Journal of Computational Physics 367 (2018) 295–321

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An improved iterative HDG approach for partial differential equations [☆]

Sriramkrishnan Muralikrishnan [a,*], Minh-Binh Tran [c], Tan Bui-Thanh [a,b]

[a] *Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, TX 78712, USA*
[b] *The Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX 78712, USA*
[c] *Department of Mathematics, University of Wisconsin, Madison, WI 53706, USA*

A R T I C L E   I N F O

Article history:
Received 14 November 2017
Received in revised form 23 March 2018
Accepted 18 April 2018
Available online 23 April 2018

Keywords:
Iterative solvers
Schwarz methods
Hybridized Discontinuous Galerkin methods
Transport equation
Shallow water equation
Convection–diffusion equation

A B S T R A C T

We propose and analyze an iterative high-order hybridized discontinuous Galerkin (iHDG) discretization for linear partial differential equations. We improve our previous work [45] in several directions: 1) the improved iHDG approach converges in a finite number of iterations for the scalar transport equation; 2) it is unconditionally convergent for both the linearized shallow water system and the convection–diffusion equation; 3) it has improved stability and convergence rates; 4) we uncover a relationship between the number of iterations and time stepsize, solution order, meshsize and the equation parameters. This allows us to choose the time stepsize such that the number of iterations is approximately independent of the solution order and the meshsize; and 5) we provide both strong and weak scalings of the improved iHDG approach up to 16,384 cores. A connection between iHDG and time integration methods such as parareal and implicit/explicit methods are discussed. Extensive numerical results for linear (and nonlinear) PDEs are presented to verify the theoretical findings.

1. Introduction

Originally developed [1] for the neutron transport equation, first analyzed in [2,3], the discontinuous Galerkin (DG) method has been studied extensively for virtually all types of partial differential equations (PDEs) [4–8]. This is due to the fact that DG combines advantages of finite volume and finite element methods. As such, it is well-suited to problems with large gradients including shocks and with complex geometries, and large-scale simulations demanding parallel implementations. In spite of these advantages, DG methods for steady state and/or time-dependent problems that require implicit time-integrators are more expensive in comparison to other existing numerical methods since they typically have many more (coupled) unknowns.

As an effort to mitigate the computational expense associated with DG methods, the hybridized (also known as hybridizable) discontinuous Galerkin (HDG) methods are introduced for various types of PDEs including Poisson-type equation [9–14], Stokes equation [15,16], Euler and Navier–Stokes equations, wave equations [17–23], to name a few. In [24–26], we have proposed an upwind HDG framework that provides a unified and a systematic construction of HDG methods for a large class of PDEs. We note that the weak Galerkin methods in [27–30] share many similar advantages with HDG.

[☆] This research was partially supported by DOE grants DE-SC0010518 and DE-SC0011118, NSF Grants NSF-DMS1620352 and RNMS (Ki-Net) 1107444. We are grateful for the supports.
[*] Corresponding author.
E-mail address: sriramkrishnan@utexas.edu (S. Muralikrishnan).

https://doi.org/10.1016/j.jcp.2018.04.033
0021-9991/© 2018 Elsevier Inc. All rights reserved.

Besides the usual DG volume unknown, HDG methods introduce extra single-valued trace unknowns on the mesh skeleton to reduce the number of coupled degrees of freedom and to promote further parallelism. This is accomplished via a Schur complement approach in which the volume unknowns on each elements are independently eliminated in parallel to provide a system of equations involving only the trace unknowns. Moreover, the trace system is substantially smaller and sparser compared to a standard DG linear system. Once the trace unknowns are solved for, the volume unknowns can be recovered in an element-by-element fashion, completely independent of each other. For small and medium sized problems the above approach is popular. However, for practically large-scale applications, where complex and high-fidelity simulations involving features with a large range of spatial and temporal scales are necessary, the trace system is still a bottleneck. In this case, matrix-free iterative solvers/preconditioners [31–33] which converge in reasonably small number of iterations are required.

Schwarz-type domain decomposition methods (DDMs) have become popular during the last three decades as they provide efficient algorithms to parallelize and to solve PDEs [34–36]. Schwarz waveform relaxation methods and optimized Schwarz methods [37–43] are among the most important subclasses of DDMs since they can be adapted to the underlying physics, and thus lead to powerful parallel solvers for challenging problems. While, scalable iterative solvers/preconditioners for the statically condensed trace system can be developed [44], DDMs and HDG have a natural connection which can be exploited to create efficient parallel solvers. We have developed and analyzed one such solver namely iterative HDG (iHDG) in our previous work [45] for elliptic, scalar and system of hyperbolic equations. Independent and similar efforts for elliptic and parabolic equations have been proposed and analyzed in [46,39,47].

In the following, we discuss in section 2 an upwind HDG framework [24] for a general class of PDEs and also our notations used in this paper. The iHDG algorithm and significant improvements over our previous work [45] are explained in section 3. The convergence of the new iHDG algorithm for the scalar and for system of hyperbolic PDEs is proved in section 4 using an energy approach. In section 5 we applied the iHDG approach for the convection–diffusion PDE considered in the first order form. The convergence and the scaling of the number of iHDG iterations with meshsize and solution order are derived. Section 6 presents various steady and time dependent examples, in both two and three spatial dimensions, to support the theoretical findings. We also present both strong and weak scaling results of our algorithm up to 16,384 cores in section 6. We finally conclude the paper in section 7.

2. Upwind HDG framework

In this section we briefly review the upwind HDG framework for a general system of linear PDEs and introduce necessary notations. To begin, let us consider the following system of first order PDEs

$$\partial_t \mathbf{u} + \sum_{k=1}^{d} \partial_k \mathbf{F}_k(\mathbf{u}) + \mathbf{C}\mathbf{u} := \partial_t \mathbf{u} + \sum_{k=1}^{d} \partial_k (\mathbf{A}_k \mathbf{u}) + \mathbf{C}\mathbf{u} = \mathbf{f}, \quad \text{in } \Omega, \tag{1}$$

where d is the spatial dimension (which, for the clarity of the exposition, is assumed to be $d = 3$ whenever a particular value of the dimension is of concern, but the result is also valid for $d = \{1, 2\}$), $\mathbf{F}_k$ the kth component of the flux vector (or tensor) $\mathbf{F}$, $\mathbf{u}$ the unknown solution with values in $\mathbb{R}^m$, and $\mathbf{f}$ the forcing term. For simplicity, we assume that the matrices $\mathbf{A}_k$ and $\mathbf{C}$ are continuous across Ω. Here, ∂_k stands for the kth partial derivative, in which k represents the kth component of a vector/tensor, and ∂_t is the temporal derivative. We adopt a semi-discretization strategy in which the HDG method is employed to discretize the spatial derivatives, while standard discretizations such as backward Euler and/or Crank–Nicolson are used for the temporal derivative.

Let us start with notations and conventions. We partition $\Omega \in \mathbb{R}^d$, an open and bounded domain, into N_{el} non-overlapping elements $K_j, j = 1, \ldots, N_{\text{el}}$ with Lipschitz boundaries such that $\Omega_h := \cup_{j=1}^{N_{\text{el}}} K_j$ and $\overline{\Omega} = \overline{\Omega}_h$. The meshsize h is defined as $h := \max_{j \in \{1,\ldots,N_{\text{el}}\}} diam\left(K_j\right)$. 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 outward normal vector on the boundary ∂K of element K (also denoted as K^-) and $\mathbf{n}^+ = -\mathbf{n}^-$ as the outward 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 convenience, we denote by $\mathcal{E}_h^\partial$ the sets of all boundary faces on $\partial\Omega$, by $\mathcal{E}_h^o := \mathcal{E}_h \setminus \mathcal{E}_h^\partial$ the set of all interior faces, and $\partial\Omega_h := \{\partial K : K \in \Omega_h\}$.

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 the inner product from which the norm is coming. We also denote the ε-weighted norm of a function u as $\|u\|_{\varepsilon,K} := \left\|\sqrt{\varepsilon} u\right\|_K$ 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 ($\mathbf{u}_i, i = 1, \ldots, m$) of $\mathbf{u}$. Moreover, we define $(\mathbf{u}, \mathbf{v})_{\Omega_h} := \sum_{K \in \Omega_h} (\mathbf{u}, \mathbf{v})_K$ 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. We conventionally use $\mathbf{u}$ ($\mathbf{v}$ and $\hat{\mathbf{u}}$) for the numerical solution and $\mathbf{u}^e$ for the exact solution.

We denote by $\mathcal{P}^p(K)$ the space of polynomials of degree at most p on a domain K. Next, we introduce two discontinuous piecewise polynomial spaces

$$\mathbf{V}_h(\Omega_h) := \left\{\mathbf{v} \in \left[L^2(\Omega_h)\right]^m : \mathbf{v}|_K \in \left[\mathcal{P}^p(K)\right]^m, \forall K \in \Omega_h\right\},$$
$$\boldsymbol{\Lambda}_h(\mathcal{E}_h) := \left\{\boldsymbol{\lambda} \in \left[L^2(\mathcal{E}_h)\right]^m : \boldsymbol{\lambda}|_e \in \left[\mathcal{P}^p(e)\right]^m, \forall e \in \mathcal{E}_h\right\},$$

and similar spaces $\mathbf{V}_h(K)$ and $\boldsymbol{\Lambda}_h(e)$ on K and e by replacing Ω_h with K and $\mathcal{E}_h$ with e. For scalar-valued functions, we denote the corresponding spaces as

$$V_h(\Omega_h) := \left\{v \in L^2(\Omega_h) : v|_K \in \mathcal{P}^p(K), \forall K \in \Omega_h\right\},$$
$$\Lambda_h(\mathcal{E}_h) := \left\{\lambda \in L^2(\mathcal{E}_h) : \lambda|_e \in \mathcal{P}^p(e), \forall e \in \mathcal{E}_h\right\}.$$

Following [24], an upwind HDG discretization for (1) in each element K involves the DG local unknown $\mathbf{u}$ and the extra "trace" unknown $\hat{\mathbf{u}}$ such that

$$(\partial_t \mathbf{u}, \mathbf{v})_K - (\mathbf{F}(\mathbf{u}), \nabla\mathbf{v})_K + \left\langle \hat{\mathbf{F}}(\mathbf{u}, \hat{\mathbf{u}}) \cdot \mathbf{n}, \mathbf{v}\right\rangle_{\partial K} + (\mathbf{C}\mathbf{u}, \mathbf{v})_K = (\mathbf{f}, \mathbf{v})_K, \tag{2a}$$
$$\left\langle \left[\!\left[\hat{\mathbf{F}}(\mathbf{u}, \hat{\mathbf{u}}) \cdot \mathbf{n}\right]\!\right], \boldsymbol{\mu}\right\rangle_e = \mathbf{0}, \quad \forall e \in \mathcal{E}_h^o, \tag{2b}$$

where we have defined the "jump" operator for any quantity $(\cdot)$ as $[\![(\cdot)]\!] := (\cdot)^- + (\cdot)^+$. We also define the "average" operator $\{\!\{(\cdot)\}\!\}$ via $2\{\!\{(\cdot)\}\!\} := [\![(\cdot)]\!]$. For simplicity, we have ignored the fact that equations (2a), (2b) must hold for all test functions $\mathbf{v} \in \mathbf{V}_h(K)$ and $\boldsymbol{\mu} \in \boldsymbol{\Lambda}_h(e)$ respectively. This is implicitly understood throughout the paper. Here, the HDG flux is defined as

$$\hat{\mathbf{F}} \cdot \mathbf{n} = \mathbf{F}(\mathbf{u}) \cdot \mathbf{n} + |\mathbf{A}|\left(\mathbf{u} - \hat{\mathbf{u}}\right), \tag{3}$$

with the matrix $\mathbf{A} := \sum_{k=1}^d \mathbf{A}_k \mathbf{n}_k = \mathbf{R}\mathbf{S}\mathbf{R}^{-1}$, and $|\mathbf{A}| := \mathbf{R}|\mathbf{S}|\mathbf{R}^{-1}$. Here $\mathbf{n}_k$ is the kth component of the outward normal vector $\mathbf{n}$ and $|\mathbf{S}|$ represents a matrix obtained by taking the absolute value of the main diagonal of the matrix $\mathbf{S}$. We have assumed that $\mathbf{A}$ admits an eigen-decomposition, and this is valid for a large class of PDEs of Friedrichs' type [48].

The typical procedure for computing HDG solution requires three steps. We first solve (2a) for the local solution $\mathbf{u}$ as a function of $\hat{\mathbf{u}}$. It is then substituted into the conservative algebraic equation (2b) on the mesh skeleton to solve for the unknown $\hat{\mathbf{u}}$. Finally, the local unknown $\mathbf{u}$ is computed, as in the first step, using $\hat{\mathbf{u}}$ from the second step. Since the number of trace unknowns $\hat{\mathbf{u}}$ is less than the DG unknowns $\mathbf{u}$ [26], HDG is more advantageous. For large-scale problems, however, the trace system on the mesh skeleton could be large and iterative solvers are necessary. In the following we construct an iterative solver that takes advantage of the HDG structure and the domain decomposition method.

3. iHDG methods

To reduce the cost of solving the trace system, our previous effort [45] is to break the coupling between $\hat{\mathbf{u}}$ and $\mathbf{u}$ in (2) by iteratively solving for $\mathbf{u}$ in terms of $\hat{\mathbf{u}}$ in (2a), and $\hat{\mathbf{u}}$ in terms of $\mathbf{u}$ in (2b). We name this approach iterative HDG (iHDG) method, and now let us call it iHDG-I to distinguish it from the approach developed in this paper. From a linear algebra point of view, iHDG-I can be considered as a block Gauss–Seidel approach for the system (2) that requires only independent element-by-element and face-by-face local solves in each iteration. However, unlike conventional Gauss–Seidel schemes which are purely algebraic, the convergence of iHDG-I [45] does not depend on the ordering of unknowns. From the domain decomposition point of view, thanks to the HDG flux, iHDG can be identified as an optimal Schwarz method in which each element is a subdomain. Using an energy approach, we have rigorously shown the convergence of iHDG-I for the transport equation, the linearized shallow water equation and the convection–diffusion equation [45]. For the sake of completeness we provide the iHDG-I algorithm below. The approximation of the HDG solution at the $(k+1)$th iteration is governed by the local equation (2a) as

$$\left(\partial_t \mathbf{u}^{k+1}, \mathbf{v}\right)_K - \left(\mathbf{F}\left(\mathbf{u}^{k+1}\right), \nabla\mathbf{v}\right)_K + \left\langle \mathbf{F}\left(\mathbf{u}^{k+1}\right) \cdot \mathbf{n} + |\mathbf{A}|\,(\mathbf{u}^{k+1} - \hat{\mathbf{u}}^k), \mathbf{v}\right\rangle_{\partial K} + \left(\mathbf{C}\mathbf{u}^{k+1}, \mathbf{v}\right)_K = (\mathbf{f}, \mathbf{v})_K, \tag{4}$$

where the weighted trace $|\mathbf{A}|\,\hat{\mathbf{u}}^k$ is computed using information at the k-iteration via the conservation condition (2b), i.e.,

$$\left\langle |\mathbf{A}|\,\hat{\mathbf{u}}^k, \boldsymbol{\mu}\right\rangle_{\partial K} = \left\langle \left\{\!\!\left\{|\mathbf{A}|\,\mathbf{u}^k\right\}\!\!\right\} + \left\{\!\!\left\{\mathbf{F}\left(\mathbf{u}^k\right) \cdot \mathbf{n}\right\}\!\!\right\}, \boldsymbol{\mu}\right\rangle_{\partial K}. \tag{5}$$

Algorithm 1 summarizes the iHDG-I approach. Nevertheless, *a number of questions need to be addressed for the iHDG-I approach*. First, with the upwind flux it theoretically takes infinite number of iterations to converge for the scalar transport equation. Second, it is conditionally convergent for the linearized shallow water system; in particular, it blows up for fine meshes

Algorithm 1 The iHDG-I approach.

```
Ensure: Given initial guess $\mathbf{u}^0$, compute the weighted trace $|\mathbf{A}|\,\hat{\mathbf{u}}^0$ using (5).
1: while not converged do
2:     Solve the local equation (4) for $\mathbf{u}^{k+1}$ using the weighted trace $|\mathbf{A}|\,\hat{\mathbf{u}}^k$.
3:     Compute $|\mathbf{A}|\,\hat{\mathbf{u}}^{k+1}$ using (5).
4:     Check convergence. If yes, exit, otherwise set $k = k + 1$ and continue.
5: end while
```

and/or large time stepsizes. Furthermore, we have not been able to estimate the number of iterations as a function of time stepsize, solution order, and meshsize. Third, it is also conditionally convergent for the convection–diffusion equation, especially in the diffusion-dominated regime.

The iHDG approach constructed in this paper, which we call iHDG-II, overcomes all the aforementioned shortcomings. In particular, it converges in a finite number of iterations for the scalar transport equation and is unconditionally convergent for both the linearized shallow water system and the convection–diffusion equation. Moreover, compared to our previous work [45], we provide several additional findings: 1) we make a connection between iHDG and the parareal method, which reveals interesting similarities and differences between the two methods; 2) we show that iHDG can be considered as a *locally implicit* method, and hence being somewhat in between fully explicit and fully implicit approaches; 3) for both the linearized shallow water system and the convection–diffusion equation, using an asymptotic approximation, we uncover a relationship between the number of iterations and time stepsize, solution order, meshsize and the equation parameters. This allows us to choose the time stepsize such that the number of iterations is approximately independent of the solution order and the meshsize; 4) we show that iHDG-II has improved stability and convergence rates over iHDG-I; and 5) we provide both strong and weak scalings of the iHDG-II approach up to 16,384 cores.

