Journal of Computational Physics 230 (2011) 3695–3718

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics

N.C. Nguyen [a,*], J. Peraire [a], B. Cockburn [b]

[a] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 24 May 2010
Received in revised form 15 January 2011
Accepted 25 January 2011

Keywords:
Finite element method
Discontinuous Galerkin methods
Hybrid/mixed methods
Superconvergence
Postprocessing
Acoustics
Elastodynamics

ABSTRACT

We present a class of hybridizable discontinuous Galerkin (HDG) methods for the numerical simulation of wave phenomena in acoustics and elastodynamics. The methods are fully implicit and high-order accurate in both space and time, yet computationally attractive owing to their following distinctive features. First, they reduce the globally coupled unknowns to the approximate trace of the velocity, which is defined on the element faces and single-valued, thereby leading to a significant saving in the computational cost. In addition, all the approximate variables (including the approximate velocity and gradient) converge with the optimal order of $k+1$ in the L^2-norm, when polynomials of degree $k \geqslant 0$ are used to represent the numerical solution and when the time-stepping method is accurate with order $k+1$. When the time-stepping method is of order $k+2$, superconvergence properties allows us, by means of local postprocessing, to obtain better, yet inexpensive approximations of the displacement and velocity at any time levels for which an enhanced accuracy is required. In particular, the new approximations converge with order $k+2$ in the L^2-norm when $k \geqslant 1$ for both acoustics and elastodynamics. Extensive numerical results are provided to illustrate these distinctive features.

1. Introduction

The numerical solution of wave phenomena in acoustics, elastodynamics and electromagnetics has found important applications in many areas of engineering and science such as aerospace, geophysics, civil engineering, mechanical engineering, telecommunication, medicine, and biology. Examples of applications include noise reduction, stealth technology, seismic and earthquake, nondestructive testing, antenna design, the detection of hidden targets, radar, satellite, nanophotonic devices, optical fibers, waveguides, and medical imaging. The wide range of applications has led to the development of many computational techniques for solving hyperbolic systems of partial differential equations (PDEs) governing wave phenomena.

The finite element method has been among the most popular techniques for the spatial discretization of wave propagation problems due to its ability to handle complex geometries and inhomogeneous materials, provide high-order accuracy, as well as perform h/p adaptivity. There are several spatial discretization strategies within the finite element method. They include continuous Galerkin/Petrov–Galerkin methods, spectral element methods, mixed finite element methods, extended finite element methods, and discontinuous Galerkin/Petrov–Galerkin methods. Each method has its own strengths and weaknesses that make it ideal for some applications, but not the best choice for others.

* Corresponding author.
E-mail address: cuongng@mit.edu (N.C. Nguyen).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.01.035

For instance, discontinuous Galerkin methods [2–4,18,19,24,23,25] work well on arbitrary meshes, result in stable high-order accurate (low dispersion) discretizations of hyberbolic systems, allow for a simple and unambiguous imposition of boundary conditions, and are very flexible to parallelization and adaptivity. One major criticism of many DG methods is that they have too many degrees of freedom due to nodal duplication at the element boundary interfaces. However, when used with explicit time-stepping schemes, DG methods [7,19,28,30] provide block-diagonal mass matrices to be inverted, which results in very low storage and efficient numerical schemes for wave propagation problems. One major disadvantage with this approach is that the timestep size is restricted by the smallest element in the mesh and the degree of polynomials used in representing the numerical solution. Even a few small elements can render the timestep size so small that it actually leads to very high computational cost. The timestep restriction is not only applicable to DG methods, but in fact to any numerical methods with explicit time integration. On the other hand, when used with implicit time-stepping schemes to alleviate the timestep restriction, many existing DG methods result in a discrete system of too many globally coupled degrees of freedom [39,40]. Still, there are a variety of applications for which implicit schemes would be much more efficient than explicit schemes (and, of course, vice versa).

Regardless of time integration schemes and spatial discretization methods, there are two general approaches for solving second-order hyperbolic equations. One popular approach is to use finite element approximation in space and discretize the second-order time derivative directly by using finite difference, leading to the well-known class of finite element methods for spatial discretization and Newmark methods for temporal discretization. However, this approach is only second-order accurate in time and it is difficult to extend it beyond second-order accuracy. The second approach is to transform the second-order hyperbolic equations into a first-order hyperbolic system of differential equations. One then discretizes the spatial derivatives to obtain an ordinary differential equation (ODE) system, which can be discretized in time by ODE techniques such as linear multistep methods, Runge–Kutta methods, or even DG in time. This approach is very popular for explicit time integration since the mass matrix is either block-diagonal by using DG discretization in space [7,8,19,28,30] or can be reduced to an approximate block-diagonal matrix by using mass lumping techniques [20]. However, the approach seems less appealing to implicit time integration since it results in a matrix system which might be considerably larger than the matrix system obtained with the continuous Galerkin (CG) method. Indeed, the use of the mixed Raviart–Thomas method for the scalar wave equation leads to a global matrix whose size is the degrees of freedom of the approximate gradient vector [27]. This computational inefficiency is the main motivation behind the development of a class of mixed finite elements [5] to be used with an explicit scheme through mass lumping.

In this paper, we present a class of hybridizable discontinuous Galerkin (HDG) methods for spatial discretization of the first-order formulation of acoustic and elastic wave equations. Both the backward difference formula (BDF) schemes and diagonally implicit Runge–Kutta (DIRK) methods are used for time integration. The resulting methods are fully implicit, unstructured, and high-order accurate in both space and time; yet they are computationally attractive because the only globally coupled unknown is the numerical trace of the velocity field. Since the numerical trace is defined on the element faces and single-valued, the HDG methods may have significantly less global degrees of freedom than other DG methods using implicit time integration. Another attractive feature of the HDG methods is that they yield optimal convergence of order $k+1$ in the L^2-norm for all the approximate variables and possess some superconvergence properties for other approximate quantities. Based on these convergence properties we develop new local postprocessing schemes to obtain better approximations of the displacement and velocity at the time levels for which an enhanced accuracy is required. Not only less expensive to compute than the original approximations the new approximations converge faster with order $k+2$ in the L^2-norm when $k \geqslant 1$ for both acoustics and elastodynamics, when time-stepping methods are of order $k+2$. Extensive numerical experiments are provided to illustrate those features and show that the present method outperforms the standard CG-Newmark method for smooth problems.

This work is a continuation of our recent effort [34,35,37,32,38] on the development of HDG methods for solving time-dependent partial differential equations. The first HDG method was introduced for diffusion–reaction problems [13] and later analyzed in [9,16,15]. Several HDG methods are subsequently developed for Biharmonic equations [10], linear and nonlinear convection–diffusion problems [34,35,11], linear elasticity [43], Stokes flows [12,36,17] (see also [14] for the error analysis), incompressible Navier–Stokes equations [37,32], and compressible Euler and Navier–Stokes equations [38]. An overview of recent developments in HDG methodology for fluid dynamics is given in [33].

The article is organized as follows. In Sections 2 and 3, we devote our discussion to acoustic and elastic wave equations, respectively. In each of the two sections, we introduce the HDG method, prove stability and uniqueness of the numerical solution, describe the implementation and local postprocessing, extend the method to treat unbounded domains, and present numerical results to assess the convergence and accuracy of the method. Finally, in Section 4, we provide some concluding remarks on future work.

2. The acoustic wave equation

2.1. Problem statement

Let $\Omega \in \mathbb{R}^d$ be a bounded domain with Lipschitz continuous boundary $\partial\Omega$ and let $T > 0$ be a final time. We consider the following acoustic wave equation:

$$\rho\frac{\partial^2 u}{\partial t^2} - \nabla\cdot(\mathbf{A}\nabla u) = f, \quad \text{in } \Omega\times(0,T]. \tag{1}$$

Here u is the scalar variable and f is a given source term. We assume that the scalar coefficient ρ satisfies $\rho(\mathbf{x}) > 0$ for all $\boldsymbol{x}\in\Omega$ and that the matrix-valued coefficient $\mathbf{A}\in\mathbb{R}^{d\times d}$ is symmetric positive-definite, namely, $\boldsymbol{p}^T\mathbf{A}\boldsymbol{p} > 0$ for all $\boldsymbol{p}\in\mathbb{R}^d$.

We next introduce the velocity $v = u_t$ and the gradient $\boldsymbol{q} = \nabla u$. We then write (1) into a system of first-order equations as:

$$\begin{aligned} &\frac{\partial \boldsymbol{q}}{\partial t} - \nabla v = 0, \quad \text{in } \Omega\times(0,T],\\ &\rho\frac{\partial v}{\partial t} - \nabla\cdot\mathbf{A}\boldsymbol{q} = f, \quad \text{in } \Omega\times(0,T]. \end{aligned} \tag{2}$$

We supplement this system with boundary conditions:

$$\begin{aligned} &v = g_D, \quad \text{on } \partial\Omega_D,\\ &\mathbf{A}\boldsymbol{q}\cdot\boldsymbol{n} = g_N, \quad \text{on } \partial\Omega_N, \end{aligned} \tag{3}$$

and initial conditions:

$$\begin{aligned} &v(\boldsymbol{x}, t=0) = v_0(\boldsymbol{x}),\\ &\boldsymbol{q}(\boldsymbol{x}, t=0) = \boldsymbol{q}_0(\boldsymbol{x}). \end{aligned} \tag{4}$$

Here $\partial\Omega_D$ and $\partial\Omega_N$ are two disjoint parts of the boundary $\partial\Omega$ such that $\partial\Omega = \overline{\partial\Omega_D\cup\partial\Omega_N}$.

2.2. Approximation spaces

Let $\mathcal{T}_h$ be a collection of disjoint elements that partition Ω. We denote by $\partial\mathcal{T}_h$ the set $\{\partial K : K\in\mathcal{T}_h\}$. For an element K of the collection $\mathcal{T}_h$, $F = \partial K\cap\partial\Omega$ is the boundary face if the $d-1$ Lebesgue measure of F is nonzero. For two elements K^+ and K^- of the collection $\mathcal{T}_h$, $F = \partial K^+\cap\partial K^-$ is the interior face between K^+ and K^- if the $d-1$ Lebesgue measure of F is nonzero. Let $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$ denote the set of interior and boundary faces, respectively. We denote by $\mathcal{E}_h$ the union of $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$.

Let $\mathcal{P}_k(D)$ denote the set of polynomials of degree at most k on a domain D. We are going to use the following discontinuous finite element spaces:

$$\begin{aligned} W_h &= \left\{w\in L^2(\Omega) : w|_K\in W(K),\ \forall K\in\mathcal{T}_h\right\},\\ \boldsymbol{V}_h &= \left\{\boldsymbol{p}\in(L^2(\Omega))^d : \boldsymbol{p}|_K\in\boldsymbol{V}(K),\ \forall K\in\mathcal{T}_h\right\}. \end{aligned}$$

Some appropriate choices for the local space $W(K)\times\boldsymbol{V}(K)$ on K include:

$$W(K)\times\boldsymbol{V}(K) \equiv \begin{cases} \mathcal{P}_k(K)\times(\mathcal{P}_k(K))^d,\\ \mathcal{P}_{k-1}(K)\times(\mathcal{P}_k(K))^d,\\ \mathcal{P}_k(K)\times\left((\mathcal{P}_k(K))^d + \boldsymbol{x}\,\mathcal{P}_k(K)\right). \end{cases}$$

These spaces correspond to the equal-order elements, the BDM elements [6], and the RT elements [41], respectively. In addition, we introduce a traced finite element space:

$$M_h = \left\{\mu\in L^2(\mathcal{E}_h) : \mu|_F\in\mathcal{P}_k(F),\ \forall F\in\mathcal{E}_h\right\}.$$

We also set $M_h(g_D) = \{\mu\in M_h : \mu = \mathrm{P}g_D \text{ on } \partial\Omega_D\}$, where P denotes the L^2-projection into the space $\{\mu|_{\partial\Omega}, \forall\mu\in M_h\}$.

For functions $\boldsymbol{w}$ and $\boldsymbol{v}$ in $(L^2(D))^d$, we denote $(\boldsymbol{w},\boldsymbol{v})_D = \int_D \boldsymbol{w}\cdot\boldsymbol{v}$. For functions w and v in $L^2(D)$, we denote $(w,v)_D = \int_D wv$ if D is a domain in $\mathbb{R}^d$ and $\langle w,v\rangle_D = \int_D wv$ if D is a domain in $\mathbb{R}^{d-1}$. We finally introduce:

$$(w,v)_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h}(w,v)_K, \quad \langle\mu,\eta\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h}\langle\mu,\eta\rangle_{\partial K},$$

for w, v defined on $\mathcal{T}_h$ and μ, η defined on $\partial\mathcal{T}_h$.

2.3. HDG method with backward difference formulas

We begin by considering the governing Eq. (2) on one element K of $\mathcal{T}_h$. Multiplying (2) by a test function $(\boldsymbol{r},w)\in\boldsymbol{V}(K)\times W(K)$ and integrating by parts we find an approximation $(\boldsymbol{q}_h, v_h)\in\boldsymbol{V}_h\times W_h$ such that for all $K\in\mathcal{T}_h$:

$$\begin{aligned} &\left(\frac{\partial\boldsymbol{q}_h}{\partial t},\boldsymbol{r}\right)_K + (v_h,\nabla\cdot\boldsymbol{r})_K - \langle\hat{v}_h,\boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial K} = 0, \quad \forall\boldsymbol{r}\in\boldsymbol{V}(K),\\ &\left(\rho\frac{\partial v_h}{\partial t},w\right)_K + (\mathbf{A}\boldsymbol{q}_h,\nabla w)_K - \langle\mathbf{A}\hat{\boldsymbol{q}}_h\cdot\boldsymbol{n},w\rangle_{\partial K} = (f,w)_K, \quad \forall w\in W(K). \end{aligned} \tag{5}$$

Here the numerical traces $\hat{\boldsymbol{q}}_h$ and $\hat{v}_h$ are approximations to $\boldsymbol{q}$ and v over ∂K, respectively. This system is then discretized in time using backward difference formulaes (BDF) for the discretization of the time derivative. For instance, using the Backward–Euler scheme at time level t^n with timestep Δt^n we obtain:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\boldsymbol{q}_h^n, \boldsymbol{r})_K + (v_h^n, \nabla\cdot\boldsymbol{r})_K - \langle \hat{v}_h^n, \boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial K} = \frac{1}{\Delta t^n}(\boldsymbol{q}_h^{n-1}, \boldsymbol{r})_K,\\
&\frac{1}{\Delta t^n}(\rho v_h^n, w)_K + (\mathbf{A}\boldsymbol{q}_h^n, \nabla w)_K - \langle \mathbf{A}\hat{\boldsymbol{q}}_h^n\cdot\boldsymbol{n}, w\rangle_{\partial K} = (f^n, w)_K + \frac{1}{\Delta t^n}(\rho v_h^{n-1}, w)_K,
\end{aligned} \tag{6}$$

for all $(\boldsymbol{r}, w)\times\boldsymbol{V}(K)\times W(K)$. Here we denote $v_h^n = v_h(t^n)$ and $\boldsymbol{q}_h^n = \boldsymbol{q}_h(t^n)$.

By summing the above system over all the elements and enforcing the continuity of the normal component of the numerical flux $\hat{\boldsymbol{q}}_h$, we arrive at the following problem: Find $(\boldsymbol{q}_h^n, v_h^n, \hat{v}_h^n)\in\boldsymbol{V}_h\times W_h\times M_h(g_D^n)$ such that:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\boldsymbol{q}_h^n, \boldsymbol{r})_{\mathcal{T}_h} + (v_h^n, \nabla\cdot\boldsymbol{r})_{\mathcal{T}_h} - \langle \hat{v}_h^n, \boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = \frac{1}{\Delta t^n}(\boldsymbol{q}_h^{n-1}, \boldsymbol{r})_{\mathcal{T}_h},\\
&\frac{1}{\Delta t^n}(\rho v_h^n, w)_{\mathcal{T}_h} + (\mathbf{A}\boldsymbol{q}_h^n, \nabla w)_{\mathcal{T}_h} - \langle \mathbf{A}\hat{\boldsymbol{q}}_h^n\cdot\boldsymbol{n}, w\rangle_{\partial\mathcal{T}_h} = (f^n, w)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho v_h^{n-1}, w)_{\mathcal{T}_h},\\
&\langle \mathbf{A}\hat{\boldsymbol{q}}_h^n\cdot\boldsymbol{n}, \mu\rangle_{\partial\mathcal{T}_h} = \langle g_N^n, \mu\rangle_{\partial\Omega_N},
\end{aligned} \tag{7}$$

for all $(\boldsymbol{r}, w, \mu)\in\boldsymbol{V}_h\times W_h\times M_h(0)$, where $(v_h^0, w)_{\mathcal{T}_h} = (v_0, w)_{\mathcal{T}_h}, \forall w\in W_h$, and $(\boldsymbol{q}_h^0, \boldsymbol{r})_{\mathcal{T}_h} = (\boldsymbol{q}_0, \boldsymbol{r})_{\mathcal{T}_h}, \forall\boldsymbol{r}\in\boldsymbol{V}_h$. Note that the Dirichlet boundary condition is enforced by requiring that $\hat{v}_h^n\in M_h(g_D^n)$ and that the Neumann boundary condition is enforced by the last equation in (7). We complete the HDG method (7) by defining $\hat{\boldsymbol{q}}_h^n$ in terms of the other unknowns as:

$$\mathbf{A}\hat{\boldsymbol{q}}_h^n = \mathbf{A}\boldsymbol{q}_h^n - \tau(v_h^n - \hat{v}_h^n)\boldsymbol{n}, \quad \text{on } \partial\mathcal{T}_h. \tag{8}$$

Here τ is the *local stabilization parameter* which has an important effect on both the stability and accuracy of the numerical scheme. The selection of the value of the parameter τ will be described below.

