

EHDG: AN EXPONENTIALLY CONVERGENT ITERATIVE SOLVER FOR HDG DISCRETIZATIONS OF HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS. *

SRIRAMKRISHNAN MURALIKRISHNAN*, MINH-BINH TRAN[†], AND TAN BUI-THANH[‡]

Abstract. We present a scalable and efficient iterative solver for high-order hybridized discontinuous Galerkin (HDG) discretizations of hyperbolic partial differential equations. It is an interplay between domain decomposition methods and HDG discretizations. In particular, the method is a fixed-point approach that requires only independent element-by-element local solves in each iteration. As such, it is well-suited for current and future computing systems with massive concurrencies. We rigorously show that the proposed method is exponentially convergent in the number of iterations for transport and linearized shallow water equations. Furthermore, the convergence is independent of the solution order. Various 2D and 3D numerical results for steady and time-dependent problems are presented to verify our theoretical findings.

Key words. Iterative solvers, Discontinuous Galerkin methods, Hybridized Discontinuous Galerkin methods, Shallow water equation, hyperbolic equations, Scalable solvers

1. Introduction. The discontinuous Galerkin (DG) method was originally developed by Reed and Hill [8] for the neutron transport equation, first analyzed in [5], and then has been extended to other problems governed by partial differential equations (PDEs) [3]. Roughly speaking, DG combines advantages of classical finite volume and finite element methods. In particular, it has the ability to treat solutions with large gradients including shocks, it provides the flexibility to deal with complex geometries, and it is highly parallelizable due to its compact stencil.

However, for steady state problems or time-dependent ones that require implicit time-integrators, DG methods typically have many more (coupled) unknowns compared to the other existing numerical methods, and hence more expensive in general.

In order to mitigate the computational expense associated with DG methods, Cockburn, coauthors, and others have introduced hybridizable (also known as hybridized) discontinuous Galerkin (HDG) methods for various types of PDEs, see e.g. [2, 7]. The upwind HDG framework proposed in [1] provides a unified and a systematic construction of HDG methods for a large class of PDEs. In HDG discretizations, the coupled unknowns are single-valued traces introduced on the mesh skeleton, i.e. the faces, and the resulting matrix is substantially smaller and sparser compared to standard DG approaches. Once they are solved for, the usual DG unknowns can be recovered in an element-by-element fashion, completely independent of each other. Nevertheless, the trace system is still a bottleneck for practically large-scale applications, where complex and high-fidelity simulations involving features with a large range of spatial and temporal scales are necessary.

Meanwhile, Schwarz-type domain decomposition methods (DDMs) have been introduced as procedures to parallelize and solve partial differential equations numerically, where each iteration involves the solutions of the original equations on smaller subdomains [6]. Among the many DDMs, Schwarz waveform relaxation methods and

¹Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712, USA.

²Department of Mathematics, University of Wisconsin, Madison, WI 53706, USA.

³Department of Aerospace Engineering and Engineering Mechanics, and the Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA.

*This research was partially supported by DOE grants DE-SC0010518 and DE-SC0011118. We are grateful for the supports.

optimized Schwarz methods [9], have attracted substantial attention over the past decade since they can be adapted to the physics of the underlying problems and thus lead to very efficient parallel solvers for challenging problems. We view the HDG method as an extreme DDM approach in which each subdomain is an element.

While either HDG community or DDM community can contribute individually towards advancing its own field, the potential for a true breakthrough may lie in bringing together the advances from both sides and in exploiting opportunities at their interfaces. Unlike [4], we blend the HDG method and optimized Schwarz idea to produce a efficient and scalable iterative approach for HDG methods. One of the main features of the proposed approach is that it has exponential convergence rate, and for that reason we term it as eHDG. The method can be viewed as a fixed-point approach that requires only independent element-by-element local solves in each iteration. As such, it is well-suited for current and future computing systems with massive concurrencies. We rigorously show that the proposed method is exponentially convergent in the number of iterations for transport and linearized shallow water equations. Furthermore, the convergence is independent of the solution order. The theoretical findings will be verified on various 2D and 3D numerical results for steady and time-dependent problems.