We now present a detailed construction of the iHDG-II approach. We define the approximate solution for the volume variables at the $(k+1)$th iteration using the local equation (2a) as

$$\left(\partial_t \mathbf{u}^{k+1}, \mathbf{v}\right)_K - \left(\mathbf{F}\left(\mathbf{u}^{k+1}\right), \nabla \mathbf{v}\right)_K + \left\langle \mathbf{F}\left(\mathbf{u}^{k+1}\right) \cdot \mathbf{n} + |\mathbf{A}|\left(\mathbf{u}^{k+1} - \hat{\mathbf{u}}^{k,k+1}\right), \mathbf{v}\right\rangle_{\partial K} + \left(\mathbf{C}\mathbf{u}^{k+1}, \mathbf{v}\right)_K = (\mathbf{f}, \mathbf{v})_K\,, \tag{6}$$

where the weighted trace $|\mathbf{A}|\,\hat{\mathbf{u}}^{k,k+1}$ is computed from (2b) using volume unknown in element K at the $(k+1)$th iteration, i.e. $\left(\mathbf{u}^{k+1}\right)^-$, and volume solution of the neighbors at the (k)th iteration, i.e. $\left(\mathbf{u}^k\right)^+$:

$$\left\langle 2\,|\mathbf{A}|\,\hat{\mathbf{u}}^{k,k+1}, \boldsymbol{\mu}\right\rangle_{\partial K} = \left\langle |\mathbf{A}|\left\{\left(\mathbf{u}^{k+1}\right)^- + \left(\mathbf{u}^k\right)^+\right\}, \boldsymbol{\mu}\right\rangle_{\partial K} + \left\langle \mathbf{F}\left\{\left(\mathbf{u}^{k+1}\right)^-\right\} \cdot \mathbf{n}^- + \mathbf{F}\left\{\left(\mathbf{u}^k\right)^+\right\} \cdot \mathbf{n}^+, \boldsymbol{\mu}\right\rangle_{\partial K}. \tag{7}$$

Algorithm 2 summarizes the iHDG-II approach. Compared to iHDG-I, iHDG-II improves the coupling between $\hat{\mathbf{u}}$ and $\mathbf{u}$ while still avoiding intra-iteration communication between elements. The trace $\hat{\mathbf{u}}$ is double-valued during the course of iterations for iHDG-II and in the event of convergence it becomes single valued upto a specified tolerance. Another principal difference is that while the well-posedness of iHDG-I elemental local solves is inherited from the original HDG counterpart, *it has to be shown for iHDG-II*. This is due to the new way of computing the weighted trace in (7) that involves $\mathbf{u}^{k+1}$, and hence changing the structure of the local solves. Similar and independent work for HDG methods for elliptic/parabolic problems have appeared in [46,39,47]. Here, we are interested in pure hyperbolic equations/systems and convection–diffusion equations. Unlike existing matrix-based approaches, our convergence analysis is based on an energy approach that exploits the variational structure of HDG methods. Moreover we provide, both rigorous and asymptotic, relationships between the number of iterations and time stepsize, solution order, meshsize and the equation parameters. We also make connection between our proposed iHDG-II approach with parareal and time integration methods. Last but not least, our framework is more general: indeed it recovers the contraction factor results in [46] for elliptic equations as one of the special cases.

Algorithm 2 The iHDG-II approach.

```
Ensure: Given initial guess $\mathbf{u}^0$, compute the weighted trace $|\mathbf{A}|\,\hat{\mathbf{u}}^{0,1}$ using (7).
1: while not converged do
2:     Solve the local equation (6) for $\mathbf{u}^{k+1}$ using the weighted trace $|\mathbf{A}|\,\hat{\mathbf{u}}^{k,k+1}$.
3:     Compute $|\mathbf{A}|\,\hat{\mathbf{u}}^{k+1,k+2}$ using (7).
4:     Check convergence. If yes, exit, otherwise set $k = k + 1$ and continue.
5: end while
```

4. iHDG-II for linear hyperbolic PDEs

In this section we show that iHDG-II improves upon iHDG-I in many aspects discussed in section 3. The PDEs of interest are (steady and time dependent) transport equation, and the linearized shallow water system [45].

4.1. Transport equation

Let us start with the (steady) transport equation

$$\boldsymbol{\beta} \cdot \nabla u^e = f \quad \text{in } \Omega, \tag{8a}$$

$$u^e = g \quad \text{on } \partial\Omega^-, \tag{8b}$$

where $\partial\Omega^-$ is the inflow part of the boundary $\partial\Omega$, and again u^e denotes the exact solution. *Note that $\boldsymbol{\beta}$ is assumed to be continuous across the mesh skeleton.*

Applying the iHDG-II Algorithm 2 to the upwind HDG discretization [24] for (8) we obtain the approximate solution u^{k+1} at the $(k+1)$th iteration restricted on each element K via the following independent local solve:

$$-\left(\left(u^{k+1}\right)^-, \nabla \cdot (\boldsymbol{\beta} v)\right)_K + \left\langle \boldsymbol{\beta} \cdot \mathbf{n}^- \left(u^{k+1}\right)^- + |\boldsymbol{\beta} \cdot \mathbf{n}| \left\{\left(u^{k+1}\right)^- - \hat{u}^{k,k+1}\right\}, v \right\rangle_{\partial K} = (f, v)_K, \tag{9}$$

where the weighted trace $|\boldsymbol{\beta} \cdot \mathbf{n}|\, \hat{u}^{k,k+1}$ is computed using information from the previous iteration and current iteration as

$$2\,|\boldsymbol{\beta} \cdot \mathbf{n}|\, \hat{u}^{k,k+1} = \left\{\boldsymbol{\beta} \cdot \mathbf{n}^- \left(u^{k+1}\right)^- + \boldsymbol{\beta} \cdot \mathbf{n}^+ \left(u^k\right)^+\right\} + |\boldsymbol{\beta} \cdot \mathbf{n}| \left\{\left(u^{k+1}\right)^- + \left(u^k\right)^+\right\}. \tag{10}$$

Next we study the convergence of the iHDG-II method (9), (10). Since (8) is linear, it is sufficient to show that the algorithm converges to the zero solution for the homogeneous equation with zero forcing $f = 0$ and zero boundary condition $g = 0$. Let us define ∂K^{out} as the outflow part of ∂K, i.e. $\boldsymbol{\beta} \cdot \mathbf{n}^- \geq 0$ on ∂K^{out}, and ∂K^{in} as the inflow part of ∂K, i.e. $\boldsymbol{\beta} \cdot \mathbf{n}^- < 0$ on ∂K^{in}. First, we will prove the well-posedness of the local solver (9).

Lemma 1. *Assume $-\nabla \cdot \boldsymbol{\beta} \geq \alpha > 0$, i.e. (8) is well-posed. Then the local solver (9) of the iHDG-II algorithm for the transport equation is well-posed.*

Proof. Taking $v = \left(u^{k+1}\right)^-$ in (9), substituting (10) in (9) and applying homogeneous forcing condition yield

$$\begin{aligned} -\left(\left(u^{k+1}\right)^-, \nabla \cdot \left\{\boldsymbol{\beta}\left(u^{k+1}\right)^-\right\}\right)_K &+ \frac{1}{2}\left\langle \left(\boldsymbol{\beta} \cdot \mathbf{n}^- + |\boldsymbol{\beta} \cdot \mathbf{n}|\right)\left(u^{k+1}\right)^-, \left(u^{k+1}\right)^- \right\rangle_{\partial K} \\ &= \frac{1}{2}\left\langle \left(\boldsymbol{\beta} \cdot \mathbf{n}^+ + |\boldsymbol{\beta} \cdot \mathbf{n}|\right)\left(u^k\right)^+, \left(u^{k+1}\right)^- \right\rangle_{\partial K}. \end{aligned} \tag{11}$$

Since

$$\left(\left(u^{k+1}\right)^-, \nabla \cdot \left\{\boldsymbol{\beta}\left(u^{k+1}\right)^-\right\}\right)_K = \left(\left(u^{k+1}\right)^-, \nabla \cdot \boldsymbol{\beta}\left(u^{k+1}\right)^-\right)_K + \left(\left(u^{k+1}\right)^-, \boldsymbol{\beta} \cdot \nabla\left(u^{k+1}\right)^-\right)_K,$$

integrating by parts the second term on the right hand side

$$\begin{aligned} \left(\left(u^{k+1}\right)^-, \nabla \cdot \left\{\boldsymbol{\beta}\left(u^{k+1}\right)^-\right\}\right)_K &= \left(\left(u^{k+1}\right)^-, \nabla \cdot \boldsymbol{\beta}\left(u^{k+1}\right)^-\right)_K \\ &\quad - \left(\left(u^{k+1}\right)^-, \nabla \cdot \left\{\boldsymbol{\beta}\left(u^{k+1}\right)^-\right\}\right)_K + \left\langle \boldsymbol{\beta} \cdot \mathbf{n}^-\left(u^{k+1}\right)^-, \left(u^{k+1}\right)^- \right\rangle_{\partial K}, \end{aligned}$$

yields the following identity, after rearranging the terms

$$\left(\left(u^{k+1}\right)^-, \nabla \cdot \left\{\boldsymbol{\beta}\left(u^{k+1}\right)^-\right\}\right)_K = \left(\left(u^{k+1}\right)^-, \frac{\nabla \cdot \boldsymbol{\beta}}{2}\left(u^{k+1}\right)^-\right)_K + \frac{1}{2}\left\langle \boldsymbol{\beta} \cdot \mathbf{n}^-\left(u^{k+1}\right)^-, \left(u^{k+1}\right)^- \right\rangle_{\partial K}. \tag{12}$$

Using (12) in (11) we get

$$\left\|\left(u^{k+1}\right)^-\right\|^2_{\frac{-\nabla\cdot\boldsymbol{\beta}}{2}, K} + \left\|\left(u^{k+1}\right)^-\right\|^2_{|\boldsymbol{\beta}\cdot\mathbf{n}|/2, \partial K} = \frac{1}{2}\left\langle \left(\boldsymbol{\beta} \cdot \mathbf{n}^+ + |\boldsymbol{\beta} \cdot \mathbf{n}|\right)\left(u^k\right)^+, \left(u^{k+1}\right)^- \right\rangle_{\partial K}. \tag{13}$$

In equation (13) all the terms on the left hand side are positive. Since $\left(u^k\right)^+$ is the "forcing" for the local equation, by taking $\left(u^k\right)^+ = 0$ the only solution possible is $\left(u^{k+1}\right)^- = 0$ and hence the local solver is well-posed. $\square$

Having proved the well-posedness of the local solver we can now proceed to prove the convergence of Algorithm 2 for the transport equation.

Theorem 1. *Assume $-\nabla\cdot\boldsymbol{\beta} \geq \alpha > 0$, i.e. (8) is well-posed. There exists $J \leq N_{el}$ such that the iHDG-II algorithm for the homogeneous transport equation converges to the HDG solution in J iterations.*

Proof. Using (13) from Lemma 1 and $\boldsymbol{\beta}\cdot\mathbf{n}^+ > 0$ on ∂K^{in}, $\boldsymbol{\beta}\cdot\mathbf{n}^+ \leq 0$ on ∂K^{out} we can write

$$\left\|\left(u^{k+1}\right)^-\right\|^2_{\frac{-\nabla\cdot\boldsymbol{\beta}}{2},K} + \left\|\left(u^{k+1}\right)^-\right\|^2_{|\boldsymbol{\beta}\cdot\mathbf{n}|/2,\partial K} = \left\langle |\boldsymbol{\beta}\cdot\mathbf{n}| u^k_{\text{ext}}, \left(u^{k+1}\right)^-\right\rangle_{\partial K^{\text{in}}}, \tag{14}$$

where u^k_{ext} is either the physical boundary condition or the solution of the neighboring element that shares the same inflow boundary ∂K^{in}.

Consider the set $\mathcal{K}^1$ of all elements K such that ∂K^{in} is a subset of the physical inflow boundary $\partial\Omega^{\text{in}}$ on which we have $u^k_{\text{ext}} = 0$ for all $k \in \mathbb{N}$. We obtain from (14) that

$$\left\|\left(u^{k+1}\right)^-\right\|^2_{\frac{-\nabla\cdot\boldsymbol{\beta}}{2},K} + \left\|\left(u^{k+1}\right)^-\right\|^2_{|\boldsymbol{\beta}\cdot\mathbf{n}|/2,\partial K} = 0, \tag{15}$$

which implies $u^1 = 0$ on $K \in \mathcal{K}^1$, i.e. our iterative solver is exact on $K \in \mathcal{K}^1$ at the first iteration.

Next, let us define $\Omega_h^1 := \Omega_h$ and

$$\Omega_h^2 = \Omega_h^1 \backslash \mathcal{K}^1.$$

Consider the set $\mathcal{K}^2$ of all K in Ω_h^2 such that ∂K^{in} is either (possibly partially) a subset of the physical inflow boundary $\partial\Omega^{\text{in}}$ or (possibly partially) a subset of the outflow boundary of elements in $\mathcal{K}^1$. This implies, on $\partial K^{\text{in}} \in \mathcal{K}^2$, $u^k_{\text{ext}} = 0$ for all $k \in \mathbb{N}\setminus\{1\}$. Thus, $\forall K \in \mathcal{K}^2$, we have

$$\left\|\left(u^{k}\right)^-\right\|^2_{\frac{-\nabla\cdot\boldsymbol{\beta}}{2},K} + \left\|\left(u^{k}\right)^-\right\|^2_{|\boldsymbol{\beta}\cdot\mathbf{n}|/2,\partial K} = 0, \quad \forall k \in \mathbb{N}\setminus\{1\}, \tag{16}$$

which implies $u^2 = 0$ in $K \in \mathcal{K}^2$, i.e. our iterative solver is exact on $K \in \mathcal{K}^2$ at the second iteration.

Repeating the same argument, we can construct subsets $\mathcal{K}^j \subset \Omega_h$, on which the iterative solution on $K \in \mathcal{K}^j$ is the exact HDG solution at the jth iteration. Since the number of elements N_{el} is finite, there exists $J \leq N_{\text{el}}$ such that $\Omega_h = \cup_{j=1}^J \mathcal{K}^j$. It follows that the iHDG-II algorithm provides exact HDG solution on Ω_h after J iterations. □

Remark 1. Compared to iHDG-I [45], which requires an infinite number of iterations to converge, iHDG-II needs finite number of iterations for convergence. The key to the improvement is the stronger coupling between $\hat{\mathbf{u}}$ and $\mathbf{u}$ by using $\left(\mathbf{u}^{k+1}\right)^-$ in (7) instead of $\left(\mathbf{u}^{k}\right)^-$. The proof of Theorem 1 also shows that iHDG-II automatically marches the flow, i.e., each iteration yields the HDG solution exactly for a group of elements. Moreover, the marching process is automatic (i.e. does not require an ordering of elements) and adapts to the velocity field $\boldsymbol{\beta}$ under consideration.

4.2. Time-dependent transport equation

In this section we first comment on a space–time formulation of the iHDG methods and compare it with the parareal methods studied in [49] for the time-dependent scalar transport equation. Then we consider the semi-discrete version of iHDG combined with traditional time integration schemes and compare it with the fully implicit and explicit DG/HDG schemes.

4.2.1. Comparison of space–time iHDG and parareal methods for the scalar transport equation

Space–time finite element methods have been studied extensively for the past several years both in the context of continuous and discontinuous Galerkin methods [50–54] and HDG methods [55]. Parareal methods, on the other hand, were first introduced in [56] and various modifications have been proposed and studied (see [57–61] and references therein).

In the scope of our work, we compare our methods with the parareal scheme proposed in [49] for the scalar advection equation. Let us start with the following ordinary differential equation

$$\frac{du}{dt} = f \quad \text{in } (0,T), \ \ u(0) = g, \tag{17}$$

for some positive constant $T > 0$.

Corollary 1. *Suppose we discretize the temporal derivative in (17) using the iHDG-II method with the upwind flux and the elements K_j are ordered such that K_j is on the left of K_{j+1}. At iteration k, $u^k\big|_{K_j}$ converges to the HDG solution $u|_{K_j}$ for $j \leq k$.*

Proof. Since (17) can be considered as 1D transport equation (8) with velocity $\boldsymbol{\beta} = 1$, the proof follows directly from Theorem 1 and induction. □

Note that the iHDG scheme can be considered as a parareal algorithm in which the fine propagator is taken to be the local solver (6) and the coarse propagator corresponds to the conservation condition (7). However, unlike existing parareal algorithms, the coarse propagator of iHDG-parareal is dependent on the fine propagator. Moreover, Corollary 1 says that after k iterations the iHDG-parareal solution converges up to element k, a feature common to the parareal algorithm studied in [49]. For time dependent hyperbolic PDEs, the space–time iHDG method again can be understood as parareal approach, and in this case, a layer of space–time elements converges after each iHDG-parareal iteration (see Remark 1). See Fig. 1 and Table 1 of section 6 for a demonstration in 2D where either x or y is considered as "time". It should be pointed out that the specific parareal method in [49] exactly traces the characteristics, and hence may take less iterations to converge than the iHDG-parareal method, but this is only true if the forward Euler discretization in time, upwind finite difference in space, and $CFL = 1$ are used with constant advection velocity.