Once v_h^n is available we compute u_h^n by simultaneously seeking $u_h^n\in W_h$ such that:

$$\frac{1}{\Delta t^n}(u_h^n, w)_{\mathcal{T}_h} = (v_h^n, w)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(u_h^{n-1}, w)_{\mathcal{T}_h}, \quad \forall w\in W_h. \tag{9}$$

Here the initial solution u_h^0 is given by $(u_h^0, w)_{\mathcal{T}_h} = (u_0, w)_{\mathcal{T}_h}, \forall w\in W_h$, where u_0 is the initial data. Note that the above system can be inverted at the element level thanks to the discontinuous nature of the space W_h.

The application of higher-order backward difference formulas (BDFs) leads to systems similar to (7), so that our discussion is relevant to higher-order time differencing as well. The HDG method can also work with other implicit time-stepping methods such as the diagonally implicit Runge–Kutta (DIRK) methods as discussed below.

2.4. HDG method with DIRK schemes

We consider the following DIRK(q,p) formulas [1] written in the form of Butcher's table for time integration:

$$\begin{array}{cccccc}
a_{11} & 0 & & \dots & 0 & c_1\\
a_{21} & a_{22} & 0 & \dots & 0 & c_2\\
\dots & & & \dots & & \dots\\
a_{q1} & a_{q2} & & \dots & a_{qq} & c_q\\
\hline
b_1 & b_2 & & \dots & b_q &
\end{array} \tag{10}$$

where q denotes the number of stages and p denotes the order of the method. For instance, the DIRK(2,3) and DIRK(3,4) schemes proposed by Crouzeix [22] are known to be A-stable. In addition, we shall also use the A-stable DIRK(5,5) scheme proposed by Cooper and Sayfy [21].

We are now ready to describe the HDG method for spatial discretization and the DIRK(q,p) method for time integration. This is done by applying the DIRK(q,p) method to integrate the ODE system (5) in time and enforcing the continuity of the normal component of the numerical flux. To simplify notation we write $t^{n,\,i}$ for $t^n + c_i\Delta t^n$, $\boldsymbol{y}_h^n$ for $(\boldsymbol{q}_h^n, v_h^n)$, and $\boldsymbol{y}_h^{n,i}$ for $(\boldsymbol{q}_h^{n,i}, v_h^{n,i}) = (\boldsymbol{q}_h(t^{n,i}), v_h(t^{n,i}))$. The numerical solution $\boldsymbol{y}_h^{n+1} = (\boldsymbol{q}_h^{n+1}, v_h^{n+1})$ at time level $n+1$ given by the DIRK(q,p) method is computed as follows:

$$\boldsymbol{y}_h^{n+1} = \boldsymbol{y}_h^n + \Delta t^n\sum_{i=1}^{q} b_i\boldsymbol{f}_{h,i}, \tag{11}$$

where

$$\begin{aligned}
\boldsymbol{f}_{h,1} &= \frac{\boldsymbol{y}_h^{n,1} - \boldsymbol{y}_h^n}{a_{11}\Delta t^n},\\
\boldsymbol{f}_{h,2} &= \frac{\boldsymbol{y}_h^{n,2} - \boldsymbol{y}_h^n}{a_{22}\Delta t^n} - \frac{a_{21}}{a_{22}}\boldsymbol{f}_{h,1},\\
&\cdots\\
\boldsymbol{f}_{h,q} &= \frac{\boldsymbol{y}_h^{n,q} - \boldsymbol{y}_h^n}{a_{qq}\Delta t^n} - \sum_{j=1}^{q-1}\frac{a_{qj}}{a_{qq}}\boldsymbol{f}_{h,j}.
\end{aligned} \tag{12}$$

The intermediate states $\boldsymbol{y}_h^{n,i} = (\boldsymbol{q}_h^{n,i}, v_h^{n,i})$, $1 \leqslant i \leqslant q$, are determined as follows: $(\boldsymbol{q}_h^{n,i}, v_h^{n,i}, \hat{v}_h^{n,i}) \in \boldsymbol{V}_h \times W_h \times M_h(g(t^{n,i})_D)$ satisfies:

$$\begin{aligned}
&\frac{1}{a_{ii}\Delta t^n}\left(\boldsymbol{q}_h^{n,i}, \boldsymbol{r}\right)_{\mathcal{T}_h} + \left(v_h^{n,i}, \nabla\cdot\boldsymbol{r}\right)_{\mathcal{T}_h} - \left\langle \hat{v}_h^{n,i}, \boldsymbol{r}\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h} = \left(\boldsymbol{p}_h^{n,i}, \boldsymbol{r}\right)_{\mathcal{T}_h},\\
&\frac{1}{a_{ii}\Delta t^n}\left(\rho v_h^{n,i}, w\right)_{\mathcal{T}_h} + \left(\mathbf{A}\boldsymbol{q}_h^{n,i}, \nabla w\right)_{\mathcal{T}_h} - \left\langle \mathbf{A}\hat{\boldsymbol{q}}_h^{n,i}\cdot\boldsymbol{n}, w\right\rangle_{\partial\mathcal{T}_h} = \left(f(t^{n,i}), w\right)_{\mathcal{T}_h} + \left(\rho z_h^{n,i}, w\right)_{\mathcal{T}_h},\\
&\left\langle \mathbf{A}\hat{\boldsymbol{q}}_h^{n,i}\cdot\boldsymbol{n}, \mu\right\rangle_{\partial\mathcal{T}_h} = \left\langle g_N(t^{n,i}), \mu\right\rangle_{\partial\Omega_N},
\end{aligned} \tag{13}$$

where

$$\hat{\boldsymbol{q}}_h^{n,i} = \boldsymbol{q}_h^{n,i} - \tau\left(v_h^{n,i} - \hat{v}_h^{n,i}\right)\boldsymbol{n}, \quad \text{on } \partial\mathcal{T}_h, \tag{14}$$

and the terms $\boldsymbol{s}_h^{n,i} = (\boldsymbol{p}_h^{n,i}, z_h^{n,i})$, $1 \leqslant i \leqslant q$, on the right-hand side of (13) are given by

$$\begin{aligned}
\boldsymbol{s}_h^{n,1} &= \frac{\boldsymbol{y}_h^n}{a_{11}\Delta t^n},\\
\boldsymbol{s}_h^{n,2} &= \frac{\boldsymbol{y}_h^n}{a_{22}\Delta t^n} + \frac{a_{21}}{a_{22}}\left(\frac{\boldsymbol{y}_h^{n,1}}{a_{11}\Delta t^n} - \boldsymbol{s}_h^{n,1}\right),\\
&\cdots\\
\boldsymbol{s}_h^{n,q} &= \frac{\boldsymbol{y}_h^n}{a_{qq}\Delta t^n} + \sum_{j=1}^{q-1}\frac{a_{qj}}{a_{qq}}\left(\frac{\boldsymbol{y}_h^{n,j}}{a_{jj}\Delta t^n} - \boldsymbol{s}_h^{n,j}\right).
\end{aligned} \tag{15}$$

We note that the resulting system (13) at each ith stage of the DIRK(q,p) method is very similar to the system (7) of the Backward–Euler method. Therefore, most of our discussion for the Backward–Euler method would be relevant to the DIRK(q,p) method.

2.5. Stability and uniqueness for the backward-euler method

We now show that the numerical solution obtained by the HDG scheme (7) is stable and unique.

Proposition 1. *Assume that* $\mathbf{A}$ *is symmetric positive-definite and piecewise-constant. Any solution of the equations defining the HDG method, (7), satisfies the following energy identity:*

$$\frac{1}{2}\left(\mathbf{A}\boldsymbol{q}_h^m, \boldsymbol{q}_h^m\right)_{\mathcal{T}_h} + \frac{1}{2}\left(\rho v_h^m, v_h^m\right)_{\mathcal{T}_h} + \Theta_h^m = \frac{1}{2}\left(\mathbf{A}\boldsymbol{q}_h^0, \boldsymbol{q}_h^0\right)_{\mathcal{T}_h} + \frac{1}{2}\left(\rho v_h^0, v_h^0\right)_{\mathcal{T}_h} + \Phi_h^m, \tag{16}$$

for all $m \geqslant 1$, *where,*

$$\begin{aligned}
\Theta_h^m &:= \sum_{n=1}^m \Delta t^n \left\langle \tau\left(v_h^n - \hat{v}_h^n\right), \left(v_h^n - \hat{v}_h^n\right)\right\rangle_{\partial\mathcal{T}_h} + \frac{1}{2}\sum_{n=1}^m \left(\mathbf{A}\left(\boldsymbol{q}_h^n - \boldsymbol{q}_h^{n-1}\right), \left(\boldsymbol{q}_h^n - \boldsymbol{q}_h^{n-1}\right)\right)_{\mathcal{T}_h} + \frac{1}{2}\sum_{n=1}^m \left(\rho\left(v_h^n - v_h^{n-1}\right), \left(v_h^n - v_h^{n-1}\right)\right)_{\mathcal{T}_h},\\
\Psi_h^m &:= \sum_{n=1}^m \Delta t^n \left(\left(f^n, v_h^n\right)_{\mathcal{T}_h} + \left\langle g_N^n, \hat{v}_h^n\right\rangle_{\partial\Omega_N}\right).
\end{aligned}$$

Moreover, if we assume the following positivity condition:

$$\tau > 0, \quad \text{on } \partial\mathcal{T}_h, \tag{17}$$

then the HDG method (7) has a unique solution $(\boldsymbol{q}_h^n, v_h^n, \hat{v}_h^n)$ *for any timestep* $n \geqslant 1$.

Proof. To prove the energy identity, we begin by substituting the definition of the numerical flux $\hat{\boldsymbol{q}}_h$ (8) into (7) to obtain:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\boldsymbol{q}_h^n, \boldsymbol{v})_{\mathcal{T}_h} + (v_h^n, \nabla\cdot\boldsymbol{v})_{\mathcal{T}_h} - \langle \hat{v}_h^n, \boldsymbol{v}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = \frac{1}{\Delta t^n}(\boldsymbol{q}_h^{n-1}, \boldsymbol{v})_{\mathcal{T}_h},\\
&\frac{1}{\Delta t^n}(\rho v_h^n, w)_{\mathcal{T}_h} - (\nabla\cdot\mathbf{A}\boldsymbol{q}_h^n, w)_{\mathcal{T}_h} + \langle \tau(v_h^n - \hat{v}_h^n), w\rangle_{\partial\mathcal{T}_h} = (f^n, w)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho v_h^{n-1}, w)_{\mathcal{T}_h},\\
&\langle \mathbf{A}\boldsymbol{q}_h^n\cdot\boldsymbol{n} - \tau(v_h^n - \hat{v}_h^n), \mu\rangle_{\partial\mathcal{T}_h} = \langle g_N^n, \mu\rangle_{\partial\Omega_N}.
\end{aligned} \tag{18}$$

Next, we take $\boldsymbol{v} = \mathbf{A}\boldsymbol{q}_h^n, w = v_h^n, \mu = \hat{v}_h^n$ in (18) and add the resulting equations up to obtain:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\mathbf{A}\boldsymbol{q}_h^n, \boldsymbol{q}_h^n)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho v_h^n, v_h^n)_{\mathcal{T}_h} + \langle \tau(v_h^n - \hat{v}_h^n), (v_h^n - \hat{v}_h^n)\rangle_{\partial\mathcal{T}_h}\\
&\quad = \frac{1}{\Delta t^n}(\mathbf{A}\boldsymbol{q}_h^{n-1}, \boldsymbol{q}_h^n)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho v_h^{n-1}, v_h^n)_{\mathcal{T}_h} + (f^n, v_h^n)_{\mathcal{T}_h} + \langle g_N^n, \hat{v}_h^n\rangle_{\partial\Omega_N}.
\end{aligned}$$

Since $ab = a^2/2 + b^2/2 - (a-b)^2/2$ and since A is symmetric, we readily obtain:

$$\begin{aligned}
&\frac{1}{2\Delta t^n}(\mathbf{A}\boldsymbol{q}_h^n, \boldsymbol{q}_h^n)_{\mathcal{T}_h} + \frac{1}{2\Delta t^n}(\rho v_h^n, v_h^n)_{\mathcal{T}_h} + \langle \tau(v_h^n - \hat{v}_h^n), (v_h^n - \hat{v}_h^n)\rangle_{\partial\mathcal{T}_h}\\
&\quad = \frac{1}{2\Delta t^n}(\mathbf{A}\boldsymbol{q}_h^{n-1}, \boldsymbol{q}_h^{n-1})_{\mathcal{T}_h} - \frac{1}{2\Delta t^n}(\mathbf{A}(\boldsymbol{q}_h^n - \boldsymbol{q}_h^{n-1}), (\boldsymbol{q}_h^n - \boldsymbol{q}_h^{n-1}))_{\mathcal{T}_h}\\
&\qquad + \frac{1}{2\Delta t^n}(\rho v_h^{n-1}, v_h^{n-1})_{\mathcal{T}_h} - \frac{1}{2\Delta t^n}(\rho(v_h^n - v_h^{n-1}), (v_h^n - v_h^{n-1}))_{\mathcal{T}_h} + (f^n, v_h^n)_{\mathcal{T}_h} + \langle g_N^n, \hat{v}_h^n\rangle_{\partial\Omega_N}.
\end{aligned}$$

To get the energy identity, we only have to multiply the above equation by Δt^n and add on n.

Let us now prove the existence and uniqueness of the approximate solution. Since the formulation (18) defines a square and linear system, the existence and uniqueness follow if we show that $\boldsymbol{q}_h^n = \boldsymbol{0}$, $v_h^n = 0$, and $\hat{v}_h = 0$ when the right-hand side is set to zero. In this case, the energy identity gives:

$$\frac{1}{\Delta t^n}(\mathbf{A}\boldsymbol{q}_h^n, \boldsymbol{q}_h^n)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho v_h^n, v_h^n)_{\mathcal{T}_h} + \langle \tau(v_h^n - \hat{v}_h^n), (v_h^n - \hat{v}_h^n)\rangle_{\partial\mathcal{T}_h} = 0. \tag{19}$$

This implies that $\boldsymbol{q}_h^n = \boldsymbol{0}$, $v_h^n = 0$, and $\hat{v}_h^n = v_h^n = 0$ since $\tau > 0, \rho > 0$, and $\mathbf{A}$ is symmetric positive-definite. This completes the proof. □

The energy identity (16) gives us a way to choose τ. In particular, on the basis of dimensional analysis τ should be chosen as:

$$\tau = \frac{\rho}{t_c} = \rho\omega_c, \tag{20}$$

where t_c is the characteristic timescale and $\omega_c = 1/t_c$ is the characteristic frequency.

2.6. Implementation

There are two different approaches for implementing the HDG method. We briefly describe the first approach and leave the second approach to Sub Section 3.3. For simplicity of exposition, we consider a fixed timestep $\Delta t^n = \Delta t$. We first note that the discretization of the system of Eq. (18) gives rise to a matrix equation of the form:

$$\begin{bmatrix} \mathbb{A} & -\mathbb{B}^T & -\mathbb{C}^T \\ \mathbb{B} & \mathbb{D} & \mathbb{E}^T \\ \mathbb{C} & \mathbb{E} & \mathbb{G} \end{bmatrix} \begin{bmatrix} Q^n \\ V^n \\ \Lambda^n \end{bmatrix} = \begin{bmatrix} \mathbb{A}Q^{n-1} \\ F^n + \mathbb{M}V^{n-1} \\ L^n \end{bmatrix}. \tag{21}$$

Here Q^n, V^n, and Λ^n represent the vector of degrees of freedom for $\boldsymbol{q}_h^n$, v_h^n, and $\hat{v}_h^n$, respectively.