2. Notations for HDG discretizations. In this section we introduce common notations and conventions to be used in the following sections where we propose and rigorously analyze the eHDG approach for scalar and systems of hyperbolic PDEs in both steady and time-dependent cases. Let us partition an open and bounded domain $\Omega \in \mathbb{R}^d$ into N_{el} non-overlapping elements $K_j, j = 1, \dots, N_{\text{el}}$ with Lipschitz boundaries such that $\Omega_h := \cup_{j=1}^{N_{\text{el}}} K_j$ and $\bar{\Omega} = \bar{\Omega}_h$. Here, h is defined as $h := \max_{j \in \{1, \dots, N_{\text{el}}\}} \text{diam}(K_j)$. We denote the skeleton of the mesh by $\mathcal{E}_h := \cup_{j=1}^{N_{\text{el}}} \partial K_j$, the set of all (uniquely defined) faces e . We conventionally identify \mathbf{n}^- as the normal vector on the boundary ∂K of element K (also denoted as K^-) and $\mathbf{n}^+ = -\mathbf{n}^-$ as the normal vector of the boundary of a neighboring element (also denoted as K^+). Furthermore, we use \mathbf{n} to denote either \mathbf{n}^- or \mathbf{n}^+ in an expression that is valid for both cases, and this convention is also used for other quantities (restricted) on a face $e \in \mathcal{E}_h$.

For simplicity in writing we define $(\cdot, \cdot)_K$ as the L^2 -inner product on a domain $K \in \mathbb{R}^d$ and $\langle \cdot, \cdot \rangle_K$ as the L^2 -inner product on a domain K if $K \in \mathbb{R}^{d-1}$. We shall use $\|\cdot\|_K := \|\cdot\|_{L^2(K)}$ as the induced norm for both cases and the particular value of K in a context will indicate which inner product the norm is coming from. We also denote the ε -weighted norm of a function u as $\|u\|_{\varepsilon, K} := \|\sqrt{\varepsilon}u\|_K$ for any positive ε . We shall use boldface lowercase letters for vector-valued functions and in that case the inner product is defined as $(\mathbf{u}, \mathbf{v})_K := \sum_{i=1}^m (\mathbf{u}_i, \mathbf{v}_i)_K$, and similarly $\langle \mathbf{u}, \mathbf{v} \rangle_K := \sum_{i=1}^m \langle \mathbf{u}_i, \mathbf{v}_i \rangle_K$, where m is the number of components $(\mathbf{u}_i, i = 1, \dots, m)$ of \mathbf{u} . Moreover, we define $(\mathbf{u}, \mathbf{v})_\Omega := \sum_{K \in \Omega_h} (\mathbf{u}, \mathbf{v})_K$ and $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{E}_h} := \sum_{e \in \mathcal{E}_h} \langle \mathbf{u}, \mathbf{v} \rangle_e$ whose induced (weighted) norms are clear, and hence their definitions are omitted. We employ boldface uppercase letters, e.g. \mathbf{L} , to denote matrices and tensors. In addition, subscripts are used to denote the components of vectors, matrices, and tensors. We define $\mathcal{P}^p(K)$ as the space of polynomials of degree at most p on a domain K .

3. Construction of eHDG methods for linear hyperbolic PDEs. In this section, we define eHDG methods for scalar and system of hyperbolic PDEs. For the clarity in exposition, we consider the transport equation and linearized shallow water system, and extension of the proposed approach to other hyperbolic PDEs is

straightforward. To begin, let us consider the transport equation

$$\boldsymbol{\beta} \cdot \nabla u = f \quad \text{in } \Omega, \quad (3.1a)$$

$$u = g \quad \text{on } \partial\Omega^-, \quad (3.1b)$$

where $\partial\Omega^-$ is the inflow part of the boundary $\partial\Omega$. An upwind HDG discretization [1] for (3.1) consists of the following local equation for each element K

$$-(u, \nabla \cdot (\boldsymbol{\beta} v))_K + \langle \boldsymbol{\beta} \cdot \mathbf{n} u + |\boldsymbol{\beta} \cdot \mathbf{n}| (u - \hat{u}), v \rangle_{\partial K} = (f, v)_K, \quad \forall v \in \mathcal{P}^p(K), \quad (3.2)$$

and conservation conditions on all edges e in the mesh skeleton \mathcal{E}_h :

$$\langle \llbracket \boldsymbol{\beta} \cdot \mathbf{n} u + |\boldsymbol{\beta} \cdot \mathbf{n}| (u - \hat{u}) \rrbracket, \mu \rangle_e = 0, \quad \forall \mu \in \mathcal{P}^p(e).$$

