High-Order Non-Conforming Discontinuous Galerkin Methods for the Acoustic Conservation Equations

Johannes Heinz,* Peter Munch,† Manfred Kaltenbacher‡

Abstract

This work compares two Nitsche-type approaches to treat non-conforming triangulations for a high-order discontinuous Galerkin (DG) solver for the acoustic conservation equations. The first approach (point-to-point interpolation) uses inexact integration with quadrature points prescribed by a primary element. The second approach uses exact integration (mortaring) by choosing quadratures depending on the intersection between non-conforming elements. In literature, some excellent properties regarding performance and ease of implementation are reported for point-to-point interpolation. However, we show that this approach can not safely be used for DG discretizations of the acoustic conservation equations since, in our setting, it yields spurious oscillations that lead to instabilities. This work presents a test case in that we can observe the instabilities and shows that exact integration is required to maintain a stable method. Additionally, we provide a detailed analysis of the method with exact integration. We show optimal spatial convergence rates globally and in each mesh region separately. The method is constructed such that it can natively treat overlaps between elements. Finally, we highlight the benefits of non-conforming discretizations in acoustic computations by a numerical test case with different fluids.

1 Introduction

The main benefit of non-conforming interfaces (NCIs) is the ability to handle arbitrary element connections. In acoustic simulations, we require different element sizes in different regions of a triangulation, e.g., due to wave propagation through inhomogeneous media. NCIs can realize the jump in element sizes without the use of transition regions which usually contain strongly distorted elements [1, 2]. This way, it is possible to reduce degrees of freedom (DoFs) needed without introducing errors related to elements with bad quality [3, 2]. Additionally, algorithms that can handle NCIs can simplify mesh generation since it is possible to generate the grids in a modular way [1].

Overlapping elements further reduce the difficulties in mesh generation since they can be constructed without paying attention to adjacent regions at all. One famous example of this is the overset grid method [4]; a structural mesh serves as the background mesh in which complex geometries can be embedded. This is done by overlaying the corresponding meshes and deleting the elements of the background mesh that completely overlap the embedded mesh.

Besides mentioned advantages of NCIs, some applications, like a rotating fan, require NCIs. To compute the aeroacoustic sound field, we need two mesh regions, one of which is rotating, see e.g. [5]. A conforming mesh at each time step is not feasible; using NCIs in such applications is the obvious solution. However, this requires the non-conforming interface to lie precisely on top of each other, which is only possible using curved elements, cf. [6]. A slightly different approach is to use methods that can also handle element overlaps between the triangulations. This way, the fixed and rotating domain can still pick values for the fluxes at the overlapping boundaries, with the difference that these values are defined inside elements of the other triangulation.

There exist three different ways to handle non-conformities. The most common method in literature is the Mortar method, first introduced by Bernadi *et al.* [7]. The Mortar method is a projection-based method that typically uses Lagrange multipliers to enforce coupling; this requires additional DoFs at the interface. Coupling of the second order wave equation using Mortar methods has been successfully applied in [3].

---

*Institute of Mechanics and Mechatronics, TU Wien, Vienna, Austria (`johannes.heinz@tuwien.ac.at`)

†High-Performance Scientific Computing, University of Augsburg, Augsburg, Germany (`peter.muench@uni-a.de`)

‡Institute of Fundamentals and Theory in Electrical Engineering (IGTE), TU Graz, Graz, Austria (`manfred.kaltenbacher@tugraz.at`)

Another way to couple non-conforming meshes is through interpolation-based methods, such as INTERNODES (INTERpolation for NOn-conforming DEcompositionS) [8].

Nitsche [9] presented the idea of including Dirichlet boundary conditions (DBCs) in the weak form. Methods using this idea are consequently named Nitsche-type methods. Discontinuous Galerkin (DG) schemes use this idea at all element boundaries already [10]. Therefore, we believe that using Nitsche-type methods to couple meshes via NCIs is the most natural way to couple DG schemes. Here we can distinguish between schemes that use exact and inexact integration. For methods with inexact integration, we use integration points dictated by elements on the NCI and evaluate needed quantities in the non-conforming attached elements. Hermann *et al.* [11] used this approach in a two dimensional DG setting for seismic waves on meshes with possibly different element types. Laughton *et al.* [12] refer to this method as point-to-point interpolation method. Methods using exact integration collect integration rules on the intersections between the connected elements. This procedure is commonly referred to as "mortaring", and the intersections are often called "mortars" (without any relation to the Mortar method). Nitsche-type mortaring has been successfully applied in a FEM setting by e.g. [13, 14] for the inhomogeneous wave equation and Maxwell equations, respectively. The procedure of mortaring is the same for Mortar and Nitsche-type mortaring methods. The difference in both methods is that Nitsche-type methods enforce the coupling point-wise (via numerical fluxes). On the other hand, Mortar methods enforce the coupling via an integral (using Lagrange multipliers).

Laughton *et al.* [12] compared the Nitsche-type mortaring method to the point-to-point interpolation method regarding performance and accuracy in a DG setting. The advantage of point-to-point interpolation over methods with mortaring is its ease of implementation [12]. For the compressible Euler equations in two dimensions, it is shown that point-to-point interpolation outperforms the method with mortaring, considering polynomial degrees between 3 and 7 [12]. We expect the performance to close up for long run-times on static triangulations (the quadrature rules of the intersections and the mapping of obtained integration points have to be setup only once). However, we suspect the performance to diverge even more on moving meshes, where the intersections and the mappings have to be updated every time step or Runge–Kutta stage. The disadvantage of point-to-point interpolation is that it introduces numerical errors related to aliasing. Methods using mortaring do not face this issue. To obtain similar errors for point-to-point interpolation compared to the Nitsche-type mortaring, [12] increases the number of quadrature points.

Solving the scalar acoustic wave equation utilizing conforming finite element methods (FEM) has some unattractive peculiarities. It requires specific time-stepping schemes and suffers from numerical dispersion (see e.g. [15, 2]). Transforming the acoustic wave equation to a first-order system yields the acoustic conservation equations. These acoustic conservation equations do not include a second-order temporal derivative; thus, standard time-stepping methods, such as Runge–Kutta methods, can be applied. Furthermore, the velocities of non-harmonically vibrating surfaces natively appear in governing equations, making a straightforward application of these velocities as boundary conditions (BCs) possible. Additionally, conservation laws are ideally suited for finite volume or DG discretizations [16], and it is possible to find less dispersive schemes by adding numerical diffusion using numerical fluxes.

We applied the point-to-point interpolation method in [17] and showed that it provides optimal rates of convergence in space. Later, we observed instabilities for some element configurations using this method. Within this work, we show that for DG discretizations of the acoustic conservation equations, it is not safe to use point-to-point interpolation since the method is not only less accurate but yields spurious oscillations that lead to instabilities in some cases. To the best of the authors' knowledge, no Nitsche-type mortaring formulation exists for the acoustic conservation laws in literature. We present a test case in which mentioned instabilities occur and show that using exact integration via mortaring is a remedy.

Additionally, we provide in-depth convergence studies for the Nitsche-type mortaring approach and show optimal spatial convergence rates on the global computational domain and separately on domains with coarse and high resolution.