Due to the discontinuous nature of $\boldsymbol{V}_h \times W_h$ the matrix $[\mathbb{A} \; -\mathbb{B}^T; \mathbb{B} \; \mathbb{D}]$ is block-diagonal and invertible, so its inverse exists and is block-diagonal. Therefore, we can eliminate both Q^n and V^n to obtain a reduced globally coupled matrix equation only for Λ^n as:

$$\mathbb{K}\Lambda^n = R^n, \tag{22a}$$

where

$$\mathbb{K} = -\begin{bmatrix} \mathbb{C} & \mathbb{E} \end{bmatrix} \begin{bmatrix} \mathbb{A} & -\mathbb{B}^T \\ \mathbb{B} & \mathbb{D} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbb{C}^T \\ \mathbb{E} \end{bmatrix} + \mathbb{G}, \tag{22b}$$

and

$$R^n = L^n - \begin{bmatrix} \mathbb{C} & \mathbb{E} \end{bmatrix} \begin{bmatrix} \mathbb{A} & -\mathbb{B}^T \\ \mathbb{B} & \mathbb{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{A}Q^{n-1} \\ F^n + \mathbb{M}V^{n-1} \end{bmatrix}. \tag{22c}$$

We can thus pre-compute the stiffness matrix $\mathbb{K}$ and the following matrix product:

$$\mathbb{H} = \begin{bmatrix} \mathbb{C} & \mathbb{E} \end{bmatrix} \begin{bmatrix} \mathbb{A} & -\mathbb{B}^T \\ \mathbb{B} & \mathbb{D} \end{bmatrix}^{-1}.$$

Then the vector R^n can be inexpensively computed at every time step. In Section 3.3, we describe the second approach to form $\mathbb{K}$ and R^n without explicitly constructing $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{E}$, and $\mathbb{G}$; see also [34,35] for additional details.

2.7. Local postprocessing

We propose here a local postprocessing procedure to improve the accuracy of the numerical approximations. In particular, the postprocessed displacement and velocity will converge with order $k+2$, whenever the approximate gradient converges with the optimal order $k+1$ and the average of the original approximation superconverges with order $k+2$ [9,44]. Note that the local postprocessing is effective only when the temporal accuracy is of order $k+2$. Furthermore, our postprocessing scheme may not be effective for other DG methods such as the BR method [3], local DG method [18], IPDG method [25], and CDG method [39] because the approximate gradient of those DG methods converges with the suboptimal order k. It appears that the above-mentioned convergence and postprocessing properties are unique for the HDG method.

We first consider the local postprocessing of the displacement. On every simplex $K \in \mathcal{T}_h$, we define a new approximate displacement $u_h^{n*} \in \mathcal{P}_{k+1}(K)$ to satisfy:

$$\begin{aligned} &(\nabla u_h^{n*}, \nabla w)_K = (\boldsymbol{q}_h^n, \nabla w)_K, \quad \forall w \in \mathcal{P}_{k+1}(K), \\ &(u_h^{n*}, 1)_K = (u_h^n, 1)_K. \end{aligned} \tag{23}$$

The postprocessing (23) requires us to solve a linear system whose size is the dimension of $\mathcal{P}_{k+1}(K)$.

To postprocess the numerical solution for the new velocity v_h^{n*}, we first compute an approximation $\boldsymbol{p}_h^n \in \boldsymbol{V}(K)$ to the velocity gradient $\boldsymbol{p}(t^n) = \nabla v(t^n)$ by locally solving the below system:

$$(\nabla \boldsymbol{p}_h^n, \boldsymbol{v})_K = -(v_h^n, \nabla \cdot \boldsymbol{v})_K + \langle \hat{v}_h^n, \boldsymbol{v} \cdot \boldsymbol{n} \rangle_{\partial K}, \quad \forall \boldsymbol{v} \in \boldsymbol{V}(K). \tag{24}$$

We then find $v_h^{n*} \in \mathcal{P}_{k+1}(K)$ such that:

$$\begin{aligned} &(\nabla v_h^{n*}, \nabla w)_K = (\boldsymbol{p}_h^n, \nabla w)_K, \quad \forall w \in \mathcal{P}_{k+1}(K), \\ &(v_h^{n*}, 1)_K = (v_h^n, 1)_K. \end{aligned} \tag{25}$$

This postprocessing step is similar to (23).

The postprocessing scheme described here is an extension of the previous work [9,16,34,35]. This local postprocessing method is efficient since it can compute both u_h^{n*} and v_h^{n*} at any time step without advancing in time. As a result, the new approximations are significantly less expensive to compute than the original approximations.

Table 1
Example 1: History of convergence of the numerical approximations for a fixed ratio $h/\Delta t = 4$.

Degree	Mesh	$\|u-u_h\|_{\mathcal{T}_h}$		$\|v-v_h\|_{\mathcal{T}_h}$		$\|\mathsf{q}-\mathsf{q}_h\|_{\mathcal{T}_h}$		$\|u-u_h^*\|_{\mathcal{T}_h}$		$\|v-v_h^*\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	2	3.75e−2	–	6.15e−2	–	1.37e−1	–	2.93e−2	–	7.06e−2	–
	4	7.10e−3	2.40	1.47e−2	2.06	2.68e−2	2.35	5.25e−3	2.48	1.46e−2	2.27
	8	1.20e−3	2.56	2.99e−3	2.30	4.56e−3	2.55	7.35e−4	2.84	2.11e−3	2.80
	16	2.31e−4	2.38	6.67e−4	2.16	8.66e−4	2.40	9.54e−5	2.95	2.77e−4	2.93
	32	5.12e−5	2.17	1.61e−4	2.05	1.90e−4	2.19	1.21e−5	2.98	3.55e−5	2.96
2	2	7.29e−3	–	1.72e−2	–	3.01e−2	–	6.16e−3	–	1.71e−2	–
	4	4.80e−4	3.92	2.16e−3	2.99	2.00e−3	3.91	2.77e−4	4.48	1.99e−3	3.11
	8	4.47e−5	3.42	1.86e−4	3.54	1.84e−4	3.44	7.02e−6	5.30	1.40e−4	3.83
	16	5.24e−6	3.09	1.81e−5	3.36	2.15e−5	3.10	2.54e−7	4.79	8.73e−6	4.00
	32	6.36e−7	3.04	2.08e−6	3.12	2.61e−6	3.04	1.44e−8	4.14	5.36e−7	4.03
3	2	5.80e−4	–	1.60e−3	–	2.67e−3	–	1.97e−4	–	1.59e−3	–
	4	3.12e−5	4.22	8.22e−5	4.29	1.38e−4	4.27	4.92e−6	5.33	8.05e−5	4.30
	8	1.78e−6	4.13	5.20e−6	3.98	7.74e−6	4.16	1.37e−7	5.17	3.78e−6	4.41
	16	1.06e−7	4.07	3.32e−7	3.97	4.56e−7	4.08	4.05e−9	5.08	1.14e−7	5.05
	32	6.46e−9	4.04	2.09e−8	3.99	2.77e−8	4.04	1.24e−10	5.03	1.50e−9	6.24

2.8. First-order absorbing boundary condition

We consider solving the wave Eq. (2) in an *truncated* domain Ω by using the first-order absorbing boundary condition of the form:

$$v + \mathbf{A}\boldsymbol{q}\cdot\boldsymbol{n} = 0, \quad \text{on } \Gamma_{\text{ext}}, \tag{26}$$

where Γ_{ext} is the *exterior* boundary of the truncated domain Ω and $\partial\Omega_D \cup \partial\,\Omega_N$ denotes the *interior* boundary of Ω. The above absorbing boundary condition has been extensively studied [26,29].

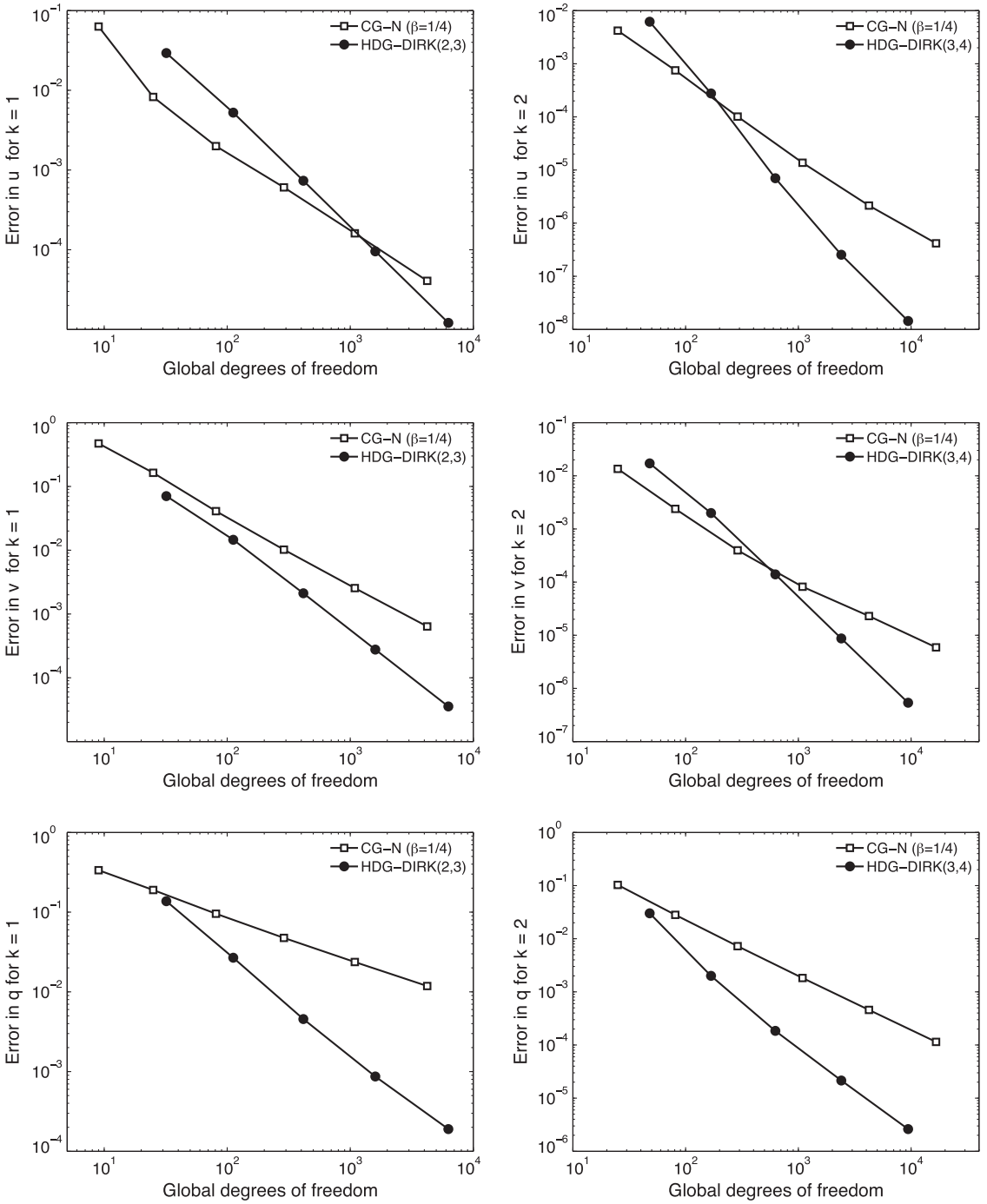

Fig. 1. Example 1: Comparison of the convergence of the L^2-errors in u (top), v (middle), and $\boldsymbol{q}$ (bottom) for the HDG–DIRK(q,p) method and the CG-N$(\beta = 1/4)$ method. The postprocessed solution was taken as the approximation for the HDG–DIRK(q,p) method.

The HDG method now seeks an approximation $(\boldsymbol{q}_h^n, v_h^n, \hat{v}_h^n) \in \boldsymbol{V}_h \times W_h \times M_h(g_D^n)$ such that:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\boldsymbol{q}_h^n, \boldsymbol{r})_{\mathcal{T}_h} + (v_h^n, \nabla\cdot\boldsymbol{r})_{\mathcal{T}_h} - \langle \hat{v}_h^n, \boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = \frac{1}{\Delta t^n}(\boldsymbol{q}_h^{n-1}, \boldsymbol{r})_{\mathcal{T}_h},\\
&\frac{1}{\Delta t^n}(\rho v_h^n, w)_{\mathcal{T}_h} + (\mathbf{A}\boldsymbol{q}_h^n, \nabla w)_{\mathcal{T}_h} - \langle \mathbf{A}\hat{\boldsymbol{q}}_h^n\cdot\boldsymbol{n}, w\rangle_{\partial\mathcal{T}_h} = (f^n, w)_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho v_h^{n-1}, w)_{\mathcal{T}_h},\\
&\langle \mathbf{A}\hat{\boldsymbol{q}}_h^n\cdot\boldsymbol{n}, \mu\rangle_{\partial\mathcal{T}_h} + \langle \hat{v}_h^n, \mu\rangle_{\Gamma_{\text{ext}}} = \langle g_N^n, \mu\rangle_{\partial\Omega_N},
\end{aligned} \tag{27}$$

for all $(\boldsymbol{r}, w, \mu) \in \boldsymbol{V}_h \times W_h \times M_h(0)$, where:

$$\mathbf{A}\hat{\boldsymbol{q}}_h^n = \mathbf{A}\boldsymbol{q}_h^n - \tau(v_h^n - \hat{v}_h^n)\boldsymbol{n}, \quad \text{on } \partial\mathcal{T}_h. \tag{28}$$

We note that the boundary condition (26) is imposed implicitly by the term $\langle \hat{v}_h^n, \mu\rangle_{\Gamma_{\text{ext}}}$ in the last equation of (27). In fact, this term is the only difference between (7) and (27). Of course, our earlier discussion such as the implementation, uniqueness and stability, and local postprocessing for (7) is also relevant to the system (27).

2.9. Numerical results

We now present several examples to demonstrate the performance of the HDG method. We shall use equal-order elements and high-order nodal basis [30] to represent the numerical approximations in all the examples.

2.9.1. Vibration of a square membrane

We consider the wave Eq. (1) on a unit square $\Omega = (0,1)\times(0,1)$ with boundary condition $v = 0$ on $\partial\Omega$ and initial condition $u(x,y,t=0) = 0$ and $v(x,y,t=0) = \sin(\pi x)\sin(\pi y)$. We set $\rho := 1$ and $\mathbf{A} := \mathbf{I}$. For $f = 0$ the problem has the following exact solution:

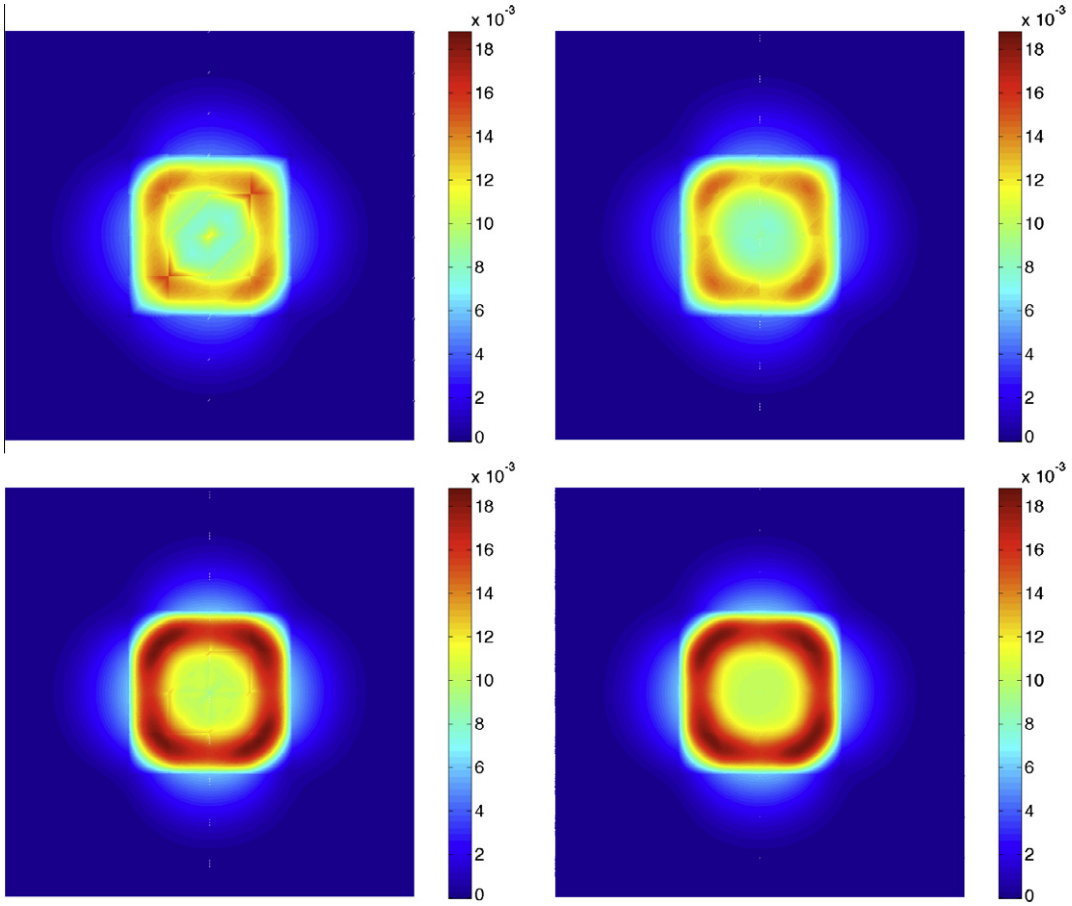

Fig. 2. Example 2: Plots of u_h (left) and u_h^* (right) at $t = 0.20$ for $k = 2$ (top) and $k = 3$ (bottom).