4.3. iHDG as a locally implicit method

In this section we discuss the relationship between iHDG and implicit/explicit HDG methods. For the simplicity of the exposition, we consider time-dependent scalar transport equation given by:

$$\frac{\partial u^e}{\partial t} + \boldsymbol{\beta} \cdot \nabla u^e = f. \tag{18}$$

We first review the implicit/explicit HDG schemes for (18), and then compare them with iHDG-II. The implicit Euler HDG scheme for (18) reads

$$\begin{aligned}\left(\frac{u^{m+1}}{\Delta t}, v\right)_K - \left(u^{m+1}, \nabla \cdot (\boldsymbol{\beta} v)\right)_K + \left\langle \boldsymbol{\beta} \cdot \mathbf{n} u^{m+1} + |\boldsymbol{\beta} \cdot \mathbf{n}|(u^{m+1} - \hat{u}^{m+1}), v\right\rangle_{\partial K} &= \left(f^{m+1} + \frac{u^m}{\Delta t}, v\right)_K, \\ \left\langle [\![|\boldsymbol{\beta} \cdot \mathbf{n}| \hat{u}^{m+1}]\!], \boldsymbol{\mu}\right\rangle_{\partial K} &= \left\langle [\![|\boldsymbol{\beta} \cdot \mathbf{n}| u^{m+1}]\!] + [\![\boldsymbol{\beta} \cdot \mathbf{n} u^{m+1}]\!], \boldsymbol{\mu}\right\rangle_{\partial K}.\end{aligned} \tag{19}$$

Here, u^{m+1} and $\hat{u}^{m+1}$ stands for the volume and the trace unknowns at the current time step, whereas u^m and $\hat{u}^m$ are the computed solutions from the previous time step. Clearly, u^{m+1} and $\hat{u}^{m+1}$ are coupled and this can be a challenge for large-scale problems.

Next let us consider an explicit HDG with forward Euler discretization in time for (18):

$$\begin{aligned}\left(\frac{u^{m+1}}{\Delta t}, v\right)_K &= \left(u^m, \nabla \cdot (\boldsymbol{\beta} v)\right)_K - \left\langle \boldsymbol{\beta} \cdot \mathbf{n} u^m + |\boldsymbol{\beta} \cdot \mathbf{n}|(u^m - \hat{u}^m), v\right\rangle_{\partial K} + \left(f^m + \frac{u^m}{\Delta t}, v\right)_K, \\ \left\langle [\![|\boldsymbol{\beta} \cdot \mathbf{n}| \hat{u}^m]\!], \boldsymbol{\mu}\right\rangle_{\partial K} &= \left\langle [\![|\boldsymbol{\beta} \cdot \mathbf{n}| u^m]\!] + [\![\boldsymbol{\beta} \cdot \mathbf{n} u^m]\!], \boldsymbol{\mu}\right\rangle_{\partial K},\end{aligned}$$

which shows that we can solve for u^{m+1} element-by-element, completely independent of each other. However, since it is an explicit scheme, the CFL restriction for stability can increase the computational cost for problems involving fast time scales and/or fine meshes.

Now applying *one iteration* of the iHDG-II scheme for the implicit HDG formulation (19) with u^m as the initial guess yields

$$\begin{aligned}\left(\frac{u^{m+1}}{\Delta t}, v\right)_K - \left(u^{m+1}, \nabla \cdot (\boldsymbol{\beta} v)\right)_K + \left\langle \boldsymbol{\beta} \cdot \mathbf{n} u^{m+1} + |\boldsymbol{\beta} \cdot \mathbf{n}|(u^{m+1} - \hat{u}^{m,m+1}), v\right\rangle_{\partial K} &= \left(f^{m+1} + \frac{u^m}{\Delta t}, v\right)_K, \\ \left\langle [\![|\boldsymbol{\beta} \cdot \mathbf{n}| \hat{u}^{m,m+1}]\!], \boldsymbol{\mu}\right\rangle_{\partial K} = \left\langle |\boldsymbol{\beta} \cdot \mathbf{n}| \left\{ \left(u^{m+1}\right)^- + \left(u^m\right)^+ \right\}, \boldsymbol{\mu}\right\rangle_{\partial K} &+ \left\langle \boldsymbol{\beta} \cdot \mathbf{n}^- \left(u^{m+1}\right)^- + \boldsymbol{\beta} \cdot \mathbf{n}^+ \left(u^m\right)^+, \boldsymbol{\mu}\right\rangle_{\partial K}.\end{aligned}$$

Compared to the explicit HDG scheme, iHDG-II requires local solves since it is *locally implicit*. As such, its CFL restriction is much less (see Fig. 2), while still having similar parallel scalability of the explicit method.[1] Indeed, Fig. 2 shows that the CFL restriction is only indirectly through the increase of the number of iterations; for CFL numbers between 1 and 5, the number of iterations varies mildly. Thus, as a locally implicit method, iHDG-II combines advantages of both explicit (e.g. matrix free and parallel scalability) and implicit (taking reasonably large time stepsize without facing instability) methods. Clearly, on convergence iHDG solution is, up to the stopping tolerance, the same as the fully-implicit solution.

[1] In fact, due to local solves, iHDG-II could provide more efficient communication and computation overlapping.

4.4. iHDG-II for system of linear hyperbolic PDEs

In this section, as an example for the system of linear hyperbolic PDEs, we consider the following linearized oceanic shallow water system [62]:

$$\frac{\partial}{\partial t}\begin{pmatrix}\phi^e\\ \Phi u^e\\ \Phi v^e\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\Phi u^e\\ \Phi\phi^e\\ 0\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\Phi v^e\\ 0\\ \Phi\phi^e\end{pmatrix}=\begin{pmatrix}0\\ f\Phi v^e-\gamma\Phi u^e+\frac{\tau_x}{\rho}\\ -f\Phi u^e-\gamma\Phi v^e+\frac{\tau_y}{\rho}\end{pmatrix} \tag{20}$$

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. We study the iHDG-II methods for this equation and compare it with the results in [45].

For the simplicity of the exposition and the analysis, let us employ the backward Euler HDG discretization for (20). Since the unknowns of interest are those at the $(m+1)$th time step, we can suppress the time index for the clarity of the exposition. Furthermore, since the system is linear it is sufficient to consider homogeneous system with zero initial condition, zero boundary condition, and zero forcing (wind stress). Applying the iHDG-II Algorithm 2 to the homogeneous system gives

$$\left(\frac{\phi^{k+1}}{\Delta t},\varphi_1\right)_K-\left(\Phi\boldsymbol{\vartheta}^{k+1},\nabla\varphi_1\right)_K+\left\langle\Phi\boldsymbol{\vartheta}^{k+1}\cdot\mathbf{n}+\sqrt{\Phi}\left(\phi^{k+1}-\hat{\phi}^{k,k+1}\right),\varphi_1\right\rangle_{\partial K}=0, \tag{21a}$$

$$\left(\frac{\Phi u^{k+1}}{\Delta t},\varphi_2\right)_K-\left(\Phi\phi^{k+1},\frac{\partial\varphi_2}{\partial x}\right)_K+\left\langle\Phi\hat{\phi}^{k,k+1}\mathbf{n}_1,\varphi_2\right\rangle_{\partial K}=\left(f\Phi v^{k+1}-\gamma\Phi u^{k+1},\varphi_2\right)_K, \tag{21b}$$

$$\left(\frac{\Phi v^{k+1}}{\Delta t},\varphi_3\right)_K-\left(\Phi\phi^{k+1},\frac{\partial\varphi_3}{\partial y}\right)_K+\left\langle\Phi\hat{\phi}^{k,k+1}\mathbf{n}_2,\varphi_3\right\rangle_{\partial K}=\left(-f\Phi u^{k+1}-\gamma\Phi v^{k+1},\varphi_3\right)_K, \tag{21c}$$

where φ_1, φ_2 and φ_3 are the test functions, and

$$\hat{\phi}^{k,k+1}=\frac{1}{2}\left\{\left(\phi^{k+1}\right)^-+\left(\phi^k\right)^+\right\}+\frac{\sqrt{\Phi}}{2}\left\{\left(\boldsymbol{\vartheta}^{k+1}\right)^-\cdot\mathbf{n}^-+\left(\boldsymbol{\vartheta}^k\right)^+\cdot\mathbf{n}^+\right\}. \tag{22}$$

Lemma 2. *The local solver (21) of the iHDG-II algorithm for the linearized shallow water equation is well-posed.*

Proof. Since $\{(\phi^k)^+, \Phi(\boldsymbol{\vartheta}^k)^+\}$ is a "forcing" to the local solver it is sufficient to set them to $\{0, \mathbf{0}\}$ and show that the only solution possible is $\{(\phi^k)^-, \Phi(\boldsymbol{\vartheta}^k)^-\} = \{0, \mathbf{0}\}$. Choosing the test functions $\varphi_1 = \phi^{k+1}$, $\varphi_2 = u^{k+1}$ and $\varphi_3 = v^{k+1}$ in (21), integrating the second term in (21a) by parts, and then summing equations in (21) altogether, we obtain

$$\begin{aligned}&\frac{1}{\Delta t}\left(\phi^{k+1},\phi^{k+1}\right)_K+\frac{\Phi}{\Delta t}\left(\boldsymbol{\vartheta}^{k+1},\boldsymbol{\vartheta}^{k+1}\right)_K+\sqrt{\Phi}\left\langle\phi^{k+1},\phi^{k+1}\right\rangle_{\partial K}+\gamma\Phi\left(\boldsymbol{\vartheta}^{k+1},\boldsymbol{\vartheta}^{k+1}\right)_K\\&-\sqrt{\Phi}\left\langle\hat{\phi}^{k,k+1},\phi^{k+1}\right\rangle_{\partial K}+\Phi\left\langle\hat{\phi}^{k,k+1},\mathbf{n}\cdot\boldsymbol{\vartheta}^{k+1}\right\rangle_{\partial K}=0.\end{aligned} \tag{23}$$

Summing (23) over all elements yields

$$\begin{aligned}&\sum_K\frac{1}{\Delta t}\left(\phi^{k+1},\phi^{k+1}\right)_K+\frac{\Phi}{\Delta t}\left(\boldsymbol{\vartheta}^{k+1},\boldsymbol{\vartheta}^{k+1}\right)_K+\gamma\Phi\left(\boldsymbol{\vartheta}^{k+1},\boldsymbol{\vartheta}^{k+1}\right)_K\\&+\sqrt{\Phi}\left\langle\phi^{k+1},\phi^{k+1}\right\rangle_{\partial K}-\sqrt{\Phi}\left\langle\hat{\phi}^{k,k+1},\phi^{k+1}\right\rangle_{\partial K}+\Phi\left\langle\hat{\phi}^{k,k+1},\mathbf{n}\cdot\boldsymbol{\vartheta}^{k+1}\right\rangle_{\partial K}=0.\end{aligned} \tag{24}$$

Substituting (22) in the above equation and canceling some terms we get,

$$\begin{aligned}&\sum_K\frac{1}{\Delta t}\left\|\left(\phi^{k+1}\right)^-\right\|_K^2+\left(\gamma+\frac{1}{\Delta t}\right)\left\|\left(\boldsymbol{\vartheta}^{k+1}\right)^-\right\|_{\Phi,K}^2+\frac{\sqrt{\Phi}}{2}\left\|\left(\phi^{k+1}\right)^-\right\|_{\partial K}^2\\&+\frac{\sqrt{\Phi}}{2}\left\|\left(\boldsymbol{\vartheta}^{k+1}\cdot\mathbf{n}\right)^-\right\|_{\Phi,\partial K}^2=\sum_{\partial K}\frac{\sqrt{\Phi}}{2}\left\langle\left\{\left(\phi^k\right)^++\sqrt{\Phi}\left(\boldsymbol{\vartheta}^k\cdot\mathbf{n}\right)^+\right\},\left(\phi^{k+1}\right)^-\right\rangle_{\partial K}\\&-\frac{\Phi}{2}\left\langle\left\{\left(\phi^k\right)^++\sqrt{\Phi}\left(\boldsymbol{\vartheta}^k\cdot\mathbf{n}\right)^+\right\},\left(\boldsymbol{\vartheta}^{k+1}\cdot\mathbf{n}\right)^-\right\rangle_{\partial K}.\end{aligned} \tag{25}$$

Since $\Phi > 0$, all the terms on the left hand side are positive. When we set $\left\{\left(\phi^k\right)^+, \Phi\left(\boldsymbol{\vartheta}^k\right)^+\right\} = \{0, \mathbf{0}\}$, i.e. the data from neighboring elements, the only solution possible is $\left\{\left(\phi^{k+1}\right)^-, \Phi\left(\boldsymbol{\vartheta}^{k+1}\right)^-\right\} = \{0, \mathbf{0}\}$ and hence the method is well-posed. $\square$

Next, our goal is to show that $\left(\phi^{k+1}, \Phi\boldsymbol{\vartheta}^{k+1}\right)$ converges to zero. To that end, let us define

$$\mathcal{C} := \frac{\mathcal{A}}{\mathcal{B}}, \quad \mathcal{A} := \frac{\max\{1, \Phi\} + \sqrt{\Phi}}{4\varepsilon}, \quad \mathcal{G} := \frac{\varepsilon\left(\max\{1, \Phi\} + \sqrt{\Phi}\right)}{4} \tag{26}$$

and

$$\mathcal{B}_1 := \left(\frac{ch}{\Delta t(p+1)(p+2)} + \frac{2\sqrt{\Phi} - (\Phi + \sqrt{\Phi})\varepsilon}{4}\right)$$
$$\mathcal{B}_2 := \left(\left(\gamma + \frac{1}{\Delta t}\right)\frac{ch}{(p+1)(p+2)} + \frac{2\sqrt{\Phi} - (1 + \sqrt{\Phi})\varepsilon}{4}\right), \mathcal{B} := \min\{\mathcal{B}_1, \mathcal{B}_2\},$$

where $0 < c \le 1$, $\varepsilon > 0$ are constants. We also need the following norms:

$$\left\|\left(\phi^k, \boldsymbol{\vartheta}^k\right)\right\|^2_{\Omega_h} := \left\|\phi^k\right\|^2_{\Omega_h} + \left\|\boldsymbol{\vartheta}^k\right\|^2_{\Phi,\Omega_h},$$
$$\left\|\left(\phi^k, \boldsymbol{\vartheta}^k \cdot \mathbf{n}\right)\right\|^2_{\mathcal{E}_h} := \left\|\phi^k\right\|^2_{\mathcal{E}_h} + \left\|\boldsymbol{\vartheta}^k \cdot \mathbf{n}\right\|^2_{\Phi,\mathcal{E}_h}.$$

Theorem 2. *Assume that the meshsize h, the time step Δt and the solution order p are chosen such that $\mathcal{B} > 0$ and $\mathcal{C} < 1$, then the approximate solution at the kth iteration $\left(\phi^k, \boldsymbol{\vartheta}^k\right)$ converges to zero in the following sense*

$$\left\|\left(\phi^k, \boldsymbol{\vartheta}^k \cdot \mathbf{n}\right)\right\|^2_{\mathcal{E}_h} \le \mathcal{C}^k \left\|\left(\phi^0, \boldsymbol{\vartheta}^0 \cdot \mathbf{n}\right)\right\|^2_{\mathcal{E}_h},$$
$$\left\|\left(\phi^k, \boldsymbol{\vartheta}^k\right)\right\|^2_{\Omega_h} \le \Delta t\,(\mathcal{A} + \mathcal{G}\mathcal{C})\,\mathcal{C}^{k-1} \left\|\left(\phi^0, \boldsymbol{\vartheta}^0 \cdot \mathbf{n}\right)\right\|^2_{\mathcal{E}_h},$$

where $\mathcal{C}$, $\mathcal{A}$ and $\mathcal{G}$ are defined in (26).

