A hierarchical scale separation approach for the hybridized discontinuous Galerkin method

Accepted Manuscript

A hierarchical scale separation approach for the hybridized discontinuous Galerkin method

Jochen Schütz, Vadym Aizinger

PII: S0377-0427(16)30628-8
DOI: http://dx.doi.org/10.1016/j.cam.2016.12.018
Reference: CAM 10942

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 1 January 2016
Revised date: 11 August 2016

Please cite this article as: J. Schütz, V. Aizinger, A hierarchical scale separation approach for the hybridized discontinuous Galerkin method, *Journal of Computational and Applied Mathematics* (2016), http://dx.doi.org/10.1016/j.cam.2016.12.018

Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 00 (2016) 1–7
Journal Logo

A hierarchical scale separation approach for the hybridized discontinuous Galerkin method

Jochen Schütz

Faculty of Sciences, Hasselt University, Campus Diepenbeek, Agoralaan Gebouw D, 3590 Diepenbeek, Belgium

Vadym Aizinger*

Chair of Applied Mathematics 1, University of Erlangen-Nürnberg, Cauerstraße 11, 91058 Erlangen, Germany

Abstract

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.

Keywords: hybridized discontinuous Galerkin method, hierarchical scale separation, convection-diffusion equation, p-multigrid method

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

*Corresponding author
Email address: aizinger@math.fau.de (Vadym Aizinger)

freedom for higher order discretizations, 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 multigrid method that relies on 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 evaluate the performance of the method on a range of benchmark problems of varying complexity. The sensitivity of the method with respect to the diffusion coefficient and the stabilization parameter is investigated for selected test cases. The combination of the two aforementioned numerical techniques both aiming to enhance the computational performance may cover a lot of ground toward our goal: Making DG discretizations more competitive with the classical finite element and finite volume methods for a range of important applications.

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 $\Omega \subset \mathbb{R}^2$,

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

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 exterior unit normal to $\partial\Omega$.

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

$$\begin{aligned} \boldsymbol{\sigma} &= \nabla u, \\ \nabla\cdot(\mathbf{c}u - \varepsilon\boldsymbol{\sigma}) &= f. \end{aligned}$$

The presentation of the HDG method below follows closely [6]. First, we need to define the standard DG approximation space V_h on a triangulation $\Omega = \bigcup_{k=1}^{\mathtt{ne}}\Omega_k$ as

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

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

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

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

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

respectively. (Obviously, at the physical boundary of the domain, only the first expression is well 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

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

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 distinct feature of the HDG method is the fact that only the degrees of freedom contained in 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

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

where α is a stabilization parameter. 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

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

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

$$\widehat{\mathbf{c}u - \varepsilon\boldsymbol{\sigma}} := \mathbf{c}\lambda_h - \alpha(\lambda_h - u^-)\mathbf{n} - \varepsilon\boldsymbol{\sigma}_h^-, \tag{2}$$

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

$$a_h(\lambda_h, \mu_h) := \left\langle [\![\widehat{\mathbf{c}u(\lambda_h) - \varepsilon\boldsymbol{\sigma}}(\lambda_h)]\!], \mu_h\right\rangle_\Gamma = -\langle [\![\alpha u_{nh}\,\mathbf{n} - \varepsilon\boldsymbol{\sigma}_{nh}]\!], \mu_h\rangle_\Gamma =: b(\mu_h), \qquad \forall \mu_h \in M_h. \tag{3}$$

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 $O(1)$.*

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

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

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 $\lambda_h \in M_h$ admits a decoupling into *coarse* and *fine* scales $\bar{\lambda}_h + \lambda'_h$. The hierarchical basis could be a Taylor as in [1] or a Legendre basis as in the present case. We equate here the coarse scales to the piecewise constant part of our approximate solution. Then (3) yields a linear system of equations that, after subdividing into 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 systems associated to $\bar{A}$ and the block-diagonal version of A'. For that, 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 given in Alg. 1 (see again [1]) reads:

$\boldsymbol{\lambda}^{(0)} = (\bar{\boldsymbol{\lambda}}^{(0)},\ \boldsymbol{\lambda}'^{(0)})^T$ – 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 the above algorithm consists of a 'coarse grid' part acting on the space of piecewise polynomials and a block-Jacobi smoother dealing with fine scales. 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, the results of our numerical studies indicate that even the simplest choices made in this work deliver efficient and robust performance on a range of test problems.

