Received: 12 November 2018 | Revised: 13 October 2019 | Accepted: 22 February 2020

DOI: 10.1002/nme.6343

RESEARCH ARTICLE

WILEY

High-order cut discontinuous Galerkin methods with local time stepping for acoustics

Svenja Schoeder[1] | Simon Sticko[2] | Gunilla Kreiss[3] | Martin Kronbichler[1]

[1]Institute for Computational Mechanics, Technical University of Munich, Garching, Germany

[2]Department of Mathematics and Mathematical Statistics, Umeå University, Umeå, Sweden

[3]Division of Scientific Computing, Department of Information Technology, Uppsala University, Uppsala, Sweden

Correspondence
Svenja Schoeder, Institute for Computational Mechanics, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany.
Email: schoeder@lnm.mw.tum.de

Summary

We propose a method to solve the acoustic wave equation on an immersed domain using the hybridizable discontinuous Galerkin method for spatial discretization and the arbitrary derivative method with local time stepping (LTS) for time integration. The method is based on a cut finite element approach of high order and uses level set functions to describe curved immersed interfaces. We study under which conditions and to what extent small time step sizes balance cut instabilities, which are present especially for high-order spatial discretizations. This is done by analyzing eigenvalues and critical time steps for representative cuts. If small time steps cannot prevent cut instabilities, stabilization by means of cell agglomeration is applied and its effects are analyzed in combination with local time step sizes. Based on two examples with general cuts, performance gains of the LTS over the global time stepping are evaluated. We find that LTS combined with cell agglomeration is most robust and efficient.

KEYWORDS

acoustic wave equation, arbitrary derivative time integration, cell-merging, cut finite element method, hybridizable discontinuous Galerkin, local time stepping

1 | INTRODUCTION

The discontinuous Galerkin (DG) method is a favorable numerical tool to simulate wave propagation (see, eg, References 1 and 2 for a general overview). Although DG methods employ more degrees of freedom compared to continuous Galerkin (CG) methods on the same mesh, they have been shown to outperform CG in several test cases, especially for higher orders of accuracy.[3,4] In addition, DG combines low dispersion errors over a wide range of wave numbers with dissipation on the underresolved scales. This is highly attractive for hyperbolic problems because any wave number is either transported accurately or annihilated.[2,5] Just as continuous finite elements, DG relies not only on the discretization of the solution but also on the discretization of the physical domain on which the solution of the transport problem is sought.

In this article, we investigate some basic properties, namely, accuracy, stability, and time-step restrictions, of an immersed approach. The physical domain is not defined by the mesh, but a background mesh is created and the physical domain is defined implicitly, for example, by a level set function. The correct consideration of the physical domain is achieved by elementwise integration on the background mesh but only considering the regions overlapping with the physical domain (see Reference 6 for an overview on immersed methods and cut finite elements). Immersed methods are motivated by the fact that generating a high-quality conforming mesh, where interfaces and boundaries conform with element edges, can be challenging and computationally expensive, especially in 3D and for high-order methods (HHOs).

The argument is even stronger if, for example, the problem under consideration includes time-dependent geometric features. A conforming approach requires either distorting the mesh or remeshing. The first option may destroy the mesh quality, while the second involves repeating a potentially expensive procedure. A similar situation arises when solving an inverse problem with unknown geometric features. Immersed methods were successfully combined with CG methods as shown in References 7 and 8 for the Poisson equation, in Reference 9 for the Stokes problem, and in Reference 10 for the acoustic wave equation. In Reference 11, cut elements for the acoustic wave equation of higher order are presented.

A problem accompanying cut elements and immersed methods is an ill-conditioning of the finite element discretization in the presence of small cuts. For explicit time integration, this leads to a severe time step limitation or even instability. The ill-conditioning in both the mass and stiffness matrices is caused by basis functions that only have a small support in the physical domain. To handle these issues, a vast variety of stabilization techniques has been presented in the literature. One approach is to use problem-tailored preconditioners to counteract cut-induced ill-conditioning.[12,13] However, this does not solve the problem of severe time step restrictions when using explicit time stepping. A related alternative to preconditioning is to rescale basis functions with small support, as is done for DG in Reference 14. References 15 and 16 considered an unstabilized hybridizable DG discretization in the context of cut elements. We note that combinations of immersed methods and hybridizable discontinuous Galerkin (HDG) are an active field of research. Related are also so called hybrid HHO, which can be seen as a generalization of HDG methods. Cut-HHO methods have been discussed in References 17 and 18. Another approach is to penalize jumps in the derivatives of the solution between elements by a ghost penalty.[8-10,19] This bounds the solution in a cut element in terms of the well-defined solution on the neighbor, which limits growth from one element to the next. This stabilization was discussed for DG methods in particular in Reference 20, together with related forms of ghost penalty stabilization. Basis functions with small support can also be dropped completely by replacing them by an extension of the basis functions of a neighboring element. This approach is known as cell agglomeration or cell merging and is presented in Reference 21-26 in combination with higher order finite element and DG methods for cut problems applied to elliptic problems and flow equations. The cell agglomeration can be thought of as a strong enforcement of the continuity of the solution field to larger neighboring elements, while penalizing jumps in derivatives represent a weak enforcement of the continuity.

Numerical methods for the acoustic wave equation, especially in combination with explicit time stepping, rely on the Courant-Friedrichs-Lewy (CFL) condition for stability. This condition relates characteristic sizes for temporal and spatial discretization as well as the wave speed:[27] the smaller the element size and the faster the wave, the smaller the required time step. For meshes with varying element sizes or varying wave speeds, global time stepping schemes have to use the smallest, that is, the most critical, time step size. As a consequence, most elements are operated on a time step size smaller than necessary and computational overhead is introduced. Local time stepping (LTS) schemes can be utilized to advance each element of the discretization with its optimal time step. In the context of immersed methods, a natural approach is to use a small time step for cut elements with small support in the physical domain, thereby counteracting cut instabilities. When combining the finite volume method with explicit time stepping, the so-called h-box method can be utilized,[28] where fluxes are evaluated on reference grid cell lengths rather than on small cut faces. For the h-box method, second-order accuracy has been achieved.[29] Another approach to circumvent time step restrictions is the so-called implicit-explicit (IMEX) time integration. Here, the major part of the elements is integrated explicitly, whereas the critical elements are integrated implicitly.[30,31] This allows for larger time steps than in a purely explicit global time stepping scheme. Large time steps require more iterations or stronger preconditioners for solving the implicit system and hence the computational gains are limited, as pointed out for the acoustic wave equation in Reference 32.

In this work, we propose an HDG method using LTS based on an arbitrary derivative (ADER) time integration[33-35] to solve the acoustic wave equation on an immersed domain. The method is designed to be compatible with efficient explicit time integrators based on evaluation of the right-hand side with sum factorization techniques in the bulk of the domain, according to the methods described in References 32,36-39. The implementation of HDG allows for high polynomial degrees of the shape functions, which are especially critical in the context of cut instabilities. A first result is that the usage of LTS with small time steps cannot alone stabilize all kinds of cut instabilities. A second contribution is an investigation of cell agglomeration, which can complement the scheme in the presence of critical cut scenarios. We propose two variants of agglomeration, which differ in how the auxiliary trace variables defined on element faces are handled in the agglomeration step. The stabilizing effects of the cell agglomeration are evaluated by an eigenvalue analysis of the semidiscrete system. In addition, critical time step sizes are evaluated for prototype cut scenarios without and with agglomeration. In all our experiments, agglomeration yields stable simulations, albeit sometimes with small time steps. Furthermore, the computational gains of LTS in contrast to global time stepping are determined for a general cut scenario.

The article is structured as follows. We start with the spatial discretization of the acoustic wave equation using the HDG method[32,40] in Section 2.1. Next, the time discretization of the semi-discrete system relying on ADER is presented in Section 2.2. The respective ADER LTS is explained in Section 2.3. High-order integration in cut elements is explained in Section 2.4. Next, the stability of the proposed method is studied in Section 3 performing eigenvalue and critical time step size analyses. In Section 3.1, the cell agglomeration approach is presented for our method and in Section 3.2, the basic and two stabilized schemes are compared based on skew cuts. The convergence of the proposed methods is shown in Section 3.3. In Section 4, two general cut scenarios are studied and the computational performance is analyzed. Section 5 concludes this work.

2 | DISCRETIZATION OF THE WAVE EQUATION

We solve the acoustic wave equation on a two-dimensional domain Ω subject to Dirichlet and Neumann boundary conditions on $\Gamma_\mathrm{D} \cup \Gamma_\mathrm{N} = \Gamma = \partial\Omega$, respectively. Written as a first-order system in terms of the pressure p and the particle velocity $\boldsymbol{v}$, the problem reads

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{\rho}\nabla p = \boldsymbol{0}, \tag{1}$$

$$\frac{\partial p}{\partial t} + c^2\rho\nabla \cdot \boldsymbol{v} = 0. \tag{2}$$

Therein, c and ρ denote the speed of sound and the mass density, respectively. They are temporally constant but generally vary in space. The first equation represents the conservation of linear momentum, while the second represents the conservation of mass. Time differentiation of Equation (2) and insertion of Equation (1) yields the acoustic wave equation in its standard second order formulation. The initial conditions for the velocity and the pressure at time $t = 0$ are given on the domain Ω

$$\boldsymbol{v}(\boldsymbol{x}, t = 0) = \boldsymbol{v}_0(\boldsymbol{x}),$$
$$p(\boldsymbol{x}, t = 0) = p_0(\boldsymbol{x}).$$

The boundary conditions read

$$p = p_\mathrm{D} \ \text{ on } \ \Gamma_\mathrm{D},$$
$$\boldsymbol{v} \cdot \boldsymbol{n} = 0 \ \text{ on } \ \Gamma_\mathrm{N}.$$

For convenience, only the homogeneous Neumann boundary condition is stated here. The extension to the inhomogeneous case is straightforward.

2.1 | Spatial discretization

Consider Figure 1A,B. The physical domain Ω is immersed in a background mesh $\mathcal{T}_\mathrm{B}$, consisting of quadrilaterals K. Let $\mathcal{T}$ denote the mesh consisting of the smallest set of elements in $\mathcal{T}_\mathrm{B}$ covering Ω:

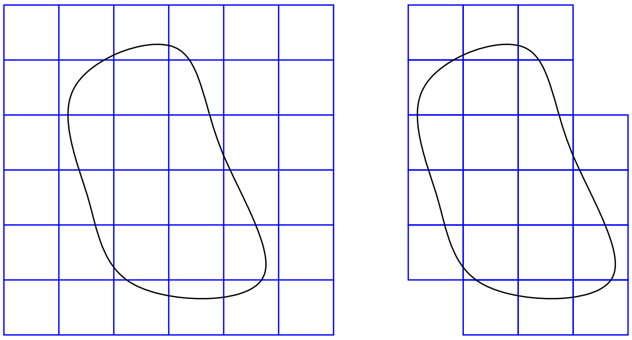
$$\mathcal{T} = \left\{K \in \mathcal{T}_\mathrm{B} \, : \, T \cap \Omega \neq \emptyset\right\}.$$

Furthermore, we let $\Omega_\mathcal{T}$ denote the so-called fictitious domain, consisting of the union of all elements K in the mesh $\mathcal{T}$:

$$\Omega_\mathcal{T} = \bigcup_{K\in\mathcal{T}} K.$$

Under the assumption that the mesh is sufficiently fine, each cell and face in the mesh is cut at most once by the domain's boundary Γ.

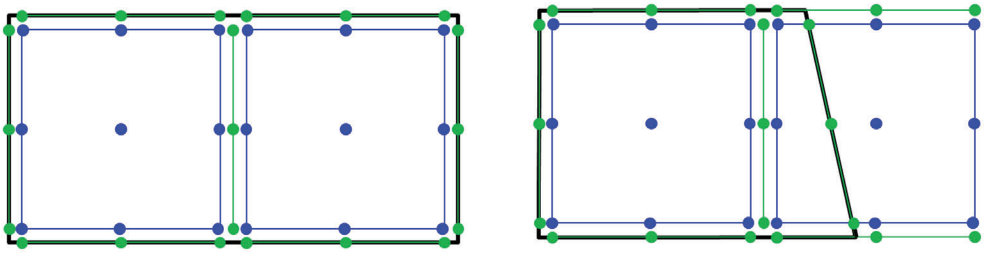
For discretization of the solution variables, pressure p, and velocity $\boldsymbol{v}$, the HDG method is used. Just as in conventional DG approaches, the solution variables are approximated by functions p_h and $\boldsymbol{v}_h$, consisting of piecewise polynomials with



(A) The domain Ω immersed in a background mesh $\mathcal{T}_\mathrm{B}$.

(B) Mesh $\mathcal{T}$ and fictitious domain $\Omega_\mathcal{T}$.

FIGURE 1 Discretization of the physical domain Ω. A, The domain immersed in a background mesh $\mathcal{T}_\mathcal{B}$. B, Mesh $\mathcal{T}$ and fictitious domain $\Omega_\mathcal{T}$ [Colour figure can be viewed at wileyonlinelibrary.com]



(A) Standard HDG degrees of freedom

(B) HDG degrees of freedom for the immersed approach

FIGURE 2 Degrees of freedom for standard HDG and for the immersed approach: the physical domain is given by the black line. Blue dots are element-wise defined degrees of freedom for velocity and pressure, which are discontinuous between elements. Green dots are face degrees of freedom along the element faces. They are as well defined along the face cutting through an element. A, Standard HDG degrees of freedom. B, HDG degrees of freedom for the immersed approach

discontinuities between elements

$$\boldsymbol{V}_h = \left\{ \boldsymbol{v}_h \in (L_2(\Omega_\mathcal{T}))^d \, : \, \boldsymbol{v}_h|_K \in (\mathcal{P}_k(K))^d, \forall K \in \mathcal{T} \right\}, \tag{3}$$

$$P_h = \{ p_h \in L_2(\Omega_\mathcal{T}) \, : \, p_h|_K \in \mathcal{P}_k(K), \forall K \in \mathcal{T} \}, \tag{4}$$

with $d = 2$ for the two-dimensional setup considered here. The polynomial degree is k and $\mathcal{P}_k$ are nodal Lagrange polynomials of tensor product form interpolated in the Gauss-Lobatto points.