Proof. Using Cauchy–Schwarz and Young's inequalities for the terms on the right hand side of (25) and simplifying we have

$$\begin{aligned}
&\sum_K \frac{1}{\Delta t}\left\|\left(\phi^{k+1}\right)^-\right\|^2_K + \left(\gamma + \frac{1}{\Delta t}\right)\left\|\left(\boldsymbol{\vartheta}^{k+1}\right)^-\right\|^2_{\Phi,K} + \frac{\sqrt{\Phi}}{2}\left\|\left(\phi^{k+1}\right)^-\right\|^2_{\partial K} \\
&+ \frac{\sqrt{\Phi}}{2}\left\|\left(\boldsymbol{\vartheta}^{k+1} \cdot \mathbf{n}\right)^-\right\|^2_{\Phi,\partial K} \le \sum_{\partial K} \frac{\Phi + \sqrt{\Phi}}{4\varepsilon}\left\|\left(\phi^k\right)^+\right\|^2_{\partial K} \\
&+ \frac{1 + \sqrt{\Phi}}{4\varepsilon}\left\|\left(\boldsymbol{\vartheta}^k \cdot \mathbf{n}\right)^+\right\|^2_{\Phi,\partial K} + \frac{\varepsilon(\Phi + \sqrt{\Phi})}{4}\left\|\left(\phi^{k+1}\right)^-\right\|^2_{\partial K} \\
&+ \frac{\varepsilon(1 + \sqrt{\Phi})}{4}\left\|\left(\boldsymbol{\vartheta}^{k+1} \cdot \mathbf{n}\right)^-\right\|^2_{\Phi,\partial K}.
\end{aligned} \tag{27}$$

An application of inverse trace inequality [63] for tensor product elements gives

$$\left(\phi^{k+1}, \phi^{k+1}\right)_K \ge \frac{2ch}{d(p+1)(p+2)}\left\langle\phi^{k+1}, \phi^{k+1}\right\rangle_{\partial K}, \tag{28a}$$

$$\left(\boldsymbol{\vartheta}^{k+1}, \boldsymbol{\vartheta}^{k+1}\right)_K \ge \frac{2ch}{d(p+1)(p+2)}\left\langle\boldsymbol{\vartheta}^{k+1}, \boldsymbol{\vartheta}^{k+1}\right\rangle_{\partial K}, \tag{28b}$$

where d is the spatial dimension which in this case is 2 and $0 < c \le 1$ is a constant.[2] Inequality (28), together with (27), implies

[2] Note that for simplices we can use the trace inequalities in [64] and it will change only the constants in the proof.

$$\sum_{\partial K}\left[\left(\frac{ch}{\Delta t(p+1)(p+2)}+\frac{2\sqrt{\Phi}-(\Phi+\sqrt{\Phi})\varepsilon}{4}\right)\left\|\left(\phi^{k+1}\right)^-\right\|^2_{\partial K}\right.$$
$$\left.+\left(\left(\gamma+\frac{1}{\Delta t}\right)\frac{ch}{(p+1)(p+2)}+\frac{2\sqrt{\Phi}-(1+\sqrt{\Phi})\varepsilon}{4}\right)\left\|\left(\boldsymbol{\vartheta}^{k+1}\cdot\mathbf{n}\right)^-\right\|^2_{\Phi,\partial K}\right]$$
$$\le\sum_{\partial K}\left[\frac{\Phi+\sqrt{\Phi}}{4\varepsilon}\left\|\left(\phi^{k}\right)^+\right\|^2_{\partial K}+\frac{1+\sqrt{\Phi}}{4\varepsilon}\left\|\left(\boldsymbol{\vartheta}^{k}\cdot\mathbf{n}\right)^+\right\|^2_{\Phi,\partial K}\right], \tag{29}$$

which then implies

$$\left\|\left(\phi^{k+1},\boldsymbol{\vartheta}^{k+1}\cdot\mathbf{n}\right)\right\|^2_{\mathcal{E}_h}\le\mathcal{C}\left\|\left(\phi^{k},\boldsymbol{\vartheta}^{k}\cdot\mathbf{n}\right)\right\|^2_{\mathcal{E}_h},$$

where the constant $\mathcal{C}$ is computed as in (26). Therefore

$$\left\|\left(\phi^{k+1},\boldsymbol{\vartheta}^{k+1}\cdot\mathbf{n}\right)\right\|^2_{\mathcal{E}_h}\le\mathcal{C}^{k+1}\left\|\left(\phi^{0},\boldsymbol{\vartheta}^{0}\cdot\mathbf{n}\right)\right\|^2_{\mathcal{E}_h}. \tag{30}$$

On the other hand, inequalities (27) and (30) imply

$$\left\|\left(\phi^{k+1},\boldsymbol{\vartheta}^{k+1}\right)\right\|^2_{\Omega_h}\le\Delta t\left(\mathcal{A}+\mathcal{G}\mathcal{C}\right)\mathcal{C}^k\left\|\left(\phi^{0},\boldsymbol{\vartheta}^{0}\cdot\mathbf{n}\right)\right\|^2_{\mathcal{E}_h}$$

and this ends the proof. □

We now derive an explicit relation between the number of iterations k, the meshsize h, the solution order p, the time step Δt and the mean flow geopotential height Φ. First, we need to find an ε which makes $\mathcal{C}<1$. From (26) we obtain the following inequality for ε

$$\frac{\frac{\max\{1,\Phi\}+\sqrt{\Phi}}{4\varepsilon}}{\left(\frac{ch}{\Delta t(p+1)(p+2)}+\frac{2\sqrt{\Phi}-(\max\{1,\Phi\}+\sqrt{\Phi})\varepsilon}{4}\right)}<1. \tag{31}$$

A sufficient condition for the denominator to be positive and existence of a real $\varepsilon>0$ to the above inequality (31) is

$$\frac{\frac{ch}{\Delta t(p+1)(p+2)}}{\max\{1,\Phi\}+\sqrt{\Phi}}>\frac{1}{2}. \tag{32}$$

This allows us to find an $\varepsilon>0$ that satisfies the inequality (31) for all Φ. In particular, we can pick

$$\varepsilon=\frac{\frac{2ch}{\Delta t(p+1)(p+2)}+\sqrt{\Phi}}{\max\{1,\Phi\}+\sqrt{\Phi}}. \tag{33}$$

Using this value of ε in definition (26) we get

$$\mathcal{C}=\left(\frac{\frac{\max\{1,\Phi\}+\sqrt{\Phi}}{\sqrt{\Phi}}}{1+\frac{2ch}{\sqrt{\Phi}\Delta t(p+1)(p+2)}}\right)^2$$

and since the numerator is always greater than 1, the necessary and sufficient condition for the convergence of the algorithm is given by

$$\frac{1}{\left(1+\frac{2ch}{\sqrt{\Phi}\Delta t(p+1)(p+2)}\right)^{2k}}\xrightarrow{k\to\infty}0.$$

Using binomial theorem and neglecting higher order terms we get

$$k=\mathcal{O}\left(\frac{\Delta t(p+1)(p+2)\sqrt{\Phi}}{4ch}\right). \tag{34}$$

Note that if we choose Δt similar to explicit method, i.e. $\Delta t=\mathcal{O}\left(\frac{h}{p^2\sqrt{\Phi}}\right)$ [65], $k=\mathcal{O}(1)$ independent of h and p. With this result in hand we are now in a better position to understand the stability of iHDG-I and iHDG-II algorithms for the linearized shallow water system. For unconditional stability of the iterative algorithms under consideration, we need $\mathcal{B} > 0$ in (26) independent of h, p and Δt. There are two terms in $\mathcal{B}$: $\mathcal{B}_1$ coming from ϕ and $\mathcal{B}_2$ from $\boldsymbol{\vartheta}$ or $\boldsymbol{\vartheta} \cdot \mathbf{n}$. We can write $\mathcal{B}$ in Theorem 3.6 of [45] for iHDG-I also as[3]

$$\mathcal{B}_1 := \left(\frac{ch}{\Delta t(p+1)(p+2)} + \frac{2\sqrt{\Phi} - (\Phi + \sqrt{\Phi})\varepsilon}{2} \right) \tag{35}$$

$$\mathcal{B}_2 := \left(\left(\gamma + \frac{1}{\Delta t} \right) \frac{ch}{(p+1)(p+2)} - \frac{(1+\sqrt{\Phi})\varepsilon}{2} \right), \mathcal{B} := \min\{\mathcal{B}_1, \mathcal{B}_2\}. \tag{36}$$

Note that for both iHDG-I and iHDG-II algorithms we have the stability in ϕ independent of h, p and Δt, since we can choose ε sufficiently small independent of h, p and Δt and make $\mathcal{B}_1 > 0$ in (35) and (26). However, from (36) we have to choose ε as a function of $(h, p, \Delta t)$ in order to have $\mathcal{B}_2 > 0$, and hence iHDG-I lacks the mesh independent stability in the term associated with $\boldsymbol{\vartheta}$. This explains the instability observed in [45] for fine meshes and/or large time steps. Since $\mathcal{B}_2$ in (26) can be made positive with a sufficiently small ε, independent of h, p and Δt, iHDG-II is always stable: a significant advantage over iHDG-I.

5. iHDG-II for linear convection–diffusion PDEs

5.1. First order form

In this section we apply the iHDG-II Algorithm 2 to the following first order form of the convection–diffusion equation:

$$\kappa^{-1}\boldsymbol{\sigma}^e + \nabla u^e = 0 \quad \text{in } \Omega, \tag{37a}$$

$$\nabla \cdot \boldsymbol{\sigma}^e + \boldsymbol{\beta} \cdot \nabla u^e + \nu u^e = f \quad \text{in } \Omega. \tag{37b}$$

We assume that (37) is well-posed, i.e.,

$$\nu - \frac{\nabla \cdot \boldsymbol{\beta}}{2} \geq \lambda > 0. \tag{38}$$

Though this is not a restriction, we take constant diffusion coefficient for the simplicity of the exposition. An upwind HDG numerical flux [24] is given by

$$\hat{\mathbf{F}} \cdot \mathbf{n} = \begin{bmatrix} \hat{u}\mathbf{n}_1 \\ \hat{u}\mathbf{n}_2 \\ \hat{u}\mathbf{n}_3 \\ \boldsymbol{\sigma} \cdot \mathbf{n} + \boldsymbol{\beta} \cdot \mathbf{n} u + \tau\left(u - \hat{u}\right) \end{bmatrix}, \tag{39}$$

where $\tau = \frac{1}{2}\left(\alpha - \boldsymbol{\beta} \cdot \mathbf{n}\right)$ and $\alpha = \sqrt{|\boldsymbol{\beta} \cdot \mathbf{n}|^2 + 4}$. Similar to the previous sections, it is sufficient to consider the homogeneous problem. Applying the iHDG-II Algorithm 2 we have the following iterative scheme

$$\kappa^{-1}\left(\boldsymbol{\sigma}^{k+1}, \boldsymbol{\tau}\right)_K - \left(u^{k+1}, \nabla \cdot \boldsymbol{\tau}\right)_K + \left\langle \hat{u}^{k,k+1}, \boldsymbol{\tau} \cdot \mathbf{n}\right\rangle_{\partial K} = 0, \tag{40a}$$

$$-\left(\boldsymbol{\sigma}^{k+1}, \nabla v\right)_K - \left(u^{k+1}, \nabla \cdot (\boldsymbol{\beta} v) - \nu v\right)_K + \left\langle \boldsymbol{\beta} \cdot \mathbf{n} u^{k+1} + \boldsymbol{\sigma}^{k+1} \cdot \mathbf{n} + \tau(u^{k+1} - \hat{u}^{k,k+1}), v\right\rangle_{\partial K} = 0, \tag{40b}$$

where

$$\hat{u}^{k,k+1} = \frac{\left\{\left(\boldsymbol{\sigma}^{k+1} \cdot \mathbf{n}\right)^- + \left(\boldsymbol{\sigma}^k \cdot \mathbf{n}\right)^+\right\} + \left\{\boldsymbol{\beta} \cdot \mathbf{n}^- \left(u^{k+1}\right)^- + \boldsymbol{\beta} \cdot \mathbf{n}^+ \left(u^k\right)^+\right\}}{\alpha} + \frac{\left\{\tau^- \left(u^{k+1}\right)^- + \tau^+ \left(u^k\right)^+\right\}}{\alpha}. \tag{41}$$

Lemma 3. *The local solver (40) of the iHDG-II algorithm for the convection–diffusion equation is well-posed.*

Proof. The proof is similar to the one for the shallow water equation and hence is given in the Appendix A. □

Now, we are in a position to prove the convergence of the algorithm. For $\varepsilon, h > 0$ and $0 < c \leq 1$ given, define

$$\mathcal{C}_1 := \frac{(\|\boldsymbol{\beta} \cdot \mathbf{n}\|_{L^\infty(\partial K)} + \bar{\tau})(\bar{\tau} + 1)}{2\varepsilon\alpha_*}, \ \mathcal{C}_2 := \frac{(\bar{\tau} + 1)}{2\varepsilon\alpha_*}, \tag{42}$$

[3] This can be obtained by using Young's inequality with ε in the proof of Theorem 3.6 in [45].

$$\mathcal{C}_3 := \frac{\varepsilon\bar{\tau}(1+\bar{\tau}+\|\boldsymbol{\beta}\cdot\mathbf{n}\|_{L^\infty(\partial K)})}{2\alpha_*},\ \mathcal{C}_4 := \frac{\varepsilon(1+\bar{\tau}+\|\boldsymbol{\beta}\cdot\mathbf{n}\|_{L^\infty(\partial K)})}{2\alpha_*}, \tag{43}$$

$$\mathcal{D} := \frac{\mathcal{A}}{\mathcal{B}},\ \mathcal{A} = \max\{\mathcal{C}_1, \mathcal{C}_2\},\ \mathcal{E} := \frac{\max\{\mathcal{C}_3, \mathcal{C}_4\}}{\min\{\kappa^{-1}, \lambda\}},\ \mathcal{F} := \frac{\mathcal{A}}{\min\{\kappa^{-1}, \lambda\}}, \tag{44}$$

$$\mathcal{B}_1 := \frac{2ch\kappa^{-1}}{d(p+1)(p+2)} + \frac{1}{2\bar{\alpha}} - \mathcal{C}_4,\ \mathcal{B}_2 := \frac{2ch\lambda}{d(p+1)(p+2)} + \frac{1}{\bar{\alpha}} - \mathcal{C}_3, \tag{45}$$

$$\mathcal{B} := \min\{\mathcal{B}_1, \mathcal{B}_2\}, \tag{46}$$

where $\bar{\tau} := \|\tau\|_{L^\infty(\partial\Omega_h)}$, $\bar{\alpha} := \|\alpha\|_{L^\infty(\partial\Omega_h)}$, and $\alpha_* := \inf_{\partial K \in \partial\Omega_h} \alpha$. As in the previous section we need the following norms

$$\left\|\left(\boldsymbol{\sigma}^k, u^k\right)\right\|^2_{\Omega_h} := \left\|\boldsymbol{\sigma}^k\right\|^2_{\Omega_h} + \left\|u^k\right\|^2_{\Omega_h}, \quad \left\|\left(\boldsymbol{\sigma}^k\cdot\mathbf{n}, u^k\right)\right\|^2_{\mathcal{E}_h} := \left\|\boldsymbol{\sigma}^k\cdot\mathbf{n}\right\|^2_{\mathcal{E}_h} + \left\|u^k\right\|^2_{\mathcal{E}_h}.$$

Theorem 3. *Suppose that the meshsize h and the solution order p are chosen such that $\mathcal{B} > 0$ and $\mathcal{D} < 1$, the algorithm (40a)–(41) converges in the following sense*

$$\left\|\left(\boldsymbol{\sigma}^k\cdot\mathbf{n}, u^k\right)\right\|^2_{\mathcal{E}_h} \le \mathcal{D}^k \left\|\left(\boldsymbol{\sigma}^0\cdot\mathbf{n}, u^0\right)\right\|^2_{\mathcal{E}_h},$$
$$\left\|\left(\boldsymbol{\sigma}^k, u^k\right)\right\|^2_{\Omega_h} \le (\mathcal{E}\mathcal{D}+\mathcal{F})\mathcal{D}^{k-1} \left\|\left(\boldsymbol{\sigma}^0\cdot\mathbf{n}, u^0\right)\right\|^2_{\mathcal{E}_h},$$

where $\mathcal{D}$, $\mathcal{E}$ and $\mathcal{F}$ are defined in (44).

Proof. The proof is similar to the one for the shallow water equation and hence is given in the Appendix B. □

Similar to the discussion in section 4.4, one can show that

$$k = \mathcal{O}\left(\frac{d(p+1)(p+2)}{8\bar{\alpha}ch\min\left\{\kappa^{-1}, \lambda\right\}}\right). \tag{47}$$

For time-dependent convection–diffusion equation, we discretize the spatial differential operators using HDG. For the temporal derivative, we use implicit time stepping methods, again with either backward Euler or Crank–Nicolson method for simplicity. The analysis in this case is almost identical to the one for steady state equation except that we now have an additional L^2-term $\left(u^{k+1}, v\right)_K/\Delta t$ in the local equation (40b). This improves the convergence of the iHDG-II method. Indeed, the convergence analysis is the same except we now have $\lambda + 1/\Delta t$ in place of λ. In particular we have the following estimation

$$k = \mathcal{O}\left(\frac{d(p+1)(p+2)}{8\bar{\alpha}ch\min\left\{\kappa^{-1}, (\lambda+1/\Delta t)\right\}}\right).$$

Remark 2. Similar to the shallow water system if we choose $\Delta t = \mathcal{O}\left(\frac{h}{p^2}\right)$ then the number of iterations is independent of h and p. This is more efficient than the iterative hybridizable IPDG method for the parabolic equation in [47], which requires $\Delta t = \mathcal{O}(\frac{h^2}{p^4})$ in order to achieve constant iterations. The reason is perhaps due the fact that hybridizable IPDG is posed directly on the second order form whereas HDG uses the first order form. While iHDG-I has mesh independent stability for only u (see [45, Theorem 4.1]), iHDG-II does for both u and $\boldsymbol{\sigma}$; an important improvement.

6. Numerical results

In this section various numerical results verifying the theoretical results are provided for the transport equation, the shallow water equation, and the convection–diffusion equation in both two- and three-dimensions.

6.1. Transport equation

We consider the same 2D and 3D test cases in [45, sections 5.1.1 and 5.1.2]. For the 2D test case we consider $f = 0$ and $\boldsymbol{\beta} = (1 + sin(\pi y/2), 2)$ in (8). The domain is $[0, 2] \times [0, 2]$ and the inflow boundary conditions are given by

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

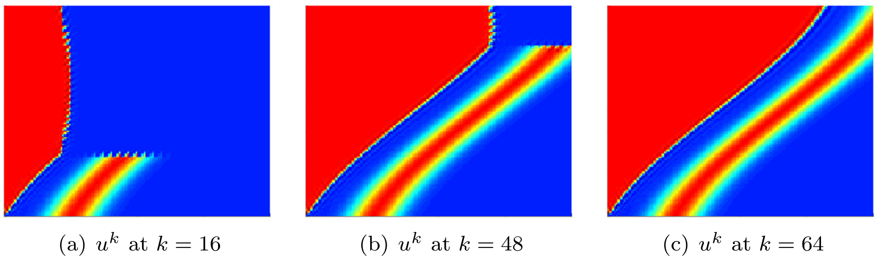
(a) u^k at $k = 16$ (b) u^k at $k = 48$ (c) u^k at $k = 64$

Fig. 1. Evolution of the iterative solution for the 2D transport equation using the iHDG-II algorithm.

Table 1
The number of iterations taken by the iHDG-II algorithm for the transport equation in 2D and 3D settings.