Remark 3. *In a triangulation, the number of edges is generally greater than the number of elements. Therefore, the coarse-scale system of the HDG method is larger than the coarse-scale system of the DG method. Nevertheless, for higher order polynomial approximations, the HDG method produces a much smaller (overall) global linear system.*

4. Numerical results

In this section, we present numerical results demonstrating the performance of the hierarchical scale separation applied to the HDG method. All coarse-scale linear systems are solved using a GMRES method with 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. For selected test problems, we also numerically investigate the sensitivity of the HSS iteration with respect to the choice of stabilization parameter α (see Eq. (2)).

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

$$u(x, y) = \sin(2\pi x)\sin(2\pi y). \tag{4}$$

Domain Ω is the unit square partitioned into a regular triangular mesh. In Fig. 1, 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 convergence behavior for varying polynomial degrees. Fig. 2 illustrates the sensitivity of the HSS iteration to the choice of the stabilization parameter α in (2). The following observations can be made from these findings:

- Rather surprisingly, the number of iterations does not seem to depend substantially on the value of ε. E.g., the number of iterations for polynomial degree $p = 4$ on the highest resolution mesh (Fig. 1) goes up from 100 to 200 as ε decreases by two orders of magnitude. Similar tendency is also seen for polynomial degree $p = 7$. This dependence was quite pronounced in the case of the non-hybridized HSS-realization [1]. It is well 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 alleviate many 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.
- For this problem, the HSS iteration appears to be mesh-independent for two larger values of the diffusion coefficient; a clear increase of the number of iterations with the increasing mesh resolution is detected for the smallest value of $\varepsilon = 0.01$. The same pattern also holds for the more complex test problems as demonstrated in the remainder of this section.
- As already stated, the results in Fig 1 were obtained using stabilization parameter $\alpha = 1$ independent of the mesh resolution without observing any convergence problems for the HSS iteration. This fact is particularly intriguing since, for a non-hybridized DG method, choosing $\alpha = \mathcal{O}\left(\frac{1}{\Delta x}\right)$ was necessary in order to get a convergent HSS algorithm. This increases the numerical diffusion – a problem already reported in [1].
- To investigate the influence of the choice of α on the number of iterations, we perform a series of tests with polynomial degree $p = 4$ and different values of diffusion coefficient ε. Plotting the number of HSS iterations versus α in Fig. 2 we observe only slight α-sensitivity for both larger diffusion coefficients; this dependence becomes pronounced for $\varepsilon = 0.01$ and large values of α.

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 and mesh are the same as in Sec. 4.2. The source term f is chosen in such a way that the exact solution u is given as the boundary layer solution

$$u(x, y) = \left(x + \frac{e^{\frac{x}{\varepsilon}} - 1}{1 - e^{\frac{1}{\varepsilon}}}\right)\left(y + \frac{e^{\frac{y}{\varepsilon}} - 1}{1 - e^{\frac{1}{\varepsilon}}}\right), \tag{5}$$

see also [7]. An image of this solution can be seen in Fig. 3. 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 $p = 1$ (left), $p = 4$ (middle), and $p = 7$ (right) to investigate the performance of the HSS algorithm. Convergence of the HSS iterations is shown in Fig. 4. In Fig. 5, the number of HSS iterations is plotted vs. α for $\varepsilon \in \{1, 0.1, 0.01\}$ and $p = 4$. The results are very similar to those given in Fig. 2 for the previous test case. The similarities to the results presented in Sec. 4.1 indicate that the convergence of the HSS iteration is not sensitive to the complexity of the test case.

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 $u(x, y) = \sin(2\pi(x - y))$. The solution of the test case and the domain geometry are illustrated in Fig. 6. For obvious reasons, we call this geometry *pacman-like*. We observe that the conclusions drawn from the previous test case also hold here (see Fig. 7), so the domain geometry and the mesh do not appear to have a significant influence on the efficiency of the HSS algorithm.

4.4. Convection equation: Simple domain

Our ultimate interest is to apply the methodology presented here to the compressible Euler or Navier-Stokes systems. 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, once again Ω is the unit square. A pure convection equation necessitates a special outflow ($\mathbf{c} \cdot \mathbf{n} > 0$) boundary treatment. We achieve this 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

$$u(x, y) = \sin(2\pi(x - y)). \tag{6}$$

This time, we compare HDG to its completely equivalent, i.e., $u_h^{DG} = u_h^{HDG}$ DG counterpart. The HSS iteration convergence is shown in Fig. 8 for the HDG and in Fig. 9 for the DG method. The following observations can be made:

- Obviously, the number of iterations until convergence is reached is not mesh-independent anymore, neither for DG nor for HDG. This is in good agreement with well-known results from multigrid theory. Note that we do not attempt to improve this convergence behavior by, e.g., ordering the mesh elements along the characteristics.
- 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 the convergence starts and is, in fact, very fast. We do not yet have an explanation for this phenomenon, nor do we know at this time how to exploit it to improve the convergence.
- $\alpha = 1$ is chosen as before for the HDG method. This time, the choice seems to be sensitive: Increasing α results in a divergence of the HSS iteration. In contrast, $\alpha = 1$ is not possible for the DG method, where $\alpha = 2$ is used instead.
- In order to investigate the sensitivity of the HSS algorithm with respect to a choice of α for a pure convection problem, we plot the number of iterations vs. α in Fig. 10. The results there for polynomial degree $p = 4$ indicate a rather simple convergence pattern for the DG method with the number of iterations monotonically increasing for $\alpha \geq 1.2$ and a sharply delineated area of fast convergence for the HDG method around the same value of α. Similar behavior is also apparent for polynomial degree $p = 7$, where, because of long computation times, only the HDG values are shown. This suggests that a suitable choice of α in HDG can produce a much more efficient linear solver than in the case of the DG discretization. The reasons for this behavior might lie in the fact that α changes the flux approximation at the element boundaries, thus it is related to the 'upwind' bias – a very relevant parameter in a convection setting. The exact mechanism of this influence is not clear at this point and is a subject of further study.
- The norm of the residual for the DG method is monotonously decreasing, however very slow. This is in good agreement with findings in [1].

4.5. Convection equation: Unstructured mesh

To substantiate our findings for the pure convection case, we compute yet another example, now with a ring-shaped domain geometry $\Omega = \{(x, y) \mid \frac{1}{2} \leq x^2 + y^2 \leq 1\}$. We choose $\mathbf{c} = (1, 1)^T$ and set the inflow boundary conditions in such a way that the solution is $u(x, y) = e^{\cos(x-y)}$ (see Fig. 11 for an illustration). Results are plotted in Fig. 12 for the HDG and in Fig. 13 for the DG method. We can confirm our findings from the previous section with the exception that the convergence behavior of the HSS iteration for the HDG method is somewhat less irregular than before.

5. Conclusion and outlook

The 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 significantly influence the convergence of the scheme, thus differing in this regard from method's behavior in non-hybridized cases. Furthermore, for pure convection, a suitable choice of the HDG stabilization parameter α guarantees a much faster convergence of the HSS algorithm than of its DG counterpart. However, some non-monotonic convergence behavior for higher order polynomial approximations has been detected in the limiting case of no diffusion.

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; particularly appealing features of the HSS scheme include a very simple algorithm and a near 'black-box' usage. Good convergence results obtained without any tuning for the limiting hyperbolic case deserve a separate study; in particular, we are looking into implementing and testing the HSS/HDG combination for the time-dependent compressible Euler and Navier-Stokes equations.

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*. Springer Series in Computational Mathematics New York, 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.