Inspired by the upwind HDG approach [1] and the optimized Schwarz method [9], we introduce an eHDG iterative method for the transport equation (3.1) as in Algorithm 1. In particular, the approximate solution u^{k+1} at the $(k+1)$ th iteration restricted on element K is defined as the solution of the following local equation, $\forall v \in \mathcal{P}^p(K)$,

$$-(u^{k+1}, \nabla \cdot (\boldsymbol{\beta} v))_K + \langle \boldsymbol{\beta} \cdot \mathbf{n} u^{k+1} + |\boldsymbol{\beta} \cdot \mathbf{n}| (u^{k+1} - \hat{u}^k), v \rangle_{\partial K} = (f, v)_K, \quad (3.3)$$

where, by introducing the average operator as $2 \llbracket (\cdot) \rrbracket := (\cdot)^- + (\cdot)^+$, we define

$$\hat{u}^k := \llbracket u^k \operatorname{sgn}(\boldsymbol{\beta} \cdot \mathbf{n}) \rrbracket + \llbracket u^k \rrbracket. \quad (3.4)$$

Algorithm 1 eHDG solver for transport equation (3.1)

Ensure: Given initial guess u^0 , compute the initial trace \hat{u}^0 using (3.4).

- 1: **while** not converged **do**
 - 2: Solve the local equation (3.3) for u^{k+1} using trace \hat{u}^k
 - 3: Compute \hat{u}^{k+1} using (3.4).
 - 4: Check convergence. If yes, **exit**, otherwise **continue**
 - 5: **end while**
-

Since (3.1) is linear, it is sufficient to show that eHDG converges for the homogeneous equation with zero forcing f and zero boundary condition g .

THEOREM 3.1. *Assume $-\nabla \cdot \boldsymbol{\beta} \geq \alpha > 0$, i.e. (3.1) is well-posed. The above eHDG for homogeneous transport equation (3.1) converges exponentially with respect to the number of iterations k . In particular, there exist $J \leq N_{el}$ such that*

$$\|u^k\|_{\frac{-\nabla \cdot \boldsymbol{\beta}}{2}, \Omega_h}^2 + \|u^k\|_{|\boldsymbol{\beta} \cdot \mathbf{n}|, \mathcal{E}_h}^2 \leq \frac{C(k)}{2^k} \|u^0\|_{|\boldsymbol{\beta} \cdot \mathbf{n}|, \mathcal{E}_h}^2, \quad (3.5)$$

where $C(k)$ is a polynomial in k of order at most J and is independent of p .

We next consider the following oceanic linearized shallow water systems

$$\frac{\partial}{\partial t} \begin{pmatrix} \phi \\ \Phi u \\ \Phi v \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \Phi u \\ \Phi \phi \\ 0 \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} \Phi v \\ 0 \\ \Phi \phi \end{pmatrix} = \begin{pmatrix} 0 \\ f \Phi v - \gamma \Phi u + \frac{\tau_x}{\rho} \\ -f \Phi u - \gamma \Phi v + \frac{\tau_y}{\rho} \end{pmatrix} \quad (3.6)$$

where $\phi = gH$ is the geopotential height with g and H being the gravitational constant and the perturbation of the free surface height, $\Phi > 0$ is a constant mean

flow geopotential height, $\boldsymbol{\vartheta} := (u, v)$ is the perturbed velocity, $\gamma \geq 0$ is the bottom friction, $\boldsymbol{\tau} := (\tau_x, \tau_y)$ is the wind stress, and ρ is the density of the water. Here, $f = f_0 + \beta(y - y_m)$ is the Coriolis parameter, where f_0 , β , and y_m are given constants.