2 Governing Equations

The wave equation reads

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \rho\nabla\cdot\left(\frac{1}{\rho}\nabla p\right) = f \quad \text{in } \Omega\times[0,T], \tag{1}$$

on a domain $\Omega \subset \mathbb{R}^d$ of dimension d and in a time interval $[0, T]$. Here, p is the acoustic pressure, c is the sound speed, and the underlying material's density is ρ. The wave equation is a reformulation of the acoustic conservation equations of momentum and mass

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} + \nabla p = 0 \quad \text{in } \Omega \times [0, T], \tag{2}$$

$$\frac{1}{c^2} \frac{\partial p}{\partial t} + \rho \nabla \cdot \boldsymbol{u} = F \quad \text{in } \Omega \times [0, T], \tag{3}$$

$$p = g_p \quad \text{on } \partial \Omega_p^{\mathrm{D}}, \tag{4}$$

$$\boldsymbol{u} = \boldsymbol{g}_u \quad \text{on } \partial \Omega_u^{\mathrm{D}}, \tag{5}$$

$$\rho c \boldsymbol{u} \cdot \boldsymbol{n} = Y p \quad \text{on } \partial \Omega^{\mathrm{Y}}. \tag{6}$$

At boundaries we can apply pressure DBCs (4), velocity DBCs (5), and admittance BCs (6) by setting the normal component of the velocity and a certain admittance Y.

3 Numerical Method

3.1 Notation

The physical domain Ω is represented by the computational domain $\Omega_h(t) = \bigcup_{i=1}^{N_{\mathrm{el}}} \Omega_{e_i} \in \mathbb{R}^d$, with the space dimension d. Within this work, it consists of N_{el} possibly overlapping rectangular/hexahedral finite elements and is bounded by $\Gamma_h = \partial \Omega_h$. A finite element spans Ω_e and is bounded by $\partial \Omega_e$. The solution is continuous inside elements and, due to the nature of DG, discontinuous between elements. The acoustic particle velocity $\boldsymbol{u}$ and acoustic pressure p are subject to the broken polynomial spaces $\mathcal{V}_h$ for the corresponding test and trial functions

$$\mathcal{V}_h^u = \left\{ \boldsymbol{u}_h \in [L_2(\Omega_h)]^d : \boldsymbol{u}_h(\boldsymbol{x})|_{\Omega^e} = \tilde{\boldsymbol{u}}(\boldsymbol{\xi})|_{\tilde{\Omega}^e} \in [\mathcal{P}_{k_u}(\tilde{\Omega}^e)]^d, \forall e \in [1, N_{\mathrm{el}}] \right\}, \tag{7}$$

$$\mathcal{V}_h^p = \left\{ p_h \in L_2(\Omega_h) : p_h|_{\Omega^e} = \tilde{p}_h|_{\tilde{\Omega}^e} \in \mathcal{P}_{k_p}(\tilde{\Omega}^e), \forall e \in [1, N_{\mathrm{el}}] \right\}. \tag{8}$$

Here $\mathcal{P}_k$ is the space of polynomial functions with order k on a reference element. Coordinates in the physical space are $\boldsymbol{x} = (x_1, ..., x_d)^T$; their representation on a reference element are $\boldsymbol{\xi} = (\xi_1, ..., \xi_d)^T$. To transfer between $\boldsymbol{x}$ and $\boldsymbol{\xi}$ a bidirectional mapping

$$\boldsymbol{\varphi} : \begin{cases} \Omega^e \to \tilde{\Omega}^e \\ \boldsymbol{x} \mapsto \boldsymbol{\xi} = \boldsymbol{\varphi}(\boldsymbol{x}, \Omega^e) \end{cases}, \quad \boldsymbol{\varphi}^{-1} : \begin{cases} \tilde{\Omega}^e \to \Omega^e \\ \boldsymbol{\xi} \mapsto \boldsymbol{x} = \boldsymbol{\varphi}^{-1}(\boldsymbol{\xi}, \Omega^e) \end{cases}, \tag{9}$$

can be used. The discrete representations of the continuous pressure and velocity fields in the reference space read

$$\tilde{\boldsymbol{u}}_h^e(\boldsymbol{\xi}) = \sum_{i=1}^{n_{N^{k_u}}} N_i^{k_u}(\boldsymbol{\xi}) \boldsymbol{u}_i^e, \quad \tilde{p}_h^e(\boldsymbol{\xi}) = \sum_{i=1}^{n_{N^{k_p}}} N_i^{k_p}(\boldsymbol{\xi}) p_i^e, \tag{10}$$

with the number of shape functions n_{N^k} defined on a volume element, e.g. for the pressure in the one dimensional case $n_{N^{k_p}} = k_p + 1$. The shape functions N_i^k are constructed by Lagrange polynomials of degree k. Within this work, the same polynomial orders k for velocity and pressure ($k_{\boldsymbol{u}} = k_p$) are utilized. We denote the interior information of an element Ω^e with $(\cdot)^-$ and the exterior information of adjacent elements with $(\cdot)^+$. Consequently, the current element (from now on called "primary element") is denoted as Ω_-^e, and the neighboring elements (or "secondary elements") are described as Ω_+^e. The outward pointing normal vectors of the primary element are $\boldsymbol{n}^-$, since facets of primary and secondary elements coincide, $\boldsymbol{n} = \boldsymbol{n}^- = -\boldsymbol{n}^+$. Accordingly, any scalar or vectorial quantity b is implicitly defined on the primary element if no superscript explicitly assigns it to the primary or secondary element $b = b^-$. We choose the notation for the averaging operator $\{\!\{\cdot\}\!\}$, jump operator $[\cdot]$, and normal jump operator $[\![\cdot]\!]$ according to Bassi *et al.* [18, 19]. They are $\{\!\{b\}\!\} = (b^- + b^+)/2$, $[b] = b^- - b^+$, and $[\![b]\!] = b^- \otimes n^- + b^+ \otimes n^+$. Hereinafter, the integrals are written in the compressed notation $(a, b)_{\Omega_e} = \int_{\Omega_e} a \cdot b \; \mathrm{d}\Omega$ and $(a, b)_{\partial \Omega_e} = \int_{\partial \Omega_e} a \cdot b \; \mathrm{d}\Gamma$, where the operator $\cdot$ indicates an inner product and a represents an arbitrary quantity of the same dimension as b. All operators are given in the notation considering element boundaries; therefore, each facet becomes a primary and

secondary element. For numerical integration, we employ Gaussian quadrature. On an element of spatial dimension d we use $n_q = (k+1)^d$ quadrature points for the volume integrals and $n_q = (k+1)^{d-1}$ on element faces. Boundary conditions are applied using a mirror principle, cf. [16]. While pressure and velocity DBCs are defined as

$$p^+ = -p^- + 2g_p; \quad \boldsymbol{u}^+ = \boldsymbol{u}^- \quad \text{on } \partial\Omega_p^{\mathrm{D}}, \tag{11}$$

$$\boldsymbol{u}^+ = -\boldsymbol{u}^- + 2\boldsymbol{g}_u; \quad p^+ = p^- \quad \text{on } \partial\Omega_u^{\mathrm{D}}. \tag{12}$$

Admittance BCs read

$$\boldsymbol{u}^+ = \left(\frac{2Y}{\rho c}p^- - \boldsymbol{u}^- \cdot \boldsymbol{n}\right)\boldsymbol{n}; \quad p^+ = p^- \quad \text{on } \partial\Omega^{\mathrm{Y}}. \tag{13}$$

Reflecting BCs and first-order absorbing BCs [20] (ABC) are achieved by setting the admittance to $Y = 0$ and $Y = 1$, respectively. If the first order ABC is insufficient, a corresponding perfectly matched layer (PML) formulation is provided in [21] for conforming FEM formulations.

3.2 Spatial Discretization

The numerical method, without non-conformities, has been described briefly in [22]. Within this section, we will recall it in a more detailed manner to be able to extend the formulation.