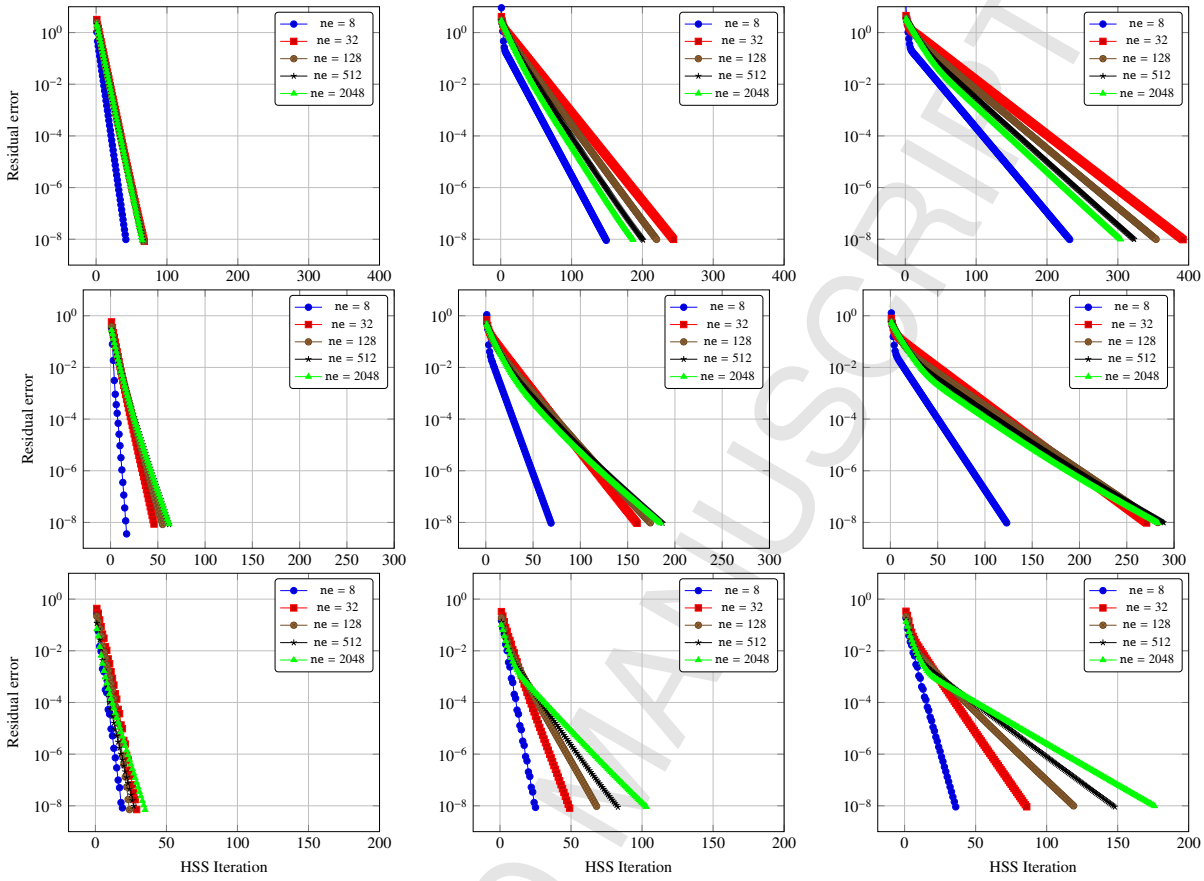

Figure 1. Simple test case with solution (4) and $\mathbf{c} = (1, 2)^T$. From top to bottom: $\varepsilon = 1, 0.1, 0.01$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

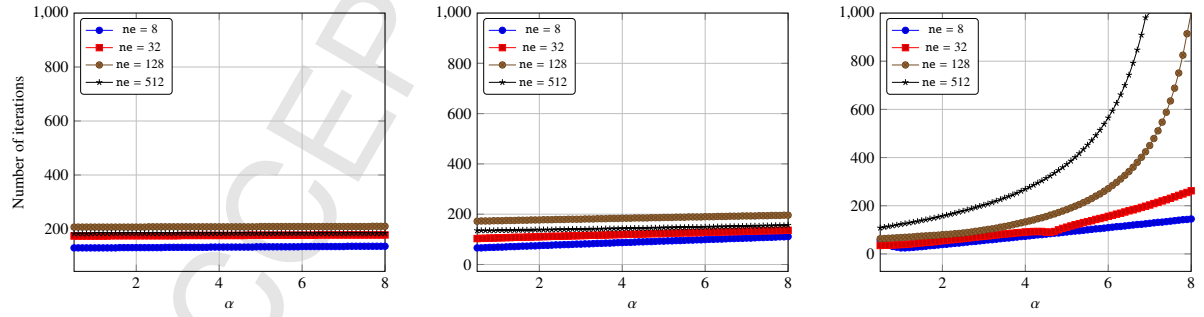

Figure 2. Simple test case with solution (4). $\mathbf{c} = (1, 2)^T$, the chosen polynomial order is $p = 4$. From left to right: $\varepsilon = 1, 0.1, 0.01$. We plot the size of α versus the number of iterations that the HSS algorithm needs to converge.

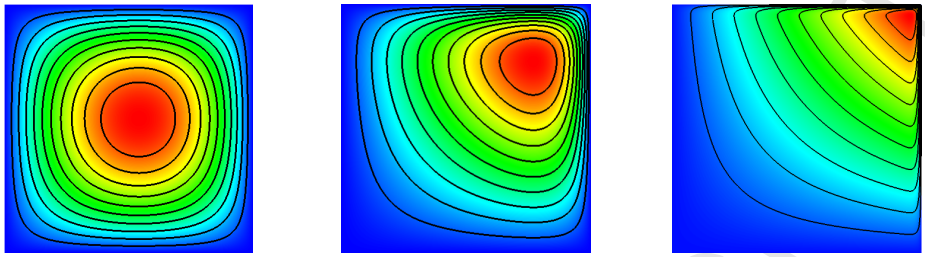

Figure 3. Boundary layer test case: Solution for $\varepsilon = 1$, 0.1 and 0.01 (left to right).