Again, for simplicity of the exposition and analysis, let us employ the backward Euler discretization for temporal derivatives and HDG for spatial ones. Since the unknowns of interest are those at the $(m+1)$ th time step, we can suppress the time index for clarity of the exposition. Furthermore, since the system (3.6) is linear, a similar argument as in the previous sections shows that it is sufficient to consider homogeneous system with zero initial condition, boundary condition, and forcing. An eHDG algorithm can be proposed for the homogeneous system as follows

$$\begin{aligned} \left(\frac{\phi^{k+1}}{\Delta t}, \varphi_1 \right)_K - \left(\Phi \boldsymbol{\vartheta}^{k+1}, \nabla \varphi_1 \right)_K + \left\langle \Phi \boldsymbol{\vartheta}^{k+1} \cdot \mathbf{n} + \sqrt{\Phi} (\phi^{k+1} - \hat{\phi}^k), \varphi_1 \right\rangle_{\partial K} &= 0, \\ \left(\frac{\Phi u^{k+1}}{\Delta t}, \varphi_2 \right)_K - \left(\Phi \phi^{k+1}, \frac{\partial \varphi_2}{\partial x} \right)_K + \left\langle \Phi \hat{\phi}^k \mathbf{n}_1, \varphi_2 \right\rangle_{\partial K} &= (f \Phi v^{k+1} - \gamma \Phi u^{k+1}, \varphi_2)_K, \\ \left(\frac{\Phi v^{k+1}}{\Delta t}, \varphi_3 \right)_K - \left(\Phi \phi^{k+1}, \frac{\partial \varphi_3}{\partial y} \right)_K + \left\langle \Phi \hat{\phi}^k \mathbf{n}_2, \varphi_3 \right\rangle_{\partial K} &= (-f \Phi u^{k+1} - \gamma \Phi v^{k+1}, \varphi_3)_K, \end{aligned}$$

where φ_1, φ_2 and φ_3 are the test functions, and similar to the transport equation we define

$$\hat{\phi}^k = \{\{\phi^k\}\} + \sqrt{\Phi} \{\{\boldsymbol{\vartheta}^k \cdot \mathbf{n}\}\}.$$

Our goal is to show that $(\phi^{k+1}, \Phi \boldsymbol{\vartheta}^{k+1})$ converges to zero.

THEOREM 3.2. *With a suitable mesh size h , the time step Δt and the order p , then the approximate solution at the k th iteration $(\phi^k, \boldsymbol{\vartheta}^k)$ decays exponentially and the rate depends on h, p and Δt .*

4. Numerical results. In this section various numerical results supporting the theoretical results are provided for 2D and 3D transport equations and the linearized shallow water equation.

4.1. 2D steady state transport equation with smooth solution. In this example we choose $\boldsymbol{\beta} = (y, x)$. Also we take the forcing and the exact solution to be of the following form:

$$u^e = \frac{1}{\pi} \sin(\pi x) \cos(\pi y), \quad (4.1a)$$

$$f = y \cos(\pi x) \cos(\pi y) - x \sin(\pi x) \sin(\pi y). \quad (4.1b)$$

Here the domain Ω is $[0, 1] \times [0, 1]$ with $x = 0$ and $y = 0$ as inflow boundaries. A structured quadrilateral mesh is used for all the numerical simulations performed. Figure 4.1 shows the h -convergence of the HDG discretization with eHDG iterative solver. The convergence is optimal i.e. $(p+1)$ for a polynomial order p . The tolerance criteria for the eHDG solver is set as follows:

$$\|u^k - u_e\|_{L^2(\Omega)} - \|u^{k-1} - u_e\|_{L^2(\Omega)} < 10^{-10}. \quad (4.2)$$

Thus the successive difference in L^2 norm of error between numerical solution and exact solution is used as a criteria for tolerance in this case.

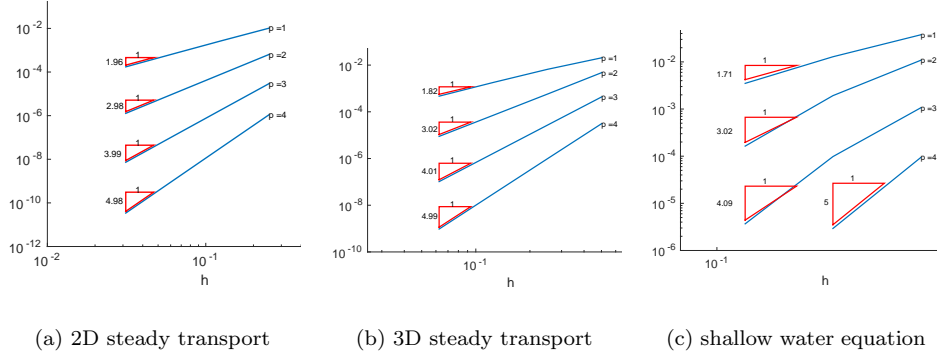


Fig. 4.1: h-convergence of the HDG method using the eHDG solver.