The semi-discrete system is obtained as usual (cf. [16]). The governing equations are multiplied by the test functions $\boldsymbol{w}_h$ and q_h, and integrated over the computational domain Ω_h. For DG schemes, it is crucial to perform the integration by parts to ensure boundary terms exist. With this, we end up with a corresponding weak formulation. For given equations, it is also possible to perform a second integration by parts to obtain the strong formulation, cf. [16], which is used in [23]. Eventually, numerical fluxes (denoted by the superscript *) are introduced into the boundary integrals. This results in the semi-discrete system of equations

$$\left(\boldsymbol{w}_h, \frac{\partial \boldsymbol{u}_h}{\partial t}\right)_{\Omega^e} - \left(\frac{1}{\rho}\nabla \cdot \boldsymbol{w}_h, p_h\right)_{\Omega^e} + \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, p_h^*\right)_{\partial\Omega^e} = 0 \qquad \forall \boldsymbol{w}_h \in \mathcal{V}_h^u, \tag{14}$$

$$\left(q_h, \frac{\partial p_h}{\partial t}\right)_{\Omega^e} - \left(\rho c^2 \nabla q_h, \boldsymbol{u}_h\right)_{\Omega^e} + \left(\rho c^2 q_h \cdot \boldsymbol{n}, \boldsymbol{u}_h^*\right)_{\partial\Omega^e} = \left(c^2 q_h, f\right)_{\Omega^e} \quad \forall q_h \in \mathcal{V}_h^p. \tag{15}$$

We use Lax–Friedrichs fluxes, as also done in [24, 23, 25, 26],

$$\begin{aligned} p_h^* &= \{\!\{p_h\}\!\} + \frac{\tau}{2}[\![\boldsymbol{u}_h]\!], \\ \boldsymbol{u}_h^* &= \{\!\{\boldsymbol{u}_h\}\!\} + \frac{\gamma}{2}[\![p_h]\!]. \end{aligned} \tag{16}$$

The penalty parameters τ and γ are derived using the Rankine–Hugoniot condition when solving for the Riemann solution [27, 23], resulting in $\tau = \rho c$ and $\gamma = \frac{1}{\rho c}$. These penalty parameters are consistent in terms of a dimension analysis which demands $\tau \sim \rho c$ and $\gamma \sim \frac{1}{\rho c}$. Element boundaries are either located inside the domain $\partial\Omega_{\mathrm{inner}}^e$, at non-conforming boundaries Γ_{NCI}^e, or subject to BCs ($\partial\Omega_p^{\mathrm{D},e}$, $\partial\Omega_u^{\mathrm{D},e}$, or $\partial\Omega^{\mathrm{Y},e}$). The explicit notation of the discretization at element boundaries reads

$$\begin{aligned} \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, p_h^*\right)_{\partial\Omega^e} &= \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, p_h^*\right)_{\partial\Omega_{\mathrm{inner}}^e} + \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, g_p\right)_{\partial\Omega_p^{\mathrm{D},e}} + \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, p_h^- + \tau(\boldsymbol{u}_h^- - \boldsymbol{g}_u)\right)_{\partial\Omega_u^{\mathrm{D},e}} \\ &\quad + \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, p_h^- + \tau\left(\boldsymbol{u}_h^- \cdot \boldsymbol{n} - \frac{Y}{\rho c}p_h^-\right)\right)_{\partial\Omega^{\mathrm{Y},e}} + \left(\frac{1}{\rho}\boldsymbol{w}_h \cdot \boldsymbol{n}, p_{h,\mathrm{NCI}}^*(p_h^-, p_h^+, \boldsymbol{u}_h^-, \boldsymbol{u}_h^+)\right)_{\Gamma_{\mathrm{NCI}-}^e}, \end{aligned} \tag{17}$$

and

$$\begin{aligned} \left(\rho c^2 q_h \boldsymbol{n}, \boldsymbol{u}_h^*\right)_{\partial\Omega^e} &= \left(\rho c^2 q_h \boldsymbol{n}, \boldsymbol{u}_h^*\right)_{\partial\Omega_{\mathrm{inner}}^e} + \left(\rho c^2 q_h \boldsymbol{n}, \boldsymbol{u}_h^- + \gamma(p_h^- - g_p)\right)_{\partial\Omega_p^{\mathrm{D},e}} + \left(\rho c^2 q_h \boldsymbol{n}, \boldsymbol{g}_u\right)_{\partial\Omega_u^{\mathrm{D},e}} \\ &\quad + \left(c q_h, 2Y p_h^-\right)_{\partial\Omega^{\mathrm{Y},e}} + \left(\rho c^2 q_h \boldsymbol{n}, \boldsymbol{u}_{h,\mathrm{NCI}}^*(p_h^-, p_h^+, \boldsymbol{u}_h^-, \boldsymbol{u}_h^+)\right)_{\Gamma_{\mathrm{NCI}-}^e}. \end{aligned} \tag{18}$$
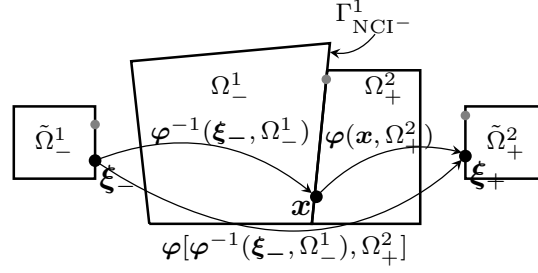
Figure 1: Point-to-point interpolation: Shown is the mapping of a exemplary quadrature point $\boldsymbol{\xi}_-$ (associated to the primary element Ω^1_-) to the the non-conformingly connected secondary element Ω^2_+. Adapted from [17].

In this notation $\Gamma^e_{\mathrm{NCI}-}$ are the faces of the primary elements. A more detailed description of how integrals are computed in the non-conforming case is provided in Section 3.3 and 3.4.

To be able to consider different materials, we have to adapt the fluxes at the NCIs. To this end, we use the LDG fluxes with special self-adapting upwinding parameters and penalty terms as an additional stabilization mechanism to increase the numerical diffusion [23]

$$
\begin{aligned}
p^*_{h,\mathrm{NCI}} &= p^-_h - \frac{\tau^-}{\tau^- + \tau^+}\, [p] + \frac{\tau^- \tau^+}{\tau^- + \tau^+}\, [\![\boldsymbol{u}_h]\!], \\
\boldsymbol{u}^*_{h,\mathrm{NCI}} &= \boldsymbol{u}^-_h - \frac{\gamma^-}{\gamma^- + \gamma^+}\, [\boldsymbol{u}_h] + \frac{\gamma^- \gamma^+}{\gamma^- + \gamma^+}\, [\![p_h]\!].
\end{aligned}
\tag{19}
$$

We can see, that the fluxes simplify to the Lax–Friedrichs fluxes of (16) for homogenous materials, i.e. $c^- \rho^- = c^+ \rho^+$ and therefore, $\gamma^- = \gamma^+$ and $\tau^- = \tau^+$.