HDG additionally introduces the trace of the pressure λ on the faces of elements as an auxiliary variable. Its approximation λ_h is found in a space that is polynomial on faces and discontinuous between faces

$$L_h = \{ \lambda_h \in L_2(\mathcal{F}) \, : \, \lambda_h|_F \in \mathcal{P}_k(F), \forall F \in \mathcal{F} \}, \tag{5}$$

with $\mathcal{F}$ being the union of all element faces F in $\mathcal{T}$. Here, $\mathcal{F}$ consists of three subsets:

$$\mathcal{F} = \mathcal{F}^\mathrm{i} \cup \mathcal{F}^\mathrm{b} \cup \mathcal{F}^\Gamma. \tag{6}$$

In Equation (6), $\mathcal{F}^\mathrm{i}$ are the inner faces and $\mathcal{F}^\mathrm{b}$ are the boundary faces of the mesh. In addition, $\mathcal{F}^\Gamma$ denotes the parts of the immersed boundary that falls within the intersected elements:

$$\mathcal{F}^\Gamma = \{ \Gamma \cap K \, : \, K \in \mathcal{T} \}, \tag{7}$$

which we shall refer to as immersed boundary faces. Figure 2 visualizes the typical HDG degrees of freedom on a standard mesh and for an immersed approach. In contrast to the pressure and velocity field, the trace field is single-valued between elements.

To derive an HDG discretization, the first step is to multiply Equations (1) and (2) with test functions and integrate by parts. Then, the pressure in the boundary integrals is replaced by the new variable λ_h for the definition of the numerical flux. Also, a stabilization term is introduced when evaluating the velocity in boundary integrals. Continuity of the normal component of the velocity over element boundaries is enforced to close the problem. Dirichlet boundary conditions are introduced by an L_2 projection P of the prescribed Dirichlet values p_{D} onto the trace field, that is, $\lambda_h \in L_h(p_{\mathrm{D}})$ with $L_h(p_{\mathrm{D}}) = \{\lambda_h \in L_h : \lambda_h = \mathsf{P}p_{\mathrm{D}} \text{ on } \Gamma_{\mathrm{D}}\}$. For a detailed derivation of HDG for the acoustic wave equation, the reader is referred to the publications.[32,40] Here, only the resulting semidiscrete scheme is presented for the problem at hand: Find $(\boldsymbol{v}_h, p_h, \lambda_h) \in \boldsymbol{V}_h \times P_h \times L_h(p_{\mathrm{D}})$ such that for all $(\boldsymbol{w}, q, \mu) \in \boldsymbol{V}_h \times P_h \times L_h(0)$

$$\left(\frac{\partial \boldsymbol{v}_h}{\partial t}, \boldsymbol{w}\right)_{\mathcal{T}\cap\Omega} - \left(\frac{1}{\rho}p_h, \nabla\cdot\boldsymbol{w}\right)_{\mathcal{T}\cap\Omega} + \left\langle \frac{1}{\rho}\lambda_h, \boldsymbol{w}\cdot\boldsymbol{n}\right\rangle_{\mathcal{G}} = 0, \tag{8}$$

$$\left(\frac{\partial p_h}{\partial t}, q\right)_{\mathcal{T}\cap\Omega} - \left(c^2\rho\boldsymbol{v}_h, \nabla q\right)_{\mathcal{T}\cap\Omega} + \left\langle c^2\rho\boldsymbol{v}_h\cdot\boldsymbol{n} + c^2\rho\tau(p_h - \lambda_h), q\right\rangle_{\mathcal{G}} = 0, \tag{9}$$

$$\langle \boldsymbol{v}_h\cdot\boldsymbol{n}, \mu\rangle_{\mathcal{G}} + \langle \tau(p_h - \lambda_h), \mu\rangle_{\mathcal{G}} = 0. \tag{10}$$

The abbreviations $(\cdot,\cdot)_X$ and $\langle\cdot,\cdot\rangle_Y$ denote the L_2 inner products over the two- and one-dimensional domains X and Y, respectively. In Equations (9) and (10), τ is an HDG-specific stabilization parameter and is typically chosen as $\tau = 1/\rho c$. Note that the boundary integrals are taken over the set $\mathcal{G}$, which we define as

$$\mathcal{G} = \{\partial(K\cap\Omega) : K \in \mathcal{T}\}. \tag{11}$$

The definition in Equation (11) should be interpreted in the sense that interior faces are counted twice. That is, for $K_1, K_2 \in \mathcal{T}$ sharing a face $F = K_1 \cap K_2$, $F\cap\Omega$ appears twice in $\mathcal{G}$. The homogeneous Neumann boundary condition is implicitly enforced by Equation (10). The integration on the intersected elements is carried out as presented in Reference 41. This means that we integrate over Ω rather than $\Omega_{\mathcal{T}}$, see Section 2.4.

The space-discrete, time-continuous matrix form reads

$$\begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} + \begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} + \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \boldsymbol{\Lambda} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{12}$$

$$\mathbb{I}\boldsymbol{V} + \mathbb{J}\boldsymbol{P} + \mathbb{G}\boldsymbol{\Lambda} = 0, \tag{13}$$

with the vectors $\boldsymbol{V}, \boldsymbol{P}$, and $\boldsymbol{\Lambda}$ containing the degrees of freedom of $\boldsymbol{v}_h, p_h$, and λ_h. Note that Equations (12) and (13) can be understood globally with matrices and vectors acting on all degrees of freedom or as elementwise equations due to the discontinuous nature of the ansatz spaces.

2.2 | Temporal discretization

ADER explicit time integration is used to propagate the system (12)-(13) in time. ADER is an explicit time integration scheme of arbitrary high order, which has been successfully applied to a large variety of hyperbolic problems in combination with DG and recently also HDG.[33-35,37,42-44]

First, the trace variable $\boldsymbol{\Lambda}$ is eliminated from Equation (12) by Equation (13). Next, time integration from t_i to t_{i+1} yields

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \underbrace{\left(\begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} - \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix} \right)}_{\mathbb{K}} \int_{t_i}^{t_{i+1}} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} \mathrm{d}t. \tag{14}$$

The matrix $\mathbb{K}$ is introduced to abbreviate the expression in the brackets and we shall refer to it as the stiffness matrix.

The main steps to derive an approximation to the time integral over $\boldsymbol{V}$ and $\boldsymbol{P}$ using ADER are to express the solution fields for velocity and pressure as a truncated Taylor expansion in time starting from the last known state. Then, the wave equation is used to replace time derivatives by space derivatives and the solution fields are projected onto the degrees of freedom. Finally, the time integral is easily evaluated since the only remaining time dependency is on time itself but the solution variables are evaluated at the last known state. Hence, the sought integral from Equation (14) can be given as

$$\int_{t_i}^{t_{i+1}} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} \mathrm{d}t = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \sum_{j=0}^{k+1} \frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!} (-1)^j \int_K \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}. \tag{15}$$

Here, k is the polynomial degree of the ansatz functions and the Taylor series is truncated after the $(k+1)$th term in order to get the same accuracy in time as in space. The matrix $\mathbb{N}$ contains the ansatz functions in matrix form to relate solution fields and values of the degrees of freedom

$$\begin{bmatrix} \boldsymbol{v}_h \\ p_h \end{bmatrix} = \mathbb{N} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix}.$$

The operator $\mathbb{S}$ is defined as

$$\mathbb{S} = \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix},$$

and represents the spatial derivatives as in the acoustic wave equation written as a first-order system, that is,

$$\frac{\partial}{\partial t} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = -\mathbb{S} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix},$$

is the same as the wave equation written as a first order system in Equations (1) and (2).

As already mentioned in Section 2.1 for Equations (12) and (13), Equations (14) and (15) can also be understood as global equations or as elementwise equations, respectively, because of the discontinuity of the ansatz spaces for the pressure, velocity, and trace fields.

Inserting Equation (15) into (14) results in an explicit time stepping scheme with HDG for the spatial discretization and explicit ADER time integration for the temporal discretization yielding order $(k+1)$ accurate results in space and time. For a detailed derivation of ADER HDG for the acoustic wave equation, see Reference 35.

2.3 | Local time stepping

ADER can easily be extended to LTS as described in References 45 and 35. Because of the discontinuous solution spaces of velocity and pressure, the only exchange of information between elements is carried out via the trace variable on the element faces and hence the application of the matrices

$$\begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix},$$

in Equation (14). For elements working on different time step sizes, it is necessary to ensure the correct information transport as explained in the following.

First, time step sizes are assigned to elements according to a characteristic measure of their size h. For uncut elements, h is the minimal edge length. For cut elements, the shortest face length is used as characteristic element length h. Optimal time step sizes are determined for all elements K according to the CFL stability criterion using a prescribed Courant number Cr

$$\Delta t_K = Cr \frac{h_K}{ck},$$

where a dependence on k is taken into account. For ADER time integration, $Cr = 0.1$ is generally a good choice.[35]

Quantification of time step restrictions due to small cuts is enabled by a reference Courant number

$$Cr_{\text{ref}} = \frac{ck\Delta t_K}{h_{\text{inner}}}, \tag{16}$$

which is calculated with the time step of the considered element but with reference to the element size h_{inner} of a standard uncut inner element. Reference Courant number Cr_{ref} and Courant number Cr can be converted using

$$Cr_{\text{ref}} = Cr\frac{h_K}{h_{\text{inner}}}.$$

For computational efficiency, the time step sizes of elements are cast to multiples of the minimal occurring time step size Δt_{min}. Elements with the same time step size Δt_C make up a cluster C. The time step sizes of the clusters fulfill

$$\Delta t_C = ((C-1)\cdot M + 1)\cdot \Delta t_{\text{min}}, \tag{17}$$

where C denotes the associated cluster and M is a user defined input on the allowed difference between clusters. Also, the cluster distribution is smoothed such that the time step sizes of neighboring elements differ by at most $M \cdot \Delta t_{\text{min}}$. Generally, too high a difference in the time step size between neighboring elements can induce instabilities in the temporal discretization.

Figure 3 explains the order of actions for the LTS on a one-dimensional example consisting of five elements. Three small elements are assigned to cluster $C = 1$ with time step size $\Delta t_{C=1}$ and two large elements are assigned to cluster $C = 2$ with time step size $\Delta t_{C=2} = 2 \cdot \Delta t_{C=1}$. To update from $t = 0$ to $t = \Delta t_{C=2}$, the elements of cluster $C = 1$ perform two time steps, while the elements of cluster $C = 2$ only perform one time step. Generally, clusters with a smaller time step advance first, which is why the first step is the evaluation of the elements of cluster $C = 1$ and the flux evaluation between elements of the same cluster. For the evaluation of the flux between elements K^3 and K^4, that is, at the cluster boundary, element K^4 has to evaluate the flux contribution for the time interval $t \in [0, \Delta t_{C=1}]$. It is used to complete the update of element K^3 but it is also stored for the subsequent update of K^4 in a so called flux memory variable $\mathcal{M}$. The second step is similar to the first one: all elements of $C = 1$ advance in time. The flux contribution from element K^4 to K^3 for the time interval $t \in [\Delta t_{C=1}, 2 \cdot \Delta t_{C=1}]$ is evaluated and summed into the flux memory variable $\mathcal{M}^{K^4}$ of K^4. In the third step, the elements of cluster $C = 2$ advance from $t = 0$ to $t = \Delta t_{C=2}$ by evaluation of themselves and under consideration of the memorized flux contribution from element K^3. After that, all elements are on the same time level $t = \Delta t_{C=2}$ and the next global time step from $t = \Delta t_{C=2}$ to $t = 2 \cdot \Delta t_{C=2}$ is performed analogously.

Generally, the elements of a cluster are updated if they fulfill the update criterion

$$t_C + \Delta t_C \leq \min(t_N + \Delta t_N) \quad \forall \text{ neighboring clusters } N,$$

stating that their progress would be shorter than or equal to that of all their neighbors. The temporal interval required for flux evaluation between neighboring elements of different clusters is

$$[t_1, t_2] = [\max(t_C, t_N), \min(t_C + \Delta t_C, t_N + \Delta t_N)].$$

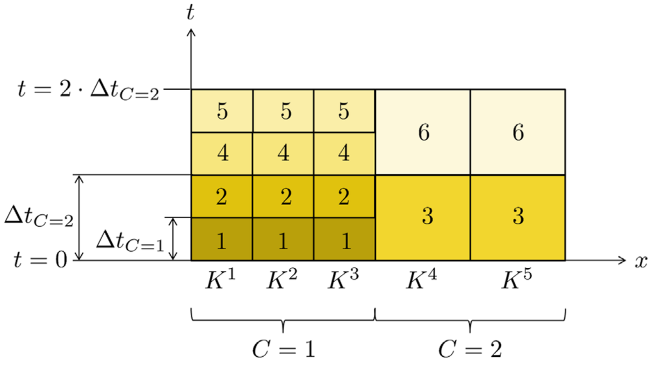


FIGURE 3 Elements with clustered time step sizes and their advancement in time with LTS [Colour figure can be viewed at wileyonlinelibrary.com]

All the elements of the cluster to be updated evaluate the time integral of the velocity and pressure degrees of freedom similar to Equation (15) but considering the relevant time interval

$$\boldsymbol{\Psi}^C = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \sum_{j=0}^{k+1} \frac{(t_2 - t_C)^{j+1} - (t_1 - t_C)^{j+1}}{(j+1)!} (-1)^j \int_{K^C} \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix},$$

while the neighboring elements of cluster N evaluate

$$\boldsymbol{\Psi}^N = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \sum_{j=0}^{k+1} \frac{(t_2 - t_N)^{j+1} - (t_1 - t_N)^{j+1}}{(j+1)!} (-1)^j \int_{K^N} \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}.$$

The interior element contributions are evaluated according to

$$\boldsymbol{\Phi}^C = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \sum_{j=0}^{k+1} \frac{(\Delta t_C)^{j+1}}{(j+1)!} (-1)^j \int_{K^C} \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}.$$

The quantities $\boldsymbol{\Psi}^C$ and $\boldsymbol{\Psi}^N$ are then used for the flux contributions when applying $\mathbb{K}$, while the vector $\boldsymbol{\Phi}^C$ is used to evaluate the contributions stemming from the element integrals. Flux contributions are also stored in the flux memory variable $\mathcal{M}^N$ for the subsequent update of neighbors. If the active elements also contain flux contributions from previous updates of neighboring elements, their flux memories must be considered. The update of the element variables is finalized by

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \quad \mathbb{K} \quad \left\{ \boldsymbol{\Psi}^C, \boldsymbol{\Psi}^N, \boldsymbol{\Phi}^C, \mathcal{M}^C \right\}.$$

The notation is symbolic indicating the correct conjunction of flux and internal terms from $\mathbb{K}$ with the different state variables. For a more detailed explanation and a more extensive description of the state variables, the flux memory variables, and the combination of the quantities with $\mathbb{K}$ (see Reference 35).