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

This solution represents the vibration of the square membrane under an initial velocity. The final time is $T = 1$.

We consider triangular meshes that are obtained by splitting a regular $n \times n$ Cartesian grid into a total of $2n^2$ triangles, giving uniform element sizes of $h = 1/n$. On these meshes, we consider polynomials of degree k to represent all the approximate variables using a high-order nodal basis [30] within each element. For temporal discretization we use the DIRK(2,3), DIRK(3,4), and DIRK(5,5) schemes for $k = 1$, 2, and 3, respectively. The stabilization parameter is set to $\tau = 1$.

We present in Table 1 the L^2 errors and associated orders of convergence for the numerical approximations at the final time $t = 1$. These results are obtained for a fixed ratio $h/\Delta t = 4$. We observe that the approximate displacement, velocity, and gradient converge with the optimal order $k + 1$ for $k = 1, 2, 3$. Moreover, both the postprocessed displacement and velocity converge with order $k + 2$ which are the same order as the DIRK schemes used for time integration. The fact that the HDG method yields optimal convergence for the approximate gradient has an important advantage since many other DG methods provide suboptimal convergence of order k for the approximate gradient. Equally important is the fact that both the postprocessed displacement and velocity converge with the order $k + 2$, which is one order higher than the original approximations. Moreover, since the local postprocessing is performed at the element level and only at the timestep where higher accuracy is desired, it adds very little to the overall computational cost. As a result, with the HDG method, the $(k + 2)$-convergent solution can be computed at the cost of a DG approximation using polynomials of degree k only.

2.9.2. Comparison with the continuous Galerkin–Newmark method

The continuous Galerkin–Newmark (CG-N) method is widely used to solve the acoustic and elastic wave equations. The CG-N method employs continuous Galerkin (CG) finite elements for spatial discretization and Newmark schemes for time integration. The unconditionally stable second-order Newmark scheme with $\beta = 1/4$ [31] is the most popular scheme of the Newmark family. Below we compare the HDG–DIRK(q,p) method with the CG-N$(\beta = 1/4)$ method.

We consider the previous test case to assess the performance of the HDG–DIRK(q,p) method and the CG-N$(\beta = 1/4)$ method based on the accuracy versus the global degrees of freedom. Since the DIRK(q,p) scheme requires q linear system solves

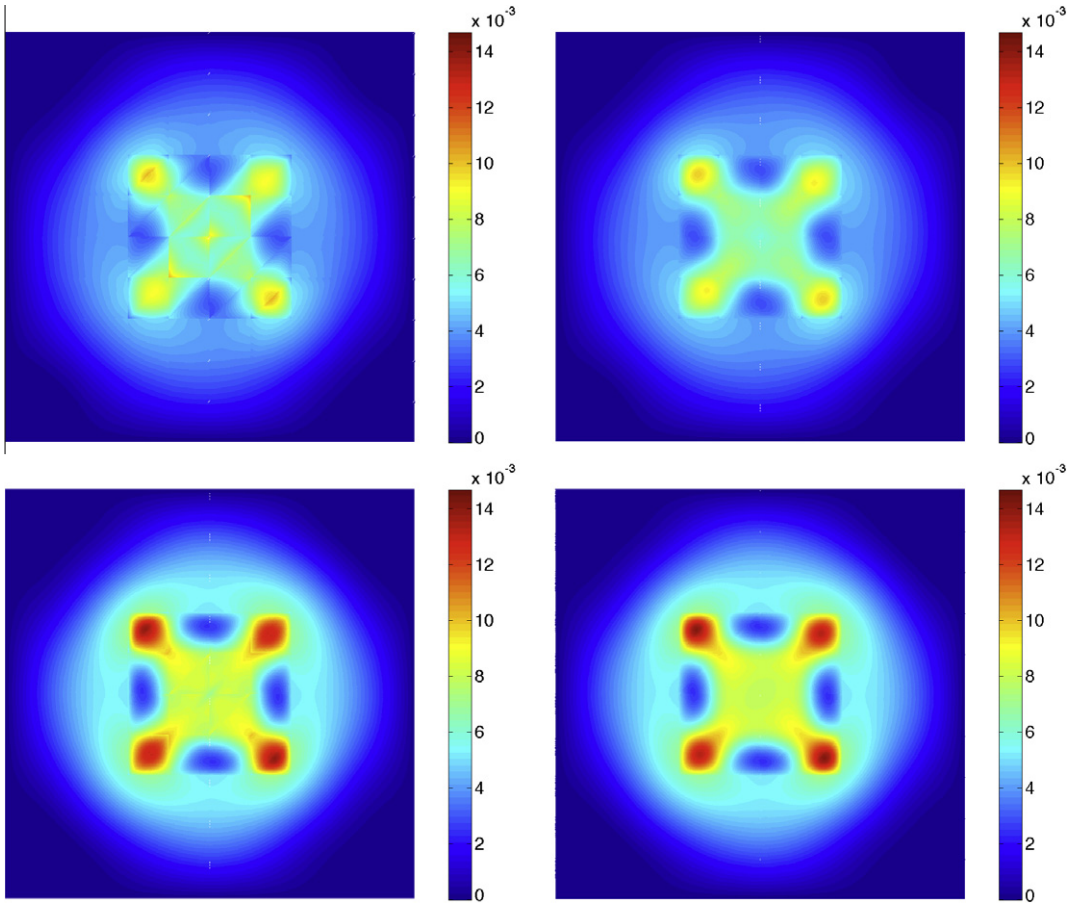

Fig. 3. Example 2: Plots of u_h (left) and u_h^* (right) at $t = 0.25$ for $k = 2$ (top) and $k = 3$ (bottom).

per one timestep, we take $h/\Delta t = 4q$ for the CG-N($\beta = 1/4$) method to guarantee that both the HDG–DIRK(q,p) method and the CG-N($\beta = 1/4$) method have exactly the same total number of linear system solves. Moreover, since the local postprocessing is inexpensive, we take the postprocessed solution as the approximation for the HDG–DIRK(q,p) method and compare it with the approximate solution of the CG-N($\beta = 1/4$) method.

Fig. 1 shows the L^2 errors in u, v, and $\boldsymbol{q}$ as a function of the global degrees of freedom. Thanks to its faster convergence rate the HDG–DIRK(q,p) method quickly outperforms the CG-N($\beta = 1/4$) method as the global degrees of freedom increase. In particular, as the problem size becomes larger than 10^3, the HDG–DIRK(q,p) method produce smaller errors than the CG-N($\beta = 1/4$) method for both $k = 1$ and $k = 2$. For the same problem size, the approximate gradient of the HDG–DIRK(q,p) method is significantly more accurate than that of the CG-N($\beta = 1/4$) method.

2.9.3. Inhomogeneous wave speed

The problem has $\Omega = (0,1) \times (0,1)$, $f = 0$, $T = 0.25$, and $\mathbf{A} = \mathbf{I}$. The density $\rho(\boldsymbol{x})$ is set to 1 if $\boldsymbol{x} \in \Omega_1 = (0.3,0.7) \times (0.3,0.7)$ and 1/9 if $\boldsymbol{x} \in \Omega_2 \equiv \Omega \backslash \Omega_1$. Therefore, the wave speed is equal to 1 in Ω_1 and 3 in Ω_2. The boundary conditions are $v = 0$ on $\partial\Omega$. The initial conditions are given by $u(\boldsymbol{x}, t = 0) = 0$ and $v(\boldsymbol{x}, t = 0) = 2e^{-500((x-0.5)^2+(y-0.5)^2)}$.

We use a regular 10×10 Cartesian grid with meshsize $h = 0.1$ for spatial discretization and employ the BDF3 scheme with timestep $\Delta t = 0.0025$ for temporal discretization. We use the DIRK(2,3) method to obtain the numerical solution at the first and second time levels. We choose $\tau = 1$. We present in Fig. 2 the plots of the approximate displacement u_h and the postprocessed displacement u_h^* at time $t = 0.2$ for $k = 2$ and $k = 3$. Fig. 3 shows the same results at time $t = 0.25$. The discontinuity in the material can be clearly seen from the plots of the numerical solution at time $t = 0.2$. We observe that the approximate displacement is significantly improved as k increases from 2 to 3. Moreover, the local postprocessing does also enhance the accuracy of the solution the since u_h^* is clearly superior to u_h. These results illustrate the effectiveness of the local postprocessing for inhomogeneous media.

2.9.4. Reflection of waves from a L-shaped domain

This problem is taken from [8]. We consider a L-shaped domain $\Omega = (0,1)^2 \backslash (0.7,1)^2$, $f = 0, \rho = 1, \mathbf{A} = \mathbf{I}$ and $T = 0.45$. The boundary and initial conditions are kept the same as those in the previous example.

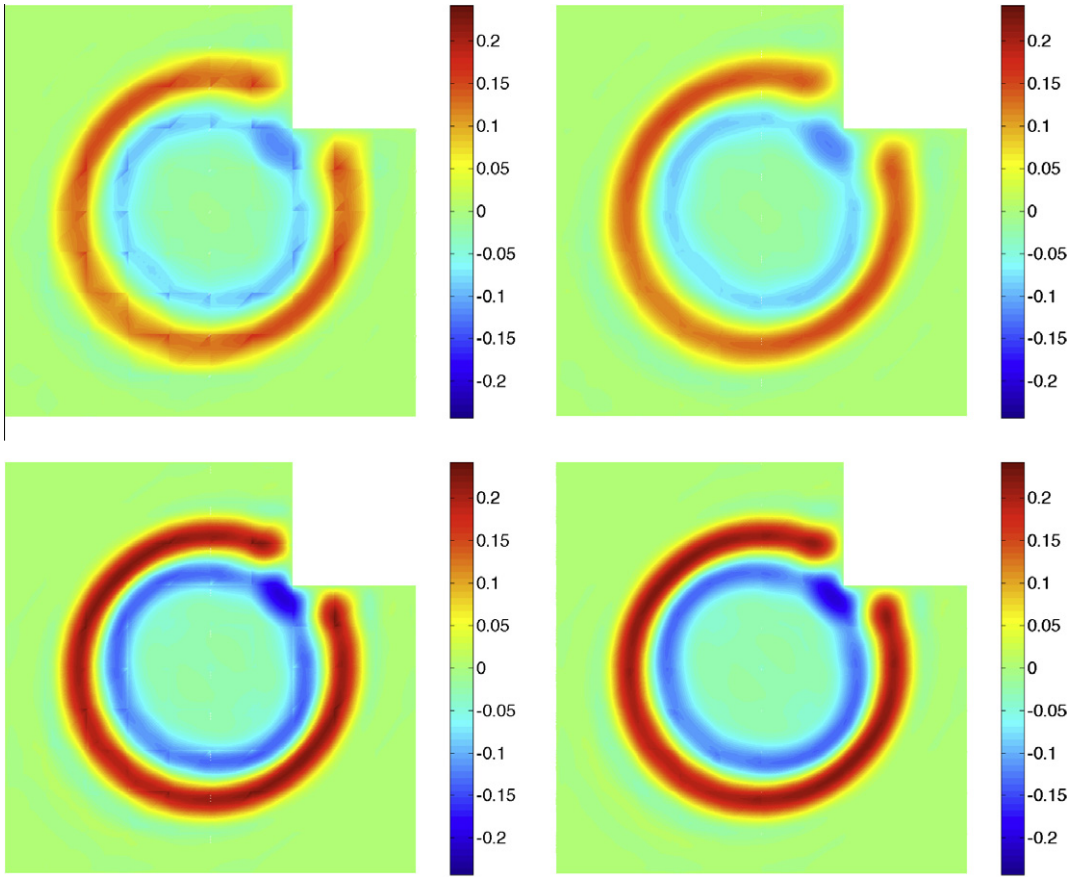

Fig. 4. Example 3: Plots of v_h (left) and v_h^* (right) at $t = 0.30$ for $k = 2$ (top) and $k = 3$ (bottom).

We use a regular Cartesian grid with meshsize $h = 0.1$ for spatial discretization and the BDF3 scheme with timestep $\Delta t = 0.005$ for temporal discretization. We set $\tau = 1$. We present in Fig. 4 the plots of the approximate velocity v_h and the postprocessed velocity v_h^* at time $t = 0.3$ for $k = 2$ and $k = 3$. Fig. 5 shows the same results at time $t = 0.45$. We see that the waves begin to reflect from the corner $(0.7, 0.7)$ at time $t = 0.3$. After time $t = 0.3$, the waves form a second circular wave centered at the corner $(0.7, 0.7)$. Again we observe that increasing k leads to a significant improvement of the numerical solution and that the local postprocessing does enhance the accuracy. Hence, our local postprocessing is effective even for irregular domains such as the L-shaped domain.

2.9.5. Scattering of plane wave from an airfoil

This example illustrates the performance of the HDG method for curved geometry in an unbounded domain. We consider the acoustic scattering from the NACA 0012 airfoil of an incident planar wave of the form:

$$u^i(x, y, t) = \sin(\boldsymbol{k} \cdot \boldsymbol{x} - c|\boldsymbol{k}|t),$$

where c is the wave speed and $\boldsymbol{k} = (k_x, k_y)$ is the wave number with $|\boldsymbol{k}| = \sqrt{k_x^2 + k_y^2}$. For our particular problem we consider $k_x = 20$, $k_y = 0$, and $c = 1$, which represents the plane wave along the x direction. The scattered wave u satisfies the homogenous wave Eq. (1) ($f = 0, \rho = 1$, and $\mathbf{A} = c^2\mathbf{I}$) and the following boundary condition on the airfoil surface Γ_a:

$$\nabla u \cdot \boldsymbol{n} = -\nabla u^i \cdot \boldsymbol{n}, \quad \text{on } \Gamma_a.$$

We study the scattered field in an truncated domain $\Omega_{\text{ext}} = (-2,3) \times (-2,2)$ and apply the first-order absorbing boundary condition on its boundary as:

$$\frac{\partial u}{\partial t} + \nabla u \cdot \boldsymbol{n} = 0, \quad \text{on } \Gamma_{\text{ext}}.$$

The initial condition is $u(\boldsymbol{x}, t = 0) = 0$ and $\partial u(\boldsymbol{x}, t = 0)/\partial t = 0$. The computational domain and associated grid is given in Fig. 6.

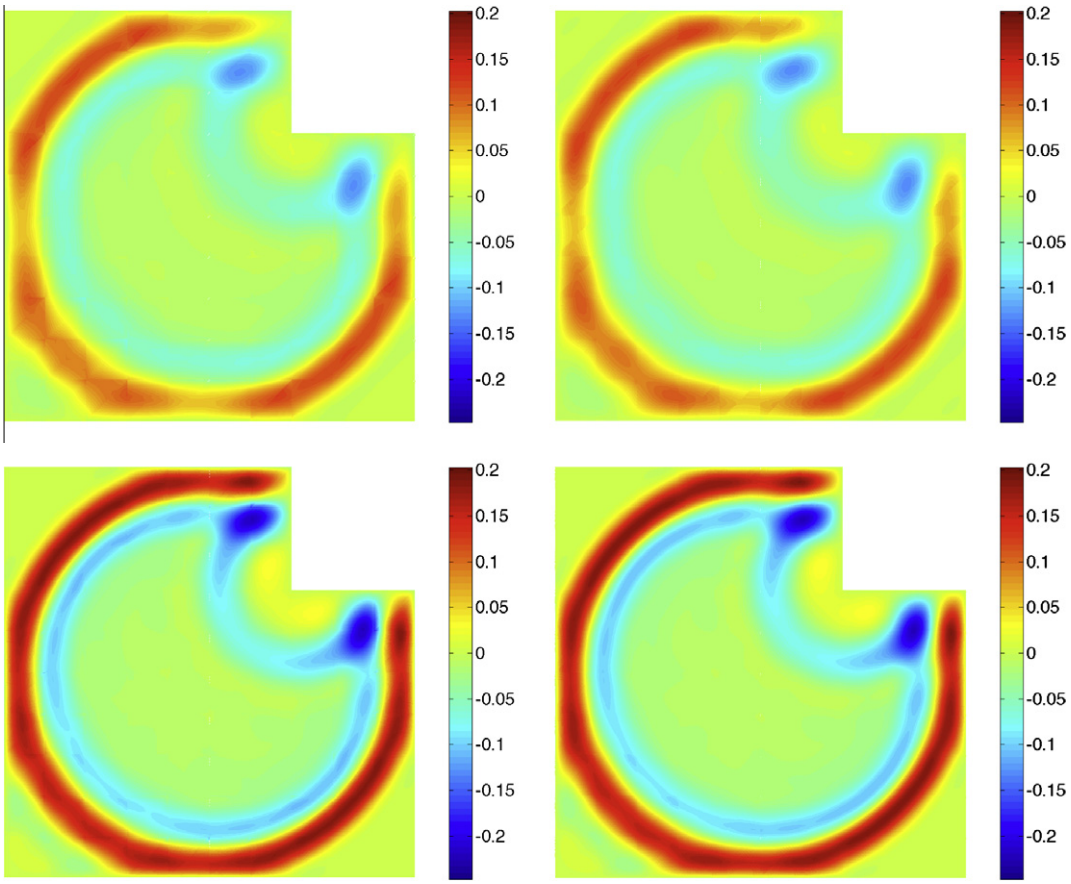

Fig. 5. Example 3: Plots of v_h (left) and v_h^* (right) at $t = 0.45$ for $k = 2$ (top) and $k = 3$ (bottom).