3.3   Point-to-Point Interpolation

Non-conformity can be easily handled by the evaluation of all quantities in consistent quadrature points [11, 12]; i.e., we have to evaluate fluxes at the same point in the physical space. The primary element dictates the used quadrature points; see Figure 1. For conforming DG this leads to the same quadrature points in the reference space $\boldsymbol{\xi}_- = \boldsymbol{\xi}_+$. However, if non-conformities in the mesh are present, quadrature points that correspond to the same coordinate in the physical space differ, and we have to find quadrature points on the secondary element as

$$
\boldsymbol{\xi}^+ = \boldsymbol{\varphi}(\boldsymbol{x}, \Omega^e_+).
\tag{20}
$$

Therefore, we can explicitly state that an arbitrary physical flux $\mathcal{F}^*_h$ is a function of arbitrary quantities b, evaluated at the same physical coordinates $\boldsymbol{x} = \varphi^{-1}(\boldsymbol{\xi}^-, \Omega^e_-)$ (provided by the integration rule of the corresponding primary element face)

$$
\mathcal{F}^*_h(b^-, b^+) = \mathcal{F}^*_h\left(b^-(\boldsymbol{\xi}^-), b^+(\boldsymbol{\xi}^+)\right).
\tag{21}
$$

The integration over a non-conforming face of a primary element subsequently reads

$$
\int_{\partial\Omega} \mathcal{F}^*(b^-, b^+)\, \mathrm{d}\Gamma_{\mathrm{NCI}-} \approx \sum_{q=1}^{n_q} w_q \mathcal{F}^*_h(b^-(\boldsymbol{\xi}^-), b^+(\boldsymbol{\xi}^+))|J_q|.
\tag{22}
$$

w_q are the weights of the Gauss quadrature, and the Jacobi determinants $|J_q|$ in quadrature points q correspond to the primary element face.
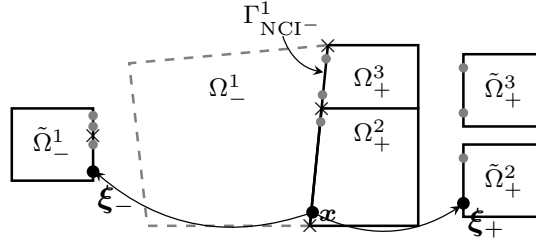
Figure 2: Nitsche-type mortaring: In contrast to point-to-point interpolation, quadrature points are not dictated by the primary element. Instead, the mortars between elements are computed, and quadrature points, weights, and Jacobians correspond to the mortar patches. This way, there is no aliasing, and values form a smooth representation in each quadrature. Intersections are computed between the face of the primary element and the secondary volume elements to ensure the method works for overlaps without modification.

3.4   Nitsche-Type Mortaring

In contrast to point-to-point interpolation, Nitsche-type mortaring computes the integration over primary element faces at NCIs as the sum of collected quadrature rules on the mortars, see Figure 2. Thus the integral computes as

$$\int_{\partial\Omega} \mathcal{F}^*(b^-, b^+) \, d\Gamma_{\text{NCI}^-} \approx \sum_{m=1}^{n_m} \sum_{q=1}^{n_q} w_q \mathcal{F}_h^*(b^-(\boldsymbol{\xi}^-), b^+(\boldsymbol{\xi}^+)) |J_q^m|. \tag{23}$$

In (23), n_m is the number of found intersections. The Jacobi determinants $|J_q^m|$ in quadrature point q is now determined on mortar m. This way, the integral on the NCI is computed exactly (if enough quadrature points n_q are chosen) without aliasing. Constructing the mortars is more challenging to implement and reduces performance [12]. In our formulation, intersections are computed between the face of a primary element and the secondary volume elements, cf. Figure 2; in 2D simulations, intersections are computed between a quadrilateral and a line. This way, we can seamlessly handle overlaps between elements (see Section 5.1.3).

4   Remarks on Implementation

Our implementations will be freely available as a part of the open source software project `EXADG` [28] and the software library `deal.II` [29].

In the case of point-to-point interpolation, we are collecting all quadrature points on the NCIs mapped to the physical space. After that, we perform a global search based on distributed bounding boxes for secondary elements that hold these integration points and store the corresponding quadrature points in the reference space. In each time step, we evaluate pressure and velocities on the secondary elements and use the results to compute the fluxes over the NCIs. If a quadrature point is found on multiple elements, we use the average value in the computation of the fluxes. This approach works on curved elements without further ado.

In the case of Nitsche-type mortaring, we first create the mortars between the primary and secondary elements. We are computing the $d-1$ dimensional intersections between d and $d-1$ geometric entities. The $d-1$ element is a face of a primary element. This way, quadrature rules are defined on the primary element faces, independent if elements overlap or not. Thus, our implementation is the same in case of element overlaps or standard NCIs. In our implementations, we extract the vertices of the non-conforming faces of the primary elements and all vertices of possibly connected secondary elements. We then use `CGAL` [30] to compute the inter-dimensional intersections and eventually create mapped quadrature rules on each mortar patch. The rest of the implementation is nearly the same as for the point-to-point interpolation: Additionally to the quadrature points, we store the Jacobi determinants at the quadrature points. Normal vectors are not stored; instead, we use the negative normal vectors of the primary element during flux evaluation. We evaluate pressure and velocity on the primary and secondary elements in each time step and use the stored Jacobi determinants to compute and test the fluxes over the NCIs. This approach limits us to non-curved elements at NCIs. Since `CGAL` is working with triangular/tetrahedral elements, mortar patches are always triangular in the 3D case (even if the patch could be rectangular). Note that the number of created mortars
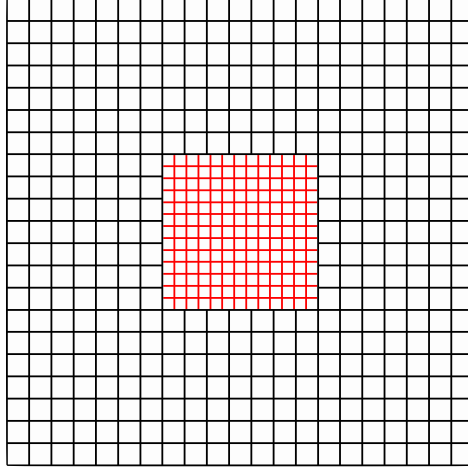
Figure 3: Computational mesh consisting of two mesh regions which are connected via a non-conforming interface.

and thus the number of quadrature points highly depends on the element configuration.

5 Numerical Results

This section provides numerical results using point-to-point interpolation and Nitsche-type mortaring. First, we show instabilities related to non-smooth representations of values at NCIs for the point-to-point interpolation method. We show that these instabilities do not occur if we use Nitsche-type mortaring instead. For Nitsche-type mortaring, we provide in-depth convergence results for different mesh regions, quantify the error introduced at the NCI, and provide results for a test case with heterogeneous material. Additionally, we show that the method also works if elements are overlapping. From now on, all spatial values are given in m.

5.1 Vibrating Membrane

To be able to compute errors exactly, we use the test case of a vibrating membrane which has also been used, amongst others, in [24, 31]. For this test case, the analytical solution at each time t is known, and for a two-dimensional domain, it reads for the pressure

$$p = \cos(M\sqrt{2}\pi t)\sin(M\pi x)\sin(M\pi y), \tag{24}$$

and for the acoustic particle velocity

$$\boldsymbol{u} = \frac{-\sin(M\sqrt{2}\pi t)}{\sqrt{2}} \begin{pmatrix} \cos(M\pi x)\sin(M\pi y) \\ \sin(M\pi x)\cos(M\pi y) \end{pmatrix}, \tag{25}$$

assuming no acoustic loads $F = 0 \text{ kg} \cdot \text{m}^{-3} \cdot \text{s}^{-1}$, as well as $\rho = 1 \text{ kg} \cdot \text{m}^{-3}$ and $c = 1 \text{ m} \cdot \text{s}^{-1}$. Our simulations' computational domain Ω consists of two mesh regions connected via NCIs, as shown in Figure 3. The outer region Ω_o is a rectangular domain with a rectangular hole in which the inner region Ω_i is embedded. Thus, $\Omega = \Omega_\text{o} \cup \Omega_\text{i}$ and within the following $\Omega_\text{o} = (0, 0) \times (0.1, 0.1) \setminus \Omega_\text{i}$ and $\Omega_\text{i} = (1/30, 1/30) \times (2/30, 2/30)$. We use $M = 120$ modes, which leads to $p = 0$ Pa at the boundaries of the computational domain Ω and we apply homogenous pressure DBCs $g_p = 0$ Pa. All computations use the low storage Runge–Kutta version RKC84 [32]. The CFL condition

$$\Delta t = \frac{\text{Cr}}{k^{1.5}} \frac{h_\text{min}}{c_\text{max}}, \tag{26}$$

gives the required time step size Δt needed for a stable temporal discretization. We use the minimum edge length h_min of all existing elements, the polynomial degree k, and the largest value of the speed of sound c_max for its computation. To account for the different spacing between Legendre–Gauss–Lobatto (used as quadrature points) we are using the