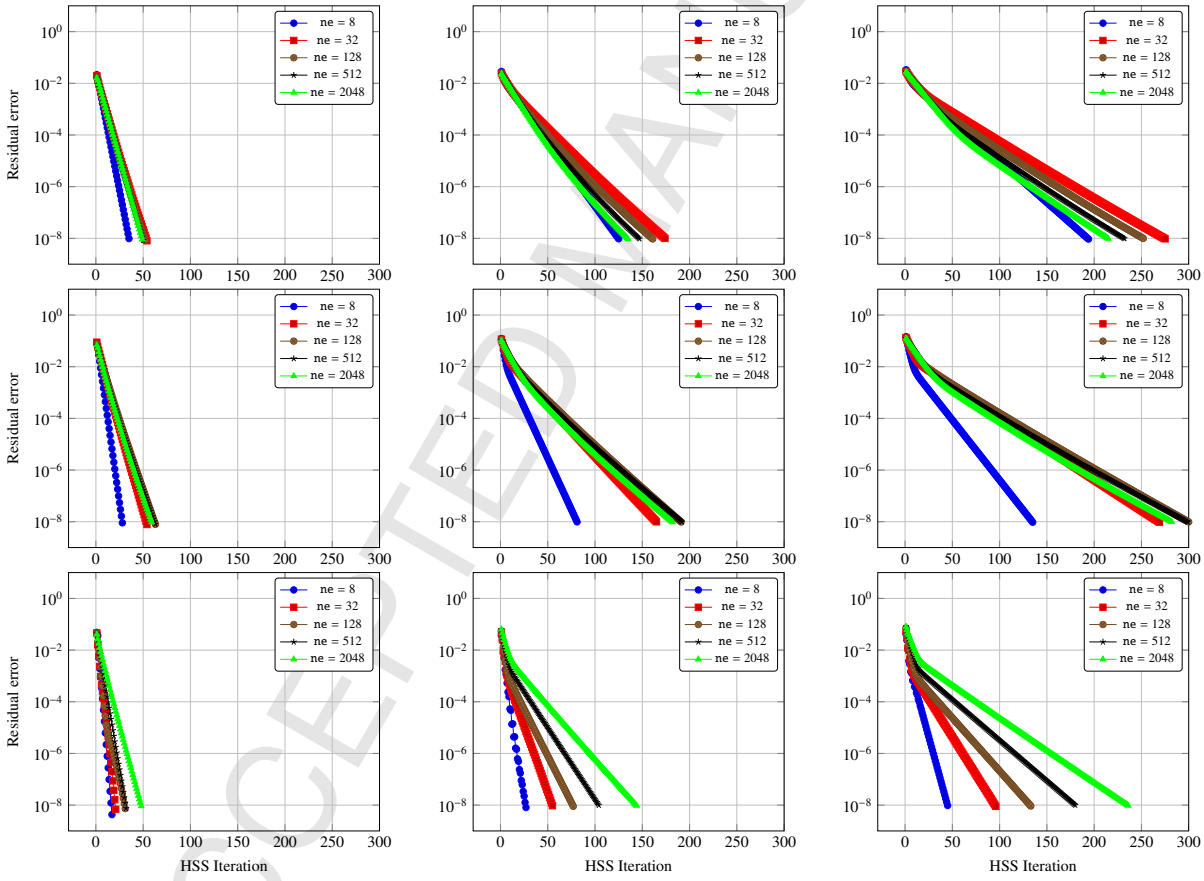

Figure 4. Boundary layer test case with solution (5) an. $\mathbf{c} = (1, 1)^T$. From top to bottom: $\varepsilon = 1, 0.1, 0.01$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