Spatial and temporal discretization of the acoustic wave equation using HDG and ADER LTS are implemented in C++ with the finite element library `deal.II`.[46]

2.4 | High-order integration on cut elements

In order to realize an immersed approach, it is necessary to integrate over the part of an element or face that is within the physical domain. For a given element, K, or face, F, a quadrature rule over $K \cap \Omega$ or $F \cap \Omega$ is required. Since we consider higher order elements, these quadrature rules need to be of high order and need to take the curvature of the boundaries into account. This is a nontrivial problem and a lot of research has been focused on this topic, see, for example, References 47-50. For the results in this article, we have used the algorithm by Saye.[41] This algorithm creates quadrature rules from a level set function by utilizing the implicit function theorem and one-dimensional root-finding. The immersed quadratures are generated from a one-dimensional quadrature. For the experiments in this article, a 1D Gauss-Legendre quadrature with $p + 1$ quadrature points has been used. An advantageous property of the algorithm[41] is that it guarantees that the constructed quadrature has positive weights.

3 | STABILITY

In order to gain insight into the basic properties of the spatial discretization, the semidiscrete operator is examined. Inserting Equation (13) into (12) results in

$$\begin{aligned} \frac{\partial}{\partial t} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} &= -\begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} - \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} \\ &=: \mathbb{O} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix}, \end{aligned}$$

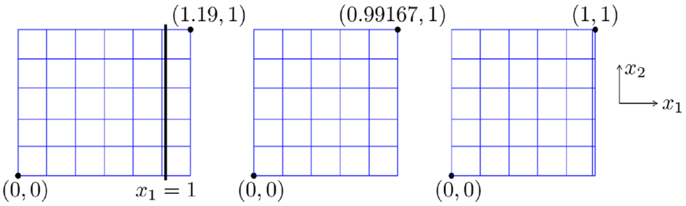
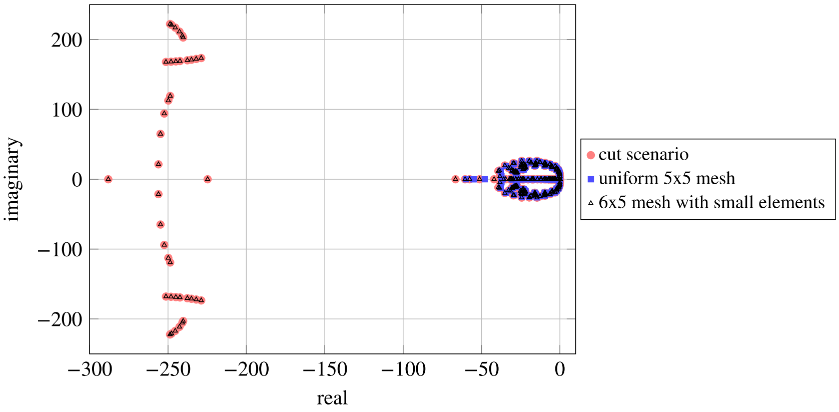


FIGURE 4 Three different setups for an eigenvalue analysis. In the first scenario, the physical boundary cuts through a regular background mesh. In the second setup, a uniform 5×5 mesh is studied. The third test case recreates the first scenario but without an immersed boundary but by one layer of small elements [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 5 Comparison of eigenvalue distributions for the three setups shown in Figure 4 [Colour figure can be viewed at wileyonlinelibrary.com]



with $\mathbb{O}$ abbreviating the spatially discrete operator. The eigenvalues of $\mathbb{O}$ are denoted v_i. They are studied for three different setups as shown in Figure 4. The first setup consists of a two-dimensional physical domain $\Omega = [0, 1] \times [0, 1]$ defined over a background mesh of 6×5 elements on $[0, 1.19] \times [0, 1]$ and a cut at $x_1 = 1$, such that cut elements with shortest face length $h_{\text{cut}} = 8.33 \cdot 10^{-3}$ are created. The second setup consists only of the 5×5 regular elements of the first setup. For comparison, the last setup is a physical domain $\Omega = [0, 1] \times [0, 1]$ on a non-uniform 6×5 mesh recreating mesh characteristics as in the first setup but without cutting. Here and in the following numerical examples, the material parameters are set to $\rho = 1$ and $c = 1$.

The eigenvalues of the semidiscrete operator $\mathbb{O}$ for shape functions of polynomial degree $k = 1$ are calculated for the three setups and are shown in Figure 5. The eigenvalues with real part close to zero represent the physical transport modes, while large negative real parts correspond to the damping of high frequencies. Characteristic of DG,[2] all eigenvalues are in the left complex half plane, which is necessary for stability. The eigenvalues of the discretization on the uniform 5×5 elements mesh are close to the origin with the maximal absolute value of 60. The two other setups approximately replicate these eigenvalues and have additional eigenvalues with absolute values between 200 and 300, which are associated with the degrees of freedom in the cut elements or in the small elements, respectively.

Setting a reasonable time step can be understood as scaling the eigenvalues of the semidiscrete operator $\mathbb{O}$ to fit into the stability region of the time integrator. For the uniform mesh, a larger time step size can be used compared to the two other test cases, because the absolute values are smaller by a factor of ≈ 5. Hence, the critical time step size will scale similarly in dependence on the specific shape of the stability region of the time integrator. Figure 5 also visualizes the motivation for LTS. Usually, the elementwise CFL stability criterion is adduced when motivating LTS, which is of course valid. LTS can also be understood as scaling the most critical eigenvalues more strongly as compared to the less critical eigenvalues. A natural choice for the time step size in the third scenario with one layer of small elements is a small time step fulfilling the CFL stability criterion in the smallest, that is, the most critical elements. In combination with LTS, a larger time step size is used in the larger elements. Comparison of the eigenvalue spectra between the third and the first setups reveals that their characteristics are very similar. Thus, one could follow the same approach for time discretization of the cut scenario, using h for determination of time step sizes in the standard elements and h_{cut} in the cut elements. Thereby, the ill-conditioning introduced by the cut is countered by the small time step size. Counteracting eigenvalues of large absolute values by small time steps works as long as the cut does not create eigenvalues with positive real parts.

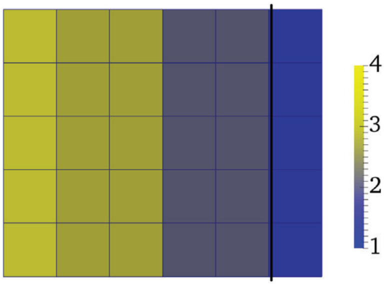


FIGURE 6 Exemplary cluster distributions for the straight cut scenario with $h_{\text{cut}} = 8.3 \cdot 10^{-3}$

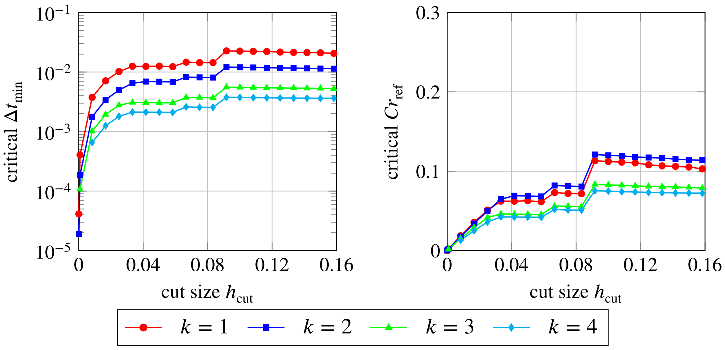


FIGURE 7 Critical time step sizes and corresponding critical reference Courant numbers for the straight cut as shown in the left panel of Figure 4. The uncut mesh size is $h = 0.2$ [Colour figure can be viewed at wileyonlinelibrary.com]

An analysis of the critical time step size for the cut test setup from Figure 4 for different cut sizes h_{cut} and several polynomial degrees is carried out next. The upper right vertex is no longer fixed at $(\hat{x}_1, \hat{x}_2) = (1.19, 1)$ with $h_{\text{cut}} = 8.3 \cdot 10^{-3}$ but set to

$$(\hat{x}_1, \hat{x}_2) = \left(1.2 - \frac{6}{5} h_{\text{cut}}, 1\right),$$

to create cuts of varying size. The analytic solution is given as the vibration of a square membrane in the mth mode

$$p = \sin(m\pi x_1) \sin(m\pi x_2) \cos\left(m\sqrt{2}\pi t\right), \tag{18}$$

$$\boldsymbol{v} = -\frac{1}{\sqrt{2}} \sin\left(m\sqrt{2}\pi t\right) \begin{bmatrix} \cos(m\pi x_1) \sin(m\pi x_2) \\ \sin(m\pi x_1) \cos(m\pi x_2) \end{bmatrix}. \tag{19}$$

The number of membrane modes is set to $m = 1$. For time discretization, the LTS is used with an allowed cluster difference of $M = 1$. The smallest time step Δt_{min} is set in the cut elements. The cluster distribution is then set up automatically as explained in Section 2.3. Figure 6 shows an exemplary cluster distribution for $h_{\text{cut}} = 8.3 \cdot 10^{-3}$. In the spatial discretization, polynomial degrees $k = 1, 2, 3, 4$ are used for the shape functions. The critical time step is determined by finding the largest time step for which the L_2 pressure error norm at $t = 1$ is smaller than 0.5. This tolerance is chosen because it is approximately twice the L_2 pressure norm of the initial field. Note that this indication for instability is somewhat arbitrary and more of a pragmatic investigation of stability rather than a mathematical approach. Figure 7 presents the critical time steps determined in this experiment. The recorded critical time step is the minimal time step Δt_{min} that occurs in the cut elements. In addition, the reference Courant number Cr_{ref} according to the definition (16) is plotted. The critical time step size varies significantly with the cut size h_{cut}. Especially for cut sizes smaller than $h_{\text{cut}} = 0.02$, the critical time step must be over-proportionally decreased. For larger cuts, the curves flatten out except for jumps. The jumps are related to the cluster setup. For small cuts, four clusters are used (see also Figure 6), for larger cuts, the cluster number is reduced corresponding to the jumps. The mesh allows for a maximum of four time step clusters (compare Figure 6), which is taken advantage of for small cuts. For larger cuts, the number of clusters is decreased to two, that is, a small time step only in the cut elements. For small cuts, the reference Courant number depends approximately linearly on the cut size which is expected because it is related to the inner element size and not the cut size. The Courant number is $Cr \approx 0.5$ for the smallest cut and $k = 1$ and decreases with increasing cut to $Cr \in [0.09, 0.15]$ for $h_{\text{cut}} = 0.16$.

TABLE 1 Maximal absolute value and maximal real part of the eigenvalues ν_i of the spatial operator $\mathbb{O}$

	$h_{\text{cut}} = 5.83 \cdot 10^{-3}$		$h_{\text{cut}} = 8.33 \cdot 10^{-3}$	
Degree	$\max_i \lvert\nu_i\rvert$	$\max_i(\text{Re}(\nu_i))$	$\max_i \lvert\nu_i\rvert$	$\max_i(\text{Re}(\nu_i))$
$k = 1$	$4.65 \cdot 10^2$	$1.96 \cdot 10^{-12}$	$3.34 \cdot 10^2$	$3.64 \cdot 10^{-13}$
$k = 2$	$1.05 \cdot 10^3$	$2.58 \cdot 10^{-10}$	$7.91 \cdot 10^2$	$3.50 \cdot 10^{-10}$
$k = 3$	$1.80 \cdot 10^3$	$8.38 \cdot 10^{-6}$	$1.35 \cdot 10^3$	$7.46 \cdot 10^{-7}$
$k = 4$	$2.72 \cdot 10^3$	$6.17 \cdot 10^{-2}$	$2.04 \cdot 10^3$	$3.62 \cdot 10^{-3}$

To understand the origins of the necessity to decrease the time step size as strongly for the small cuts, the maximal absolute $\max_i(\text{Re}(\nu_i))$ as well as the maximal real part $\max_i|\nu_i|$ of the eigenvalues of the spatial operator $\mathbb{O}$ are shown for $h_{\text{cut}} = 5.83 \cdot 10^{-3}$ and $h_{\text{cut}} = 8.33 \cdot 10^{-3}$ in Table 1. The maximal absolute $\max_i|\nu_i|$ already highlighted in the eigenvalue spectra appears to increase for higher polynomial degrees and smaller cut sizes. However, the growth is not orders of magnitude. A moderate decrease of the time step size can balance the high absolute values of the eigenvalues to compress the eigenvalue spectrum into the stability region of the HDG ADER LTS method. A decrease in the time step size, however, cannot balance positive real parts of the eigenvalues. For noncritical cut sizes and moderate polynomial degrees, the maximal real part is zero within a roundoff accuracy but it grows significantly with decreasing cut size and increasing polynomial degree which is traced back to the combination of roundoff and ill-conditioning of element matrices. Note that the inverses are based on LU factorization of the mass matrices $\mathbb{A}, \mathbb{M}$, and $\mathbb{G}$ using the LAPACK functions `dgetrf_` and `dgetrs_` which, while robust for general matrices, are subject to roundoff errors proportional to the condition number times the machine epsilon. Critical time step sizes are still found because the stability criterion is evaluated at $t = 1$ and hence after a finite number of time steps. In the limit of infinitely many time steps, positive real parts of the eigenvalues would always make it impossible to find a stable time step size.