Figure 4.2 shows the convergence history of the eHDG solver in the log-linear scale. As proved in Theorem 3.1 the eHDG is exponential convergent in the iteration k . Also the stagnation region observed near the end of each curve is due to the fact that for a particular mesh size h and polynomial order p we can achieve only as much accuracy as prescribed by the HDG discretization error and cannot go beyond that. The numerical results for different solution orders also verify the fact that the convergence of eHDG method is independent of the polynomial order p . This can also be seen from the 4th column of Table 4.1.

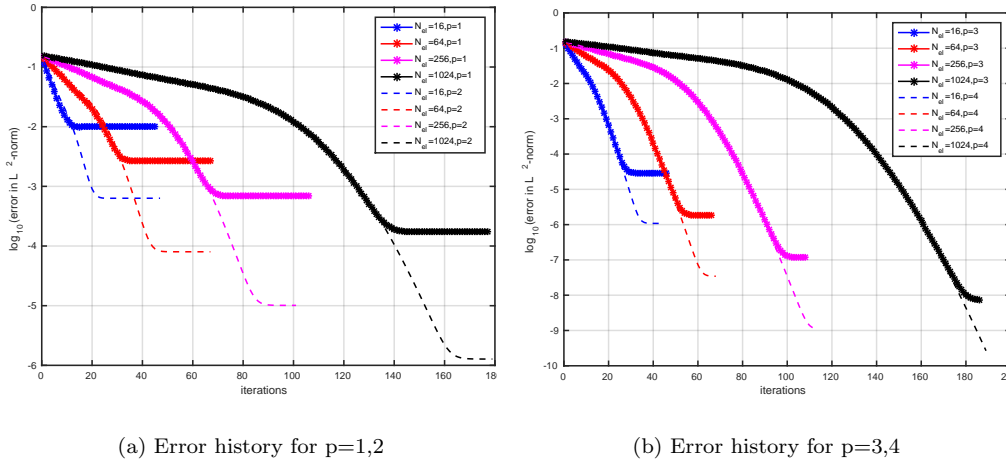


Fig. 4.2: Convergence of eHDG for different h and p for 2D transport smooth solution.

Table 4.1: *Iterations for 2D transport equation with smooth and discontinuous solutions and 3D steady state transport equation*

| $N_{el}(2D)$ | $N_{el}(3D)$ | p | 2D smooth | 2D discontinuous | 3D steady |
|--------------|--------------|---|-----------|------------------|-----------|
| 16 | 8 | 1 | 45 | 59 | 33 |
| 64 | 64 | 1 | 67 | 84 | 51 |
| 256 | 512 | 1 | 107 | 129 | 79 |
| 1024 | 4096 | 1 | 177 | 209 | 130 |
| 16 | 8 | 2 | 47 | 61 | 39 |
| 64 | 64 | 2 | 67 | 87 | 51 |
| 256 | 512 | 2 | 101 | 133 | 76 |
| 1024 | 4096 | 2 | 179 | 214 | 131 |
| 16 | 8 | 3 | 46 | 65 | 39 |
| 64 | 64 | 3 | 66 | 92 | 49 |
| 256 | 512 | 3 | 108 | 135 | 79 |
| 1024 | 4096 | 3 | 186 | 211 | 136 |
| 16 | 8 | 4 | 45 | 66 | 35 |
| 64 | 64 | 4 | 68 | 90 | 51 |
| 256 | 512 | 4 | 112 | 128 | 83 |
| 1024 | 4096 | 4 | 189 | 198 | 143 |

4.2. 2D steady state transport equation with discontinuous solution.