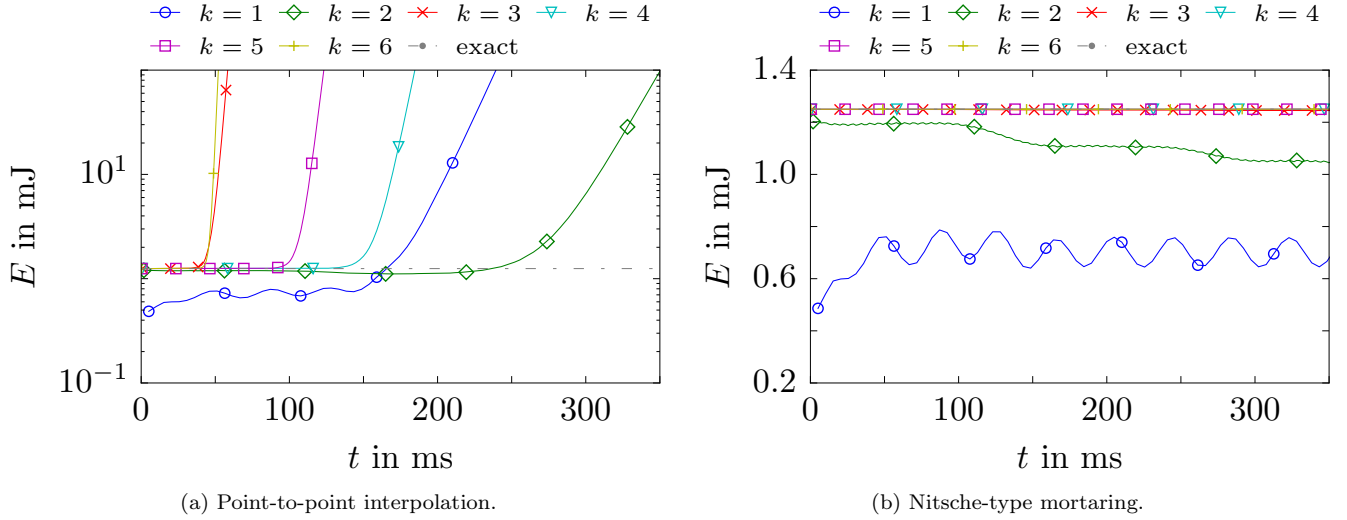(a) Point-to-point interpolation.   (b) Nitsche-type mortaring.

Figure 4: Sound energy over time computed for orders $k = 1$ to $k = 6$. The given setup for the vibrating membrane test case is perfectly energy conservative and thus energy has to be constant over time. For point-to-point interpolation, instabilities form after some time while the simulation stays stable for exact integration with Nitsche-type mortaring.

superscript 1.5 as proposed in [33]. For a detailed discussion on the CFL condition for explicit Runge–Kutta methods, we refer to [34]. From this point forward, all computations use time step sizes computed by the CFL condition with a Courant number $\mathrm{Cr} = 0.2$.

5.1.1 Instabilities

The test case perfectly conserves the sound energy

$$E = \int_\Omega \left(\frac{p^2}{2\rho c^2} + \frac{\rho(\boldsymbol{u} \cdot \boldsymbol{u})}{2} \right) \mathrm{d}\Omega. \tag{27}$$

Since the analytic solution exists, we can compute the exact sound energy contained in the system as $E_{\mathrm{exact}} = 1.25$ mJ. The mesh (cf. Figure 3) has element edge lengths of $h_{\Omega_{\mathrm{i}}} = {}^1\!/_{30.13}$ on the NCI for the inner domain and $h_{\Omega_{\mathrm{i}}} = {}^1\!/_{30.7}$ on the NCI for the outer domain. Figure 4a shows the sound energy in the system over time for point-to-point interpolation. After a certain time, instabilities manifest as an non-physical rapid increase of sound energy.

Obviously, the approach suffers from aliasing; the integration of the primary elements only includes information from each connected element if quadrature points are found in every element. One can regard this as a Dirichlet-Dirichlet approach, where the values are picked from the secondary domain instead of, e.g., an analytical function. This reasoning does not explain the observed instabilities.

The difference in the applied Dirichlet boundary values is that in the case of an analytical function, the boundary values form a smooth representation of the solution. In the case of using values from the secondary domain, boundary values are not necessarily smooth. If quadrature points are located in different elements, we might face jumps in the solution representation. While these jumps are assumed to be less distinctive in the case of continuous FE methods, the nature of the DG method intensifies this issue. Nevertheless, the same also happens in the case of continuous FE methods, in the case where whole secondary elements are not sampled by any quadrature point of the primary element. The jumps between Dirichlet values introduce spurious oscillations that eventually lead to unstable simulations. To quantify that this is indeed the source of instabilities, we tested to interpolate solution values between domains into the DoFs. This way, there are no jumps between quadrature points since the shape function of the primal element enforces a smooth representation of the values. Nevertheless, we observed instabilities in the case of high polynomial degrees. These instabilities are related to Gibbs' phenomena. Significant differences

between DoFs and the high-order shape functions lead to ringing modes, typically observed in shock capturing. Applying techniques to interpolated values that are usually used in shock capturing, such as modal filtering [16], lead to stable schemes. However, this also leads to a drop in the obtained spatial convergence rates; therefore, we are not further discussing those approaches.

According to the previous discussion, the phenomenon is not expected in the case of Nitsche-type mortaring. Since the integrals are evaluated on intersections between elements, the representation of values is forced to be continuous in each quadrature rule. The results are depicted in Figure 4b; we do not observe instabilities. From this point forward, we will only consider the Nitsche-type mortaring approach.

Note that the main focus of the test case is to show instabilities for any polynomial degree. However, the mesh resolution for low polynomial degrees k is chosen too poor for a good approximation of the primal variables. According to [35], the element size h has to be chosen such that

$$k + \frac{1}{2} > \frac{\omega h}{2} + C(\omega h)^{1/3}. \tag{28}$$

In (28), C is a constant that can be chosen unity in practice [35] and ω is the wave-number. For the vibrating membrane test case $\omega = 2\pi M$. Using the maximum element size in used triangulation, we obtain $\frac{\omega h}{2} + (\omega h)^{1/3} \approx 3.44$ for current investigations. Thus, we fulfill (28) with polynomial degrees $k > 2$, for $k = 2$ we are slightly off, and for $k = 1$ we have a substantial deviation. Consequently, we can see constant sound energy over time for orders greater than $k = 2$. For $k = 2$, we see a slight drop in energy due to numerical dispersion originating from the slightly too coarse resolution. For $k = 1$, the resolution is so poor that the sound energy can not be resolved from the beginning, and we observe non-physical oscillations. These results are in accordance to [35], in which it is reported that results might even become qualitatively incorrect for insufficient resolutions. Within the next sections the mesh resolutions are chosen such that (28) is fulfilled for $k = 1$ which successfully removes any non-physical oscillations.