Summarizing the above, it can be stated that innocuous cut scenarios increase the absolute values of the eigenvalues of $\mathbb{O}$ and small time steps or locally small time steps yield an overall stable solution scheme. Furthermore, bad cut scenarios can yield positive eigenvalues of $\mathbb{O}$, which do not appear in case of small uncut elements as in the third scenario in Figure 4. Decreasing the time step size can counteract the absolute instability only in a limited sense. Hence, the question of how to avoid positive eigenvalues naturally arises. In the literature, a vast variety of stabilization schemes is presented, for example, ghost penalty methods,[51] which aim to smoothly extend the solution field from one element to the other in a weak sense. When the continuity of the solution field and its derivatives is strongly enforced, the technique is called cell agglomeration, extending the basis functions from one element into the neighboring element. Cell agglomeration is the approach pursued in this work because it is straight forwardly combined with the HDG ADER LTS method.

Remark: Note that IMEX methods (see, eg, Reference 31) are also well suited to counteract the weak instabilities induced by the high amplitude eigenvalues of the spatial operator $\mathbb{O}$. However, explicit Runge-Kutta time integration has been shown to outperform implicit Runge-Kutta time integration significantly.[32] ADER allows to implement explicit time integration even more efficiently as shown in References 35 and 37. Also, neither LTS nor IMEX can counteract the strong instability induced by eigenvalues with positive real part, since this property is inherent in the spatial discretization and can only be overcome by a suitable spatial stabilization.

3.1 | Cell agglomeration

In Reference 23, a cell agglomeration approach is presented for two-phase flows. Its implementation is extended to simplify the agglomeration and deagglomeration in terms of the utilized data structures in Reference 24. Here, the ideas in References 23 and 24 are carried over to the HDG ADER LTS discretization.

The basic idea of cell agglomeration is to extend basis functions of an uncritical element into a neighboring element that has only small support in the physical domain Ω. In Figure 8, two elements with linear basis functions are depicted. Element K^2 contributes only with a small portion to Ω and is determined to be agglomerated to element K^1. Therefore, the basis functions of element K^1 are extended beyond the standard reference space to allow for their evaluation in the neighboring element. Element K^2 does not define its own degrees of freedom but only contributes to the integrals for the matrix entries associated with the degrees of freedom of element K^1.

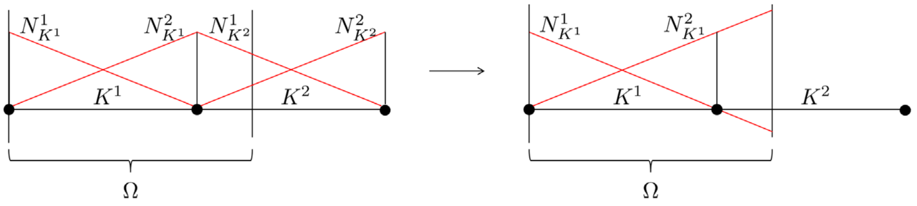


FIGURE 8 Two elements K^1 and K^2 with linear basis functions $N^i_{K^j}$ and the agglomeration approach [Colour figure can be viewed at wileyonlinelibrary.com]

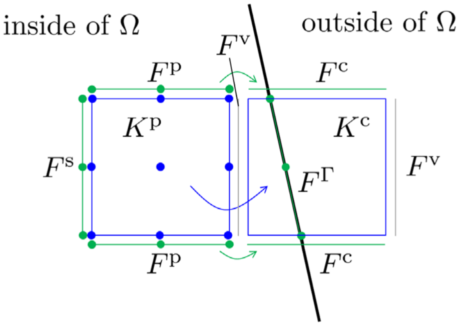


FIGURE 9 Degrees of freedom for HDG with the agglomeration approach and merged trace space in contrast to Figure 2. The arrows indicate extrapolation [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 9 repeats the scenario from Figure 2 for a two-dimensional setup and visualizes the degrees of freedom for the agglomeration approach. For the trace space on the cut faces, we propose two different strategies in Sections 3.1.1 and 3.1.2, respectively. In both strategies, the face along the physical boundary Γ defines its own degrees of freedom while the face between two merged elements is void, as the elementwise degrees of freedom for velocity and pressure are extrapolated and jump terms in DG vanish.

The elements K are divided into standard elements K^{s}, which are not involved into agglomeration, elements K^{c} which are agglomerated to others, that is, child cells, and elements K^{p} which receive contributions from others, that is, parent cells, as shown in Figure 9. The agglomerate is given as a pair of parent and child cell (or children cells) $(K^{\mathrm{p}} \cup K^{\mathrm{c}}) \cap \Omega := K^{\mathrm{a}}$. The sets are defined accordingly, that is,

$$\mathcal{T}^{\mathrm{s}} = \left\{K^{\mathrm{s}} \in \mathcal{T}\right\}, \qquad \mathcal{T}^{\mathrm{a}} = \left\{K^{\mathrm{a}} \in \mathcal{T}\right\}.$$

For the faces, a similar notation is introduced: standard faces are denoted F^{s}, parent faces receiving contributions are F^{p}, and child faces are F^{c}. Faces along the physical boundary defined by the cut interface Γ are denoted F^{Γ} and faces that are dropped, that is, void faces, are denoted F^{v}. If the physical boundary coincides with the boundary of the computational domain and hence with a face of an element, it is denoted a standard boundary face F^{sb}. If two cells are agglomerated, the face in between becomes a void face F^{v} which neither defines degrees of freedom nor contributes to integrals. An agglomerated face is defined as the pair of parent and child face in the physical domain $(F^{\mathrm{p}} \cup F^{\mathrm{c}}) \cap \Omega := F^{\mathrm{a}}$. The decision, which faces should be aggregated, differs for the two proposed stabilized schemes as detailed below. The relevant face sets are

$$\mathcal{F}^{\mathrm{s}} = \left\{F^{\mathrm{s}} \in \mathcal{F}\right\}, \qquad \mathcal{F}^{\mathrm{sb}} = \left\{F^{\mathrm{sb}} \in \mathcal{F}\right\}, \qquad \mathcal{F}^{\mathrm{a}} = \left\{F^{\mathrm{a}} \in \mathcal{F}\right\}.$$

In the present work, an element is merged to a neighbor if its intersection with the domain is smaller than a user defined tolerance (see Reference 24 for a discussion of the parameter determining when a cell should be agglomerated). In our algorithm, we merge to the neighbor which is closest in terms of the center of mass of the cut element. The function spaces for discretization of the solution fields with HDG are modified to

$$\begin{aligned}
\boldsymbol{V}_h^{\mathrm{agg}} &= \left\{\boldsymbol{v}_h \in (L_2(\Omega_{\mathcal{T}}))^d \,:\, \boldsymbol{v}_h|_K \in (\mathcal{P}_k(K))^d \;\forall\, K \in (\mathcal{T}^{\mathrm{s}} \cup \mathcal{T}^{\mathrm{a}})\right\},\\
P_h^{\mathrm{agg}} &= \left\{p_h \in L_2(\Omega_{\mathcal{T}}) \,:\, p_h|_K \in \mathcal{P}_k(K) \;\forall\, K \in (\mathcal{T}^{\mathrm{s}} \cup \mathcal{T}^{\mathrm{a}})\right\},\\
L_h^{\mathrm{agg}} &= \left\{\lambda_h \in L_2(\mathcal{F}) \,:\, \lambda_h|_F \in \mathcal{P}_k(F),\;\forall\, F \in (\mathcal{F}^{\mathrm{s}} \cup \mathcal{F}^{\mathrm{sb}} \cup \mathcal{F}^{\mathrm{a}} \cup \mathcal{F}^{\Gamma})\right\}.
\end{aligned}$$

Just as in the definitions of the function spaces in Equations (3)-(5), the polynomials $\mathcal{P}_k(F)$ are nodal Lagrange polynomials with nodes in the Gauss-Lobatto points. For $K = K^{\mathrm{a}}$, the nodes are the Gauss-Lobatto points in the parent cell K^{p} and the functions are extrapolated beyond the standard reference space. For ease of notation, the following abbreviation is introduced

$$\mathcal{G}^{\mathrm{a}} = \{\partial\left(K^s \cap \Omega\right) \, : \, K^s \in \mathcal{T}^{\mathrm{s}}\} \cup \partial\mathcal{T}^{\mathrm{a}},$$

which contains all inner faces twice (excluding void faces) similarly to $\mathcal{G}$ and also contains the faces along Γ as part of $\partial\mathcal{T}^{\mathrm{a}}$.

The weak problem statement in combination with the agglomeration approach in contrast to Equations (8)-(10) now reads: Find $(\boldsymbol{v}_h, p_h, \lambda_h) \in \boldsymbol{V}_h^{\mathrm{agg}} \times P_h^{\mathrm{agg}} \times L_h^{\mathrm{agg}}(p_{\mathrm{D}})$ such that for all $(\boldsymbol{w}, q, \mu) \in \boldsymbol{V}_h^{\mathrm{agg}} \times P_h^{\mathrm{agg}} \times L_h^{\mathrm{agg}}(0)$

$$\left(\frac{\partial \boldsymbol{v}_h}{\partial t}, \boldsymbol{w}\right)_{\mathcal{T}^{\mathrm{s}}\cup\mathcal{T}^{\mathrm{a}}} - \left(\frac{1}{\rho}p_h, \nabla\cdot\boldsymbol{w}\right)_{\mathcal{T}^{\mathrm{s}}\cup\mathcal{T}^{\mathrm{a}}} + \left\langle \frac{1}{\rho}\lambda_h, \boldsymbol{w}\cdot\boldsymbol{n}\right\rangle_{\mathcal{G}^{\mathrm{a}}} = 0, \tag{20}$$

$$\left(\frac{\partial p_h}{\partial t}, q\right)_{\mathcal{T}^{\mathrm{s}}\cup\mathcal{T}^{\mathrm{a}}} - \left(c^2\rho\boldsymbol{v}_h, \nabla q\right)_{\mathcal{T}^{\mathrm{s}}\cup\mathcal{T}^{\mathrm{a}}} + \left\langle c^2\rho\boldsymbol{v}_h\cdot\boldsymbol{n}, q\right\rangle_{\mathcal{G}^{\mathrm{a}}} + \left\langle c^2\rho\tau(p_h-\lambda_h), q\right\rangle_{\mathcal{G}^{\mathrm{a}}} = 0, \tag{21}$$

$$\langle \boldsymbol{v}_h\cdot\boldsymbol{n}, \mu\rangle_{\mathcal{G}^{\mathrm{a}}} + \langle\tau(p_h-\lambda_h), \mu\rangle_{\mathcal{G}^{\mathrm{a}}} = 0. \tag{22}$$

3.1.1 | Stabilized scheme with merged trace space

In our first variant, the trace space is based on merging all faces adjacent to merged cells. In other words, the set of agglomerated faces $\mathcal{F}^{\mathrm{a}}$ is associated with the surrounding of the agglomerated cells K^{a} which span more than one element. In this setup, if K^p is a parent of K^c, then the faces of K^c are merged to faces of K^p. This restriction forces us to avoid the agglomeration of the type illustrated in Figure 10, where cells are agglomerated diagonally. The reason for this is that the face F^c in Figure 10 cannot be extrapolated from a face of K^p. As illustrated in Figure 11, we allow for an element to be a parent of several neighbors. In rare cases, it can even be parent for a chain of children, as shown in Figure 12. However, we only allow the parent to be at most two elements away (as shown in Figure 12). The construction of an appropriate agglomeration pattern is a nontrivial task. In general, we have no proof that a working agglomeration pattern always exists. Preferably a cell with a small cut is merged with the neighbor that has the largest intersection with the domain. However, since we do not allow for the situation in Figure 10, we need to do some extra work to set up the pattern.

In this variant, the trace function space L_h^{agg} involves Lagrange polynomials in the Gauss-Lobatto points of the parent face $\mathcal{F}^{\mathrm{p}}$, similarly to the definition of the function space for cells. This selection matches the cut cells' function spaces and thus has the advantage of naturally fitting into the HDG concept including implicit system where elimination into λ_h is performed. A disadvantage of this setup is that the trace "mass" matrix $\mathbb{G}$ includes integral contributions from polynomials evaluated outside the reference element, for which a significant increase in the condition number for higher order polynomials is expected. Furthermore, the function space is somewhat restricted in case only the cells on one side of an interior face are merged but not the other.

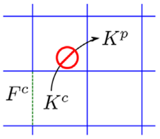


FIGURE 10 Forbidden diagonal agglomeration with the merged trace space [Colour figure can be viewed at wileyonlinelibrary.com]

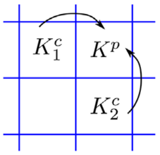


FIGURE 11 Two elements agglomerated with the same parent [Colour figure can be viewed at wileyonlinelibrary.com]

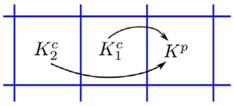


FIGURE 12 An agglomeration in two steps [Colour figure can be viewed at wileyonlinelibrary.com]

3.1.2 | Stabilized scheme with separate trace polynomials

In our second variant, we keep the faces of the original mesh, including the faces with small cuts, and let all interior faces be standard faces $\mathcal{F}^{\mathrm{s}}$ with $\mathcal{F}^{\mathrm{a}} = \emptyset$. The polynomial space $\mathcal{P}_k$ in L_h^{agg} is defined as the Lagrange polynomials in the points of Gaussian quadrature, which ensures an appropriate scaling. This implies $\mu_i(\hat{\boldsymbol{x}}_q) = \delta_{iq}$ for the ith shape function evaluated at the quadrature point $\hat{\boldsymbol{x}}_q$. Such a collocation between polynomials and quadrature points is widely used in spectral elements[52] and has been used for fast evaluation in the context of HDG methods in Reference 53. This simplifies the evaluation of the integrals, as each entry in the matrices $\mathbb{C}, \mathbb{E}, \mathbb{G}, \mathbb{I}, \mathbb{J}$ originates from a single quadrature point due to the δ_{iq} term in the summation over quadrature points. Even though the scaling of $\mathbb{G}$ might be potentially very different in cut faces compared to uncut ones for small cuts, the condition number of the matrix $\mathbb{G}$ is nearly optimal, given that the matrix is diagonal with entries consisting of the weights in Gaussian quadrature times a constant factor involving the product of τ and the length of the cut face. Furthermore, the potentially dangerous effect of $\mathbb{G}^{-1}$ with small cuts disappears when expanded via

$$\begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix},$$

yielding a contribution similar to the cell integral in the child region of an agglomerated cell. In other words, the stabilizing effect is achieved by the merged basis functions on the cells for velocity and pressure.