In this case we take $f = 0$ and $\beta = (1 + \sin(\pi y/2), 2)$. The domain Ω is $[0, 2] \times [0, 2]$ and the inflow boundary condition is given as

$$g = \begin{cases} 1 & x = 0, 0 \leq y \leq 2 \\ \sin^6(\pi x) & 0 < x \leq 1, y = 0 \\ 0 & 1 \leq x \leq 2, y = 0 \end{cases}.$$

We choose a slight different stopping criteria to avoid the exact solution:

$$\|u^k - u^{k-1}\|_{L^2(\Omega)} < 10^{-10}.$$

The evolution of solution with iterations obtained for 32×32 elements and polynomial order 4 is shown in Figure 4.3. As shown from the 5th column of Table 4.1, due to the discontinuity, the eHDG solver takes a slightly more iterations compared to the smooth solution case, but the number of iteration is still (almost) independent of the solution order. Also we observe that the solution evolves from inflow to outflow. This can be proved rigorously, but for the space limitation, the proof is omitted.

4.3. 3D steady state transport equation. In this example we choose $\beta = (z, x, y)$ in (3.1) and the following exact solution:

$$u^e = \frac{1}{\pi} \sin(\pi x) \cos(\pi y) \sin(\pi z).$$

The forcing is selected in such a way that it corresponds to the exact solution selected. Here, the domain is $[0, 1] \times [0, 1] \times [0, 1]$ with $x = 0$, $y = 0$ and $z = 0$ as inflow boundaries. A structured $16 \times 16 \times 16$ hexahedral mesh is used for all simulations. The tolerance criteria used is same as in section 4.1. Similar to the 2D example in section 4.1, we obtain the optimal convergence rates as shown in Figure 4.1(b).

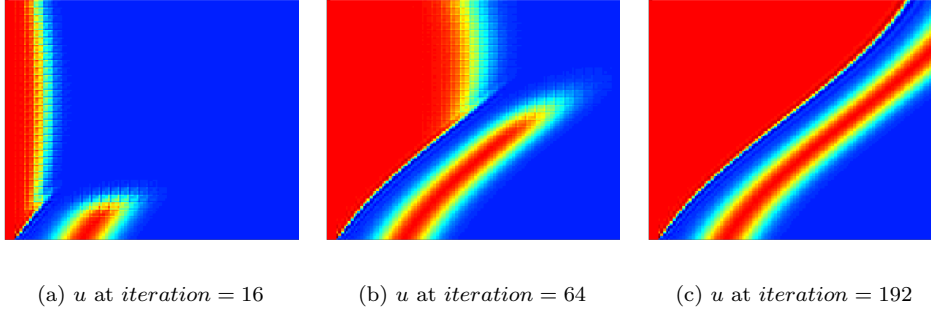


Fig. 4.3: Evolution of solution with respect to iterations for upwind HDG

The convergence is independent of the polynomial order (see also the 6th column of Table 4.1). The evolution of the eHDG solution with respect to iterations in Figure 4.4 shows the convergence of solution from inflow to outflow. Here, the solution order is $p = 4$.

4.4. 2D linearized shallow water equations. Here we consider equation (3.6), and in that we are considering a linear standing wave, which is an oceanic flow. For linear standing wave we take $\Phi = g = 1$, $f = 0$, $\gamma = 0$ (zero bottom friction), $\tau = 0$ (zero wind stress). The domain is $[0, 1] \times [0, 1]$ and wall boundary condition is applied on the boundary. The following exact solution is taken

$$\phi = \cos(\pi x) \cos(\pi y) \cos(\sqrt{2}\pi t), \quad (4.3a)$$

$$u = \frac{1}{\sqrt{2}} \sin(\pi x) \cos(\pi y) \sin(\sqrt{2}\pi t), \quad (4.3b)$$

$$v = \frac{1}{\sqrt{2}} \cos(\pi x) \sin(\pi y) \sin(\sqrt{2}\pi t). \quad (4.3c)$$

The convergence of the L^2 norm of the solution is presented in Figure 4.1. Here we have taken $\Delta t = 10^{-6}$ and 10^5 time steps in order to show the theoretical convergence rates and from Figure 4.1(c) we see that optimal convergence rate is obtained. The number of iterations required per time step in this case is constant and is always equal to 2 for all meshes and polynomial orders considered. The reason is that the initial guess for each time step is taken as the solution in the previous time step. Furthermore, the time step is small.

