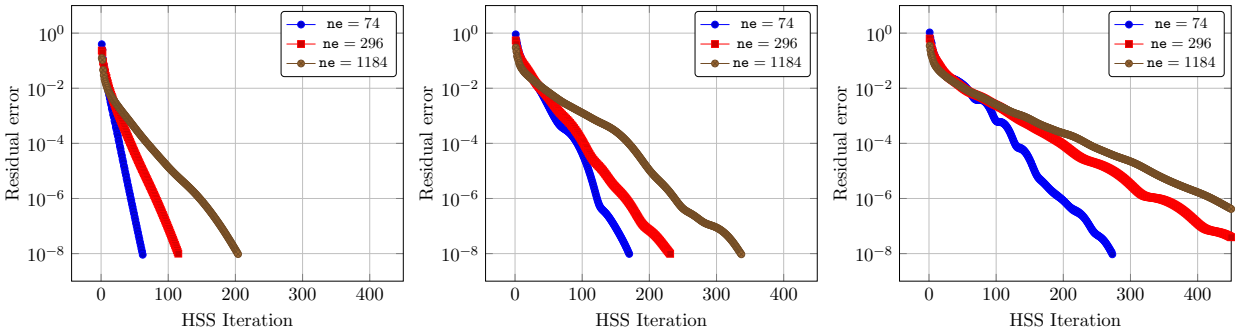

Figure 16: Convection equation with ring-shaped geometry. $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. Left plot: Polynomial order $p = 1$, middle plot: $p = 4$, right plot: $p = 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is DG with numerical stabilization $\alpha = 2$.

5 Conclusion and outlook

Our numerical tests shown in this work suggest that the HSS method is an efficient solution technique for linear equation systems arising from hybridized discontinuous Galerkin discretizations. The experimental convergence rates of the HSS-iteration compared favorably to that of the non-hybridized DG formulations;

the results appear to be unaffected by domain geometry, mesh structure, and solution complexity. The magnitude of the diffusion coefficient did not influence the convergence of the scheme, thus differing in this regard from method's behavior in non-hybridized cases. Furthermore, for pure convection, the algorithm performs better than its corresponding DG formulation, with still some inexplicable convergence behavior, though. The latter is left for future research.

Promising results obtained here for a convection-diffusion equation suggest that the HSS method, especially in the combination with an HDG discretization, is a viable alternative to conventional linear solvers; in particular, one has to note a very simple algorithm and a near 'black-box' usage of the method. A number of issues connected with the performance of the HSS scheme for the HDG method are subject of ongoing and future work. In particular, we are looking into implementing and testing the scheme for Euler and compressible Navier-Stokes equations; time-dependent problems and integration of HSS into the Newton's method for non-linear equations are other investigations being pursued.

A separate interesting research direction concerns using HSS in a combination with the traditional multigrid (either of geometric or of algebraic variety). A straightforward extension would integrate a multigrid solver for the coarse scales, a bit more sophisticated usage could employ HSS as a smoother or preconditioner within a multigrid scheme.

References

[1] V. Aizinger, D. Kuzmin, and L. Korous. Scale separation in fast hierarchical solvers for discontinuous Galerkin methods. *Applied Mathematics and Computation*, 266:838–849, 2015.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and L. Donatella Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39:1749–1779, 2002.

[3] A. Brandt. Barriers to achieving textbook multigrid efficiency (tme) in cfd. Technical Report CR-1998-207647, NASA, IC NASA Center for AeroSpace Information (CASI), Hanover, MD, 1998.

[4] F. Brezzi, J. Douglas, and L. Donatella Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47:217–235, 1985.

[5] F. Brezzi and M. Fortin. Mixed and hybrid finite element methods. 1991.

[6] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47:1319–1365, 2009.

[7] H. Egger and J. Schöberl. A hybrid mixed discontinuous Galerkin finite element method for convection-diffusion problems. *IMA Journal of Numerical Analysis*, 30:1206–1234, 2010.

[8] H. Egger and C. Waluga. hp-analysis of a hybrid dg method for stokes flow. *IMA Journal of Numerical Analysis*, 33(2):687–721, 2013.

[9] K. Fidkowski, T. Oliver, J. Lu, and D. L. Darmofal. p–Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 207:92–113, 2005.

[10] E. Katzer. Multigrid methods for hyperbolic equations. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods III*, volume 98 of *International Series of Numerical Mathematics / Internationale Schriftenreihe zur Numerischen Mathematik / Srie Internationale d Analyse Numrique*, pages 253–263. Birkhuser Basel, 1991.

[11] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228:3232–3254, 2009.

[12] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics*, 228:8841–8855, 2009.

[13] N. C. Nguyen, J. Peraire, and B. Cockburn. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics*, 230:3695–3718, 2011.

[14] E. M. Ronquist and A. T. Patera. Spectral element multigrid I. Formulation and numerical results. *Journal of Scientific Computing*, 2(4):389–406, 1987.

[15] J. Schütz and G. May. A hybrid mixed method for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 240:58–75, 2013.

[16] W. L. Wan and T. F. Chan. A phase error analysis of multigrid methods for hyperbolic equations. *SIAM Journal on Scientific Computing*, 25(3):857–880, March 2003.

universiteit hasselt

UHasselt Computational Mathematics Preprint Series

UP-16-01 *Jochen Schütz and Vadym Aizinger*, **A hierarchical scale separation approach for the hybridized discontinuous Galerkin method**, 2016