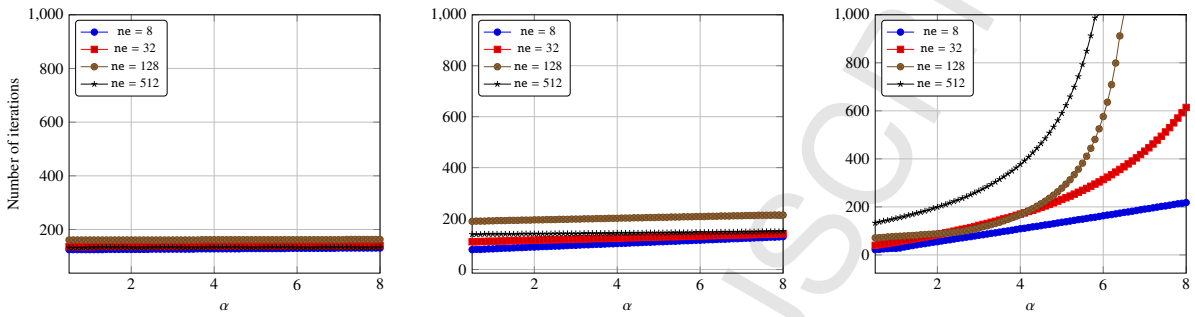

Figure 5. Boundary layer test case with solution (5). $\mathbf{c} = (1, 1)^T$, the chosen polynomial order is $p = 4$. From left to right: $\varepsilon = 1, 0.1, 0.01$. We plot the size of α versus the number of iterations that the HSS algorithm needs to converge.

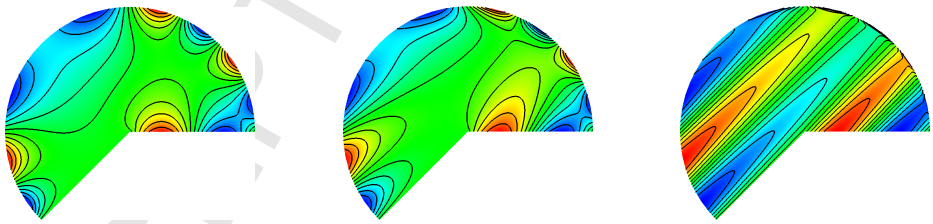

Figure 6. Test case on unstructured grid: Solution for $\varepsilon = 1$, 0.1 and 0.01 (left to right).