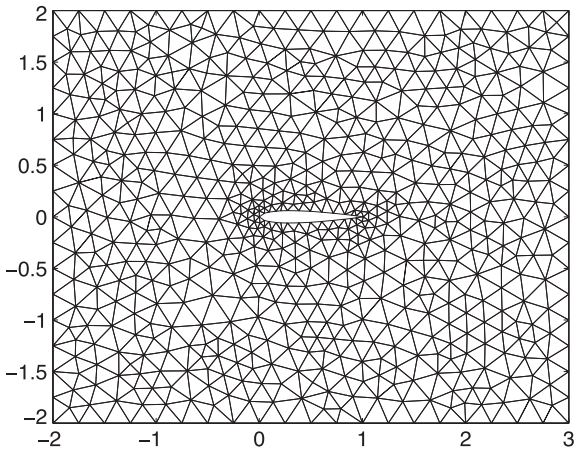

Fig. 6. Example 4: Geometry and mesh for the acoustic scattering problem.

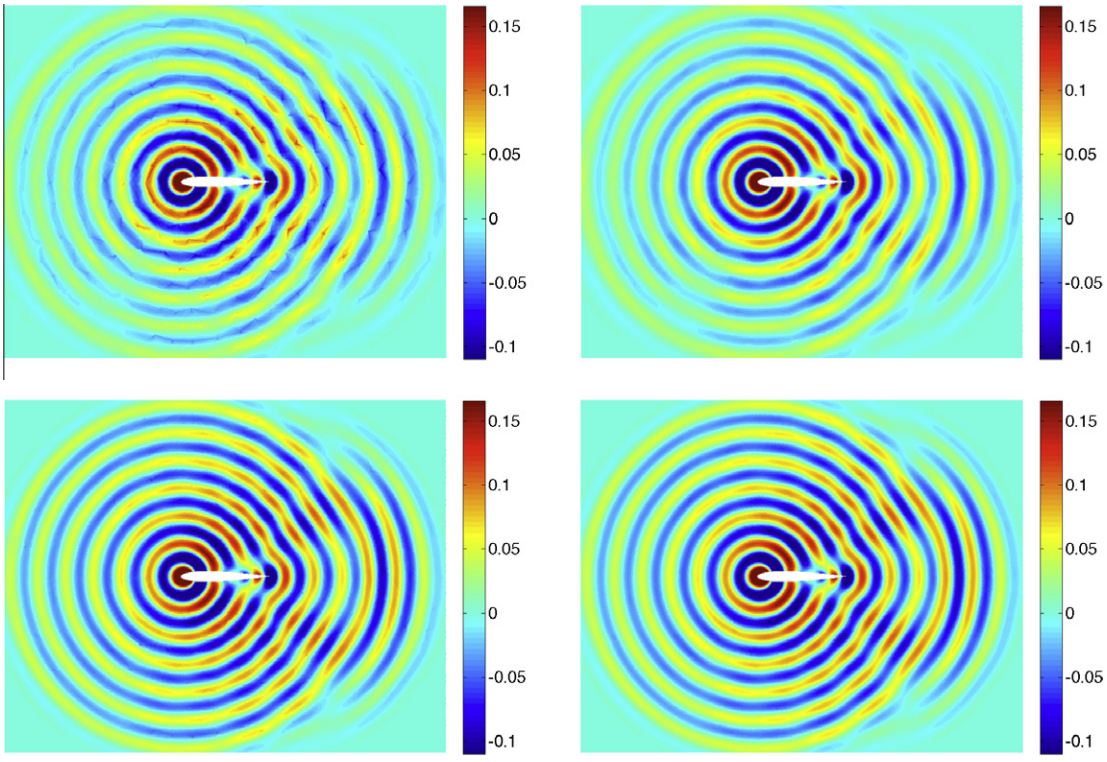

Fig. 7. Example 4: Plots of u_h (left) and u_h^* (right) at $t = 2$ for $k = 2$ (top) and $k = 3$ (bottom).

The BDF3 scheme is used to discretize the time derivatives with $\Delta t = 0.005$, while the HDG method is employed for spatial discretization. Here we choose $\tau = 1$. We present the computed results in Figs. 7 and 8 for $k = 2,3$. It is clearly seen that the local postprocessing enhances the accuracy quite significantly since the postprocessed solution for $k = 2$ looks much better than the original solution for $k = 2$ and resembles the original solution for $k = 3$. Very good results are obtained in spite of the fact that the problem has curved geometry, absorbing boundary condition, and relatively high wave number.

3. The elastic wave equations

In this section, we extend the HDG method developed in the previous section to linear elastodynamics. Linear elastodynamics has some important applications in engineering such as seismic modeling and nondestructive evaluation. The elastic

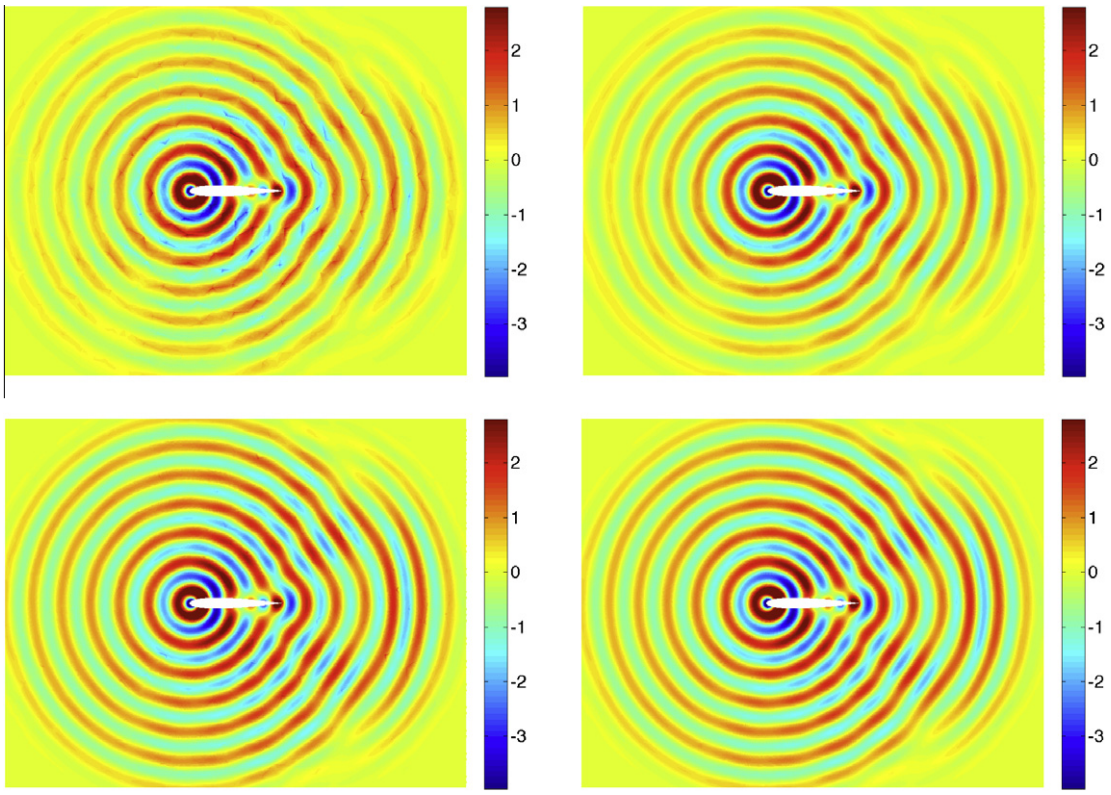

Fig. 8. Example 4: Plots of v_h (left) and v_h^* (right) at $t = 2$ for $k = 2$ (top) and $k = 3$ (bottom).

wave equations are different from the scalar acoustic wave equation in the sense that they are vectorial and have two different wave speeds, namely, pressure (primary) wave speed and shear (secondary) wave speed. However, in many cases, the elastic wave equations can be (exactly or approximately) reduced to an acoustic wave equation plus some other wave equations. Both the acoustic and elastic wave equations can be written as a first-order hyperbolic system of equations:

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \boldsymbol{A}_i \boldsymbol{w}}{\partial x_i} = \boldsymbol{f}, \tag{29}$$

where $\boldsymbol{w}$ is the vector of field variables including the velocity field and displacement gradient, and $\boldsymbol{A}_i$ are matrices consisting of material parameters. Therefore, it is natural to extend the HDG method for acoustics to treat elastodynamics.

We note that the classical DG method [42] can be used to solve the first-order formulation (29) of both the acoustic and elastic wave equations. However, the disadvantage of this DG method is that when an implicit time-stepping method is used to discretize the time derivative it will result in a large system involving the degrees of freedom of both the velocity field and displacement gradient. With the HDG method we also aim to solve the first-order system (29). However, the global system of the HDG method involves only the degrees of freedom of the approximate trace of the velocity field. This will lead to a significant saving in computational cost and memory storage for the HDG method relative to the classical DG method.

Although there are several different formulations of the elastic wave equations, we choose to develop the HDG method for the displacement gradient–velocity–pressure formulation. This method is based on an extension of our recent work for the Stokes system [14,36]. Similar development can be applied to other formulations such as the stress–velocity–pressure formulation by following the method of lines presented in this paper and the HDG framework proposed in [17] for three different formulations of the Stokes system.

3.1. Displacement gradient-velocity-pressure formulation

Let $\boldsymbol{u}$ represent the vector field of the displacement components, λ and μ the Lamé moduli, ρ the density of the elastic isotropic material, and $\boldsymbol{b}$ a time-dependent body force. Let Ω be an open bounded domain in $\mathbb{R}^d$ and T a fixed final time. The motion of the elastic isotropic body Ω is governed by

$$\rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} - \nabla \cdot [\mu \nabla \boldsymbol{u} + (\mu + \lambda)(\nabla \cdot \boldsymbol{u})\mathbf{I}] = \boldsymbol{b}, \ \text{in} \Omega \times (0, T]. \tag{30}$$

We introduce the velocity field $\boldsymbol{v} = \partial \boldsymbol{u}/\partial t$, the displacement gradient tensor $\mathbf{H} = \nabla \boldsymbol{u}$, and the hydrostatic pressure $p = (\mu + \lambda)(\nabla \cdot \boldsymbol{u})$. We then rewrite (30) as the first-order system:

$$\begin{aligned}
&\frac{\partial \mathbf{H}}{\partial t} - \nabla \boldsymbol{v} = 0, \quad \text{in } \Omega \times (0,T],\\
&\rho \frac{\partial \boldsymbol{v}}{\partial t} - \nabla \cdot (\mu \mathbf{H} + p\mathbf{I}) = \boldsymbol{b}, \quad \text{in } \Omega \times (0,T],\\
&\epsilon \frac{\partial p}{\partial t} - \nabla \cdot \boldsymbol{v} = 0, \quad \text{in } \Omega \times (0,T].
\end{aligned} \tag{31}$$

Here $\epsilon = 1/(\mu + \lambda)$, and $\mathbf{I}$ is the second-order identity tensor. Associated with this system are the boundary conditions:

$$\boldsymbol{v} = \boldsymbol{g}_D, \quad \text{on } \partial\Omega \times (0,T],$$

and initial condition:

$$\boldsymbol{v} = \boldsymbol{v}_0, \quad \mathbf{H} = \mathbf{H}_0, \quad p = p_0, \quad \text{on } \Omega \times \{t = 0\}.$$

For simplicity of exposition, we assume that $\epsilon > 0$, which in essence means that the elastic solid is either compressible or nearly incompressible. The incompressible limit $\epsilon = 0$ requires the average pressure condition and can be treated by the augmented Lagrangian method [36,17].

In addition to the finite element spaces defined in Sub Section 2.2, we introduce the following new finite element spaces:

$$\begin{aligned}
\mathbf{G}_h &= \left\{ \mathbf{N} \in \left(L^2(\mathcal{T}_h)\right)^{d\times d} : \mathbf{N}|_K \in (\mathcal{P}_k(D))^{d\times d}, \ \forall K \in \mathcal{T}_h \right\},\\
\boldsymbol{M}_h &= \left\{ \boldsymbol{\mu} \in \left(L^2(\mathcal{E}_h)\right)^{d} : \boldsymbol{\mu}|_F \in (\mathcal{P}_k(F))^{d}, \ \forall F \in \mathcal{E}_h \right\}.
\end{aligned}$$

We also set:

$$\mathsf{M}_h(\mathsf{g}) = \left\{ \boldsymbol{\mu} \in \mathsf{M}_h : \boldsymbol{\mu} = \mathsf{P}\mathsf{g} \text{ on } \partial\Omega \right\},$$

where P denotes the L^2-projection into the space $\{\boldsymbol{\mu}|_{\partial\Omega} \ \forall \boldsymbol{\mu} \in \boldsymbol{M}_h\}$. We then define volume and boundary inner products associated with $\mathbf{G}_h$ as:

$$(\mathbf{N},\mathbf{L})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\mathbf{N},\mathbf{L})_K, \quad \langle \mathbf{N},\mathbf{L}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \mathbf{N},\mathbf{L}\rangle_{\partial K},$$

for $\mathbf{N},\mathbf{L} \in (L^2(\mathcal{T}_h))^{d\times d}$. Note that $(\mathbf{N},\mathbf{L})_D$ denotes the integral of $tr(\mathbf{N}^T\mathbf{L})$ over D, where tr is the trace operator.

3.2. HDG method

We directly consider the Backward–Euler method for temporal discretization as higher-order methods admit a similar procedure. We use the same notation introduced in Sub Section 2.3 for time integration. TheHDG method then finds an approximation $(\mathbf{H}_h^n, \mathsf{v}_h^n, p_h^n, \hat{\mathsf{v}}_h^n) \in \mathbf{G}_h \times \mathsf{V}_h \times W_h \times \mathsf{M}_h(\mathsf{g}_D^n)$ at time level $t^n = n\Delta t^n$ such that:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\mathbf{H}_h^n, \mathbf{N})_{\mathcal{T}_h} + (\mathsf{v}_h^n, \nabla\cdot\mathbf{N})_{\mathcal{T}_h}\\
&\qquad - \langle \hat{\mathsf{v}}_h^n, \mathbf{N}\cdot\mathsf{n}\rangle_{\partial\mathcal{T}_h} = \frac{1}{\Delta t^n}\left(\mathbf{H}_h^{n-1}, \mathbf{N}\right)_{\mathcal{T}_h},\\
&\frac{1}{\Delta t^n}(\rho\mathsf{v}_h^n, \mathsf{w})_{\mathcal{T}_h} + (\mu\mathbf{H}_h^n + p_h^n\mathbf{I}, \nabla\mathsf{w})_{\mathcal{T}_h}\\
&\qquad - \left\langle \left(\mu\hat{\mathbf{H}}_h^n + \hat{p}_h^n\mathbf{I}\right)\cdot\mathsf{n}, \mathsf{w}\right\rangle_{\partial\mathcal{T}_h} = (\mathsf{b}^n, \mathsf{w})_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho\mathsf{v}_h^{n-1}, \mathsf{w})_{\mathcal{T}_h},\\
&\frac{1}{\Delta t^n}(\epsilon p_h^n, q)_{\mathcal{T}_h} + (\mathsf{v}_h^n, \nabla q)_{\mathcal{T}_h}\\
&\qquad - \langle \hat{\mathsf{v}}_h^n\cdot\mathsf{n}, q\rangle_{\partial\mathcal{T}_h} = \frac{1}{\Delta t^n}(\epsilon p_h^{n-1}, q)_{\mathcal{T}_h},\\
&\left\langle \left(\mu\hat{\mathbf{H}}_h^n + \hat{p}_h^n\mathbf{I}\right)\cdot\mathsf{n}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} = 0,
\end{aligned} \tag{32}$$

for all $(\mathbf{N},\boldsymbol{w},q,\boldsymbol{\mu}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times W_h \times \boldsymbol{M}_h(\boldsymbol{0})$, where

$$\mu\hat{\mathbf{H}}_h^n + \hat{p}_h^n\mathbf{I} = \mu\mathbf{H}_h^n + p_h^n\mathbf{I} - \mathbf{S}\left(\mathsf{v}_h^n - \hat{\mathsf{v}}_h^n\right)\otimes\mathsf{n}. \tag{33}$$

Here $\mathbf{S}$ is a second-order tensor consisting of stabilization parameters which will be determined below. Note that the numerical approximations at time level $t^0 = 0$ are computed as the L^2 projection of the initial conditions.

Once v_h^n is available we can compute $\mathsf{u}_h^n \in \mathsf{V}_h$ by simultaneously solving the following system:

$$\frac{1}{\Delta t^n}(\mathsf{u}_h^n, \mathsf{w})_{\mathcal{T}_h} = (\mathsf{v}_h^n, \mathsf{w})_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\mathsf{u}_h^{n-1}, \mathsf{w})_{\mathcal{T}_h}, \quad \forall \mathsf{w} \in \mathsf{V}_h. \tag{34}$$

This system can be solved at the element level thanks to the discontinuous nature of the space $\boldsymbol{V}_h$.