An alternative interpretation of this strategy is related to a pointwise evaluation of the numerical flux without an explicit definition of a global field λ_h for the trace. This restricts the choice to explicit time integration with HDG in the spirit of Reference 32. On the other hand, the choice is attractive because it avoids the increase in condition numbers for polynomials evaluated outside the reference element for the trace.

3.1.3 | Temporal discretization and time step selection

The temporal discretization of Equations (20)-(22) is analogous to the derivation in Sections 2.2 and 2.3. However, for the calculation of the optimal time step size, several possibilities to define the characteristic element size h of agglomerated cells exist. Thus far, h was determined as the shortest edge length for uncut elements and for cut elements. For an agglomerated cell, it is most intuitive to evaluate the characteristic element size based on the geometry of the agglomerated patch, which is why the edge length h and the shortest edge length of the child cut cell h_{cut} are simply added $h + h_{\mathrm{cut}}$.

The purpose of the cell agglomeration is to prevent positive eigenvalues of the operator $\mathbb{O}$. The eigenvalues are evaluated in analogy to the previously run analysis and Figure 13 shows the spectrum in the same scale as Figure 5 for the merged trace strategy (the separate trace space behaves very similarly in this example and is not shown). Apparently, the spectra of the uniform 5×5 mesh from Figure 5 and of the stabilized scheme resemble strongly. Table 2 shows the results for the maximal absolute and the maximal real part of the eigenvalues for $h_{\mathrm{cut}} = 5.83 \cdot 10^{-3}$ and $h_{\mathrm{cut}} = 8.33 \cdot 10^{-3}$ for various polynomial degrees of the ansatz functions: The largest real parts in the eigenvalues are all zero to roundoff in contrast to the real parts for the unstabilized scheme (compare Table 1) showing that the stabilization fulfills its purpose. Also, the largest eigenvalue in magnitude is smaller by a factor of ≈ 10.

The same analysis as before of how the critical time step size depends on the cut size is carried out using the agglomeration approach and the results are shown in Figure 14. For comparison, the scale is the same as in Figure 7. The time step size is almost completely independent of the cut size. It is interesting to note that the smallest time step size now occurs in the standard elements and a larger time step is chosen in the agglomerated cells.

In the following, the HDG ADER LTS without cell agglomeration but only small local time step sizes will be denoted as the *basic scheme* while additional usage of agglomeration will be indicated as either a *stabilized scheme: merged trace* for the strategy from Section 3.1.1 or a *stabilized scheme: separate trace* with the strategy from Section 3.1.2.

FIGURE 13 Eigenvalue spectrum of $\mathbb{O}$ for the cut scenario from Figure 4 but with cell agglomeration

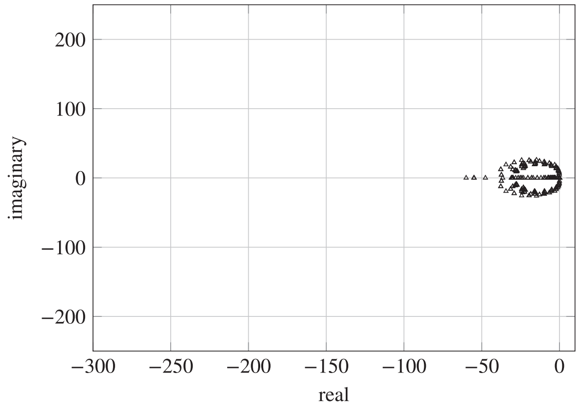
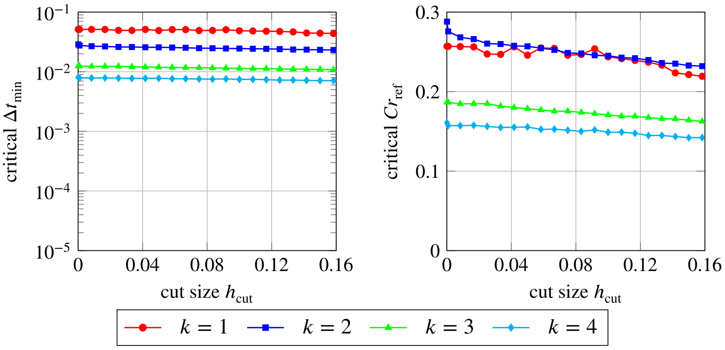


TABLE 2 Maximal absolute value and maximal real part of the eigenvalues v_i of the spatial operator $\mathbb{O}$ with agglomeration for $h_{\text{cut}} = 5.83 \cdot 10^{-3}$ and $h_{\text{cut}} = 8.33 \cdot 10^{-3}$

	$h_{\text{cut}} = 5.83 \cdot 10^{-3}$		$h_{\text{cut}} = 8.33 \cdot 10^{-3}$	
Degree	$\max_i\lvert v_i\rvert$	$\max_i(\text{Re}(v_i))$	$\max_i\lvert v_i\rvert$	$\max_i(\text{Re}(v_i))$
$k = 1$	$6.02 \cdot 10^1$	$1.57 \cdot 10^{-14}$	$6.00 \cdot 10^1$	$1.58 \cdot 10^{-14}$
$k = 2$	$1.19 \cdot 10^2$	$4.11 \cdot 10^{-14}$	$1.19 \cdot 10^2$	$4.09 \cdot 10^{-14}$
$k = 3$	$1.96 \cdot 10^2$	$9.07 \cdot 10^{-14}$	$1.96 \cdot 10^2$	$9.59 \cdot 10^{-14}$
$k = 4$	$2.89 \cdot 10^2$	$1.49 \cdot 10^{-13}$	$2.89 \cdot 10^2$	$1.36 \cdot 10^{-13}$

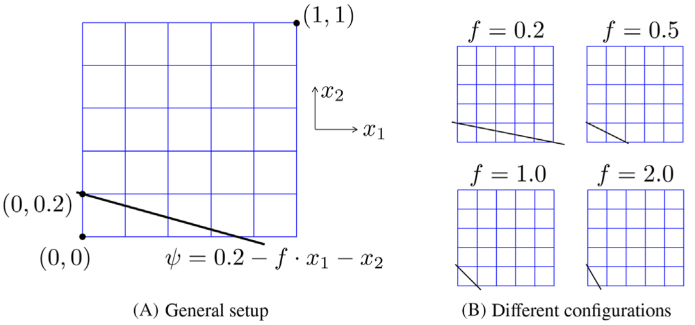
FIGURE 14 Critical time step sizes and corresponding critical reference Courant numbers for straight cut as shown in Figure 4 on the left using the agglomeration approach with merged trace space. The scales are as in Figure 7 to ease comparison [Colour figure can be viewed at wileyonlinelibrary.com]



3.2 | Skew cut

In the following, results analogous to those of the preceding section are presented but for a skew cut as shown in Figure 15 comparing the basic and the stabilized schemes. The background mesh consists of $5 \times 5 = 25$ elements on a $[0, 1] \times [0, 1]$ domain. The physical boundary at the bottom of the geometry is defined by a level set function $\psi = 0.2 - f \cdot x_1 - x_2$ with a parameter f creating different skew cut configurations. For $f > 1$, the cut element agglomerates with its right neighbor, for $f < 1$, the cut elements agglomerate with their respective top neighbors. Values $f = 0.2, 0.25, 0.333, 0.5, 1.01, 1.2, 1.5, 2, 3$ are tested. Apart from these differences, the setup is the same as in the previous experiment.

Figure 16 displays the results for the critical time step size Δt_{min} and the critical reference Courant number for the basic as well as the stabilized schemes. Without agglomeration, not all higher order simulations succeed. For $k = 3$ and $f < 1.2$ as well as for $k = 4$ and $f < 2$, no stable configurations were found. The time step range is between $8.9 \cdot 10^{-7}$ and $2.3 \cdot 10^{-2}$. The most critical cut scenario occurs for $f = 1.01$, where one element is approximately cut into two triangles and the bottom face of the second element is cut with $h_{\text{cut}} = 1.98 \cdot 10^{-3}$, which is the smallest cut face size compared to the other values of f. For the stabilized schemes, stable time steps are found for all tested configurations. For $k = 1, 2$, the critical time step is almost independent of the cut scenario. For $k = 3, 4$, the critical time step varies by a factor of ≈ 10 and ≈ 100, respectively. The two stabilized schemes perform similarly in this experiment, as is obvious from the results in Figure 16.



(A) General setup

(B) Different configurations

FIGURE 15 Simulation setup for critical time step size study with skew cut. A, General setup. B, Different configurations [Colour figure can be viewed at wileyonlinelibrary.com]

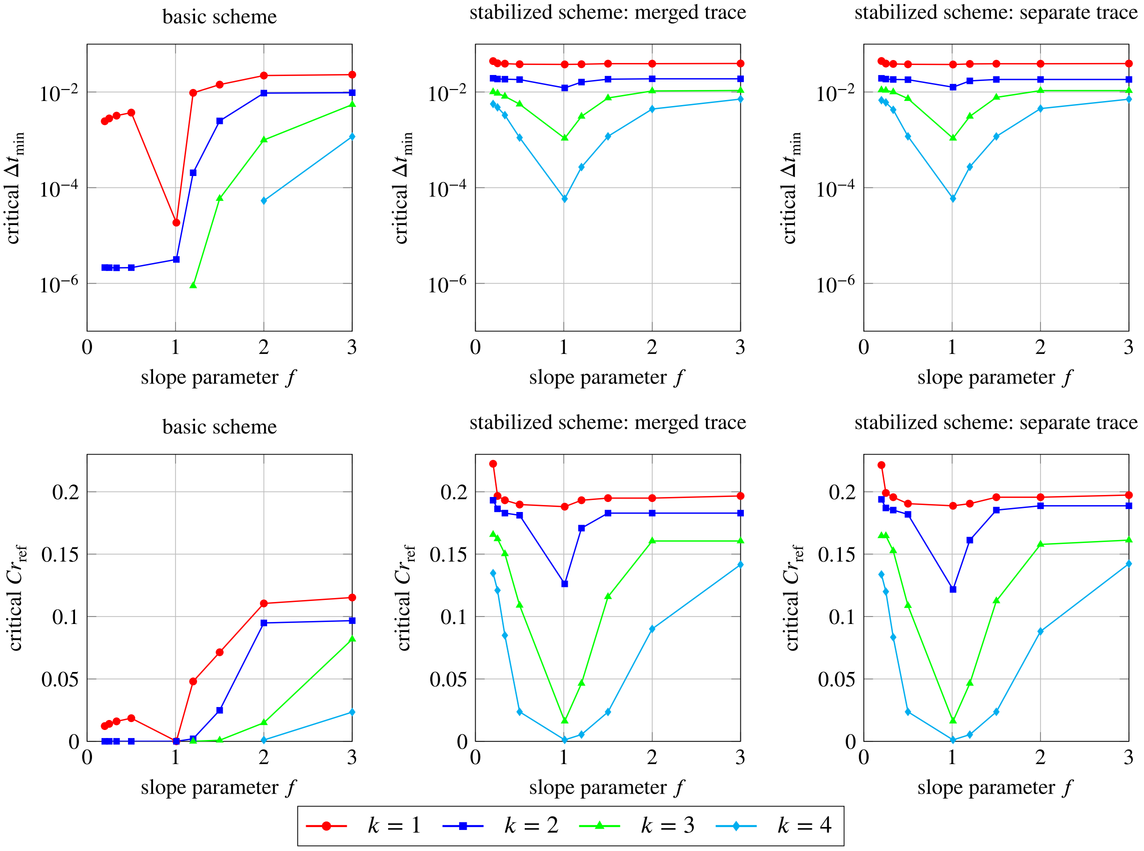


FIGURE 16 Critical time step sizes (top) and corresponding critical reference Courant numbers (bottom) for rectangular domain with skew cut as shown in Figure 15 with the basic scheme (left) and the stabilized scheme (right) [Colour figure can be viewed at wileyonlinelibrary.com]

In addition to the critical time step analysis, an eigenvalue analysis of the global operator $\mathbb{O}$ is carried out. Figure 17 plots the maximal real part of all eigenvalues $\max_i(\text{Re}(\nu_i))$ in analogy to the critical time step size plot in Figure 16. For the basic scheme, the maximal real part of the eigenvalues is zero within roundoff for polynomial degree $k = 1$. For all higher polynomial degrees, positive real parts of the eigenvalues occur and their magnitude explains why no stable time step was found as shown in Figure 16. For the stabilized schemes, the real parts of the eigenvalues lie in between $1.4 \cdot 10^{-14}$ and $1.3 \cdot 10^{-10}$, which can be assumed to be zero within the roundoff precision and the accuracy of the eigenvalue solver `dgeev_` from the LAPACK suite. The highest value occurs for $f = 1.01$ comprising the smallest cut face, which also requires the smallest time step size as shown in Figure 16.