5.1.2   Convergence Results

We use the setup of the previous section with $M = 30$ modes. However, we alter the mesh sizes compared to Figure 3. The elements in the outer domain have initial edge lengths of $h_{\text{o,initial}} = 1/30 \cdot 2$; in the inner domain initial element edge lengths are $h_{\text{i,initial}} = 1/30 \cdot 3$. We compute the relative $L2$ error for the pressure $\mathcal{E}_{p,\mathcal{R}}^{\text{L2,rel}}$ on region $\mathcal{R}$ after 1 s for different mesh refinements

$$\mathcal{E}_{p,\mathcal{R}}^{\text{L2,rel}} = \frac{\sqrt{\int_{\mathcal{R}} (p_h - p_{\text{ana}})^2 \, \mathrm{d}\Omega}}{\sqrt{\int_{\mathcal{R}} p_{\text{ana}}^2 \, \mathrm{d}\Omega}}, \tag{29}$$

with the analytical solution of the pressure p_{ana}, see Eq. (24). The velocity error $\mathcal{E}_{\boldsymbol{u},\mathcal{R}}^{\text{L2,rel}}$ is computed accordingly. Regions are either the global region Ω, the inner region Ω_{i}, or the outer region Ω_{o}. The mesh refinement is realized by replacing each quadrilateral cell by four children cells, and the corresponding edge lengths h at refinement level r compute as

$$h = \frac{h_{\text{initial}}}{2^r}. \tag{30}$$

We observe optimal convergence rates of order $k + 1$ in space on the global domain Ω, see Figure 5a. The outer domain has a coarser spatial discretization and dominates the errors on the global domain. Therefore, it is not surprising that the errors on the outer domain (Figure 5b) behave similar to the ones on the global domain (Figure 5a). The inner domain has a finer spatial discretization; thus the errors obtained in the inner domain might be shadowed by the errors obtained in the outer domain. However, errors propagate from the outer domain to the inner domain. We also observe optimal convergence rates computing the errors on the inner domain (see Figure 5c). Errors obtained in the inner and outer domain are similar. Thus errors from the outer domain entirely propagated to the inner domain after 1 s. Therefore, in practical applications, one should aim to choose mesh sizes that yield approximately the same errors in each domain. In conclusion, we obtain optimal convergence rates in all regions of the non-conforming mesh and can confidently apply the proposed method, keeping in mind that a jump in element sizes has to be justified, e.g., due to different materials.
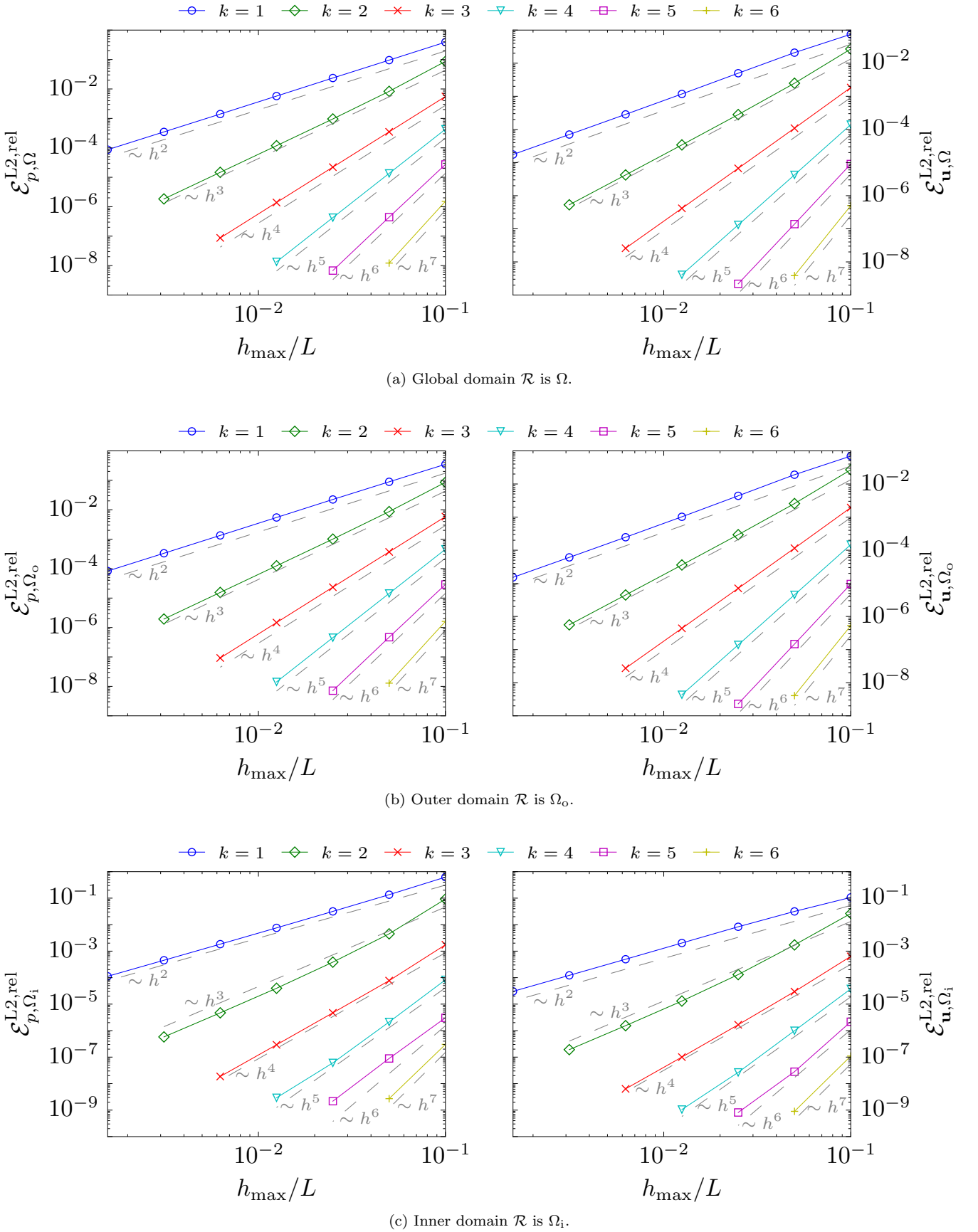
(a) Global domain $\mathcal{R}$ is Ω.

(b) Outer domain $\mathcal{R}$ is Ω_{o}.

(c) Inner domain $\mathcal{R}$ is Ω_{i}.