We obtain the following results whose proof is simlar to that of Proposition 1 and is thus omitted here for brevity.

Proposition 2. *Any solution of the* Eq. (32) *satisfies the energy identity:*

$$\frac{1}{4}(\mu \mathbf{H}_h^m, \mathbf{H}_h^m)_{\mathcal{T}_h} + \frac{1}{2}(\rho \mathsf{v}_h^m, \mathsf{v}_h^m)_{\mathcal{T}_h} + \frac{1}{2}(\epsilon p_h^m, p_h^m)_{\mathcal{T}_h} + \Theta_h^m = \frac{1}{4}\left(\mu \mathbf{H}_h^0, \mathbf{H}_h^0\right)_{\mathcal{T}_h} + \frac{1}{2}(\rho \mathsf{v}_h^0, \mathsf{v}_h^0)_{\mathcal{T}_h} + \frac{1}{2}(\epsilon p_h^0, p_h^0)_{\mathcal{T}_h} + \Psi_h^m, \tag{35}$$

where

$$\begin{aligned}
\Theta_h^m &:= \sum_{n=1}^{m} \Delta t^n \langle \mathbf{S}(\mathsf{v}_h^n - \hat{\mathsf{v}}_h^n), (\mathsf{v}_h^n \hat{\mathsf{v}}_h^n) \rangle_{\partial \mathcal{T}_h} \\
&\quad + \frac{1}{4}\sum_{n=1}^{m} \left(\mu\left(\mathbf{H}_h^n - \mathbf{H}_h^{n-1}\right), \left(\mathbf{H}_h^n - \mathbf{H}_h^{n-1}\right)\right)_{\mathcal{T}_h} \\
&\quad + \frac{1}{2}\sum_{n=1}^{m} \left(\rho(\mathsf{v}_h^n - \mathsf{v}_h^{n-1}), (\mathsf{v}_h^n - \mathsf{v}_h^{n-1})\right)_{\mathcal{T}_h} \\
&\quad + \frac{1}{2}\sum_{n=1}^{m} \left(\epsilon(p_h^n - p_h^{n-1}), (p_h^n - p_h^{n-1})\right)_{\mathcal{T}_h} \\
\Psi_h^m &:= \frac{1}{2}\left(\mu \mathbf{H}_h^0, \mathbf{H}_h^0\right)_{\mathcal{T}_h} + (\rho \mathsf{v}_h^0, \mathsf{v}_h^0)_{\mathcal{T}_h} + (\epsilon p_h^0, p_h^0)_{\mathcal{T}_h} + \sum_{n=1}^{m} \Delta t^n (\mathsf{b}^n, \mathsf{v}_h^n)_{\mathcal{T}_h}
\end{aligned}$$

for all $m \geqslant 1$. Moreover, if $\gamma > 0, \rho > 0$, and $\epsilon > 0$, and the stabilization tensor $\mathbf{S}$ satisfies:

$$\mathsf{p}^T \mathbf{S} \mathsf{p} > 0, \text{ on } \partial \mathcal{T}_h, \forall \mathsf{p} \in \mathbb{R}^d, \tag{36}$$

then the HDG method (32) has a unique solution $(\mathbf{H}_h^n, \mathsf{v}_h^n, p_h^n, \hat{\mathsf{u}}_h^n)$ for any $n \geqslant 1$.

It follows from the energy identity (35) that $\mathbf{S}$ should be chosen as:

$$\mathbf{S} = \frac{\rho}{t_c}\mathbf{I} = \rho \omega_c \mathbf{I}, \tag{37}$$

where t_c is the characteristic timescale and $\omega_c = 1/t_c$ is the characteristic frequency.

3.3. Implementation

We describe the second approach to implement the HDG method in addition to the first approach described in Sub Section 2.6. We begin with inserting (33) into (32) to obtain that $(\mathbf{H}_h^n, \mathsf{v}_h^n, p_h^n, \hat{\mathsf{v}}_h^n) \in \mathbf{G}_h \times \mathsf{V}_h \times W_h \times \mathsf{M}_h(\mathsf{g}_D^n)$ satisfies:

$$\begin{aligned}
&\frac{1}{\Delta t^n}(\mathbf{H}_h^n, \mathbf{N})_{\mathcal{T}_h} + (\mathsf{v}_h^n, \nabla \cdot \mathbf{N})_{\mathcal{T}_h} \\
&\qquad - \langle \hat{\mathsf{v}}_h^n, \mathbf{N} \cdot \mathsf{n} \rangle_{\partial \mathcal{T}_h} = \frac{1}{\Delta t^n}\left(\mathbf{H}_h^{n-1}, \mathbf{N}\right)_{\mathcal{T}_h}, \\
&\frac{1}{\Delta t^n}(\rho \mathsf{v}_h^n, \mathsf{w})_{\mathcal{T}_h} - (\nabla \cdot (\mu \mathbf{H}_h^n + p_h^n \mathbf{I}), \mathsf{w})_{\mathcal{T}_h} \\
&\qquad + \langle \mathbf{S}(\mathsf{v}_h^n - \hat{\mathsf{v}}_h^n), \mathsf{w} \rangle_{\partial \mathcal{T}_h} = (\mathsf{b}^n, \mathsf{w})_{\mathcal{T}_h} + \frac{1}{\Delta t^n}(\rho \mathsf{v}_h^{n-1}, \mathsf{w})_{\mathcal{T}_h}, \\
&\frac{1}{\Delta t^n}(\epsilon p_h^n, q)_{\mathcal{T}_h} + (\mathsf{v}_h^n, \nabla q)_{\mathcal{T}_h} \\
&\qquad - \langle \hat{\mathsf{v}}_h^n \cdot \mathsf{n}, q \rangle_{\partial \mathcal{T}_h} = \frac{1}{\Delta t^n}(\epsilon p_h^{n-1}, q)_{\mathcal{T}_h}, \\
&\langle (\mu \mathbf{H}_h^n + p_h^n \mathbf{I}) \cdot \mathsf{n} - \mathbf{S}(\mathsf{v}_h^n - \hat{\mathsf{v}}_h^n), \boldsymbol{\mu} \rangle_{\partial \mathcal{T}_h} = 0,
\end{aligned} \tag{38}$$

for all $(\mathbf{N}, \boldsymbol{w}, q, \boldsymbol{\mu}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times W_h \times \boldsymbol{M}_h(\boldsymbol{0})$. Here we may follow the approach outlined in Sub Section 2.5 to implement the HDG method. However, the HDG method can also be implemented by following the procedure described below.

We note that the first three equations of (38) can be written at the element level as:

$$\begin{aligned}
&\frac{1}{\Delta t^n}\left(\mathbf{H}_h^n,\mathbf{N}\right)_K+\left(\mathsf{v}_h^n,\nabla\cdot\mathbf{N}\right)_K=\left\langle\hat{\mathsf{v}}_h^n,\mathbf{N}\cdot\mathsf{n}\right\rangle_{\partial K}+\frac{1}{\Delta t^n}\left(\mathbf{H}_h^{n-1},\mathbf{N}\right)_K,\\
&\frac{1}{\Delta t^n}\left(\rho\mathsf{v}_h^n,\mathsf{w}\right)_K-\left(\nabla\cdot\left(\mu\mathbf{H}_h^n+p_h^n\mathbf{I}\right),\mathsf{w}\right)_K\\
&\qquad+\left\langle\mathbf{S}\left(\mathsf{v}_h^n\right),\mathsf{w}\right\rangle_{\partial K}=\left\langle\mathbf{S}\left(\hat{\mathsf{v}}_h^n\right),\mathsf{w}\right\rangle_{\partial K}\\
&\qquad+\left(\mathsf{b}^n,\mathsf{w}\right)_K+\frac{1}{\Delta t^n}\left(\rho\mathsf{v}_h^{n-1},\mathsf{w}\right)_K,\\
&\frac{1}{\Delta t^n}\left(\epsilon p_h^n,q\right)_K+\left(\mathsf{v}_h^n,\nabla q\right)_K=\left\langle\hat{\mathsf{v}}_h^n\cdot\mathsf{n},q\right\rangle_{\partial K}+\frac{1}{\Delta t^n}\left(\epsilon p_h^{n-1},q\right)_K,
\end{aligned}\tag{39}$$

for all $(\mathbf{N},\mathsf{w},q)\in(\mathcal{P}_k(K))^{d\times d}\times(\mathcal{P}_k(K))^d\times\mathcal{P}_k(K)$. The above system of equations thus defines a "local solver" $\mathcal{L}$ that maps $(\hat{\mathsf{v}}_h^n,\mathbf{H}_h^{n-1},\mathsf{v}_h^{n-1},p_h^{n-1},\mathsf{b}^n)$ to $(\mathbf{H}_h^n,\mathsf{v}_h^n,p_h^n)$ as:

$$\left(\hat{\mathsf{v}}_h^n,\mathbf{H}_h^{n-1},\mathsf{v}_h^{n-1},p_h^{n-1},\mathsf{b}^n\right)\overset{\mathcal{L}}{\mapsto}\left(\mathbf{H}_h^n,\mathsf{v}_h^n,p_h^n\right).\tag{40}$$

Therefore, if we know $\hat{\mathsf{v}}_h^n$ we can solve the local problem (39) element-by-element to obtain the numerical solution. It remains only to determine $\hat{\mathsf{v}}_h^n$. Toward this end, for any given $\boldsymbol{\eta}\in\boldsymbol{M}_h$ we set:

$$\begin{aligned}
&\left(\mathbf{H}_h^{\boldsymbol{\eta}},\mathsf{v}_h^{\boldsymbol{\eta}},p_h^{\boldsymbol{\eta}}\right):=\mathcal{L}(\boldsymbol{\eta},0,0,0,0),\\
&\left(\mathbf{H}_h^{\mathbf{H}_h^{n-1}},\mathsf{v}_h^{\mathbf{H}_h^{n-1}},p_h^{\mathbf{H}_h^{n-1}}\right):=\mathcal{L}\left(0,\mathbf{H}_h^{n-1},0,0,0\right),\\
&\left(\mathbf{H}_h^{\mathsf{v}_h^{n-1}},\mathsf{v}_h^{\mathsf{v}_h^{n-1}},p_h^{\mathsf{v}_h^{n-1}}\right):=\mathcal{L}\left(0,0,\mathsf{v}_h^{n-1},0,0\right),\\
&\left(\mathbf{H}_h^{p_h^{n-1}},\mathsf{v}_h^{p_h^{n-1}},p_h^{p_h^{n-1}}\right):=\mathcal{L}\left(0,0,0,p_h^{n-1},0\right),\\
&\left(\mathbf{H}_h^{\mathsf{b}^n},\mathsf{v}_h^{\mathsf{b}^n},p_h^{\mathsf{b}^n}\right):=\mathcal{L}(0,0,0,0,\mathsf{b}^n).
\end{aligned}\tag{41}$$

Table 2
Example 5: History of convergence of the numerical approximations with a fixed ratio $h/\Delta t=4$ for $\lambda=1$.

Degree	Mesh	$\|\mathsf{u}-\mathsf{u}_h\|_{\mathcal{T}_h}$		$\|\mathsf{v}-\mathsf{v}_h\|_{\mathcal{T}_h}$		$\|\boldsymbol{\sigma}-\boldsymbol{\sigma}_h\|_{\mathcal{T}_h}$		$\|\mathsf{u}-\mathsf{u}_h^*\|_{\mathcal{T}_h}$		$\|\mathsf{v}-\mathsf{v}_h^*\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	3.79e−4	–	1.94e−3	–	2.08e−3	–	1.74e−4	–	1.28e−3	–
	8	1.12e−4	1.76	4.51e−4	2.11	5.07e−4	2.04	2.53e−5	2.78	1.74e−4	2.88
	16	3.04e−5	1.88	1.06e−4	2.09	1.26e−4	2.01	3.27e−6	2.95	2.18e−5	2.99
	32	7.90e−6	1.94	2.60e−5	2.03	3.16e−5	2.00	4.12e−7	2.99	2.96e−6	2.89
	64	2.01e−6	1.97	6.45e−6	2.01	7.93e−6	2.00	5.16e−8	3.00	3.99e−7	2.89
2	4	5.14e−5	–	2.26e−4	–	3.27e−4	–	1.78e−5	–	2.41e−4	–
	8	8.01e−6	2.68	2.90e−5	2.96	4.21e−5	2.96	1.20e−6	3.89	7.10e−6	5.08
	16	1.10e−6	2.87	3.67e−6	2.98	5.25e−6	3.00	7.39e−8	4.02	4.53e−7	3.97
	32	1.43e−7	2.94	4.60e−7	3.00	6.54e−7	3.01	4.52e−9	4.03	2.70e−8	4.07
	64	1.82e−8	2.97	5.75e−8	3.00	8.14e−8	3.00	2.78e−10	4.02	1.68e−9	4.01

Table 3
Example 5: History of convergence of the numerical approximations with a fixed ratio $h/\Delta t=4$ for $\lambda=1000$.

Degree	Mesh	$\|\mathsf{u}-\mathsf{u}_h\|_{\mathcal{T}_h}$		$\|\mathsf{v}-\mathsf{v}_h\|_{\mathcal{T}_h}$		$\|\boldsymbol{\sigma}-\boldsymbol{\sigma}_h\|_{\mathcal{T}_h}$		$\|\mathsf{u}-\mathsf{u}_h^*\|_{\mathcal{T}_h}$		$\|\mathsf{v}-\mathsf{v}_h^*\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	Error	Order	Error	Order	Error	Order	Error	Order
1	4	3.75e−4	–	1.94e−3	–	2.2e−3	–	1.72e−4	–	1.26e−3	–
	8	1.12e−4	1.75	4.49e−4	2.11	5.41e−4	2.02	2.57e−5	2.74	1.71e−4	2.89
	16	3.04e−5	1.88	1.06e−4	2.08	1.33e−4	2.02	3.37e−6	2.93	2.13e−5	3.00
	32	7.90e−6	1.94	2.60e−5	2.03	3.33e−5	2.00	4.26e−7	2.98	2.87e−6	2.89
	64	2.01e−6	1.97	6.45e−6	2.01	8.33e−6	2.00	5.34e−8	2.99	3.85e−7	2.90
2	4	5.11e−5	–	2.24e−4	–	3.67e−4	–	1.80e−5	–	2.40e−4	–
	8	7.98e−6	2.68	2.88e−5	2.96	4.82e−5	2.93	1.22e−6	3.89	6.91e−6	5.12
	16	1.09e−6	2.87	3.66e−6	2.98	6.12e−6	2.98	7.44e−8	4.03	4.20e−7	4.04
	32	1.43e−7	2.94	4.59e−7	2.99	7.89e−7	2.96	4.52e−9	4.04	2.48e−8	4.08
	64	1.82e−8	2.97	5.75e−8	3.00	9.95e−8	2.99	2.78e−10	4.02	1.48e−9	4.07

Note that the function $(\mathbf{H}_h^{\boldsymbol{\eta}}, \mathbf{v}_h^{\boldsymbol{\eta}}, p_h^{\boldsymbol{\eta}})$ is obtained from the local problem (39) by replacing $\widehat{\mathbf{v}}_h^n$ with $\boldsymbol{\eta}$ and setting other components in the right-hand side to zero. The other functions are determined in a similar manner.

The following result is a direct consequence of the local problem (39), the decomposition (41) and the last equation of the HDG system (38).