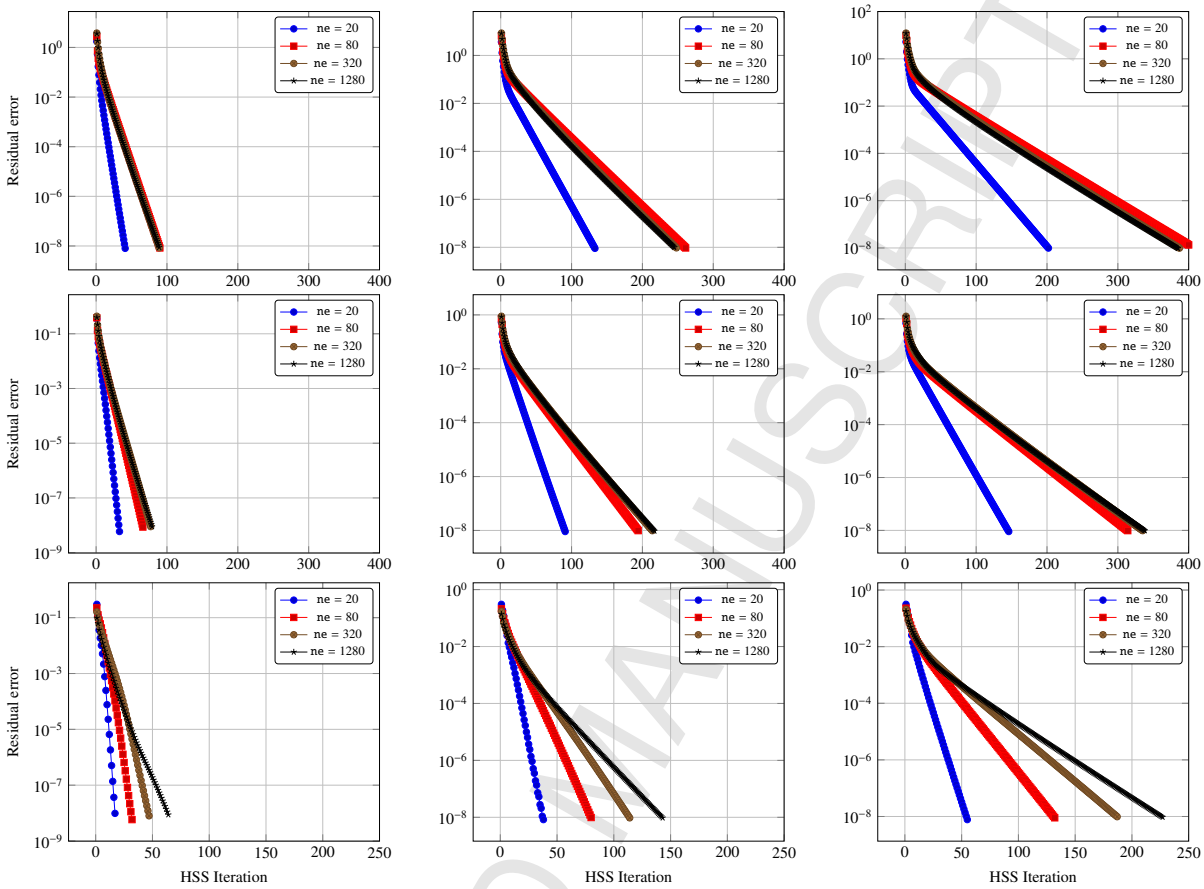

Figure 7. *Pacman* test case with $\mathbf{c} = (1,1)^T$. From top to bottom: $\varepsilon = 1, 0.1, 0.01$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual.

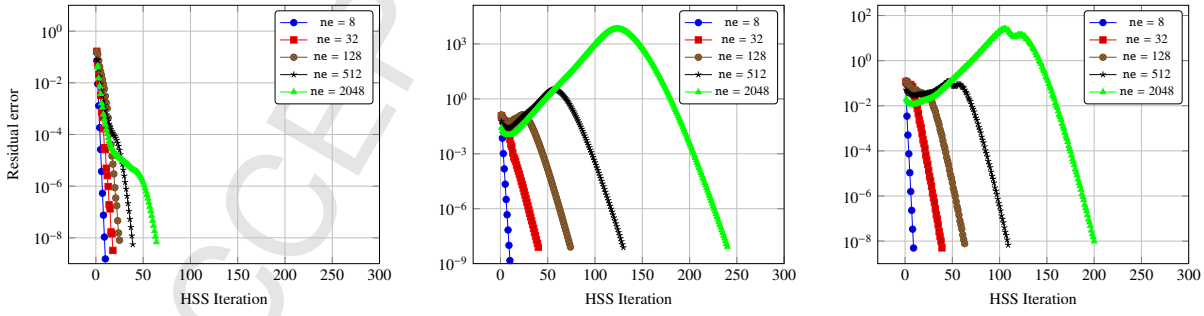

Figure 8. Convection equation with solution (6). $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is HDG with numerical stabilization $\alpha = 1$.

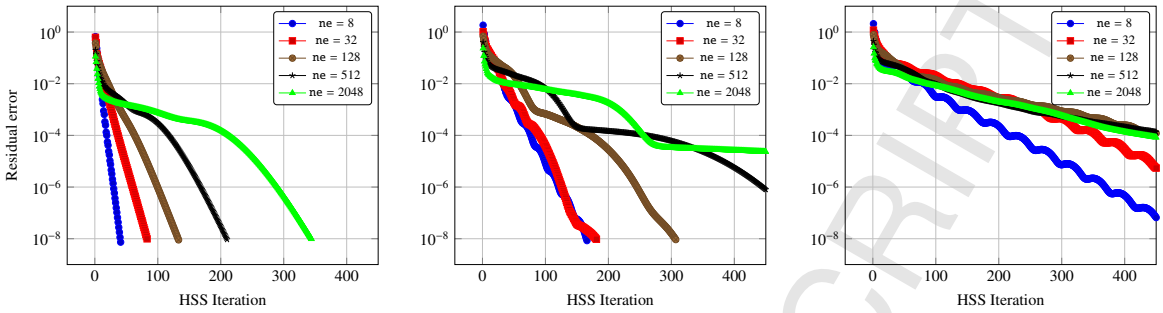

Figure 9. Convection equation with solution (6). $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is DG with numerical stabilization $\alpha = 2$.