To further quantify the effects of stabilization, the condition numbers κ of the elementwise mass matrices $\mathbb{M}$ and of the face-wise mass matrix $\mathbb{G}$ are studied because their inverses (or rather their LU factorizations) are applied in the numerical algorithm. Figure 18 plots the maximal condition number over all elements or faces, respectively, for the matrices $\mathbb{M}$ and $\mathbb{G}$ comparing the basic and the stabilized schemes. Without stabilization, the condition number reaches very high

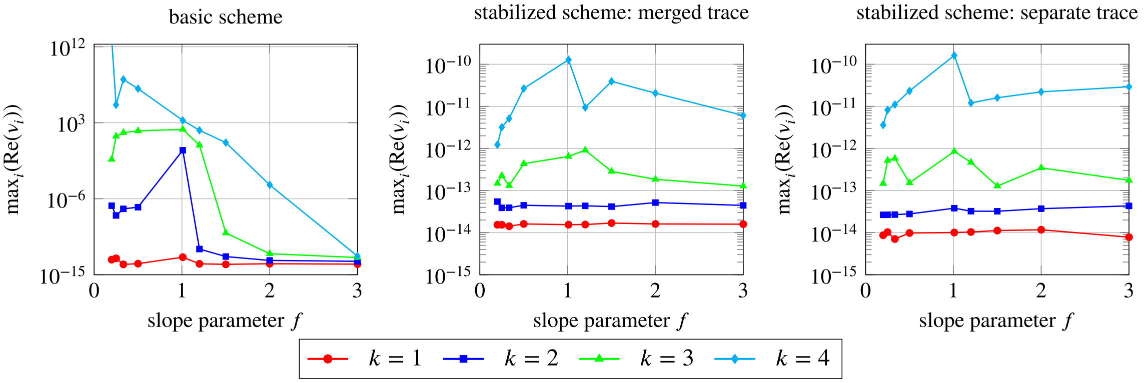


FIGURE 17 Maximal real part of the eigenvalues of $\mathbb{O}$ for rectangular domain with skew cut as shown in Figure 15. Note that the scaling of the vertical axis differs significantly between the basic and the stabilized scheme [Colour figure can be viewed at wileyonlinelibrary.com]

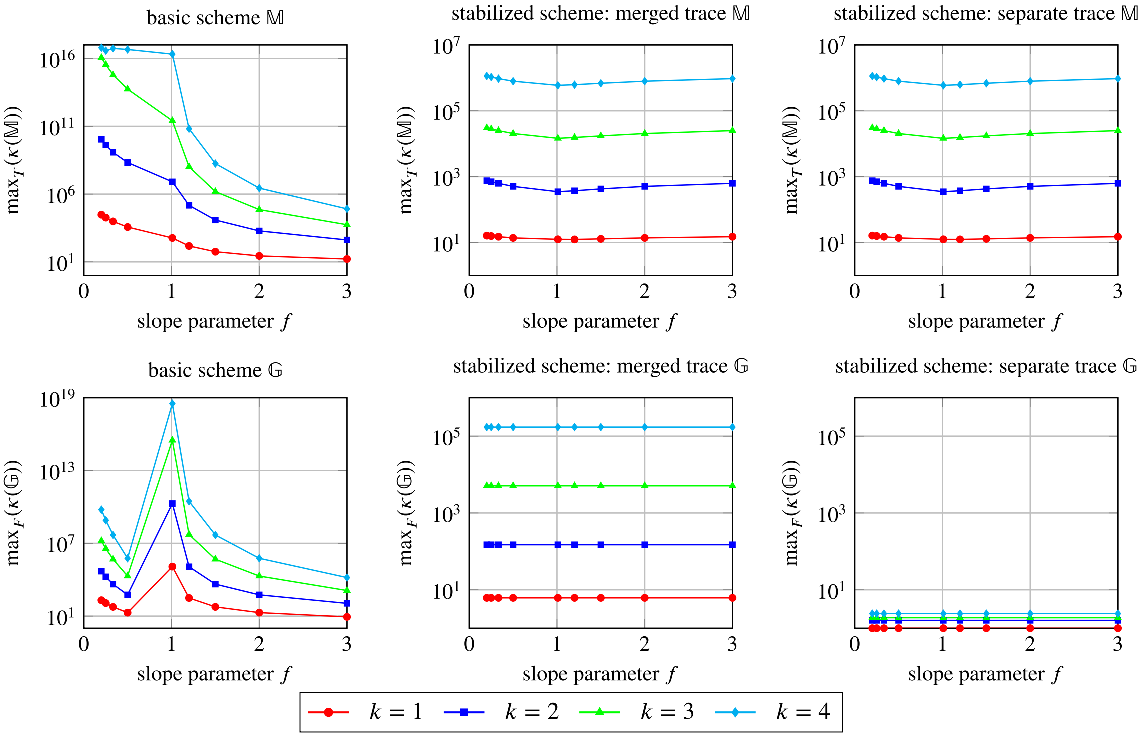


FIGURE 18 Maximal condition number of mass matrix $\mathbb{M}$ (top) and face matrix $\mathbb{G}$ (bottom) with the basic scheme (left) and stabilized (right) for rectangular domain with skew cut as shown in Figure 15 [Colour figure can be viewed at wileyonlinelibrary.com]

levels for higher polynomial degrees. For the element mass matrix $\mathbb{M}$ all cut scenarios with $f < 1$ appear to be similarly disadvantageous. For $f < 1$, neighboring edges are cut in one element, whereas opposing edges are cut for $f > 1$. For the face mass matrix $\mathbb{G}$, the cut scenario with $f = 1.01$ appears to be most disadvantageous, which is due to the fact that the smallest cut face length $h_{\mathrm{cut}} = 1.98 \cdot 10^{-3}$ occurs here. For both stabilized schemes, the condition numbers are constant for the different cut scenarios. With increasing polynomial degree, the conditioning worsens with maximal condition number of $\approx 10^6$ for $\mathbb{M}$. For reference, in standard elements and faces of this setup which are neither cut nor part of an agglomerate, the condition numbers are $\kappa(\mathbb{M}) \approx 9, 47, 75{,}108$ for polynomial degrees $k = 1, 2, 3, 4$, respectively. The condition numbers for agglomerates are much higher due to the extrapolation from the reference element to a scenario with up to twice the size. For the matrix $\mathbb{G}$, the stabilized scheme with merged trace space yields a maximal condition number of $\approx 10^5$, while for the stabilized scheme with separate trace space the condition number of $\mathbb{G}$ is close to one throughout the experiment. For comparison, for standard faces we have $\kappa(\mathbb{G}) \approx 3, 7, 9, 10$ for polynomial degrees $k = 1, 2, 3, 4$, respectively.

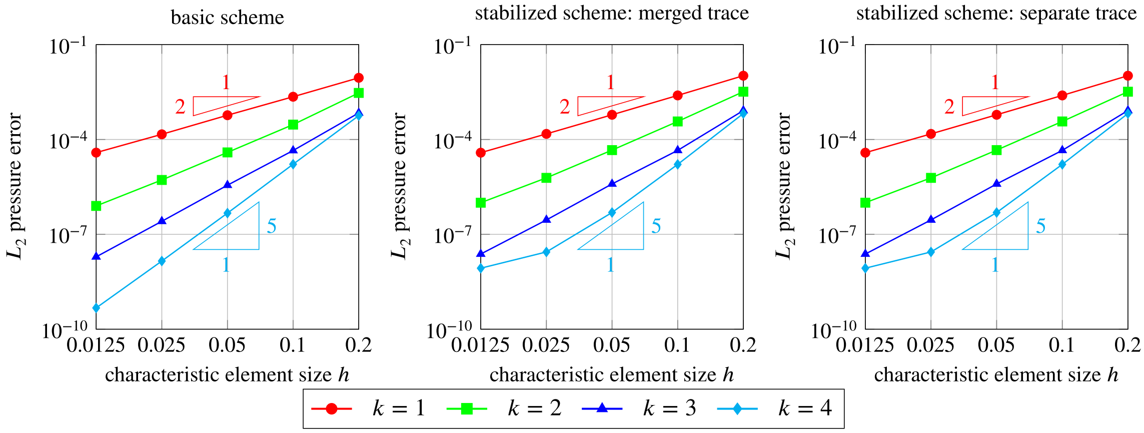


FIGURE 19 Convergence study for various polynomial degrees [Colour figure can be viewed at wileyonlinelibrary.com]

Edge length in the background mesh	Refinement factor	Edge length in agglomerated elements	Refinement factor
0.2000		0.2067	
0.1000	2.00	0.1065	1.94
0.0500	2.00	0.0579	1.83
0.0250	2.00	0.0331	1.75
0.0125	2.00	0.0207	1.60

TABLE 3 Mesh characteristics for convergence study with agglomeration

3.3 | Convergence

For completeness, a convergence study is carried out for the basic scheme and the two stabilized methods. The problem setup is the same square membrane as in the first examples. The physical domain is the unit square $\Omega = [0, 1] \times [0, 1]$ with the background mesh defined on $[0, 1.19] \times [0, 1]$ such that elements must be cut at $x_1 = 1$. The convergence study is carried out on uniformly refined meshes starting from $6 \times 5 = 30$ elements with element edge length of 0.2 in the uncut elements for the coarsest mesh. The shortest edge in the cut elements is of length $h_{\text{cut}} = 8.33 \cdot 10^{-3}$ for all levels of refinement. LTS is used with a prescribed Courant number of $Cr = 0.1$. For the basic scheme, h_{cut} is used in the cut cells to determine the time step size. The reference Courant number is $Cr_{\text{ref}} = 0.004, 0.008, 0.017, 0.033, 0.067$ for the different levels of refinement from coarse to fine, respectively. For the stabilized scheme, $h + h_{\text{cut}}$ is used in the agglomerated cells. The allowed difference in the time steps between neighboring clusters is $M = 1$ according to Equation (17). The analytic solution is the vibrating square membrane as in Equations (18) and (19) but with the number of membrane modes chosen according to the polynomial degree of the shape functions $m = k$. The initial pressure field is L_2 projected. Figure 19 presents convergence results in terms of the L_2 pressure error at $t = 1$ for the basic and the stabilized schemes. The optimal order of convergence $k + 1$ is achieved for the basic scheme as well as the stabilized schemes for all polynomial degrees. The error level is slightly higher for the stabilized schemes, which is due to the fact that the effective element sizes and the time step sizes are slightly larger. Also, it is important to note that the refinement is uniform on the background mesh but in combination with the agglomeration approach, the characteristic element size in the agglomerated layer is not exactly halved. This effect increases for finer meshes as shown in Table 3 in terms of a refinement factor comparing the face length in agglomerated elements from one level of refinement to the next. A similar effect occurs in the basic scheme for refinement in time: the minimal time step size is the same for all levels of mesh refinement because the smallest time step size stems from the cut elements where the face length does not change because it is too small to be affected by the refinement.

To conclude this section, it is stated that we proposed a basic scheme where small time step sizes balance cut instabilities to some extent and two stabilized schemes where cell agglomeration prevents strong cut instabilities, and all methods converge with optimal order. Small time steps can still be necessary with the stabilized schemes, which motivates the combination with LTS.

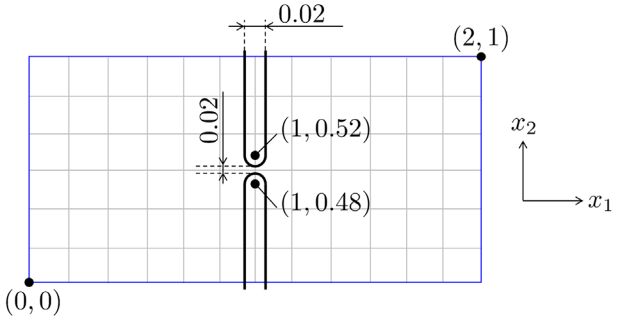


FIGURE 20 Schematic setup of a rectangular domain divided by a wall with a hole. The coarsest utilized mesh is indicated with gray lines. Note that the wall thickness is not plotted in true scale. The wall thickness is 0.02 and one element in the coarsest utilized mesh has an edge length of $1/6 \approx 0.16667$ [Colour figure can be viewed at wileyonlinelibrary.com]

3.4 | Extension to three dimensions

All developments in this section for cell merging have been applied in two space dimensions. Cell merging has previously been used in three dimensions with standard DG-methods in Reference 22. The method in this article is nontrivial to extend to 3D due to the degrees of freedom introduced on the immersed boundary. For an intersected element in 3D, K, the surface $\Gamma \cap K$ can have a quite general shape, depending on how the cut occurs. This makes it difficult to distribute degrees of freedom and define appropriate polynomials. One possibility would be to triangulate this region and distribute degrees of freedom over each triangle. However, creating a high-quality triangulation could be expensive and perhaps not possible. This is a topic of future work.

4 | GENERAL APPLICATIONS

In the following, two examples representative for general cut scenarios are presented.