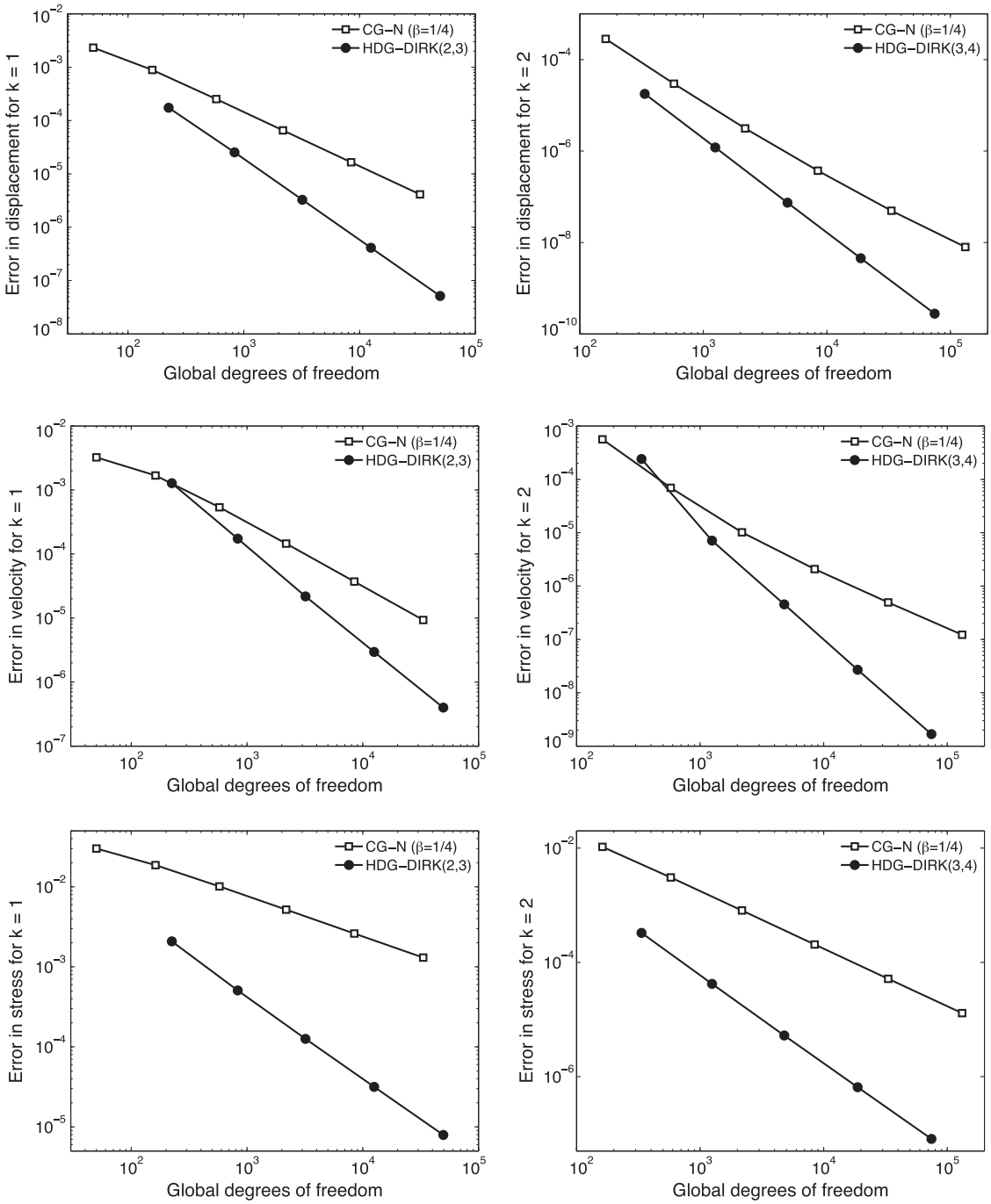

Fig. 9. Example 5: Comparison of the convergence of the L^2-errors in $\boldsymbol{u}$ (top), $\boldsymbol{v}$ (middle), and $\boldsymbol{\sigma}$ (bottom) for the HDG–DIRK(q,p) method and the CG-N$(\beta = 1/4)$ method for $\lambda = 1$. The postprocessed solution was taken as the approximation for the HDG–DIRK(q,p) method.

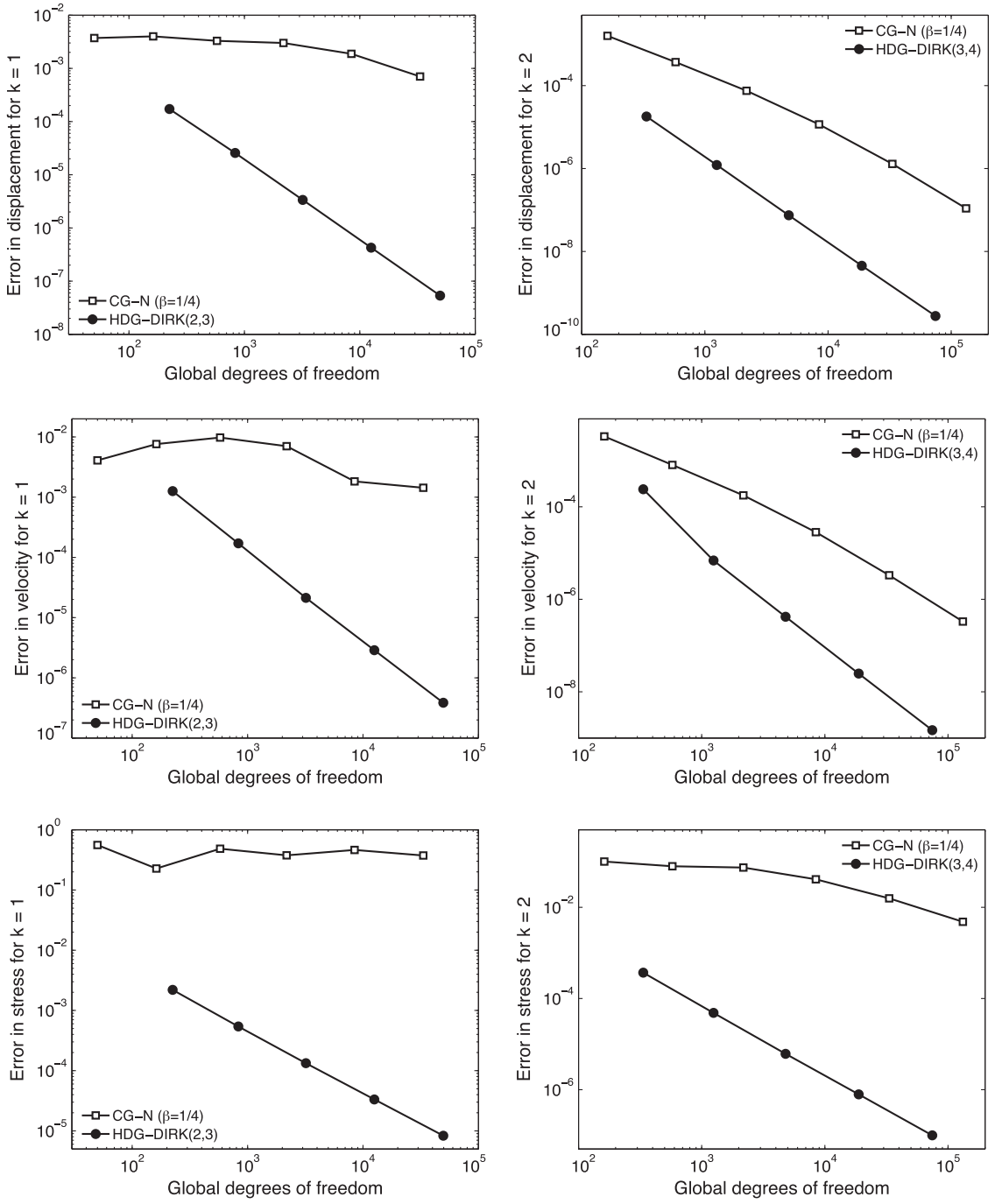

Fig. 10. Example 5: Comparison of the convergence of the L^2-errors in $\boldsymbol{u}$ (top), $\boldsymbol{v}$ (middle), and $\boldsymbol{\sigma}$ (bottom) for the HDG–DIRK(q,p) method and the CG-N($\beta = 1/4$) method for $\lambda = 1000$. The postprocessed solution was taken as the approximation for the HDG–DIRK(q,p) method.

Lemma 3.1. *We have that:*

$$
\begin{aligned}
\mathbf{H}_h^n &= \mathbf{H}_h^{\lambda^n} + \mathbf{H}_h^{\mathbf{H}_h^{n-1}} + \mathbf{H}_h^{\mathrm{v}_h^{n-1}} + \mathbf{H}_h^{p_h^{n-1}} + \mathbf{H}_h^{\mathrm{b}^n},\\
\mathrm{v}_h^n &= \mathrm{v}_h^{\lambda^n} + \mathrm{v}_h^{\mathbf{H}_h^{n-1}} + \mathrm{v}_h^{\mathrm{v}_h^{n-1}} + \mathrm{v}_h^{p_h^{n-1}} + \mathrm{v}_h^{\mathrm{b}^n},\\
p_h^n &= p_h^{\lambda^n} + p_h^{\mathbf{H}_h^{n-1}} + p_h^{\mathrm{v}_h^{n-1}} + p_h^{p_h^{n-1}} + p_h^{\mathrm{b}^n},\\
\hat{\mathrm{v}}_h^n &= \boldsymbol{\lambda}^n,
\end{aligned}
\tag{42}
$$

where $\lambda^n \in M_h(g^n)$ *is the solution of the weak formulation:*

$$a_h(\boldsymbol{\lambda}^n, \boldsymbol{\mu}) = \ell_h^n(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathsf{M}_h(0). \tag{43}$$

Here the bilinear form and linear functional are given by

$$\begin{aligned} a_h(\boldsymbol{\eta}, \boldsymbol{\mu}) &= \langle \mathbf{H}_h^{\boldsymbol{\eta}} + p_h^{\boldsymbol{\eta}}\mathbf{I} - \mathbf{S}(\mathsf{v}_h^{\boldsymbol{\eta}} - \boldsymbol{\eta}), \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h}, \\ \ell_h^n(\boldsymbol{\mu}) &= -\left\langle \left(\mathbf{H}_h^{\mathbf{H}_h^{n-1}} + p_h^{\mathbf{H}_h^{n-1}}\mathbf{I}\right)\cdot \mathsf{n} - \mathbf{S}\mathsf{v}_h^{\mathbf{H}_h^{n-1}}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} \\ &\quad - \left\langle \left(\mathbf{H}_h^{\mathsf{v}_h^{n-1}} + p_h^{\mathsf{v}_h^{n-1}}\mathbf{I}\right)\cdot \mathsf{n} - \mathbf{S}\mathsf{v}_h^{\mathsf{v}_h^{n-1}}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} \\ &\quad - \left\langle \left(\mathbf{H}_h^{p_h^{n-1}} + p_h^{p_h^{n-1}}\mathbf{I}\right)\cdot \mathsf{n} - \mathbf{S}\mathsf{v}_h^{p_h^{n-1}}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} \\ &\quad - \left\langle \left(\mathbf{H}_h^{\mathsf{b}^n} + p_h^{\mathsf{b}^n}\mathbf{I}\right)\cdot \mathsf{n} - \mathbf{S}\mathsf{v}_h^{\mathsf{b}^n}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h}, \end{aligned} \tag{44}$$

for all η, $\mu \in M_h$.

The weak formulation (43) gives rise to a matrix system of the form:

$$\mathbb{K}\ \Lambda^n = L^n, \tag{45}$$

where Λ^n is the vector of degrees of freedom of $\widehat{\mathsf{v}}_h^n$, $\mathbb{K}$ is the stiffness matrix associated with a_h, and L^n is the vector associated with ℓ_h^n. We refer to [36,17] for a detailed discussion on forming the matrix system (45) and computing the numerical solution $(\mathbf{H}_h^n, \mathsf{v}_h^n, p_h^n)$.

3.4. Local postprocessing

The local postprocessing for the elastic case is a straightforward extension of the procedure developed in Sub Section 2.7 for the acoustic case. In particular, the new approximate displacement $\mathsf{u}_h^{n*} \in (\mathcal{P}_{k+1}(K))^d$ satisfies, on every simplex $K \in \mathcal{T}_h$:

$$\begin{aligned} &\left(\nabla \mathsf{u}_h^{n*}, \nabla \mathsf{w}\right)_K = \left(\mathbf{H}_h^n, \nabla \mathsf{w}\right)_K, \quad \forall \mathsf{w} \in (\mathcal{P}_{k+1}(K))^d, \\ &\left(\mathsf{u}_h^{n*}, 1\right)_K = \left(\mathsf{u}_h^n, 1\right)_K. \end{aligned} \tag{46}$$

Note that each component of u_h^{n*} can be solved independently of each other. Hence, in effect, it is exactly the local postprocessing of Sub Section 2.7 applied to each component of the displacement field.

Using a similar technique as in the scalar case, we compute the approximate velocity gradient $\mathbf{L}_h^n \in (\mathcal{P}_k(K))^{d\times d}$ by locally solving:

$$\left(\mathbf{L}_h^n, \mathbf{N}\right)_K = -\left(\mathsf{v}_h^n, \nabla\cdot\mathbf{N}\right)_K + \left\langle \hat{\mathsf{v}}_h^n, \mathbf{N}\cdot\mathsf{n}\right\rangle_{\partial K}, \quad \forall \mathbf{N} \in (\mathcal{P}_k(K))^{d\times}. \tag{47}$$

We then define a new approximate velocity $\mathsf{v}_h^{n*} \in (\mathcal{P}_{k+1}(K))^d$ to satisfy, on every simplex $K \in \mathcal{T}_h$:

$$\begin{aligned} &\left(\nabla \mathsf{v}_h^{n*}, \nabla \mathsf{w}\right)_K = \left(\mathbf{L}_h^n, \nabla \mathsf{w}\right)_K, \quad \forall \mathsf{w} \in (\mathcal{P}_{k+1}(K))^d, \\ &\left(\mathsf{v}_h^{n*}, 1\right)_K = \left(\mathsf{v}_h^n, 1\right)_K. \end{aligned} \tag{48}$$

Again each component of $\mathbf{L}_h^n$ and v_h^{n*} can be solved independently.

Since the local postprocessing can be carried out at any particular timestep and performed at the element level, the postprocesssed displacement and velocity are very inexpensive. Note however that the postprocessing is effective only when temporal accuracy is of order $k+2$.

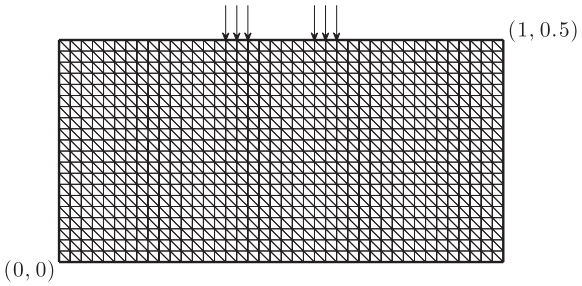

Fig. 11. Example 6: Geometry and mesh for the semi-infinite elasticity problem. The first-order absorbing boundary condition is applied on the left, right, and bottom sides, while an impulsive loading is applied over the strip $(x,y) \in [0.375, 0.425] \times \{0.5\} \cup [0.575, 0.625] \times \{0.5\}$ of the top surface.

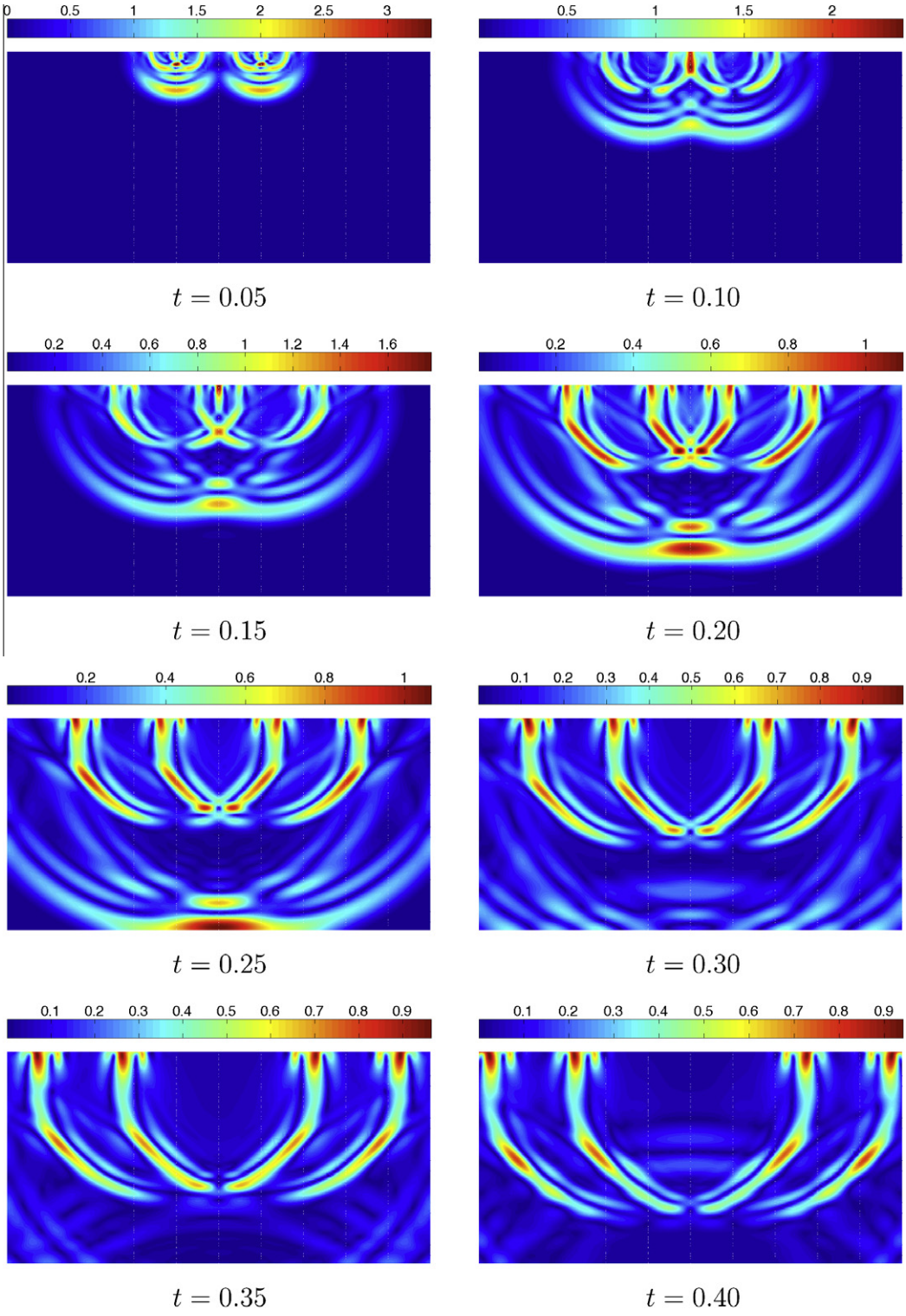

Fig. 12. Example 6: Plots of $|v_h^*| = \sqrt{(v_{hx}^*)^2 + (v_{hy}^*)^2}$ at different times for $k = 5$ and $\Delta t = 0.005$.