N_{el}(2D)	N_{el}(3D)	p	2D solution	3D solution
4×4	$2 \times 2 \times 2$	1	9	6
8×8	$4 \times 4 \times 4$	1	17	12
16×16	$8 \times 8 \times 8$	1	33	23
32×32	$16 \times 16 \times 16$	1	65	47
4×4	$2 \times 2 \times 2$	2	9	6
8×8	$4 \times 4 \times 4$	2	17	12
16×16	$8 \times 8 \times 8$	2	33	23
32×32	$16 \times 16 \times 16$	2	65	47
4×4	$2 \times 2 \times 2$	3	9	7
8×8	$4 \times 4 \times 4$	3	17	12
16×16	$8 \times 8 \times 8$	3	33	23
32×32	$16 \times 16 \times 16$	3	65	47
4×4	$2 \times 2 \times 2$	4	9	6
8×8	$4 \times 4 \times 4$	4	17	12
16×16	$8 \times 8 \times 8$	4	33	24
32×32	$16 \times 16 \times 16$	4	64	48

For the 3D test case the exact solution is given by

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

We choose $\boldsymbol{\beta} = (z, x, y)$ in (8). The forcing is selected in such a way that it corresponds to the exact solution. The domain is $[0, 1] \times [0, 1] \times [0, 1]$ with faces $x = 0$, $y = 0$ and $z = 0$ as the inflow boundaries. The inflow boundary condition is enforced using the exact solution.

The mesh consists of structured quadrilateral (2D)/hexahedral (3D) elements. Throughout the numerical section unless otherwise stated explicitly, we use the following stopping criterion

$$|\|u^k - u^e\|_{L^2(\Omega)} - \|u^{k-1} - u^e\|_{L^2(\Omega)}| < 10^{-10}, \tag{48}$$

if the exact solution is available, and

$$\|u^k - u^{k-1}\|_{L^2(\Omega)} < 10^{-10}, \tag{49}$$

if the exact solution is not available.

From Theorem 1, the theoretical number of iterations is approximately $d \times (N_{el})^{1/d}$ (where d is the dimension). It can be seen from the fourth and fifth columns of Table 1 that the numerical results agree well with the theoretical prediction. We can also see that the number of iterations is independent of solution order, which is consistent with the theoretical result Theorem 1. Fig. 1 shows the solution converging from the inflow boundary to the outflow boundary in a layer-by-layer manner. Again, the process is automatic, i.e., no prior element ordering or information about the advection velocity is required.

Now, we study the parallel performance of the iHDG algorithm. For this purpose we have implemented iHDG algorithm on top of *mangll* [66–68] which is a higher order continuous/discontinuous finite element library that supports large scale parallel simulations using MPI. The simulations are conducted on **Stampede 1** at the Texas Advanced Computing Center (TACC). Stampede 1 is a 10 petaflop supercomputer consisting of 6400 Sandy Bridge nodes. Each node consists of two 8-core Xeon E5-2680 2.7 GHz processors and one 61-core Xeon Phi SE10P KNC MIC 1.1 GHz coprocessor. It has 32 GB main memory per node (8×4 GB DDR3-1600 MHz) and the coprocessor has additional 8 GB GDDR5 memory. The interconnect

Table 2
Strong scaling results on TACC's Stampede system for the 3D transport problem.

$N_{el} = 262,144$, $p = 4$, dof = 32.768 million, Iterations = 190			
# cores	Time [s]	N_{el}/core	Efficiency [%]
64	1758.62	4096	100.0
128	883.88	2048	99.5
256	439.94	1024	99.9
512	228.69	512	96.1
1024	113.87	256	96.5
2048	56.36	128	97.5
4096	29.26	64	91.8
16384	11.38	16	59
$N_{el} = 2,097,152$, $p = 4$, dof = 262.144 million, Iterations = 382			
# cores	Time [s]	N_{el}/core	Efficiency [%]
512	3634.89	4096	100.0
1024	1788.78	2048	101.5
2048	932.495	1024	97.3
4096	447.337	512	101.5
8192	232.019	256	97.9
16384	117.985	128	92.9

Table 3
Weak scaling results on TACC's Stampede system for the 3D transport problem.

1024 N_{el}/core, $p = 4$			
# cores	Time [s]	Time ratio	Iterations ratio
4	103.93	1	1
32	217.23	2.1	2
256	439.94	4.2	4
2048	932.49	8.9	8
128 N_{el}/core, $p = 4$			
# cores	Time [s]	Time ratio	Iterations ratio
4	6.52	1	1
32	13.68	2.1	2
256	27.71	4.2	4
2048	56.37	8.6	8

is a 56 GB/s Mellanox FDR InfiniBand network in a 2-level fat-tree topology. To carry out the computations, we have used only the main processors and not the coprocessors.

Table 2 shows strong scaling results for the 3D transport problem. The parallel efficiency is over 90% for all the cases except for the case where we use 16,384 cores and 16 elements per core whose efficiency is 59%. This is due to the fact that, with 16 elements per core, the communication cost dominates the computation. Table 3 shows the weak scaling with 1024 and 128 elements/core. Since the number of iterations increases linearly with the number of elements, we can see a similar increase in time when we increase the number of elements, and hence cores.

Let us now consider the time dependent 3D transport equation with the following exact solution

$$u^e = e^{-5((x-0.35t)^2+(y-0.35t)^2+(z-0.35t)^2)},$$

where the velocity field is chosen to be $\boldsymbol{\beta} = (0.2, 0.2, 0.2)$. In Fig. 2 are the numbers of iHDG iterations taken per time step to converge versus the CFL number. As can be seen, for CFL in the range $[1, 5]$ the number of iterations grows mildly. As a result, we get a much better weak scaling results in Table 4 in comparison to the steady state case in Table 3.

6.2. Linearized shallow water system

The goal of this section is to verify the theoretical findings in section 4.4. To that extent, let us consider equation (20) with a linear standing wave, for which, $\Phi = 1$, $f = 0$, $\gamma = 0$ (zero bottom friction), $\boldsymbol{\tau} = 0$ (zero wind stress). The domain is $[0, 1] \times [0, 1]$ and the wall boundary condition is applied on the domain boundary. The following exact solution [62] is taken

$$\phi^e = \cos(\pi x)\cos(\pi y)\cos(\sqrt{2}\pi t), \tag{50a}$$

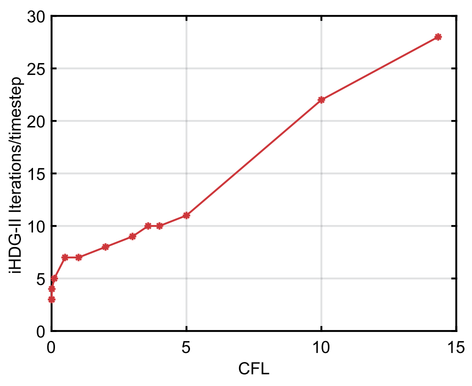

Fig. 2. CFL versus Iterations for the 3D time dependent transport.

Table 4
Weak scaling results on TACC's Stampede system for the 3D time dependent transport problem.

128 N_{el}/core, $p=4$, $\Delta t = 0.01$, $\lvert\boldsymbol{\beta}\rvert_{max} = 0.35$				
# cores	Time/Timestep [s]	Time ratio	Iterations ratio	CFL
4	1.69	1	1	0.45
32	1.91	1.1	1.1	0.9
256	2.09	1.2	1.1	1.8
2048	2.72	1.6	1.4	3.6
16384	4.68	2.8	2.1	7.2

Table 5
Comparison of iHDG-I and iHDG-II for the linearized shallow water system.

h	p	iHDG-I		iHDG-II	
		$\Delta t = 10^{-1}$	$\Delta t = 10^{-2}$	$\Delta t = 10^{-1}$	$\Delta t = 10^{-2}$
0.25	1	19	6	14	6
0.125	1	*	6	18	9
0.0625	1	*	7	32	10
0.03125	1	*	9	59	8
0.25	2	*	9	15	9
0.125	2	*	11	19	9
0.0625	2	*	13	32	11
0.03125	2	*	15	59	12
0.25	3	*	7	16	8
0.125	3	*	9	20	8
0.0625	3	*	12	31	10
0.03125	3	*	*	59	12
0.25	4	*	10	17	9
0.125	4	*	12	32	10
0.0625	4	*	*	60	9
0.03125	4	*	*	112	13

$$u^e = \frac{1}{\sqrt{2}} \sin(\pi x)\cos(\pi y)\sin(\sqrt{2}\pi t), \tag{50b}$$

$$v^e = \frac{1}{\sqrt{2}} \cos(\pi x)\sin(\pi y)\sin(\sqrt{2}\pi t). \tag{50c}$$

We use Crank–Nicolson method for the temporal discretization and the iHDG-II approach for the spatial discretization. In Table 5 we compare the number of iterations taken by iHDG-I and iHDG-II methods for two different time steps $\Delta t = 0.1$ and $\Delta t = 0.01$. Here, "$*$" indicates divergence. As can be seen from the third and fourth columns, the iHDG-I method diverges for finer meshes and/or larger time steps. This is consistent with the findings in section 4.4 where the divergence is expected because of the lack of mesh independent stability in the velocity. On the contrary, iHDG-II converges for all cases.

In Table 5, we use a series of structured quadrilateral meshes with uniform refinements such that the ratio of successive meshsizes is 1/2. The asymptotic result (34), which is valid for $\frac{h}{\Delta t(p+1)(p+2)\sqrt{\Phi}} \ll 1$, predicts that the ratio of the number

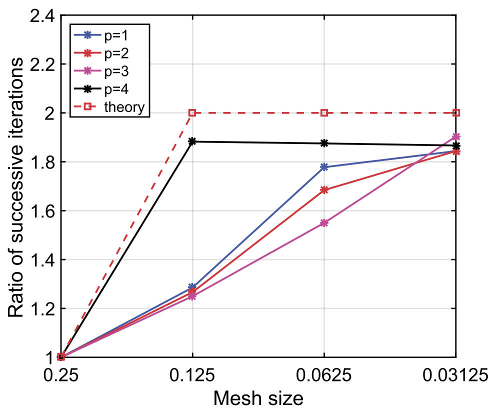

Fig. 3. Growth of iterations with meshsize h for the iHDG-II method for the linearized shallow water system. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 6
Growth of iterations with solution order p for the iHDG-II method for the linearized shallow water system.

p	Meshsize (h)				Asymptotics
	0.25	0.125	0.0625	0.03125	
2	1.07	1.06	1	1	2
3	1.14	1.11	0.97	1	3.33
4	1.21	1.78	1.87	1.9	5

of iterations required by successive refined meshes is 2, and the results in Fig. 3 confirm this prediction. The last two columns of Table 5 also confirms the asymptotic result (34) that the number of iHDG-II iterations scales linearly with the time stepsize.

Next, we study the iHDG iteration growth as the solution order p increases. The asymptotic result (34) predicts that $k = \mathcal{O}(p^2)$. In Table 6, rows 2–4 show the ratio of the number of iterations taken for solution orders $p = \{2, 3, 4\}$ over the one for $p = 1$ for four different meshsizes as in Table 5. As can be seen, the theoretical estimation is conservative.

6.3. *Nonlinear shallow water system*

In this section, we consider the nonlinear shallow water system, given by,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{f}, \tag{51}$$

where the forcing function $\mathbf{f}$, and the x-component and y-component of the flux $\mathbf{F}$ are given by

$$\mathbf{F}_x := \begin{pmatrix} Hu \\ Hu^2 + \frac{1}{2}gH^2 \\ Huv \end{pmatrix}, \quad \mathbf{F}_y := \begin{pmatrix} Hv \\ Huv \\ Hv^2 + \frac{1}{2}gH^2 \end{pmatrix}, \quad \text{and} \quad \mathbf{f} := \begin{pmatrix} 0 \\ -gHb_x \\ -gHb_y \end{pmatrix},$$

while the conservative variables $\mathbf{u}$ are defined as

$$\mathbf{u} := (H, Hu, Hv)^T .$$

Here, H is the water depth, u is the depth averaged velocity component in the x-direction, v is the depth averaged velocity component in the y-direction, b is the bathymetry, and g is the gravitational constant.

For nonlinear problems both the local solver (2a) and the conservation constraints (2b) are nonlinear. In order to tackle these problems, we apply the iHDG algorithms to solve the linearized system arising from each Newton step of the HDG system (2). For the clarity of the exposition, let us consider one generic Newton step. To begin, we define the following residuals for (2a) and (2b):

$$\mathcal{R}es = (\partial_t \mathbf{u}, \mathbf{v})_{\Omega_h} - (\mathbf{F}(\mathbf{u}), \nabla \mathbf{v})_{\Omega_h} + \left\langle \hat{\mathbf{F}}(\mathbf{u}, \hat{\mathbf{u}}) \cdot \mathbf{n}, \mathbf{v} \right\rangle_{\partial \Omega_h} + (\mathbf{C}\mathbf{u}, \mathbf{v})_{\Omega_h} - (\mathbf{f}, \mathbf{v})_{\Omega_h}, \tag{52a}$$

$$\mathcal{F}lx = \left\langle \left[\!\left[\hat{\mathbf{F}}(\mathbf{u}, \hat{\mathbf{u}}) \cdot \mathbf{n} \right]\!\right], \boldsymbol{\mu} \right\rangle_{\mathcal{E}_h}. \tag{52b}$$

Here, the hybridized Lax–Friedrichs one, used in [24] is employed, i.e.,

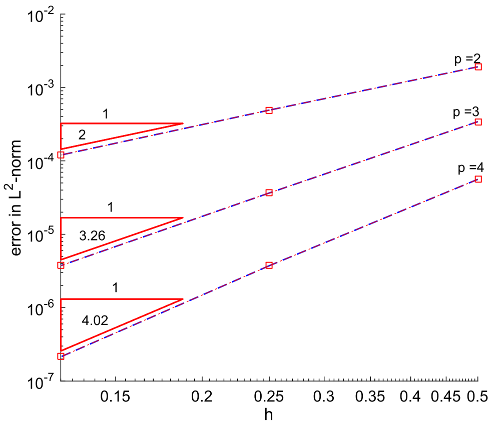

Fig. 4. Nonlinear shallow water system with translating vortex solution: convergence rate for HDG methods with iHDG-II algorithm (blue dashed line) and direct solver (red dashed squares).

$$\hat{\mathbf{F}}\left(\mathbf{u},\hat{\mathbf{u}}\right)\cdot\mathbf{n}=\mathbf{F}\left(\mathbf{u}\right)\cdot\mathbf{n}+\tau\mathbf{I}\left(\mathbf{u}-\hat{\mathbf{u}}\right),$$

with $\hat{\mathbf{u}}:=\left(\hat{H},\widehat{Hu},\widehat{Hv}\right)^T$, the stabilization $\tau=\sqrt{\hat{u}^2+\hat{v}^2}+g\hat{H}$, and $\mathbf{I}$ the corresponding identity matrix. Note that $\hat{u}$ and $\hat{v}$ are given by $\hat{u}=\frac{\widehat{Hu}}{\hat{H}}$ and $\hat{v}=\frac{\widehat{Hv}}{\hat{H}}$.

If we define a Newton step for $\mathbf{u}$ and $\hat{\mathbf{u}}$ as $\delta\mathbf{u}$ and $\delta\hat{\mathbf{u}}$, respectively, the linear system for $\delta\mathbf{u}$ and $\delta\hat{\mathbf{u}}$ resulting from each Newton step is given by (after the temporal derivative is discretized, e.g., with backward Euler or Crank–Nicolson)

$$\begin{bmatrix}\frac{\partial\mathcal{R}es}{\partial\mathbf{u}} & \frac{\partial\mathcal{R}es}{\partial\hat{\mathbf{u}}}\\ \frac{\partial\mathcal{F}lx}{\partial\mathbf{u}} & \frac{\partial\mathcal{F}lx}{\partial\hat{\mathbf{u}}}\end{bmatrix}\begin{Bmatrix}\delta\mathbf{u}\\ \delta\hat{\mathbf{u}}\end{Bmatrix}=\begin{Bmatrix}-\mathcal{R}es\\ -\mathcal{F}lx\end{Bmatrix}. \tag{53}$$

Applying iHDG-II algorithm to the linear system (53) we get

$$\frac{\partial\mathcal{R}es}{\partial\mathbf{u}}\delta\mathbf{u}^{k+1}+\frac{\partial\mathcal{R}es}{\partial\hat{\mathbf{u}}}\delta\hat{\mathbf{u}}^{k,k+1}=-\mathcal{R}es, \tag{54}$$

where $\delta\hat{\mathbf{u}}^{k,k+1}$ is determined from the conservation condition as

$$\left[\!\left[\frac{\partial\mathcal{F}lx}{\partial\hat{\mathbf{u}}}\right]\!\right]\delta\hat{\mathbf{u}}^{k,k+1}=-\left[\!\left[\mathcal{F}lx\right]\!\right]-\left(\frac{\partial\mathcal{F}lx}{\partial\mathbf{u}}\right)^{-}\left(\delta\mathbf{u}^{k+1}\right)^{-}-\left(\frac{\partial\mathcal{F}lx}{\partial\mathbf{u}}\right)^{+}\left(\delta\mathbf{u}^{k}\right)^{+}. \tag{55}$$

Once convergence is obtained, the volume and trace unknown are updated as

$$\mathbf{u}\longrightarrow\mathbf{u}+\delta\mathbf{u},$$
$$\hat{\mathbf{u}}\longrightarrow\hat{\mathbf{u}}+\delta\hat{\mathbf{u}},$$

and we can proceed with the next Newton step.