4.1 | Sound propagation through a hole in wall

A two-dimensional domain of size $[0, 2] \times [0, 1]$ is cut by a level set function representing a wall with a hole as shown in Figure 20. The wall is of thickness 0.02 and the hole is of size 0.02. The wall is of semicircular shape at the hole. Meshing of this setup with boundary conforming elements would require small elements, corresponding to a high computational expense. Here, the coarsest utilized mesh consists of $12 \times 6 = 72$ elements of edge length $h = 0.16667$. The combination of an even number of elements and the symmetry of the setup yields elements that are only cut on one of their sides. Finer meshes are obtained by uniform refinement steps r resulting in $72 \cdot 4^r$ elements and $h = 0.16667 \cdot 2^{-r}$. The initial pressure and velocity fields are

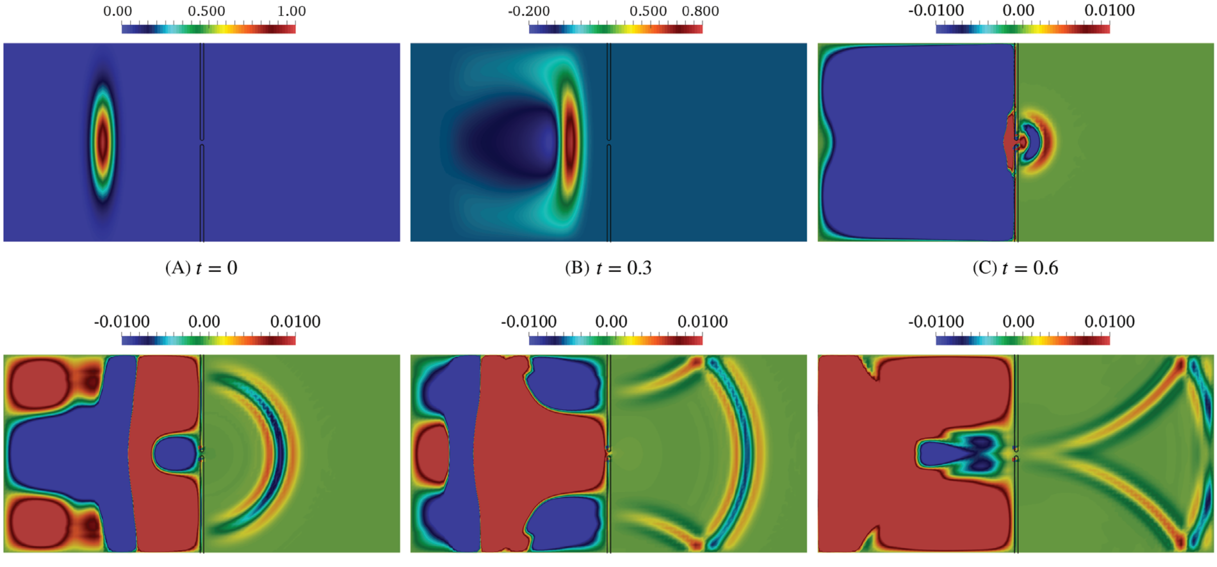
$$p_0 = \exp\left(-400 \cdot (x_1 - 0.5)^2 - 20 \cdot (x_2 - 0.5)^2\right),$$

$$\boldsymbol{v}_0 = \begin{bmatrix} \exp\left(-400 \cdot (x_1 - 0.5)^2 - 20 \cdot (x_2 - 0.5)^2\right) \\ 0 \end{bmatrix}.$$

Mass density and speed of sound are set to $\rho = 1$ and $c = 1$. The Courant number is prescribed as $Cr = 0.1$ and a cluster distribution as shown in Figure 21A is obtained. Figure 22 shows the pressure field at various points in time to give an impression of the qualitative solution behavior. The initial pressure propagates in the left part of the domain and a small share passes through the hole propagating circularly afterward.

The zero contour of the level set function is subject to a discretization error just like the solution fields. Table 4 shows the hole size calculated from the cut face sizes h_{cut} for different polynomial degrees k and refinement levels r. For coarse discretizations, the approximation of the level set function does not yield a hole in the wall. With increasing resolution, a trend to the correct value of 0.02 is observed.

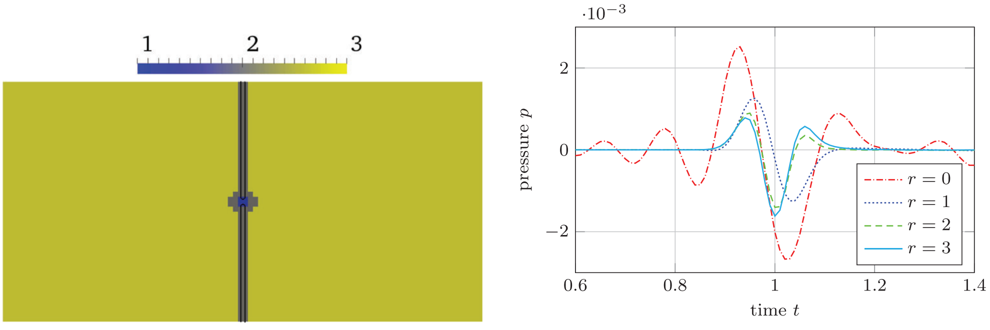
For $Cr = 0.1$, stable results are obtained for $k = 1, 2, 3, 4$ and $r = 0, 1, 2, 3$ with the basic scheme, that is, without stabilization through cell agglomeration but with LTS. To demonstrate a "convergence" in the primary solution fields, the pressure is plotted over time at the point $(1.505, 0.505)$ for $k = 4$ and refinement levels $r = 0, 1, 2, 3$. For the coarse discretizations, overshoots and oscillations are observed. Comparison of $r = 2$ and $r = 3$ only shows a slight



(A) $t = 0$ (B) $t = 0.3$ (C) $t = 0.6$

(D) $t = 0.9$ (E) $t = 1.2$ (F) $t = 1.5$

FIGURE 21 Exemplary cluster distribution (for $k = 4$ and $r = 3$) and pressure curves over time for simulations of hole in a wall with $k = 4$ and different refinement levels. A, Exemplary cluster distribution. B, Pressure over time at (1.505, 0.505)



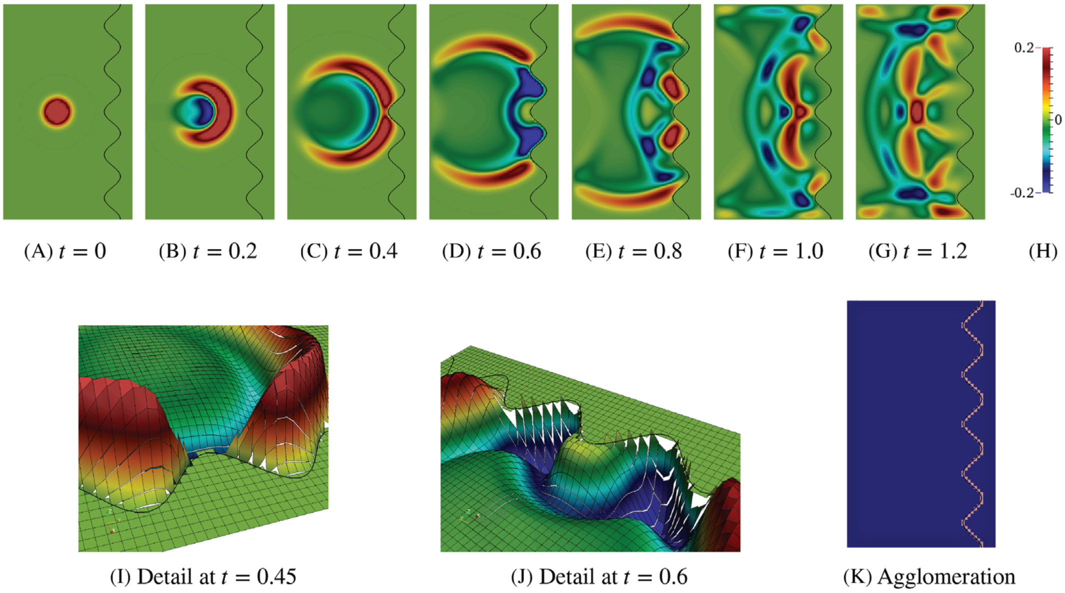
(A) Exemplary cluster distribution (B) Pressure over time at (1.505, 0.505)

FIGURE 22 Pressure field at various points in time for the hole in a wall. Note the changes in scaling at $t = 0.3$ and $t = 0.6$. A, $t = 0$. B, $t = 0.3$. C, $t = 0.6$. D, $t = 0.9$. E, $t = 1.2$. F, $t = 1.5$

	$r = 0$	$r = 1$	$r = 2$	$r = 3$
$k = 1$	–	–	–	0.0163
$k = 2$	–	–	0.0159	0.0186
$k = 3$	–	0.0165	0.0166	0.0189
$k = 4$	0.0119	0.0147	0.0196	0.0201

TABLE 4 Actual hole size resulting from discretization of level set zero contour for various polynomial degrees k and refinement levels r

difference, confirming convergence. For $r = 4$, a small cut with $h_{\text{cut}}/h < 3 \cdot 10^{-3}$ occurs and without agglomeration, only $k = 1$ yields stable results. For the unstable configuration $r = 4$ and $k = 2$, the stabilized approach with merged trace space is applied but also leads to an unstable simulation in combination with the standard input of $Cr = 0.1$ when measured against the merged edge lengths. Reducing the input Courant number to $Cr = 0.003$ is the first stable configuration. Without cell agglomeration, the solution diverges for $Cr = 0.003$. The stabilization approach with separate trace space behaves similarly but allows for a significantly larger time step—the scheme is unstable for the standard setting $Cr = 0.1$ but is stable when the Courant number is reduced to $Cr = 0.05$. This suggests that the stabilization is effective, but needs to be combined with locally small time steps, which is efficiently enabled by the proposed LTS approach.



(A) $t = 0$ (B) $t = 0.2$ (C) $t = 0.4$ (D) $t = 0.6$ (E) $t = 0.8$ (F) $t = 1.0$ (G) $t = 1.2$ (H)

(I) Detail at $t = 0.45$ (J) Detail at $t = 0.6$ (K) Agglomeration

FIGURE 23 Pressure field for sinusoidal cut at various points in time. A, $t = 0$. B, $t = 0.2$. C, $t = 0.4$. D, $t = 0.6$. E, $t = 0.8$. F, $t = 1.0$. G, $t = 1.2$. In A-G, the pressure field is shown for $k = 2$. I and J, The warped solution field obtained with $k = 1$. H, The pressure legend. K, All elements are marked that are agglomerated

4.2 | Sinusodial cut

The last numerical example to be studied is defined on a background mesh of 48 × 80 elements with size $h = 0.025$ on the domain $[0, 1.2] \times [-0.5, 1.5]$. On the right boundary, the physical domain is defined by a level set function given as

$$\psi = 0.3 \cdot \sin(5\pi x_1) + 4 \cdot (x_2 - 1) - 4.05.$$

The initial pressure and velocity fields are

$$p_0 = \exp\left(-160 \cdot \left((x_1 - 0.5)^2 + (x_2 - 0.5)^2\right)\right),$$
$$\boldsymbol{v}_0 = \exp\left(-160 \cdot \left((x_1 - 0.5)^2 + (x_2 - 0.5)^2\right)\right) \cdot \begin{bmatrix} 1 \\ x_2 - 0.5 \end{bmatrix}.$$

This cut scenario is demanding because many skew cuts as discussed in Section 3.2 are generated. For a first impression on the problem setup, Figure 23 plots pressure snapshots and solution details. Without stabilization, no stable results are obtained for a range of tested Courant numbers. For the stabilized schemes, elements are agglomerated to their left neighbor if their cut faces are smaller than 60% of a standard face as shown in Figure 23K.

4.2.1 | Stabilized scheme with merged trace space

For simulations using the stabilized scheme with merged faces and standard parameters, that is, $Cr = 0.1$ in combination with $h + h_{\text{cut}}$ as edge length and $M = 1$, no stable results are obtained. Only when a stricter time step limit is imposed, stable simulations are obtained: when using $0.1 \cdot h_{\text{cut}}$ as characteristic element size in agglomerated cells instead of $h + h_{\text{cut}}$ to determine the time step size from the Courant input, simulations with $k = 1$ and $k = 2$ are stable. For $k = 3, 4$ this and further tightening of the time step did not improve the situation. This relates to the drops in critical time steps reported for the skew cut test case in Section 3.2 for $k > 2$. We suspect that roundoff errors during the LU factorization of the matrices $\mathbb{A}$, $\mathbb{M}$, and $\mathbb{G}$ interact to create slightly positive eigenvalues in the discrete operator $\mathbb{O}$ for certain cut situations; however, we have not been able to investigate the full eigenvalue spectrum with an accurate dense eigenvalue solver due to the size of the operator with around 242 000 unknowns for $k = 4$.

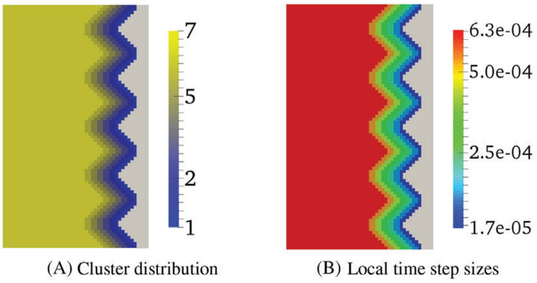


FIGURE 24 Cluster and time step size distribution for simulation of sinusoidal cut with $M = 6$. A, Cluster distribution. B, Local time step sizes

For the stable configuration with $k = 2$, the maximal time step difference M between neighboring clusters is determined. For the presented scenario $M = 6$ yields stable results, while $M \geq 7$ yields instabilities. The cluster and time step size distributions for $M = 6$ are shown in Figure 24. If all elements would advance with the smallest time step size, the total number of time steps multiplied by the number of elements is $3.4 \cdot 10^8$ to reach t_{end}. With LTS and $M = 6$, the total number of element updates is $7.0 \cdot 10^7$, which is 4.9 times less. The computational times on one processor are $8.0 \cdot 10^3$s and $1.8 \cdot 10^3$s, respectively. The real speed up is hence 4.6, which is close to the ratio of the computational work. LTS introduces slight overhead compared to global time stepping for the correct flux consideration at cluster boundaries.[35] In general, it can be said that the speed up due to LTS depends on the mesh characteristics. If the physical domain of the presented example would be extended in negative x_1 direction by more elements of standard size, the speed up would be even higher. The larger the ratio between small and standard elements (in terms of size as well as numbers), the higher the expected speed up.

4.2.2 | Stabilized scheme with separate trace space

The second variant with separate trace space according to Section 3.1.2 is stable for the standard parameters $Cr = 0.1$ in combination with $h + h_{\text{cut}}$ as edge length and $M = 1$ in the clustering for all $k = 1, 2, 3, 4$ in this scenario. This shows that keeping the trace space on the two sections of a merge cell separate is beneficial for the stability, as it avoids any increase in the condition number of $\mathbb{G}$.

5 | CONCLUSION

We have presented a method to solve the acoustic wave equation on an immersed domain using the hybridizable DG method for spatial discretization and ADER with global or LTS for temporal discretization.

We have studied and quantified the stabilizing effects of small time steps for different cut scenarios considering critical time step sizes, eigenvalue spectra, and conditioning of mass matrices for polynomial degrees $k = 1, 2, 3, 4$ based on academic examples. We encountered two kinds of cut instabilities: a weak instability is caused by eigenvalues of the semi-discrete operator with high absolute values and a strong instability is caused by positive real parts of the eigenvalues. It has turned out that in the academic examples small time step sizes can balance the cut instabilities to some extent, especially the weak instability, that is, for moderate polynomial orders or noncritical cut sizes.