3.5. Stress boundary conditions and first-order absorbing boundary conditions

We describe here a novel and systematic way for imposing boundary conditions for the stress and absorbing layer which are not naturally associated with the weak formulation of the HDG method. The treatment of such incompatible boundary

conditions is first proposed in [32]. In particular, let us consider the following boundary conditions for our elastic wave equations (30):

$$\begin{aligned}
&\mathsf{v} = \mathsf{g}_D, \quad \text{on } \partial\Omega_D \times (0,T],\\
&\left(\mu\left(\nabla\mathsf{u} + \nabla\mathsf{u}^T\right) + \lambda(\nabla\cdot\mathsf{u})\mathbf{I}\right)\cdot\mathsf{n} = \mathsf{g}_N, \quad \text{on } \partial\Omega_N \times (0,T],\\
&\mathsf{v} + \left(\mu\left(\nabla\mathsf{u} + \nabla\mathsf{u}^T\right) + \lambda(\nabla\cdot\mathsf{u})\mathbf{I}\right)\cdot\mathsf{n} = 0, \quad \text{on } \Gamma_{\text{ext}},
\end{aligned}$$

which in that order imposes the boundary conditions on the velocity, stress, and (first-order) absorbing layer. Here $\partial\Omega_D$ and $\partial\Omega_N$ are two disjoint parts of the *interior* boundary of $\partial\Omega$, and Γ_{ext} is the *exterior* boundary of $\partial\Omega$.

To incorporate the above boundary conditions, we redefine the space $\boldsymbol{M}_h(\boldsymbol{g})$ as:

$$\mathsf{M}_h(\mathsf{g}) = \{\boldsymbol{\mu} \in \mathsf{M}_h \ : \ \boldsymbol{\mu} = \mathsf{Pg} \text{ on } \partial\Omega_D\}.$$

We then replace the last equation of the HDG formulation (32) with:

$$\left\langle \left(\mu\widehat{\mathbf{H}}_h^n + \hat{p}_h^n\mathbf{I}\right)\cdot\mathsf{n}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega_N} + \left\langle \hat{\mathsf{v}}_h^n, \boldsymbol{\mu}\right\rangle_{\Gamma_{\text{ext}}} + \left\langle \left(\mu\left(\widehat{\mathbf{H}}_h^n + \left(\widehat{\mathbf{H}}_h^n\right)^T\right) + \frac{\lambda}{\mu+\lambda}\hat{p}_h^n\mathbf{I}\right)\cdot\mathsf{n}, \boldsymbol{\mu}\right\rangle_{\partial\Omega_N\cup\Gamma_{\text{ext}}} = \left\langle \mathsf{g}_N^n, \boldsymbol{\mu}\right\rangle_{\partial\Omega_N}. \tag{49}$$

In essence, this equation enforces the continuity of the normal component of the total gradient for interior faces and imposes the stress boundary condition on $\partial\Omega_N$ and the absorbing boundary condition on Γ_{ext}.

3.6. Numerical results

3.6.1. Convergence test

We consider the elastic wave Eq. (30) on a unit square $\Omega = (0,1)\times(0,1)$ with $\mu = 1$ and $\rho = 1$. The exact solution is given by

$$\begin{aligned}
u_1 &= -x^2y(2y-1)(x-1)^2(y-1)\sin(\pi t),\\
u_2 &= xy^2(2x-1)(x-1)(y-1)^2\sin(\pi t).
\end{aligned}$$

The source term $\boldsymbol{b}$ is determined from the above solution. The Dirichlet boundary data are determined as the restriction of the exact solution on the boundary. Likewise the initial data is taken as an instantiation of the exact solution at time $t = 0$. The final time is $T = 0.5$. Our triangular meshes are constructed upon regular $n \times n$ Cartesian grids ($h = 1/n$). The stabilization parameter is set to $\tau = 1$.

We present the L^2 errors and orders of convergence of the numerical approximations at the final time $t = 0.5$ in Table 2 for $\lambda = 1$ (compressible case as the Poisson ratio $\nu = 0.25$) and in Table 3 for $\lambda = 1000$ (nearly incompressible case as $\nu \approx 0.4995$), respectively. These results are obtained using the DIRK(2,3) scheme for $k = 1$ and the DIRK(3,4) scheme for $k = 2$, and a fixed ratio $h/\Delta t = 4$. We observe that the approximate displacement $\boldsymbol{u}_h$, velocity $\boldsymbol{v}_h$, and stress $\boldsymbol{\sigma}_h = \mu(\mathbf{H}_h + \mathbf{H}_h^T) + \lambda/(\lambda+\mu)p_h\mathbf{I}$ converge with the optimal order $k+1$ for $k = 1$ and $k = 2$, even for the nearly incompressible case. Optimal convergence of the stress is an important advantage of the HDG method because many quantities of engineering interest are derived from the stress. Furthermore, we observe that both the postprocessed displacement u_h^* and postprocessed velocity v_h^* converge with order $k+2$ for $k = 1$ and $k = 2$, which are one order higher than the original approximations. Since the postprocessed quantities are inexpensive to compute, the HDG method provides better convergence and accuracy than existing DG methods for the numerical solution of wave propagation in linear elastodynamics.

3.6.2. Comparison with the continuous Galerkin–Newmark method

We consider the previous example to assess the performance of the HDG–DIRK(q,p) method and the CG-N($\beta = 1/4$) method. As before we take $h/\Delta t = 4q$ for the CG-N($\beta = 1/4$) method to guarantee that both the HDG–DIRK(q,p) method and the CG-N($\beta = 1/4$) method have exactly the same total number of linear system solves. Moreover, we compare the postprocessed solution of the HDG–DIRK(q,p) method with the approximate solution of the CG-N($\beta = 1/4$) method.

Fig. 9 shows the L^2 errors in $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{\sigma}$ as a function of the global degrees of freedom for $\lambda = 1$. It is clear that the HDG–DIRK(q,p) method outperforms the CG-N($\beta = 1/4$) method for both $k = 1$ and $k = 2$. In particular, when the same polynomial degree k is used, the numerical approximations for the HDG–DIRK(q,p) method converge with one order higher than those for the CG-N($\beta = 1/4$) method. As a result, the HDG–DIRK(q,p) method provides better accuracy than the CG-N($\beta = 1/4$) method for the same number of global degrees of freedom.

Fig. 10 shows the results obtained for $\lambda = 1000$. The CG-N($\beta = 1/4$) method fails to converge for $k = 1$, while the convergence rates remain optimal for the HDG–DIRK(q,p) method. The failure of the CG-N($\beta = 1/4$) method for $k = 1$ is due to the well-known volumetric locking phenomenon which may occur for some numerical schemes when the Poisson ratio ν is very close to 0.5. For the standard CG method, the volumetric locking can be remedied by introducing the pressure variable and using either reduced integration or mixed finite element spaces such as the Taylor–Hood elements. Despite the fact that equal-degree polynomials are used to represent all the approximate variables the HDG method does not suffer from the volumetric locking phenomenon. Although the CG-N($\beta = 1/4$) method converges better for $k = 2$, its solution is significantly less accurate than that of the HDG–DIRK(q,p) method for the same number of global degrees of freedom.

3.6.3. Elastic waves in a semi-infinite medium

We consider the propagation of elastic waves in half space under an impulsive loading applied at the top surface. In order to solve this problem we consider a bounded domain $\Omega = (0,0) \times (1,0.5)$ and apply the first-order absorbing boundary condition on the left, right, and bottom sides of the domain. The impulse loading applied at the top surface is given by

$$g_N(x,y,t) = \begin{cases} (0,-30) & \text{if } x \in [0.375, 0.425] \cup [0.575, 0.625], \quad y = 0.5, t \leqslant 0.005, \\ (0,0) & \text{otherwise}, \end{cases}$$

The medium is isotropic and homogeneous with material constants $\mu = 1$ kN/m^2, $\lambda = 2$ kN/m^2, and $\rho = 1$ kg/m^3, so that the shear wave speed is $c_s = \sqrt{\mu/\rho} = 1$ m/s and the pressure wave speed is $c_p = \sqrt{(\lambda + 2\mu)/\rho} = 2$ m/s. Fig. 11 shows the geometry and finite element mesh used for the calculation.

In Fig. 12 we present the numerical solution (the magnitude of the postprocessed velocity) of this problem at different times. These results are obtained by using the DIRK(2,3) scheme with the timestep $\Delta t = 0.005$ for time integration and polynomial degree $k = 5$ for spatial discretization. The stabilization tensor $\mathbf{S}$ is set to $\tau\mathbf{I}$ with $\tau = 10$. High-order numerical simulation clearly reveals a rich structure of propagating waves. We can see two cylindrical waves (pressure wave and shear wave) propagating with two different speeds and we can also observe the Rayleigh wave propagating along the surface with a speed slightly less than the speed of shear waves. However, since our absorbing boundary condition is only first-order there are waves which reflect at the absorbing boundary and propagate back into the domain. Nevertheless, this example serves well to demonstrate the capability of HDG methods for solving seismic problems in geophysics.

4. Conclusion

We have presented HDG methods for acoustics and elastodynamics. The HDG methods possess several attractive properties in terms of the reduced degrees of freedom, higher order of convergence, and postprocessing as discussed in the Introduction section and demonstrated by several numerical experiments. Unlike many other DG methods, the HDG methods are fully implicit and yet computationally attractive in the sense that only the approximate trace of the velocity has to be solved at every timestep. Moreover, the approximate stresses, postprocessed displacement and velocity converge faster than the same quantities obtained by using other DG methods known in the literature. Although the HDG method has more global degrees of freedom for the same mesh and polynomial degree, it produces more accurate approximations than the standard CG method for the same number of global degrees of freedom.

Acknowledgements

J. Peraire and N.C. Nguyen would like to acknowledge the Singapore–MIT Alliance and the Air Force Office of Scientific Research under the MURI project on Biologically Inspired Flight and the AFOSR Grant No. FA9550-08-1-0350 for partially supporting this work. B. Cockburn would like to acknowledge the National Science Foundation for partially supporting this work through Grant No. DMS-0712955.

References

[1] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff ODEs, SIAM J. Numer. Anal. 14 (1977) 1006–1021.
[2] I. Babuška, M. Zlámal, Nonconforming elements in the finite element method with penalty, SIAM J. Numer. Anal. 10 (5) (1973) 863–875.
[3] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-stokes equations, J. Comput. Phys. 131 (2) (1997) 267–279.
[4] C. Baumann, J. Oden, A discontinuous hp finite element method for convection–diffusion problems, Comput. Methods Appl. Mech. Eng. 175 (1999) 311–341.
[5] E. Bécache, P. Joly, C. Tsogka, An analysis of new mixed finite elements for the approximation of wave propagation problems, SIAM J. Numer. Anal. 37 (4) (2000) 1053–1084.
[6] F. Brezzi Jr., J. Douglas, L.D. Marini, Variable degree mixed methods for second order elliptic problems, Mat. Apl. Comput. 4 (1985) 19–34.
[7] E.T. Chung, B. Engquist, Optimal discontinuous Galerkin methods for wave propagation, SIAM J. Numer. Anal. 44 (5) (2006) 2131–2158.
[8] E.T. Chung, B. Engquist, Optimal discontinuous Galerkin methods for the acoustic wave equation in higher dimensions, SIAM J. Numer. Anal. 47 (5) (2009) 3820–3848.
[9] B. Cockburn, B. Dong, J. Guzmán, A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems, Math. Comput. 77 (2008) 1887–1916.
[10] B. Cockburn, B. Dong, J. Guzmán, A hybridizable and superconvergent discontinuous Galerkin method for Biharmonic problems, J. Sci. Comput. 40 (1–3) (2009) 141–187.
[11] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, R. Sacco, A hybridizable discontinuous Galerkin method for steady-state convection–diffusion–reaction problems, SIAM J. Sci. Comput. 31 (5) (2009) 3827–3846.
[12] B. Cockburn, J. Gopalakrishnan, The derivation of hybridizable discontinuous Galerkin methods for Stokes flow, SIAM J. Numer. Anal. 47 (2009) 1092–1125.
[13] 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.
[14] B. Cockburn, J. Gopalakrishnan, N.C. Nguyen, J. Peraire, F.-J. Sayas, Analysis of HDGmethods for Stokes flow, Math. Comput. 80 (2011) 723–760.
[15] B. Cockburn, J. Gopalakrishnan, F.-J. Sayas, A projection-based error analysis of HDG methods, Math. Comput. 79 (2010) 1351–1367.
[16] B. Cockburn, J. Guzmán, H. Wang, Superconvergent discontinuous Galerkin methods for second-order elliptic problems, Math. Comput. 78 (2009) 1–24.
[17] B. Cockburn, N.C. Nguyen, J. Peraire, A comparison of HDG methods for Stokes flow, J. Sci. Comput. 45 (2010) 215–237.
[18] B. Cockburn, C.W. Shu, The local discontinuous Galerkin method for convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.

[19] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.
[20] G. Cohen, P. Joly, J.E. Roberts, N. Tordjman, Higher order triangular finite elements with mass lumping for the wave equation, SIAM J. Numer. Anal. 38 (6) (2001) 2047–2078.
[21] G.J. Cooper, A. Sayfy, Semiexplicit A-stable Runge–Kutta methods, Math. Comput. 33 (1979) 541–556.
[22] M. Crouzeix, Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge-Kutta, Ph.D. Thesis, Université de Paris VI, 1975.
[23] L. Demkowicz, J. Gopalakrishnan, A class of discontinuous Petrov–Galerkin methods. Part II: optimal test functions, Numer. Methods Part. Differ. Equat. 27 (2011) 70–105.
[24] L. Demkowicz, J. Gopalakrishnan, A class of discontinuous Petrov–Galerkin methods. Part I: the transport equation, Comput. Methods Appl. Mech. Eng. 199 (2010) 1558–1572.
[25] J. Douglas, T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in: Computing Methods in Applied Sciences, Lecture Notes in Physics, vol. 58, Springer, Berlin, 1976, pp. 207–216.
[26] B. Engquist, A. Majda, Absorbing boundary conditions for the numerical simulation of waves, Math. Comput. 31 (1977) 629–651.
[27] T. Geveci, On the application of mixed finite element methods to the wave equations, RAIRO Model. Math. Anal. Numer. 22 (1988) 243–250.
[28] M.J. Grote, A. Schneebeli, D. Schötzau, Discontinuous Galerkin finite element method for the wave equation, SIAM J. Numer. Anal. 44 (6) (2006) 2408–2431.
[29] T. Ha-Duong, P. Joly, On the stability analysis of boundary conditions for the wave equation by energy methods. Part I: the homogeneous case, Math. Comput. 62 (1994) 539–563.
[30] J.S. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids I. Time–domain solution of Maxwell's equations, J. Comput. Phys. 181 (1) (2002) 186–221.
[31] Thomas J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, 1987.
[32] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method the incompressible Navier–Stokes equations, J. Comput. Phys. 230 (2011) 1147–1170.
[33] N.C. Nguyen, J. Peraire, B. Cockburn, Hybridizable discontinuous Galerkin methods, in: Spectral and High Order Methods for Partial Differential Equations, Lecture Notes in Computational Science and Engineering, vol. 76, 2011, pp. 64–84.
[34] 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.
[35] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations, J. Comput. Phys. 228 (2009) 8841–8855.
[36] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for Stokes flow, Comput. Methods Appl. Mech. Eng. 199 (2010) 582–597.
[37] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations (AIAA Paper 2010-362), in: Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, Florida, Janury 2010.
[38] J. Peraire, N.C. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations (AIAA Paper 2010-363), in: Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, Florida, Janury 2010.
[39] J. Peraire, P.-O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput. 30 (4) (2008) 1806–1824.
[40] P.O. Persson, J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations, SIAM J. Sci. Comput. 30 (6) (2008) 2709–2733.
[41] P.A. Raviart, J.M. Thomas, A mixed finite element method for second-order elliptic problems, in: I. Galligani, E. Magenes (Eds.), Mathematical Aspects of the Finite Element Method, Springer, New York, 1977, pp. 292–315.
[42] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[43] S.-C. Soon, B. Cockburn, H.K. Stolarski, A hybridizable discontinuous Galerkin method for linear elasticity, Int. J. Numer. Methods Eng. 80 (8) (2009) 1058–1092.
[44] R. Stenberg, A family of mixed finite elements for the elasticity problem, Numer. Math. 53 (1988) 513–538.