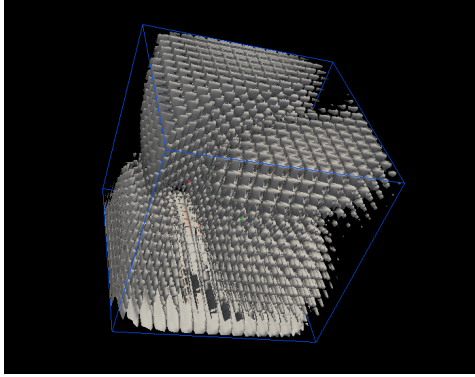
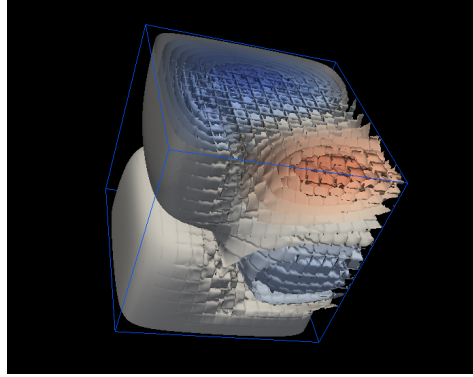
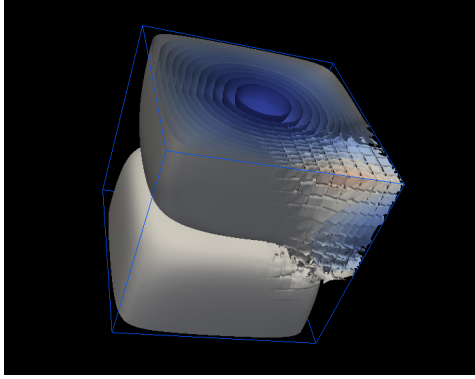
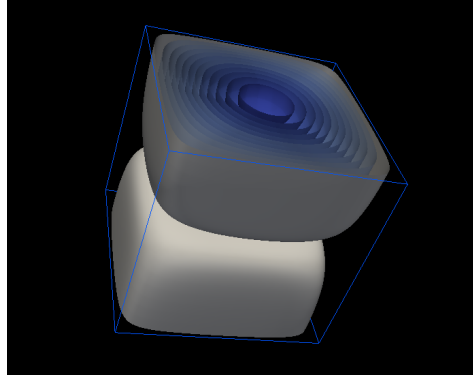
(a) u at $iteration = 1$ (b) u at $iteration = 16$ (c) u at $iteration = 48$ (d) u at $iteration = 143$

Fig. 4.4: Evolution of iterative eHDG solutions for 3D steady state transport equation.

Table 4.2: *Iterations per time step for 2D linearized shallow water equation and 3D time dependent advection for different Δt*

| $N_{el}(2D)$ | $N_{el}(3D)$ | p | 2D Shallow water | | 3D advection | |
|--------------|--------------|---|----------------------|----------------------|----------------------|----------------------|
| | | | $\Delta t = 10^{-3}$ | $\Delta t = 10^{-4}$ | $\Delta t = 10^{-3}$ | $\Delta t = 10^{-4}$ |
| 16 | 8 | 1 | 3 | 2 | 2 | 2 |
| 64 | 64 | 1 | 4 | 2 | 3 | 2 |
| 256 | 512 | 1 | 4 | 3 | 3 | 2 |
| 1024 | 4096 | 1 | 4 | 3 | 3 | 2 |
| 16 | 8 | 2 | 4 | 2 | 2 | 2 |
| 64 | 64 | 2 | 4 | 2 | 3 | 2 |
| 256 | 512 | 2 | 5 | 2 | 3 | 2 |
| 1024 | 4096 | 2 | 6 | 3 | 3 | 2 |
| 16 | 8 | 3 | 4 | 2 | 3 | 2 |
| 64 | 64 | 3 | 4 | 2 | 3 | 2 |
| 256 | 512 | 3 | 5 | 3 | 3 | 2 |
| 1024 | 4096 | 3 | 6 | 3 | 4 | 2 |
| 16 | 8 | 4 | 4 | 2 | 3 | 2 |
| 64 | 64 | 4 | 5 | 3 | 3 | 2 |
| 256 | 512 | 4 | 6 | 3 | 3 | 2 |
| 1024 | 4096 | 4 | 7 | 3 | 4 | 2 |

To compare with the 3D time-dependent advection in the next section, we choose the time step of $\Delta t = 10^{-3}$ and $\Delta t = 10^{-4}$, and tabulate the number of eHDG iterations in Table 4.2. As can be seen, the number of iterations increases slightly as we increase the solution order, and this is consistent with Theorem 3.2.