For the first example, we study the convergence of the iHDG-II solver for a translating vortex solution [69] whose exact solution is given by

$$H^e=\left[1-\frac{(\gamma-1)}{16\gamma\pi^2}\beta^2e^{2\left(1-R^2\right)}\right]^{\frac{1}{\gamma-1}},\qquad u^e=1-\beta e^{\left(1-R^2\right)}\frac{(y-y_0)}{2\pi},$$
$$v^e=\beta e^{\left(1-R^2\right)}\frac{(x-t-x_0)}{2\pi},$$

with $R^2=(x-t-x_0)^2+(y-y_0)^2$, $x_0=5$, $y_0=0$, and $\beta=5$. We take $\gamma=g=2$, and a flat bathymetry $b=b_x=b_y=0$. The domain considered is $\Omega=[3.5,5.5]\times[-1,1]$ and structured quadrilateral elements are used. The exact solution is used to enforce the boundary condition. For time discretization, we again use the Crank–Nicolson method, in which the time step is $\Delta t=10^{-4}$ and there are 100 time steps. In Fig. 4, we compare the h-convergence rates obtained with the iHDG-II algorithm and the direct solver. Results from both solvers are on top of each other with the convergence rate between p and $(p+1/2)$. In this case, each time step takes 2–3 Newton iterations, each of which takes less than 10 iHDG-II iterations. The stopping criterion for iHDG-II algorithm is based on (49) and the stopping tolerance is taken to be 10^{-10} for both iHDG-II and Newton iterations.

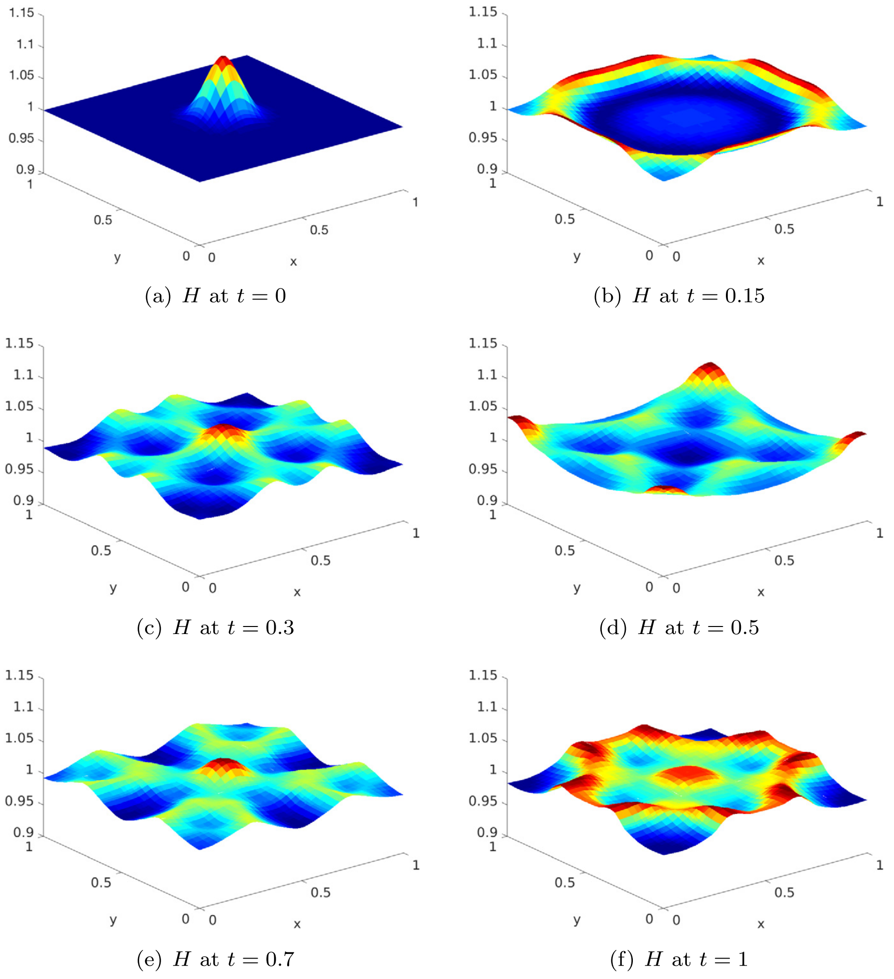

Fig. 5. Evolution of the water depth H for the water drop test case with iHDG-II algorithm.

Next, we consider the water drop problem considered in [24,70,71]. The initial conditions are

$$H(x,y,0) := 1 + 0.1\exp\left[-100(x-0.5)^2 - 100(y-0.5)^2\right],$$

and

$$Hu(x,y,0) = Hv(x,y,0) = 0,$$

that is, the flow is initially at rest.

We consider the case with flat bottom i.e., $b = b_x = b_y = 0$ and the domain of interest is $\Omega = [0,1]^2$. A structured quadrilateral mesh with 64 elements ($h = 0.125$) and solution order $p = 6$ is used. Wall boundary conditions are applied to the entire boundary $\partial\Omega$. The Crank–Nicolson method with a time step of $\Delta t = 0.0005$ is employed and the simulation is run for 2000 time steps. The time evolution of the water depth H and the depth averaged y-velocity v are shown in Figs. 5 and 6, respectively. The u velocity evolution is same as v but rotated by 90 degrees, and hence is not shown. The numerical results are comparable to those in [24,70,71]. In Fig. 7, we show the number of iHDG-II iterations taken per Newton iteration at the indicated times. The horizontal axis represents the number of Newton iterations taken from $(t - \Delta t)$ to t at the indicated times t. The markers in the vertical axis indicate the number of iHDG-II iterations taken at the corresponding Newton iteration in the horizontal axis to solve the linear system (53). The stopping tolerance is taken to be 10^{-10} for both iHDG-II and Newton iterations. As can be seen, approximately 16 Newton iterations are required per time step and the number of iHDG-II iterations decreases with each Newton iteration. The reason is that, after each Newton iteration, the initial guess (the solution from the previous Newton iteration) for iHDG iteration is improved and upon convergence Newton steps are smaller.

In Table 7 we compare the maximum number of iHDG-II iterations and Newton iterations taken per time step for different meshsizes and time steps. Similar to linear problems the number of iHDG-II iterations increases for finer mesh-

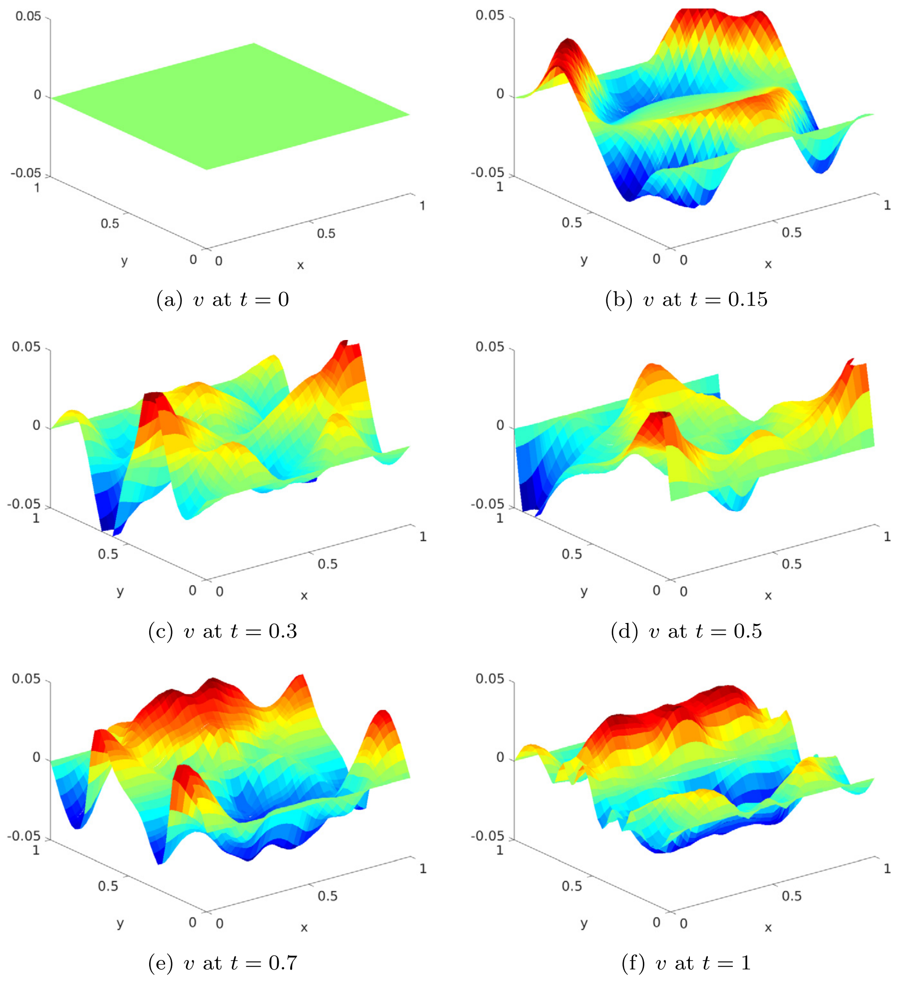

Fig. 6. Evolution of depth averaged y-velocity v for the water drop test case with iHDG-II algorithm.

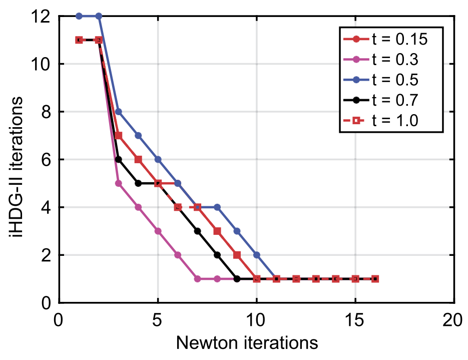

Fig. 7. Number of iHDG-II iterations taken for each Newton iteration at different times for the water drop test case with $h = 0.125$, $p = 6$ and $\Delta t = 0.0005$.

sizes, higher solution orders, and larger time steps. The number of Newton iterations on the other hand decreases for finer meshsizes and approaches a constant. This is well-known as the number of Newton iterations depends on the non-linearity of the problem and how well it is captured by the meshsize. Once the nonlinearity is captured by a particular meshsize and solution order, the number of iterations remains unchanged with further refinements [72,73]. For smaller time steps the number of Newton iterations is reduced since the solutions at two consecutive time steps are close to each other.

Table 7
Number of iHDG-II and Newton iterations for different meshsizes and time stepsizes for the water drop test case of nonlinear shallow water system.

h	p	iHDG-II		Newton	
		$\Delta t = 10^{-3}$	$\Delta t = 10^{-4}$	$\Delta t = 10^{-3}$	$\Delta t = 10^{-4}$
0.25	1	7	5	7	4
0.125	1	8	5	6	3
0.0625	1	10	5	3	3
0.03125	1	13	6	3	3
0.25	2	8	5	8	4
0.125	2	9	5	6	4
0.0625	2	12	6	6	3
0.03125	2	17	7	3	3
0.25	3	9	5	9	5
0.125	3	11	5	8	4
0.0625	3	14	6	3	3
0.03125	3	20	7	3	3
0.25	4	9	5	10	5
0.125	4	12	6	9	4
0.0625	4	18	7	5	3
0.03125	4	26	9	7	3

Table 8
Comparison of iHDG-I and iHDG-II methods for different κ.

h	p	iHDG-I			iHDG-II		
		$\kappa = 10^{-2}$	$\kappa = 10^{-3}$	$\kappa = 10^{-6}$	$\kappa = 10^{-2}$	$\kappa = 10^{-3}$	$\kappa = 10^{-6}$
0.5	1	24	23	23	17	17	17
0.25	1	30	34	35	25	25	26
0.125	1	50	55	56	35	37	38
0.0625	1	90	94	97	62	64	65
0.5	2	26	24	25	17	19	19
0.25	2	41	42	42	27	27	27
0.125	2	66	67	67	42	43	43
0.0625	2	*	109	110	67	70	71
0.5	3	27	31	31	19	19	19
0.25	3	33	33	38	24	26	27
0.125	3	*	58	60	38	39	41
0.0625	3	*	102	106	69	69	71
0.5	4	26	27	27	17	19	19
0.25	4	50	41	43	26	27	27
0.125	4	*	71	72	42	45	46
0.0625	4	*	123	125	73	78	79

6.4. *Linear convection–diffusion equation*

In this section the following exact solution for equation (37) is considered

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

The forcing is chosen such that it corresponds to the exact solution. The velocity field is chosen as $\boldsymbol{\beta} = (1+z, 1+x, 1+y)$ and we take $\nu = 1$. The domain is $[0,1]\times[0,1]\times[0,1]$. A structured hexahedral mesh is used for the simulations. The stopping criterion based on the exact solution (48) is used.

In Table 8 we report the number of iterations taken by iHDG-I and iHDG-II methods for different values of the diffusion coefficient κ. Similar to the shallow water equations, due to the lack of stability in $\boldsymbol{\sigma}$, iHDG-I diverges when κ is large for fine meshes and/or high solution orders. The iHDG-II method, on the other hand, converges for all the meshes and solution orders, and the number of iterations are smaller than that of the iHDG-I method. Next, we verify the growth of iHDG-II iterations predicted by the asymptotic result (47).

Since $\min\left\{\kappa^{-1}, \lambda\right\} = \lambda$ for all the numerical results considered here, due to (47) we expect the number of iHDG-II iterations to be independent of κ and this can be verified in Table 8. In Figs. 8(a), 8(b) and 8(c) the growth of iterations with respect to meshsize h for $\kappa = 10^{-2}, 10^{-3}$ and 10^{-6} are compared. In the asymptotic limit, for all the cases, the ratio of successive iterations reaches a value of around 1.7 which is close to the theoretical prediction 2. Hence the theoretical

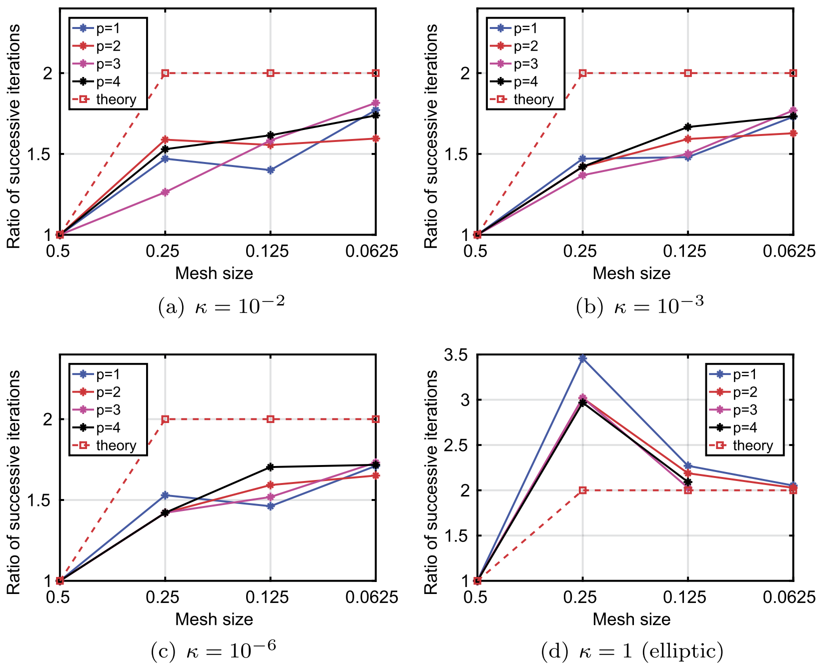

Fig. 8. Ratio of successive iterations as we refine the mesh for the iHDG-II method for different κ.

Table 9
Growth of iterations with p for the iHDG-II method for the elliptic equation.

p	Meshsize (h)				Asymptotics
	0.5	0.25	0.125	0.0625	
2	2.41	2.11	2.03	2.01	2
3	4.07	3.55	3.17	*	3.33
4	6.09	5.23	4.81	*	5

analysis predicts well the growth of iterations with respect to the meshsize h. On the other hand, columns 6–8 in Table 8 show that the iterations are almost independent of solution orders. This is not predicted by the theoretical results which indicates that the number of iterations scales like $\mathcal{O}(p^2)$. The reason is due to the convection dominated nature of the problem, for which we have shown that the number of iterations is independent of the solution order.

Finally, we consider the elliptic regime with $\kappa = 1$ and $\boldsymbol{\beta} = 0$. For this case we use the following stopping criterion based on the direct solver solution u_{direct}

$$\left\| u^k - u_{direct} \right\|_{L^2(\Omega)} < 10^{-10}.$$

As shown in Fig. 8(d) and Table 9, our theoretical analysis predicts well the relation between the number of iterations and the meshsize and the solution order. In Table 9 "$*$" represents that the scheme has reached the maximum number of iterations specified i.e. 2000 and didn't converge to the specified tolerance limit of 10^{-10} yet.

6.5. SPE10 test case

In this section we consider a benchmark problem of subsurface flow through porous media from the Tenth Society of Petroleum Comparative Solution Project (SPE10, model 2) [74]. The flow is governed by the following elliptic equation (Darcy law) written in the first-order form,

$$\begin{aligned} \boldsymbol{\sigma} &= -\kappa \nabla u \quad \text{on} \quad \Omega, \\ \nabla \cdot \boldsymbol{\sigma} &= f \quad \text{on} \quad \Omega, \end{aligned} \tag{56}$$

where $\boldsymbol{\sigma}$ is the Darcy velocity, u is the pressure and κ is the permeability.