Figure 5: Spatial convergence study for the vibrating membrane testcase with $M = 30$, defined on the rectangular domain of Figure 3 (with $h_{\mathrm{i,initial}} = {}^{1}/{}_{30\cdot3}$, and $h_{\mathrm{o,initial}} = {}^{1}/{}_{30\cdot2}$) using Nitsche-type mortaring: Shown are the relative L2 errors for pressure $\mathcal{E}_{p,\mathcal{R}}^{\mathrm{L2,rel}}$ and velocity $\mathcal{E}_{\boldsymbol{u},\mathcal{R}}^{\mathrm{L2,rel}}$ on different domains $\mathcal{R}$. The domain $\mathcal{R}$ might be the global domain Ω, the inner domain Ω_{i}, or the outer domain Ω_{o}.

| (a) Curved interface | (b) Small overlap | (c) Overset mesh |
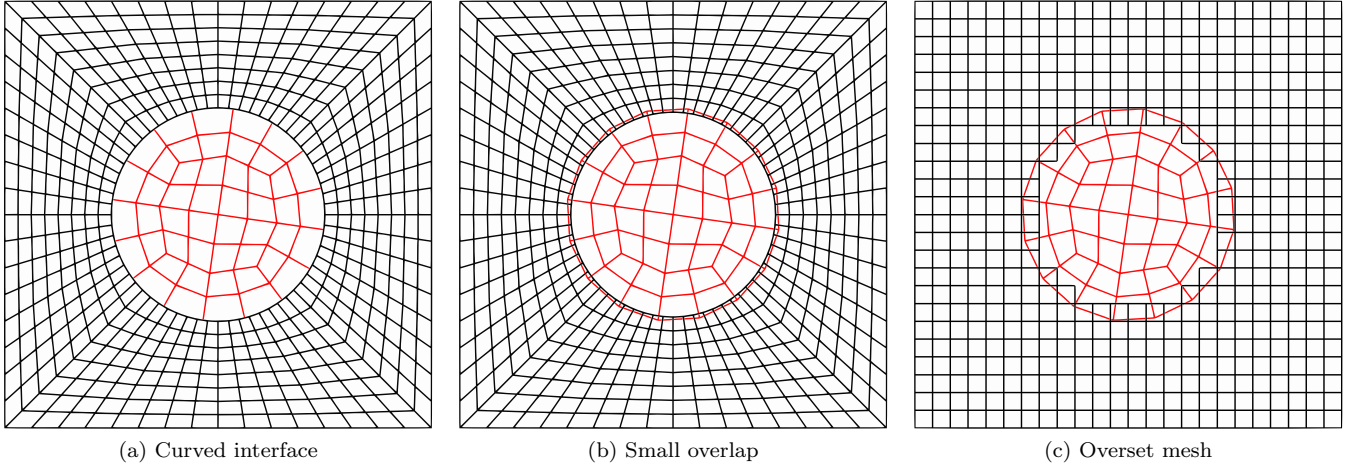
Figure 6: Three different meshes that have a circular mesh embedded in a rectangular mesh.

5.1.3 Embedding of Circular Domain

Being able to handle overlaps has two useful properties. Mesh generation gets more straightforward, and rotating interfaces can be handled without the need for curved elements. We provide results for three different grids, depicted in Figure 6, that are prototypical in the context of rotating interfaces. The rectangular domain spans $\Omega_o = [0, 0.1]^2$ and the circular domain Ω_i has a radius of 0.5.

In this particular case, it is easily possible to manually compute quadrature rules on the curved intersections since the NCI is a circle (cf. Figure 6a). Note that this does not work for arbitrary shapes in our implementations since we rely on `CGAL` to compute the intersections. Nevertheless, this approach becomes relevant for large-scale computations with sliding interfaces since the computational cost to create mortars is heavily reduced, cf. [36]. For the version with a slight overlap (cf. Figure 6b), the radius of the hole is slightly smaller than the radius of the circular domain, i.e., $0.5 - 2 \cdot 10^{-3}$. Using overset meshes (cf. Figure 6c) is particularly helpful in generating structured meshes in regions connected to complex geometries.

All meshes have similar numbers of DoFs. Note that the methodology works for arbitrary overlaps. However, the same physical fields are computed in the overlap; thus, redundant work is done if the overlap exceeds one element. Table 1 shows the errors obtained after 1 s for the vibrating membrane test case with $M = 5$ modes. In this case, we apply inhomogeneous pressure DBCs with g_p obtained from the analytical solution. We can see that the errors are in the same order of magnitude for the overlapping and overset mesh. Even though we used fewer DoFs in the overset mesh, we can see slightly better errors, with an outlier at polynomial degree $k = 5$. This relates to the element distortions in the overlapping case. The curved interface setup produces more significant errors than the overlapping setup, the most distinct deviations are for polynomial degree $k = 3$ and $k = 4$. This is not expected and needs further investigation before application to sliding rotating interfaces. One possible explanation is that

Table 1: Relative L^2 errors $\mathcal{E}_\Omega^{\mathrm{L2,rel}} = \mathcal{E}_{p,\Omega}^{\mathrm{L2,rel}} + \mathcal{E}_{\boldsymbol{u},\Omega}^{\mathrm{L2,rel}}$ for different polynomial degrees k computed on the meshes depicted in Figure 6.

	Curved interface		Overlap		Overset	
k	**DoFs**	$\mathcal{E}_\Omega^{\mathrm{L2,rel}}$	**DoFs**	$\mathcal{E}_\Omega^{\mathrm{L2,rel}}$	**DoFs**	$\mathcal{E}_\Omega^{\mathrm{L2,rel}}$
1	6,720	$2.915 \cdot 10^{-2}$	6,720	$2.779 \cdot 10^{-2}$	6,432	$2.053 \cdot 10^{-2}$
2	15,120	$8.303 \cdot 10^{-4}$	15,120	$5.444 \cdot 10^{-4}$	14,472	$4.641 \cdot 10^{-4}$
3	26,880	$4.304 \cdot 10^{-5}$	26,880	$8.536 \cdot 10^{-6}$	25,728	$7.428 \cdot 10^{-6}$
4	42,000	$1.666 \cdot 10^{-6}$	42,000	$2.220 \cdot 10^{-7}$	40,200	$2.200 \cdot 10^{-7}$
5	60,480	$6.974 \cdot 10^{-8}$	60,480	$2.328 \cdot 10^{-9}$	57,888	$2.367 \cdot 10^{-9}$
6	82,320	$2.650 \cdot 10^{-9}$	82,320	$1.286 \cdot 10^{-9}$	78,792	$1.096 \cdot 10^{-9}$

round-off errors are introduced while computing the curved intersections.

Overall, we conclude that our methodology works as expected if elements overlap.
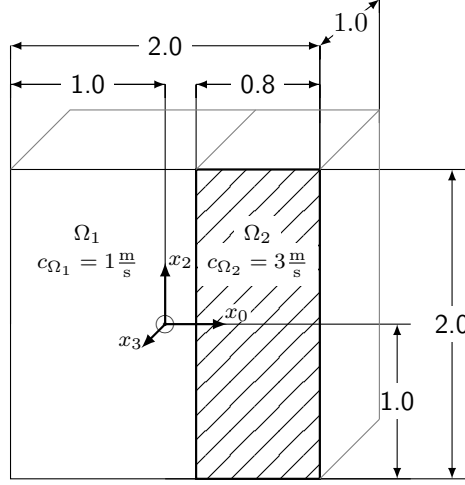
5.2 Application

Figure 7: Application: Domain with heterogeneous fluids clipped in x_1-x_2 plane. The speed of sound in both fluids differs, while the density is $\rho_{\Omega_1} = \rho_{\Omega_2} = 1 \text{ kg} \cdot \text{m}^{-3} = \text{const}$. A pressure pulse $\odot$ is located in the center of the domain as an initial condition.