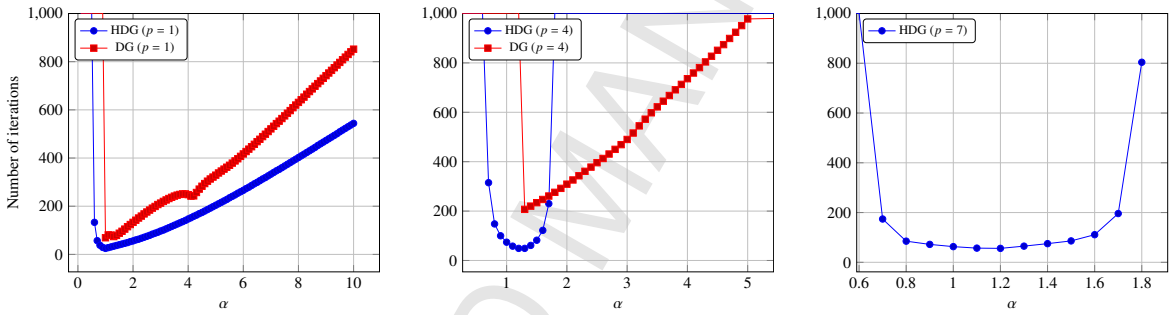

Figure 10. Convection equation with solution (6). $\mathbf{c} = (1,1)^T$ and $\varepsilon = 0$. The number of elements is $\mathtt{ne} = 128$. We plot the size of α versus the number of iterations that the HSS algorithm needs to converge. Unplotted values resulted in a divergence of the HSS algorithm. Note that for the $p = 4$ case, the choice is rather sensitive. For the $p = 7$ case, we do not report DG results because of slow convergence. However, performing some random checks on DG indicates that the convergence pattern is very similar to that of the $p = 4$ case (with a larger number of iterations of course.)

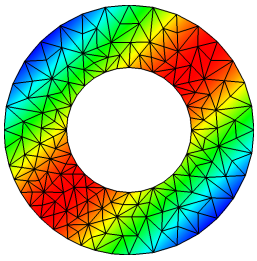

Figure 11. Convection on unstructured mesh. Plotted is medium mesh with 296 cells.

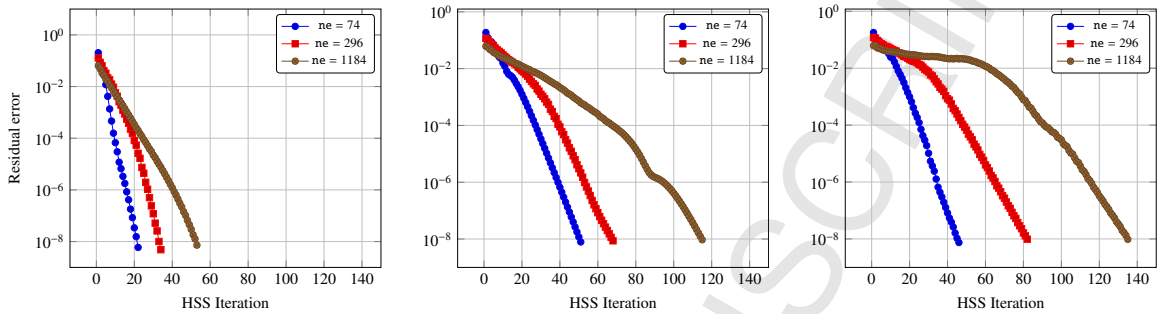

Figure 12. Convection equation with ring-shaped geometry. $\mathbf{c} = (1, 1)^T$ and $\varepsilon = 0$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is HDG with numerical stabilization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

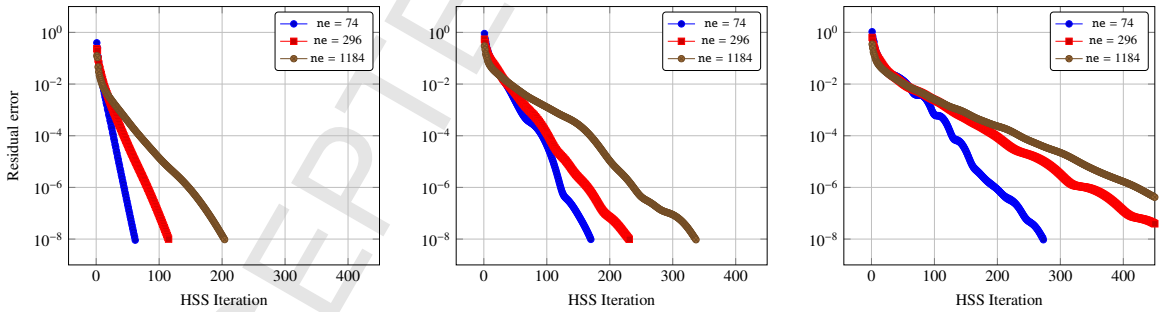

Figure 13. Convection equation with ring-shaped geometry. $\mathbf{c} = (1, 1)^T$ and $\varepsilon = 0$. From left to right: Polynomial order $p = 1, 4, 7$. Plotted is HSS iterations (k in Algorithm 1) versus the algebraic two-norm of the residual. Numerical algorithm is DG with numerical stabilization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