We consider the permeability field corresponding to the 75th layer as shown in Fig. 9(a). The permeability field varies by six orders of magnitude and is highly heterogeneous. It gives rise to extremely complex velocity fields. The considered

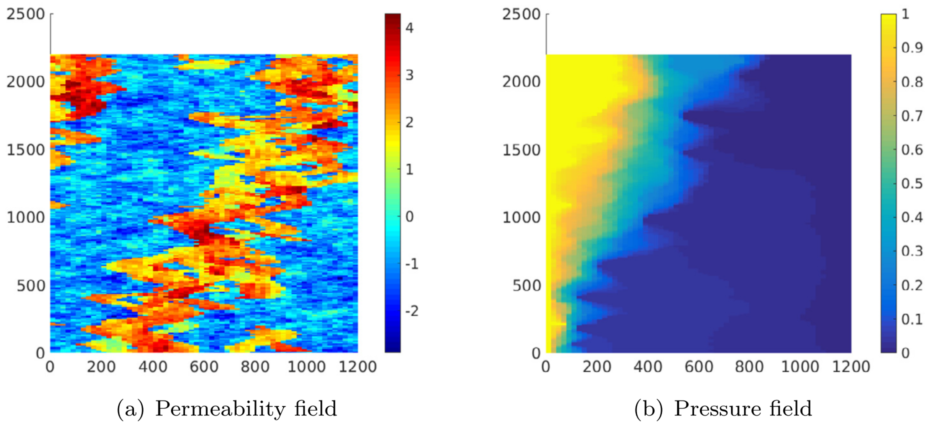

Fig. 9. SPE10 test case: (a) permeability field in log scale, and (b) pressure field from direct solver for $p = 1$ solution.

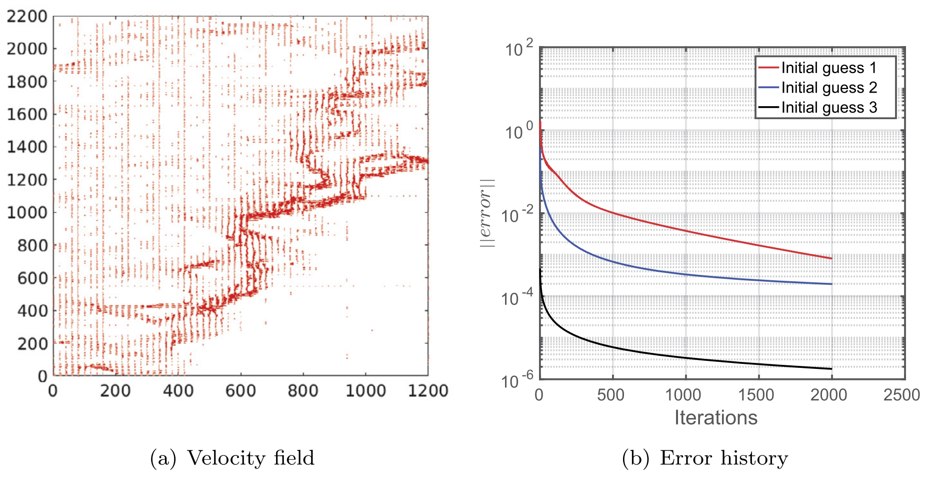

Fig. 10. SPE10 test case: (a) velocity field from direct solver for $p = 1$ solution, and (b) error history with respect to iHDG-II iterations for the three different initial guesses.

domain is 1200×2200 $[ft^2]$. The mesh is chosen as 60×220 quadrilateral elements, so that the mesh skeleton aligns with the discontinuities in the permeability and the permeability field is constant within each element. We choose $f = 0$, which corresponds to the fact that there is no source or sink. For the boundary conditions, we take the pressure on the left and right faces to be 1 and 0 respectively. On the top and bottom faces, no-flux boundary condition $\boldsymbol{\sigma} \cdot \mathbf{n} = 0$ is applied. We use the upwind HDG flux (39) with the velocity $\boldsymbol{\beta} = 0$ for this problem. In Figs. 9(b) and 10(a) we show the pressure field and the velocity field using direct solver with solution order $p = 1$.

We next consider the iHDG-II algorithm with three different initial guesses:

1. zero,
2. solution of equation (56) with the average of the original permeability field over the whole domain,
3. solution of equation (56) with the permeability field in each element given by

$$\kappa = \kappa + rand(0, 1) \times \kappa,$$

where $rand(0, 1)$ is a random number between 0 and 1, i.e. we randomly perturb the original permeability value in each element by 0%–100%.

In Fig. 11 we show the three initial guesses for the pressure and in Fig. 12 are the pressure fields corresponding to these initial guesses after 2000 iHDG-II iterations using solution order $p = 1$. Since the third initial guess is the closest to the direct solution (compared to Fig. 9(b)), the corresponding pressure field is almost the same as the direct solution, while the others are not. The conclusion is similar for the corresponding velocity fields in Figs. 13 and 14. In order to have a more quantitative comparison, we plot the relative error

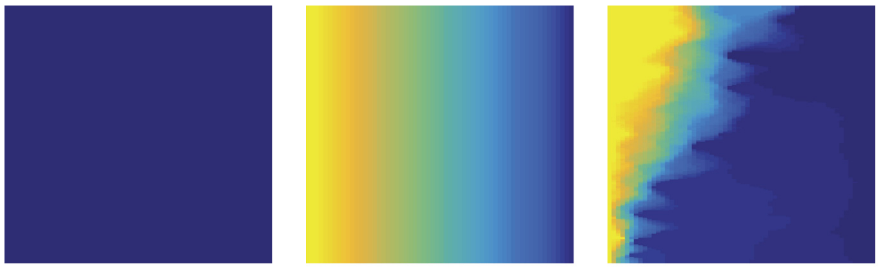
(a) Initial guess 1 (b) Initial guess 2 (c) Initial guess 3

Fig. 11. SPE10 test case: three different initial guesses for the pressure field.

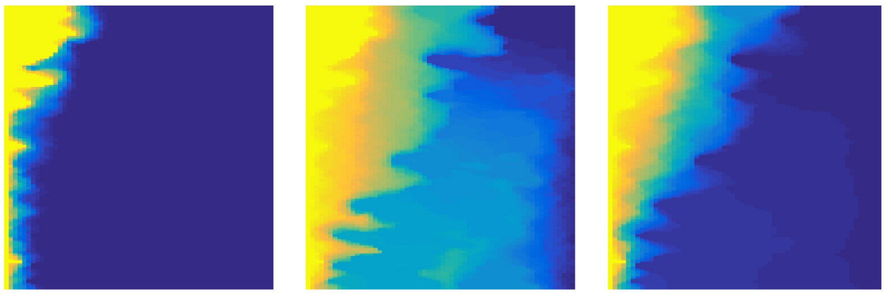
(a) iHDG solution with initial guess 1 (b) iHDG solution with initial guess 2 (c) iHDG solution with initial guess 3

Fig. 12. SPE10 test case: the pressure fields obtained with the iHDG-II algorithm after 2000 iterations with three different initial guesses.

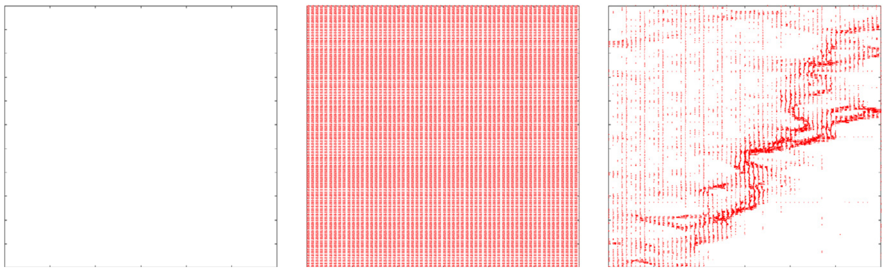
(a) Initial guess 1 (b) Initial guess 2 (c) Initial guess 3

Fig. 13. SPE10 test case: three different initial guesses for the velocity field.

$$\|error\| = \frac{\sqrt{\|\boldsymbol{\sigma}^{k+1} - \boldsymbol{\sigma}^{k}\|^2 + \|u^{k+1} - u^{k}\|^2}}{\sqrt{\|\boldsymbol{\sigma}^{k+1}\|^2 + \|u^{k+1}\|^2}}$$

in Fig. 10(b) as the number of iHDG iterations increases. For all of the initial guesses, there is a quick drop in the errors in the first few iterations and thereafter the errors decrease slowly. Compared to the other examples, this is the most challenging one, for which the iHDG algorithm, if used as a direct solver, could be ineffective. Nevertheless, this is a common feature of many of the fixed-point iterative methods like Jacobi and Gauss–Seidel [75,76]. Ongoing work is to employ iHDG as a smoother for multigrid methods or as a preconditioner for GMRES iterations, and we shall report our findings in the near future.

7. Conclusions

We have presented an iterative HDG approach which improves upon our previous work [45] in several aspects. In particular, it converges in a finite number of iterations for the scalar transport equation and is unconditionally convergent for both the linearized shallow water system and the convection–diffusion equation. Moreover, compared to our previous work [45], we provide several additional findings: 1) we make a connection between iHDG and the parareal method, which reveals interesting similarities and differences between the two methods; 2) we show that iHDG can be considered as a *locally implicit* method, and hence being somewhat in between fully explicit and fully implicit approaches; 3) for both the

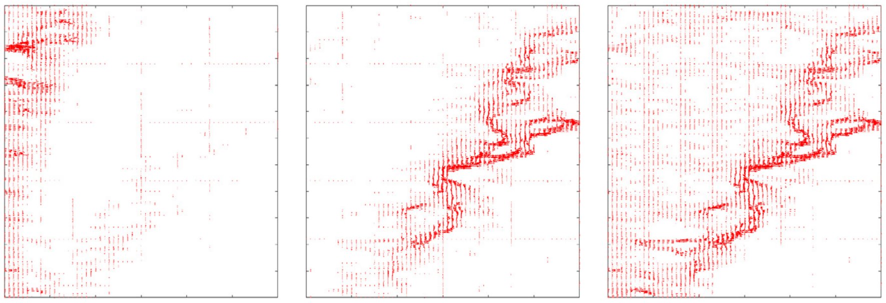
(a) iHDG solution with initial guess 1 (b) iHDG solution with initial guess 2 (c) iHDG solution with initial guess 3

Fig. 14. SPE10 test case: velocity field obtained with iHDG-II algorithm after 2000 iterations with three different initial guesses.

linearized shallow water system and the convection–diffusion equation, using an asymptotic approximation, we uncover a relationship between the number of iterations and time stepsize, solution order, meshsize and the equation parameters. This allows us to choose the time stepsize such that the number of iterations is approximately independent of the solution order and the meshsize; 4) we show that iHDG-II has improved stability and convergence rates over iHDG-I; 5) we provide both strong and weak scalings of our iHDG approach up to 16,384 cores; and 6) we show how iHDG approaches can be used as a linear solver for nonlinear problem. Ongoing work is to use iHDG algorithms as smoother for multigrid methods and as preconditioner for GMRES approaches.

Acknowledgements

We are indebted to Professor Hari Sundar for sharing his high-order finite element library **homg**, on which we have implemented the iHDG algorithms and produced numerical results. We thank Texas Advanced Computing Center (TACC) for processing our requests so quickly regarding running our simulations on Stampede. The first author would like to thank Stephen Shannon for helping in proving some results and various fruitful discussions on this topic. We thank Professor Martin Gander for pointing us to the reference [49]. We thank the anonymous referees for their useful comments which improved this paper substantially. The first author would like to dedicate this work in memory of his late grandfather Gopalakrishnan Thiruvankatam.

Appendix A. Proof of well-posedness of local solver of the iHDG-II method for the linear convection–diffusion equation

Proof. Choosing $\boldsymbol{\sigma}^{k+1}$ and u^{k+1} as test functions in (40a)–(40b), integrating the second term in (40a) by parts, using (12) for second term in (40b), and then summing up the resulting two equations we obtain

$$\kappa^{-1}\left(\boldsymbol{\sigma}^{k+1},\boldsymbol{\sigma}^{k+1}\right)_K+\left(\left\{\nu-\frac{\nabla\cdot\boldsymbol{\beta}}{2}\right\}u^{k+1},u^{k+1}\right)_K \\ +\left\langle\left(\frac{\boldsymbol{\beta}\cdot\mathbf{n}}{2}+\tau\right)u^{k+1},u^{k+1}\right\rangle_{\partial K}+\left\langle(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}-\tau u^{k+1}),\hat{u}^{k,k+1}\right\rangle_{\partial K}=0. \quad (57)$$

Substituting (41) in the above equation and simplifying some terms we get

$$\sum_K \kappa^{-1}\left\|\left(\boldsymbol{\sigma}^{k+1}\right)^-\right\|_K^2+\left(\left\{\nu-\frac{\nabla\cdot\boldsymbol{\beta}}{2}\right\}\left(u^{k+1}\right)^-,\left(u^{k+1}\right)^-\right)_K \\ +\left\langle\left\{\frac{|\boldsymbol{\beta}\cdot\mathbf{n}|^2+2}{2\alpha}\right\}\left(u^{k+1}\right)^-,\left(u^{k+1}\right)^-\right\rangle_{\partial K}+\left\langle\frac{1}{\alpha}\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-,\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\rangle_{\partial K} \\ +\left\langle\frac{\boldsymbol{\beta}\cdot\mathbf{n}^-}{\alpha}\left(u^{k+1}\right)^-,\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\rangle_{\partial K}=\sum_{\partial K}-\left\langle\frac{1}{\alpha}\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-,\left(\boldsymbol{\sigma}^{k}\cdot\mathbf{n}\right)^+\right\rangle_{\partial K} \\ -\left\langle\left\{\frac{\boldsymbol{\beta}\cdot\mathbf{n}^++\tau^+}{\alpha}\right\}\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-,\left(u^k\right)^+\right\rangle_{\partial K}+\left\langle\frac{\tau^-}{\alpha}\left(u^{k+1}\right)^-,\left(\boldsymbol{\sigma}^{k}\cdot\mathbf{n}\right)^+\right\rangle_{\partial K} \\ +\left\langle\left\{\frac{\tau^-(\boldsymbol{\beta}\cdot\mathbf{n}^++\tau^+)}{\alpha}\right\}\left(u^{k+1}\right)^-,\left(u^k\right)^+\right\rangle_{\partial K}. \quad (58)$$

Using the identity

$$\left\langle \frac{\boldsymbol{\beta}\cdot\mathbf{n}}{\alpha}u^{k+1}, \boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right\rangle_{\partial K} = \left\| \frac{1}{\sqrt{2\alpha}}\left(\boldsymbol{\beta}\cdot\mathbf{n}u^{k+1} + \boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)\right\|^2_{\partial K} - \left\langle \frac{\boldsymbol{\beta}\cdot\mathbf{n}^2}{2\alpha}u^{k+1}, u^{k+1}\right\rangle_{\partial K} - \left\langle \frac{1}{2\alpha}\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}, \boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right\rangle_{\partial K}, \quad (59)$$

and the coercivity condition (38) we can write (58) as

$$\begin{aligned}
&\sum_K \kappa^{-1}\left\|\left(\boldsymbol{\sigma}^{k+1}\right)^-\right\|^2_K + \lambda\left\|\left(u^{k+1}\right)^-\right\|^2_K + \left\|\left(u^{k+1}\right)^-\right\|^2_{1/\alpha,\partial K} \\
&+ \left\|\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\|^2_{1/2\alpha,\partial K} + \left\|\frac{1}{\sqrt{2\alpha}}\left\{\boldsymbol{\beta}\cdot\mathbf{n}^-\left(u^{k+1}\right)^- + \left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\}\right\|^2_{\partial K} \\
&\le \sum_{\partial K} -\left\langle \frac{1}{\alpha}\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-, \left(\boldsymbol{\sigma}^{k}\cdot\mathbf{n}\right)^+\right\rangle_{\partial K} - \left\langle \left\{\frac{\boldsymbol{\beta}\cdot\mathbf{n}^+ + \tau^+}{\alpha}\right\}\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-, \left(u^k\right)^+\right\rangle_{\partial K} \\
&+ \left\langle \frac{\tau^-}{\alpha}\left(u^{k+1}\right)^-, \left(\boldsymbol{\sigma}^{k}\cdot\mathbf{n}\right)^+\right\rangle_{\partial K} + \left\langle \left\{\frac{\tau^-(\boldsymbol{\beta}\cdot\mathbf{n}^+ + \tau^+)}{\alpha}\right\}\left(u^{k+1}\right)^-, \left(u^k\right)^+\right\rangle_{\partial K}. \qquad (60)
\end{aligned}$$

Since all the terms on the left hand side are positive, when we take the "forcing" to the local solver $\left\{\left(u^k\right)^+, \left(\boldsymbol{\sigma}^k\right)^+\right\} = \{0, \mathbf{0}\}$, the only solution possible is $\left\{\left(u^{k+1}\right)^-, \left(\boldsymbol{\sigma}^{k+1}\right)^-\right\} = \{0, \mathbf{0}\}$ and hence the method is well-posed. □

Appendix B. Proof of convergence of the iHDG-II method for the linear convection–diffusion equation