To avoid the strong instabilities, which cannot be opposed by small time steps, two stabilized approaches based on cell agglomeration have been proposed. The two approaches differ in how they define the function space on the traces of the HDG method, with one merging the spaces on faces and a second one that keeps them separate. The stabilized approaches improve the conditioning of the mass matrices and prohibit positive real parts of eigenvalues of the spatially discrete operator. However, skew cuts may still require a significant local decrease of the time step size for higher polynomial degrees of the shape functions.

It has been shown that both the basic approach with LTS and the approach with additional stabilization by agglomeration converge with optimal order $k + 1$ for polynomial degrees $k = 1, 2, 3, 4$.

A rectangular domain divided by a wall with a hole presents the applicability of the developed method to problems which are both hard to mesh with boundary conforming elements and would require a strong reduction in the element size and thus sacrifice efficiency with explicit time stepping. Based on an example with a sine wave as cut level set function generating general cut scenarios, the problem specific-speed up due to the LTS in contrast to global time stepping has been evaluated. The problem has been solved in five times less computational time. The considered examples show that neither the reduction of the time step nor the cell agglomeration approach alone provide efficient solutions for the problem of cut instabilities in the general case. However, the combination of cell agglomeration and small local time steps allows to combine large time steps in the bulk of the domain with small time steps only close to the boundary, thus increasing efficiency. For the stabilized approach with separate polynomials on the two parts of an edge of merged cells, all our examples could be stably simulated for all tested polynomial degrees. This was however not true for the straightforward approach with merged trace space. It is the subject of future work to analyze the difference in stability properties of the different approaches. Comparisons with a stabilization approach based on ghost penalty stabilization for both regular and face variables is also an open issue.

ACKNOWLEDGEMENTS

This research was partly supported by the Swedish Research Council Grant No. 2014-6088 and the German Research Foundation (DFG) Grant No. KR4661/2-1.

ORCID

Svenja Schoeder https://orcid.org/0000-0002-0157-6760
Martin Kronbichler https://orcid.org/0000-0001-8406-835X

REFERENCES

1. Cockburn B, Karniadakis G, Shu CW. *Discontinuous Galerkin Methods - Theory, Computation and Applications*. 1st ed. Berlin Heidelberg / Germany: Springer Verlag; 2000.
2. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Application. 54 of Texts in Applied Mathematics*. New York, NY: Springer; 2008.
3. Giorgiani G, Modesto D, Fernández-Méndez S, Huerta A. High-order continuous and discontinuous Galerkin methods for wave problems. *Int J Numer Methods Fluids*. 2013;73:883-903. https://doi.org/10.1002/fld.3828.
4. Huerta A, Angeloski A, Roca X, Peraire J. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int J Numer Methods Eng*. 2013;96:529-560. https://doi.org/10.1002/nme.4547.
5. Ainsworth M. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *J Comput Phys*. 2004;198:106-130. https://doi.org/10.1016/j.jcp.2004.01.004.
6. Burman E, Claus S, Hansbo P, Larson M, Massing A. CutFEM: discretizing geometry and partial differential equations. *Int J Numer Methods Eng*. 2015;104(7):472-501. https://doi.org/10.1002/nme.4823.
7. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: Part I. a stabilized Lagrange multiplier method. *Comput Methods Appl Mech Eng*. 2010;199(41):2680-2686. https://doi.org/10.1016/j.cma.2010.05.011.
8. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: Part II. a stabilized Nitsche method. *Appl Numer Math*. 2012;62(4):328-341. https://doi.org/10.1016/j.apnum.2011.01.008.
9. Massing A, Larson MG, Logg A, Rognes ME. A stabilized Nitsche fictitious domain method for the stokes problem. *J Sci Comput*. 2014;61(3):604-628. https://doi.org/10.1007/s10915-014-9838-9.
10. Sticko S, Kreiss G. A stabilized Nitsche cut element method for the wave equation. *Comput Methods Appl Mech Eng*. 2016;309:364-387. https://doi.org/10.1016/j.cma.2016.06.001.
11. Sticko S, Kreiss G. Higher order cut finite elements for the wave equation. *J Sci Comput*. 2019;80(3):1867-1887. https://doi.org/10.1007/s10915-019-01004-2.
12. Lehrenfeld C, Reusken A. Optimal preconditioners for Nitsche-XFEM discretizations of interface problems. *Numer Math*. 2016;135:313-332. https://doi.org/10.1007/s00211-016-0801-6.
13. Reusken A. Analysis of an extended pressure finite element space for two-phase incompressible flows. *Comput Vis Sci*. 2008;11(4-6):293-305. https://doi.org/10.1007/s00791-008-0099-8.
14. Bastian P, Engwer C. An unfitted finite element method using discontinuous Galerkin. *Int J Numer Methods Eng*. 2009;79(12):1557-1576. https://doi.org/10.1002/nme.2631.
15. Gürkan C, Sala-Lardies E, Kronbichler M, Fernández-Méndez S. Extended hybridizable discontinous Galerkin (X-HDG) for void problems. *J Sci Comput*. 2016;66(3):1313-1333. https://doi.org/10.1007/s10915-015-0066-8.
16. Gürkan C, Kronbichler M, Fernández-Méndez S. eXtended hybridizable discontinous Galerkin with Heaviside enrichment for heat bimaterial problems. *J Sci Comput*. 2016;72(2):542-567. https://doi.org/10.1007/s10915-017-0370-6.
17. Burman E, Ern A. An unfitted hybrid high-order method for elliptic interface problems. *SIAM J Numer Anal*. 2018;56(3):1525-1546. https://doi.org/10.1137/17M1154266.
18. Burman E, Ern A. A cut cell hybrid high-order method for elliptic problems with curved boundaries. In: Radu FA, Kumar K, Berre I, Nordbotten JM, Pop IS, eds. *Numerical Mathematics and Advanced Applications ENUMATH 2017*. New York, NY: Springer International Publishing; 2019:173-181.

19. Hansbo P, Larson MG, Zahedi S. A cut finite element method for a Stokes interface problem. *Appl Numer Math*. 2014;85(Suppl C):90-114. https://doi.org/10.1016/j.apnum.2014.06.009.
20. Gürkan C, Massing A. A stabilized cut discontinuous Galerkin framework for elliptic boundary value and interface problems. *Comput Methods Appl Mech Eng*. 2019;348:466-499. https://doi.org/10.1016/j.cma.2018.12.041.
21. Modisette J, Darmofal D. Toward a robust, higher-order cut-cell method for viscous flows. Paper presented at: Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition (AIAA 2010-721); 2010:1-18; Orlando, FL.
22. Johansson A, Larson M. A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary. *Numer Math*. 2013;123(4):607-628. https://doi.org/10.1007/s00211-012-0497-1.
23. Kummer F. Extended discontinuous Galerkin methods for two-phase flows: the spatial discretization. *Int J Numer Methods Eng*. 2016;109(2):259-289. https://doi.org/10.1002/nme.5288.
24. Müller B, Krämer-Eis S, Kummer F, Oberlack M. A high-order discontinuous Galerkin method for compressible flows with immersed boundaries. *Int J Numer Methods Eng*. 2016;110(1):3-30. https://doi.org/10.1002/nme.5343.
25. Badia S, Verdugo F, Martín AF. The aggregated unfitted finite element method for elliptic problems. *Comput Methods Appl Mech Eng*. 2018;336:533-553. https://doi.org/10.1016/j.cma.2018.03.022.
26. Verdugo F, Martín AF, Badia S. Distributed-memory parallelization of the aggregated unfitted finite element method. *Comput Methods Appl Mech Eng*. 2019;357:112583. https://doi.org/10.1016/j.cma.2019.112583.
27. Courant R, Friedrichs K, Lewy H. Über die partiellen Differenzengleichungen der mathematischen Physik. *Math Ann*. 1928;100(1):32-74. https://doi.org/10.1007/BF01448839.
28. Berger M, Helzel C. A simplified h-box method for embedded boundary grids. *SIAM J Sci Comput*. 2012;34(2):861-888. https://doi.org/10.1137/110829398.
29. Abgrall R, Shu C. *Handbook of Numerical Methods for Hyperbolic Problems XVIII, Applied and Modern Issues*. Amsterdam, Netherlands: Elsevier; 2017.
30. Ascher UM, Ruuth SJ, Spiteri RJ. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl Numer Math*. 1997;25(2):151-167. https://doi.org/10.1016/S0168-9274(97)00056-1.
31. Kanevsky A, Carpenter M, Gottlieb D, Hesthaven J. Application of implicit–explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes. *J Comput Phys*. 2007;225(2):1753-1781. https://doi.org/10.1016/j.jcp.2007.02.021.
32. Kronbichler M, Schoeder S, Müller C, Wall W. Comparison of implicit and explicit hybridizable discontinuous Galerkin methods for the acoustic wave equation. *Int J Numer Methods Eng*. 2016;106(9):712-739. https://doi.org/10.1002/nme.5137.
33. Schwartzkopff T, Dumbser M, Munz CD. Fast high order ADER schemes for linear hyperbolic equations. *J Comput Phys*. 2004;197:532-539. https://doi.org/10.1016/j.jcp.2003.12.007.
34. Dumbser M, Käser M. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – II. the three-dimensional isotropic case. *Geophys J Int*. 2006;167:319-336. https://doi.org/10.1111/j.1365-246X.2006.03120.x.
35. Schoeder S, Kronbichler M, Wall W. Arbitrary high-order explicit hybridizable discontinuous Galerkin methods for the acoustic wave equation. *J Sci Comput*. 2018;76:969-1006. https://doi.org/10.1007/s10915-018-0649-2.
36. Kronbichler M, Kormann K. A generic interface for parallel finite element cell-based operator application. *Comput Fluids*. 2012;63:135-147. https://doi.org/10.1016/j.compfluid.2012.04.012.
37. Schoeder S, Kormann K, Wall WA, Kronbichler M. Efficient explicit time stepping of high order discontinuous Galerkin schemes for waves. *SIAM J Sci Comput*. 2018;40(6):C803-C826. https://doi.org/10.1137/18M1185399.
38. Schoeder S, Wall WA, Kronbichler M. ExWave: a high performance discontinuous Galerkin solver for the acoustic wave equation. *Software X*. 2019;9:49-54. https://doi.org/10.1016/j.softx.2019.01.001.
39. Kronbichler M, Kormann K. Fast matrix-free evaluation of discontinuous Galerkin finite element operators. *ACM Trans Math Softw*. 2019;45(3):29:1-29:40. https://doi.org/10.1145/3325864.
40. Nguyen NC, Peraire J, Cockburn B. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *J Comput Phys*. 2011;230:3695-3718. https://doi.org/10.1016/j.jcp.2011.01.035.
41. Saye RI. High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. *SIAM J Sci Comput*. 2015;37(2):A993-A1019. https://doi.org/10.1137/140966290.
42. Schwartzkopff T, Munz CD, Toro EF. ADER: a high-order approach for linear hyperbolic systems in 2D. *J Sci Comput*. 2002;17(1):231-240. https://doi.org/10.1023/A:1015160900410.
43. Dumbser M, Schwartzkopff T, Munz CD. Arbitrary high order finite volume schemes for linear wave propagation. In: Krause E, Shokin Y, Resch M, Shokina N, eds. *Computational Science and High Performance Computing II: The 2nd Russian-German Advanced Research Workshop, Stuttgart, Germany, March 14 to 16, 2005*. Berlin, Heidelberg, Germany: Springer; 2006:129-144.
44. Dumbser M, Peshkov I, Romenski E, Zanotti O. High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: viscous heat-conducting fluids and elastic solids. *J Comput Phys*. 2016;314:824-862. https://doi.org/10.1016/j.jcp.2016.02.015.
45. Dumbser M, Käser M, Toro EF. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – V. local time stepping and *p*-adaptivity. *Geophys J Int*. 2007;171:695-717. https://doi.org/10.1111/j.1365-246X.2007.03427.x.
46. Alzetta G, Arndt D, Bangerth W, et al. The deal.II library, version 9.0. *J Numer Math*. 2018;26(4):173-184. https://doi.org/10.1515/jnma-2018-0054.
47. Müller B, Kummer F, Oberlack M. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Int J Numer Methods Eng*. 2013;96(8):512-528. https://doi.org/10.1002/nme.4569.

48. Fries TP, Omerović S, Schöllhammer D, Steidl J. Higher-order meshing of implicit geometries—Part I: integration and interpolation in cut elements. *Comput Methods Appl Mech Eng*. 2017;313:759-784. https://doi.org/10.1016/j.cma.2016.10.019.
49. Lehrenfeld C. High order unfitted finite element methods on level set domains using isoparametric mappings. *Comput Methods Appl Mech Eng*. 2016;300:716-733. https://doi.org/10.1016/j.cma.2015.12.005.
50. Engwer C, Nüßing A. Geometric reconstruction of implicitly defined surfaces and domains with topological guarantees. *ACM Trans Math Softw*. 2017;44(2):1-20. https://doi.org/10.1145/3104989.
51. Burman E. Ghost penalty. *Comptes Rendus Mathematique*. 2010;348(21):1217-1220. https://doi.org/10.1016/j.crma.2010.10.006.
52. Kopriva DA. *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. New York, NY: Springer; 2009.
53. Kronbichler M, Kormann K, Wall WA. Fast matrix-free evaluation of hybridizable discontinuous Galerkin operators. In: Radu FA, Kumar K, Berre I, Nordbotten JM, Pop I, eds. *Numerical Mathematics and Advanced Applications—ENUMATH 2017*. Vol 126. Cham: Springer; 2019:581-589.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Schoeder S, Sticko S, Kreiss G, Kronbichler M. High-order cut discontinuous Galerkin methods with local time stepping for acoustics. *Int J Numer Methods Eng*. 2020;121:2979–3003. https://doi.org/10.1002/nme.6343