4.5. 3D time dependent transport equation. In this section we consider the following time-dependent transport equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\beta u) = f, \quad (4.4)$$

and the exact solution is a Gaussian moving across the diagonal of a unit cube, i.e.,

$$u^e = e^{-5((x-0.2t)^2 + (y-0.2t)^2 + (z-0.2t)^2)},$$

Structured hexahedral mesh $8 \times 8 \times 8$ is used and the solution order is $p = 4$. The time step is chosen $\Delta t = 0.01$ and the simulation is run for 240 time steps. Figure 4.5 compares the numerical solution using the eHDG iterative solver and the exact solution. The tolerance criteria the same as in Section 4.1, and the solver always takes 9 iterations per time step. In table 4.2 we compare the iterations per time step required to converge for two smaller time step sizes. Unlike shallow water equation, transport equation with eHDG iterative solver has constant eHDG iterations as the solution increases, and this is consistent with our theoretical result in Theorem 3.5.

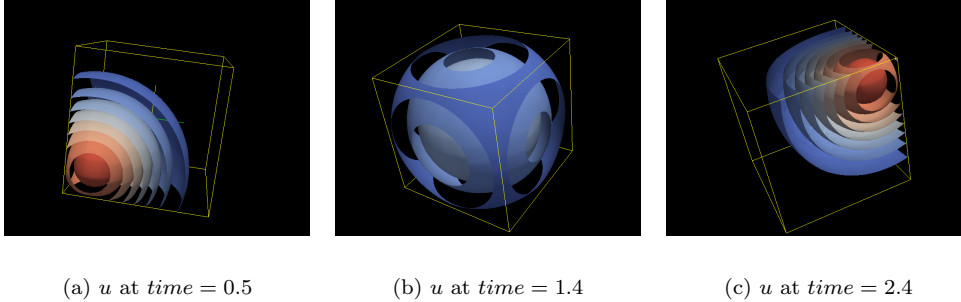


Fig. 4.5: eHDG solution for 3D time dependent transport equation

5. Conclusion. We have presented an iterative solver, namely eHDG, for HDG discretizations of hyperbolic systems. The method exploits the structure of HDG discretization and idea from domain decomposition methods. The key features of the eHDG solver are: 1) it solves independent element-by-element local equations during each iteration, 2) the number of iterations are independent of polynomial order, and 3) it achieves exponential convergence rate. These features make the eHDG solver naturally suitable for higher order HDG methods in large scale parallel environments.

- [1] T. BUI-THANH, From Godunov to a unified hybridized discontinuous Galerkin framework for partial differential equations, Journal of Computational Physics, 295 (2015), pp. 114–146.
- [2] BERNARDO COCKBURN, JAY GOPALAKRISHNAN, AND RAYTCHO LAZAROV, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal., 47 (2009), pp. 1319–1365.
- [3] BERNARDO COCKBURN, GEORGE E. KARNIADAKIS, AND CHI-WANG SHU, Discontinuous Galerkin Methods: Theory, Computation and Applications, Lecture Notes in Computational Science and Engineering, Vol. 11, Springer Verlag, Berlin, Heidelberg, New York, 2000.
- [4] MARTIN J. GANDER AND SOHEIL HAJIAN, Analysis of Schwarz methods for a hybridizable discontinuous Galerkin discretization, SIAM J. Numer. Anal., 53 (2015), pp. 573–597.
- [5] P. LESAINT AND P. A. RAVIART, On a finite element method for solving the neutron transport equation, in Mathematical Aspects of Finite Element Methods in Partial Differential Equations, C. de Boor, ed., Academic Press, 1974, pp. 89–145.
- [6] P.-L. LIONS, On the Schwarz alternating method. I, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987), SIAM, Philadelphia, PA, 1988, pp. 1–42.
- [7] N. C. NGUYEN, J. PERAIRE, AND B. COCKBURN, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations, Journal Computational Physics, 230 (2011), pp. 1147–1170.
- [8] W. H. REED AND T. R. HILL, Triangular mesh methods for the neutron transport equation, Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [9] MINH-BINH TRAN, Parallel Schwarz waveform relaxation method for a semilinear heat equation in a cylindrical domain, C. R. Math. Acad. Sci. Paris, 348 (2010), pp. 795–799.