Proof. In equation (60) omitting the last term on the left hand side and using Cauchy–Schwarz and Young's inequalities for the terms on the right hand side we get

$$\begin{aligned}
&\sum_K \kappa^{-1}\left\|\left(\boldsymbol{\sigma}^{k+1}\right)^-\right\|^2_K + \lambda\left\|\left(u^{k+1}\right)^-\right\|^2_K + \left\|\left(u^{k+1}\right)^-\right\|^2_{1/\alpha,\partial K} \\
&+ \left\|\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\|^2_{1/2\alpha,\partial K} \le \sum_{\partial K}\frac{1}{2\varepsilon}\left\langle\left\{\frac{\tau^- + 1}{\alpha}\right\}\left(\boldsymbol{\sigma}^k\cdot\mathbf{n}\right)^+, \left(\boldsymbol{\sigma}^k\cdot\mathbf{n}\right)^+\right\rangle_{\partial K} \\
&+ \frac{1}{2\varepsilon}\left\langle\left\{\frac{(1+\tau^-)(\boldsymbol{\beta}\cdot\mathbf{n}^+ + \tau^+)}{\alpha}\right\}\left(u^k\right)^+, \left(u^k\right)^+\right\rangle_{\partial K} \\
&+ \frac{\varepsilon}{2}\left\langle\left\{\frac{1+\tau^+ + \boldsymbol{\beta}\cdot\mathbf{n}^+}{\alpha}\right\}\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-, \left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\rangle_{\partial K} \\
&+ \frac{\varepsilon}{2}\left\langle\left\{\frac{\tau^-(1+\tau^+ + \boldsymbol{\beta}\cdot\mathbf{n}^+)}{\alpha}\right\}\left(u^{k+1}\right)^-, \left(u^{k+1}\right)^-\right\rangle_{\partial K}. \qquad (61)
\end{aligned}$$

We can write the above inequality as

$$\begin{aligned}
&\kappa^{-1}\left\|\left(\boldsymbol{\sigma}^{k+1}\right)^-\right\|^2_K + \lambda\left\|\left(u^{k+1}\right)^-\right\|^2_K + \frac{1}{\bar{\alpha}}\left\|\left(u^{k+1}\right)^-\right\|^2_{\partial K} + \frac{1}{2\bar{\alpha}}\left\|\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\|^2_{\partial K} \\
&\le \frac{\bar{\tau}+1}{2\varepsilon\alpha_*}\left\|\left(\boldsymbol{\sigma}^k\cdot\mathbf{n}\right)^+\right\|^2_{\partial K} + \frac{(1+\bar{\tau})(\|\boldsymbol{\beta}\cdot\mathbf{n}\|_{L^\infty(\partial K)} + \bar{\tau})}{2\varepsilon\alpha_*}\left\|\left(u^k\right)^+\right\|^2_{\partial K} \\
&+ \frac{\varepsilon(1+\bar{\tau} + \|\boldsymbol{\beta}\cdot\mathbf{n}\|_{L^\infty(\partial K)})}{2\alpha_*}\left\|\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\|^2_{\partial K} \\
&+ \frac{\varepsilon\bar{\tau}(1+\bar{\tau} + \|\boldsymbol{\beta}\cdot\mathbf{n}\|_{L^\infty(\partial K)})}{2\alpha_*}\left\|\left(u^{k+1}\right)^-\right\|^2_{\partial K}, \qquad (62)
\end{aligned}$$

where $\bar{\tau} := \|\tau\|_{L^\infty(\partial\Omega_h)}$, $\bar{\alpha} := \|\alpha\|_{L^\infty(\partial\Omega_h)}$, and $\alpha_* := \inf_{\partial K\in\partial\Omega_h}\alpha$.

By the inverse trace inequality (28) we infer from (62) that

$$\sum_{\partial K}\mathcal{B}_1\left\|\left(\boldsymbol{\sigma}^{k+1}\cdot\mathbf{n}\right)^-\right\|^2_{\partial K} + \mathcal{B}_2\left\|\left(u^{k+1}\right)^-\right\|^2_{\partial K} \le \sum_{\partial K}\left[\mathcal{C}_1\left\|\left(u^k\right)^+\right\|^2_{\partial K} + \mathcal{C}_2\left\|\left(\boldsymbol{\sigma}^k\cdot\mathbf{n}\right)^+\right\|^2_{\partial K}\right],$$

which implies

$$\left\| \left(\boldsymbol{\sigma}^{k+1} \cdot \mathbf{n}, u^{k+1} \right) \right\|_{\mathcal{E}_h}^2 \le \mathcal{D} \left\| \left(\boldsymbol{\sigma}^{k} \cdot \mathbf{n}, u^{k} \right) \right\|_{\mathcal{E}_h}^2 ,$$

where the constant $\mathcal{D}$ is computed as in (44). Therefore

$$\left\| \left(\boldsymbol{\sigma}^{k+1} \cdot \mathbf{n}, u^{k+1} \right) \right\|_{\mathcal{E}_h}^2 \le \mathcal{D}^{k+1} \left\| \left(\boldsymbol{\sigma}^{0} \cdot \mathbf{n}, u^{0} \right) \right\|_{\mathcal{E}_h}^2 . \tag{63}$$

Inequalities (62) and (63) imply

$$\left\| \left(\boldsymbol{\sigma}^{k+1}, u^{k+1} \right) \right\|_{\Omega_h}^2 \le (\mathcal{E}\mathcal{D} + \mathcal{F})\mathcal{D}^{k} \left\| \left(\boldsymbol{\sigma}^{0} \cdot \mathbf{n}, u^{0} \right) \right\|_{\mathcal{E}_h}^2 ,$$

and this concludes the proof. □

References

[1] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[2] P. LeSaint, P.A. Raviart, On a finite element method for solving the neutron transport equation, in: C. de Boor (Ed.), Mathematical Aspects of Finite Element Methods in Partial Differential Equations, Academic Press, 1974, pp. 89–145.
[3] C. Johnson, J. Pitkäranta, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation, Math. Comput. 46 (173) (1986) 1–26.
[4] J. Douglas, T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, Comput. Methods Appl. Sci. (1976) 207–216.
[5] M.F. Wheeler, An elliptic collocation-finite element method with interior penalties, SIAM J. Numer. Anal. 15 (1) (1978) 152–161.
[6] D.N. Arnold, An interior penalty finite element method with discontinuous elements, SIAM J. Numer. Anal. 19 (4) (1982) 742–760.
[7] B. Cockburn, G.E. Karniadakis, C.-W. Shu, Discontinuous Galerkin Methods: Theory, Computation and Applications, Lecture Notes in Computational Science and Engineering, vol. 11, Springer Verlag, Berlin, Heidelberg, New York, 2000.
[8] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2002) 1749–1779.
[9] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2009) 1319–1365.
[10] B. Cockburn, J. Gopalakrishnan, F.-J. Sayas, A projection-based error analysis of HDG methods, Math. Comput. 79 (271) (2010) 1351–1367.
[11] R.M. Kirby, S.J. Sherwin, B. Cockburn, To CG or to HDG: a comparative study, J. Sci. Comput. 51 (2012) 183–212.
[12] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations, J. Comput. Phys. 228 (2009) 3232–3254.
[13] B. Cockburn, B. Dong, J. Guzman, M. Restelli, R. Sacco, A hybridizable discontinuous Galerkin method for steady state convection–diffusion–reaction problems, SIAM J. Sci. Comput. 31 (2009) 3827–3846.
[14] H. Egger, J. Schoberl, A hybrid mixed discontinuous Galerkin finite element method for convection–diffusion problems, IMA J. Numer. Anal. 30 (2010) 1206–1234.
[15] B. Cockburn, J. Gopalakrishnan, The derivation of hybridizable discontinuous Galerkin methods for Stokes flow, SIAM J. Numer. Anal. 47 (2) (2009) 1092–1125.
[16] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinous Galerkin method for Stokes flow, Comput. Methods Appl. Mech. Eng. 199 (2010) 582–597.
[17] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations, J. Comput. Phys. 230 (2011) 1147–1170.
[18] D. Moro, N.C. Nguyen, J. Peraire, Navier–Stokes Solution Using Hybridizable Discontinuous Galerkin Methods, American Institute of Aeronautics and Astronautics, 2011, 2011–3407.
[19] N.C. Nguyen, J. Peraire, B. Cockburn, Hybridizable discontinuous Galerkin method for the time harmonic Maxwell's equations, J. Comput. Phys. 230 (2011) 7151–7175.
[20] L. Li, S. Lanteri, R. Perrrussel, A hybridizable discontinuous Galerkin method for solving 3D time harmonic Maxwell's equations, in: Numerical Mathematics and Advanced Applications 2011, Springer, 2013, pp. 119–128.
[21] N.C. Nguyen, J. Peraire, B. Cockburn, High-order implicit hybridizable discontinuous Galerkin method for acoustics and elastodynamics, J. Comput. Phys. 230 (2011) 3695–3718.
[22] R. Griesmaier, P. Monk, Error analysis for a hybridizable discontinous Galerkin method for the Helmholtz equation, J. Sci. Comput. 49 (2011) 291–310.
[23] J. Cui, W. Zhang, An analysis of HDG methods for the Helmholtz equation, IMA J. Numer. Anal. 34 (1) (2014) 279–295.
[24] T. Bui-Thanh, From Godunov to a unified hybridized discontinuous Galerkin framework for partial differential equations, J. Comput. Phys. 295 (2015) 114–146.
[25] T. Bui-Thanh, From Rankine–Hugoniot condition to a constructive derivation of HDG methods, in: Lecture Notes in Computational Sciences and Engineering, Springer, 2015, pp. 483–491.
[26] T. Bui-Thanh, Construction and analysis of HDG methods for linearized shallow water equations, SIAM J. Sci. Comput. 38 (6) (2016) A3696–A3719.
[27] J. Wang, X. Ye, A weak Galerkin finite element method for second-order elliptic problems, J. Comput. Appl. Math. 241 (2013) 103–115.
[28] J. Wang, X. Ye, A weak Galerkin mixed finite element method for second order elliptic problems, Math. Comput. 83 (2014) 2101–2126.
[29] Q. Zhai, R. Zhang, X. Wang, A hybridized weak Galerkin finite element scheme for the stokes equations, Sci. China Math. 58 (11) (2015) 2455–2472.
[30] L. Mu, J. Wang, X. Ye, A new weak Galerkin finite element method for the Helmholtz equation, IMA J. Numer. Anal. 35 (3) (2015) 1228–1255.
[31] J. Brown, Efficient nonlinear solvers for nodal high-order finite elements in 3D, J. Sci. Comput. 45 (1–3) (2010) 48–63.
[32] A. Crivellini, F. Bassi, An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations, Comput. Fluids 50 (1) (2011) 81–93.
[33] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.
[34] 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.
[35] P.-L. Lions, On the Schwarz alternating method, II: stochastic interpretation and order properties, in: Domain Decomposition Methods, Los Angeles, CA, 1988, SIAM, Philadelphia, PA, 1989, pp. 47–70.

[36] P.-L. Lions, On the Schwarz alternating method, III: a variant for nonoverlapping subdomains, in: Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, Houston, TX, 1989, SIAM, Philadelphia, PA, 1990, pp. 202–223.
[37] L. Halpern, Optimized Schwarz waveform relaxation: roots, blossoms and fruits, in: Domain Decomposition Methods in Science and Engineering XVIII, in: Lect. Notes Comput. Sci. Eng., vol. 70, Springer, Berlin, 2009, pp. 225–232.
[38] M.J. Gander, L. Gouarin, L. Halpern, Optimized Schwarz waveform relaxation methods: a large scale numerical study, in: Domain Decomposition Methods in Science and Engineering XIX, in: Lect. Notes Comput. Sci. Eng., vol. 78, Springer, Heidelberg, 2011, pp. 261–268.
[39] M.J. Gander, S. Hajian, Analysis of Schwarz methods for a hybridizable discontinuous Galerkin discretization, SIAM J. Numer. Anal. 53 (1) (2015) 573–597.
[40] M.-B. Tran, Parallel Schwarz waveform relaxation method for a semilinear heat equation in a cylindrical domain, C. R. Math. Acad. Sci. Paris 348 (13–14) (2010) 795–799.
[41] M.-B. Tran, A parallel four step domain decomposition scheme for coupled forward–backward stochastic differential equations, J. Math. Pures Appl. 96 (4) (2011) 377–394.
[42] M.-B. Tran, Overlapping domain decomposition: convergence proofs, in: Domain Decomposition Methods in Science and Engineering XX, Springer, 2013, pp. 493–500.
[43] M.-B. Tran, The behavior of domain decomposition methods when the overlapping length is large, Cent. Eur. J. Math. 12 (10) (2014) 1602–1614.
[44] B. Cockburn, O. Dubois, J. Gopalakrishnan, S. Tan, Multigrid for an HDG method, IMA J. Numer. Anal. (2013) 1–40.
[45] S. Muralikrishnan, M.-B. Tran, T. Bui-Thanh, iHDG: an iterative HDG framework for partial differential equations, SIAM J. Sci. Comput. 39 (5) (2017) S782–S808.
[46] M.J. Gander, S. Hajian, Block Jacobi for discontinuous Galerkin discretizations: no ordinary Schwarz methods, in: Domain Decomposition Methods in Science and Engineering XXI, Springer, 2014, pp. 305–313.
[47] M.J. Gander, S. Hajian, Analysis of Schwarz methods for a hybridizable discontinuous Galerkin discretization: the many subdomain case, arXiv:1603.04073.
[48] K.O. Friedrichs, Symmetric positive linear differential equations, Commun. Pure Appl. Math. XI (1958) 333–418.
[49] M.J. Gander, Analysis of the parareal algorithm applied to hyperbolic problems using characteristics, Bol. Soc. Esp. Mat. Apl. 42 (2008) 21–35.
[50] Z. Kaczkowski, The method of finite space–time elements in dynamics of structures, J. Tech. Phys. 16 (1) (1975) 69–84.
[51] J. Argyris, D. Scharpf, Finite elements in time and space, Nucl. Eng. Des. 10 (4) (1969) 456–464.
[52] J.T. Oden, A general theory of finite elements, II: applications, Int. J. Numer. Methods Eng. 1 (3) (1969) 247–259.
[53] C.M. Klaij, J.J. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations, J. Comput. Phys. 217 (2) (2006) 589–611.
[54] T. Ellis, J. Chan, L. Demkowicz, Robust DPG methods for transient convection–diffusion, in: Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations, Springer, 2016, pp. 179–203.
[55] S. Rhebergen, B. Cockburn, A space–time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains, J. Comput. Phys. 231 (11) (2012) 4185–4204.
[56] J.-L. Lions, Y. Maday, G. Turinici, Résolution d'edp par un schéma en temps pararéel, C. R. Acad. Sci., Ser. 1 Math. 332 (7) (2001) 661–668.
[57] I. Garrido, B. Lee, G. Fladmark, M. Espedal, Convergent iterative schemes for time parallelization, Math. Comput. 75 (255) (2006) 1403–1428.
[58] C. Farhat, M. Chandesris, Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications, Int. J. Numer. Methods Eng. 58 (9) (2003) 1397–1434.
[59] Y. Maday, G. Turinici, A parareal in time procedure for the control of partial differential equations, C. R. Math. 335 (4) (2002) 387–392.
[60] M. Minion, A hybrid parareal spectral deferred corrections method, Commun. Appl. Math. Comput. Sci. 5 (2) (2011) 265–301.
[61] M.J. Gander, S. Vandewalle, Analysis of the parareal time-parallel time-integration method, SIAM J. Sci. Comput. 29 (2) (2007) 556–578.
[62] F.X. Giraldo, T. Warburton, A high-order triangular discontinous Galerkin oceanic shallow water model, Int. J. Numer. Methods Fluids 56 (2008) 899–925.
[63] J. Chan, Z. Wang, A. Modave, J.-F. Remacle, T. Warburton, GPU-accelerated discontinuous Galerkin methods on hybrid meshes, J. Comput. Phys. 318 (2016) 142–168.
[64] T. Warburton, J.S. Hesthaven, On the constants in *hp*-finite element trace inverse inequalities, Comput. Methods Appl. Mech. Eng. 192 (25) (2003) 2765–2773.
[65] M. Taylor, J. Tribbia, M. Iskandarani, The spectral element method for the shallow water equations on the sphere, J. Comput. Phys. 130 (1) (1997) 92–108.
[66] L.C. Wilcox, G. Stadler, C. Burstedde, O. Ghattas, A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media, J. Comput. Phys. 229 (24) (2010) 9373–9396.
[67] C. Burstedde, O. Ghattas, M. Gurnis, T. Isaac, G. Stadler, T. Warburton, L.C. Wilcox, Extreme-scale AMR, in: SC10: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ACM/IEEE, 2010.
[68] C. Burstedde, O. Ghattas, M. Gurnis, E. Tan, T. Tu, G. Stadler, L.C. Wilcox, S. Zhong, Scalable adaptive mantle convection simulation on petascale supercomputers, in: SC08: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ACM/IEEE, 2008.
[69] R. Gandham, D. Medina, T. Warburton, GPU accelerated discontinuous Galerkin methods for shallow water equations, arXiv:1403.1661.
[70] O. San, K. Kara, High-order accurate spectral difference method for shallow water equations, Int. J. Res. Rev. Appl. Sci. 6 (2011) 41–54.
[71] Y. Xing, X. Zhang, Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes, J. Sci. Comput. 57 (2013) 19–41.
[72] M. Weiser, A. Schiela, P. Deuflhard, Asymptotic mesh independence of Newton's method revisited, SIAM J. Numer. Anal. 42 (5) (2005) 1830–1845.
[73] E. Allgower, K. Böhmer, Application of the mesh independence principle to mesh refinement strategies, SIAM J. Numer. Anal. 24 (6) (1987) 1335–1351.
[74] M. Christie, M. Blunt, et al., Tenth SPE comparative solution project: a comparison of upscaling techniques, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2001.
[75] W.L. Briggs, S.F. McCormick, et al., A Multigrid Tutorial, vol. 72, Siam, 2000.
[76] U. Trottenberg, C.W. Oosterlee, A. Schuller, Multigrid, Academic Press, 2000.