As pointed out, NCIs are especially desirable if different spatial resolutions are required. Imagine two fluids with different speeds of sound c. We need different element sizes to resolve the acoustic pressure up to a specific frequency. We use the test case with heterogeneous acoustic material, also simulated by [37, 38, 39]. We adapt the computational domain to show that our implementations work in the 3D case. A wave travels over the interface between two materials. At the interface, the wave is partially transmitted and partially reflected, and an additional wavefront emerges due to the Huygens–Fresnel principle. A sectional view of the setup for this test case is depicted in Figure 7. The domain $\Omega = \Omega_1 \cup \Omega_2$ spans from $\Omega = (-1, -1, -1) \times (1, 1, 1)$. In the left part of the domain the speed of sound is $c_{\Omega_1} = 1 \text{ m} \cdot \text{s}^{-1}$ while it is $c_{\Omega_2} = 3 \text{ m} \cdot \text{s}^{-1}$ in the right part. The density of both fluids is $\rho_{\Omega_1} = \rho_{\Omega_2} = 1 \text{ kg} \cdot \text{m}^{-3}$. As an initial condition, a pressure pulse is chosen

$$p(t = 0) = \text{e}^{-10^4 \, \boldsymbol{x} \cdot \boldsymbol{x}}, \tag{31}$$

$$\boldsymbol{u}(t = 0) = \boldsymbol{0}. \tag{32}$$

The test case is subject to homogenous pressure BCs.

In the right domain, we use element sizes that are three times as big compared to the left domain to resolve both domains up to the same frequency. In the left domain we use elements with maximum edge length $h_{\max,\Omega_1} = 0.0167$ and accordingly $h_{\max,\Omega_2} = 0.05$. The used polynomial degree is $k = 3$. The pressure field at different times can be seen in Figure 9.

To quantify the effect of the NCI we also run the simulations on a domain with $h_{\max,\Omega_1} = h_{\max,\Omega_2} = 0.0167$ and $h_{\max,\Omega_1} = h_{\max,\Omega_2} = 0.05$. We record the pressure at 1000 points along x_1, $x_2 = 0$, $x_3 = 0$ at $t = 0.2$ s. The discretization with the smallest mesh size $h_{\max,\Omega_1} = h_{\max,\Omega_2} = 0.0167$ serves as a reference. It is supposed to produce the most accurate solution but uses too many DoFs if we want to resolve the same frequencies in both fluids.

The recorded pressure profile is plotted in Figure 8a. Figure 8b shows a detailed view around the interface. We observe great differences to the reference for the discretization with the biggest mesh size $h_{\max,\Omega_1} = h_{\max,\Omega_2} = 0.05$. However, using the biggest and smallest mesh size for the different regions, employing the non-conforming formulation, gives a result that is in good agreement with the reference solution. In this case, the finest domain has $442,368,000$ DoFs while the domain with different element sizes has $271,974,400$ DoFs, i.e. the problem size

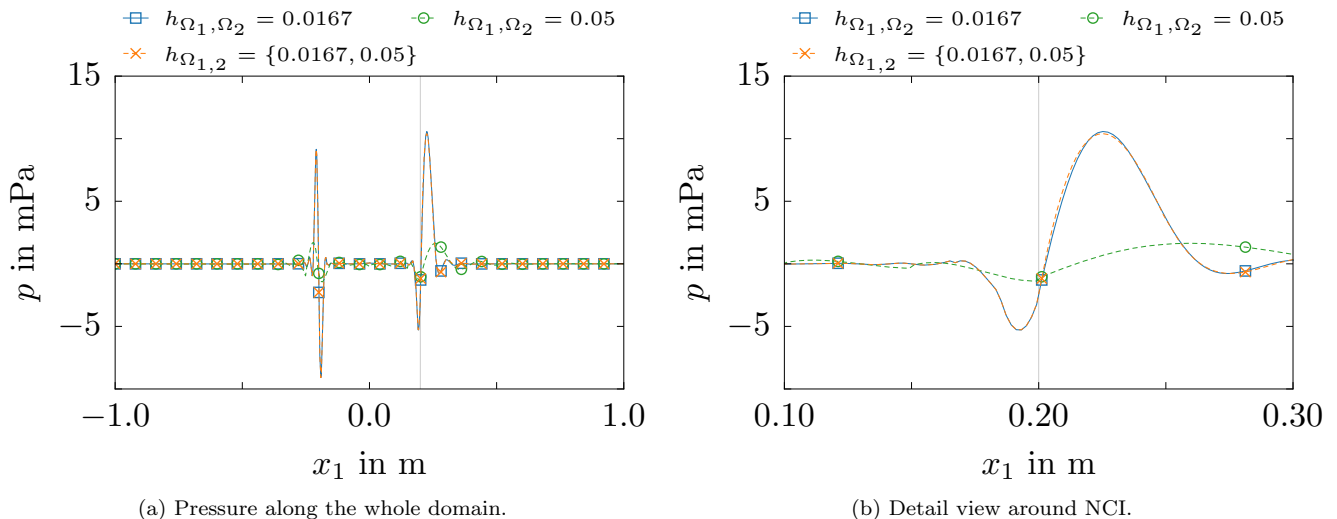(a) Pressure along the whole domain.   (b) Detail view around NCI.

Figure 8: Application: Pressure values along x_1 at $x_2 = 0, x_3 = 0$ and at $t = 0.2$ s. The position of the NCI is indicated by the vertical line.

is reduced approximately by 40% in comparison the fine problem while keeping the same accuracy. This highly encourages to use NCIs for this kind of problems to effectively reduce the number of DoFs.

6 Conclusion

Using Nitsche-type mortaring, we proposed a stable non-conforming DG discretization for the acoustic conservation laws. We showed that point-to-point interpolation is unsuitable in this setting since it introduces errors related to non-smooth representations of values in quadrature rules. Therefore, we can not avoid the expensive computations of element intersections between primary and secondary elements.

The proposed method collects integration rules on the intersections between secondary volume elements and facets of primary elements. This way, the method naturally extends to overlapping elements and is a perfect starting point for problems with rotating interfaces. The method is subject to optimal spatial convergence rates. Measuring the error region-wise, we can show that the method converges optimally in all sub-domains. Nevertheless, errors are propagating in the domain; therefore, optimal spatial convergence can only be applied in a meaningful way if triangulations are constructed such that errors are of the same magnitude in all parts of the domain. Thus, we recommend using element sizes that resolve the same frequencies in all sub-domains in acoustics. With an application, we demonstrated that this procedure efficiently reduces needed degrees of freedom while maintaining accuracy.
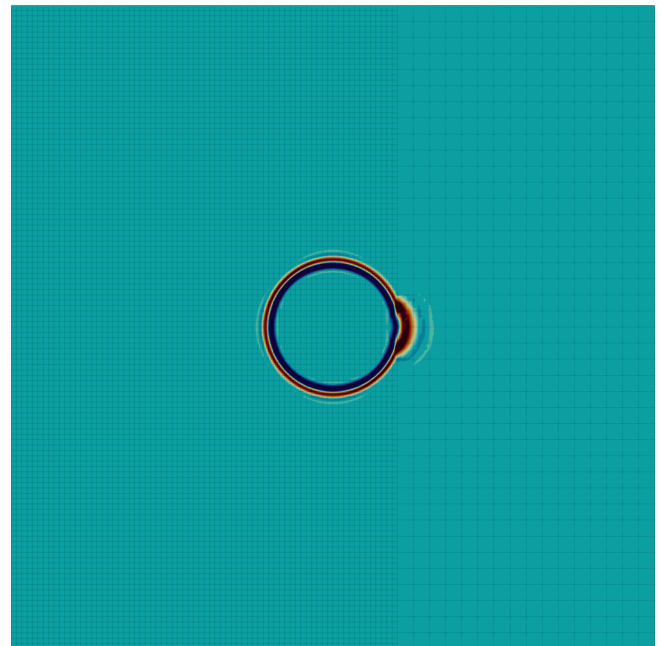
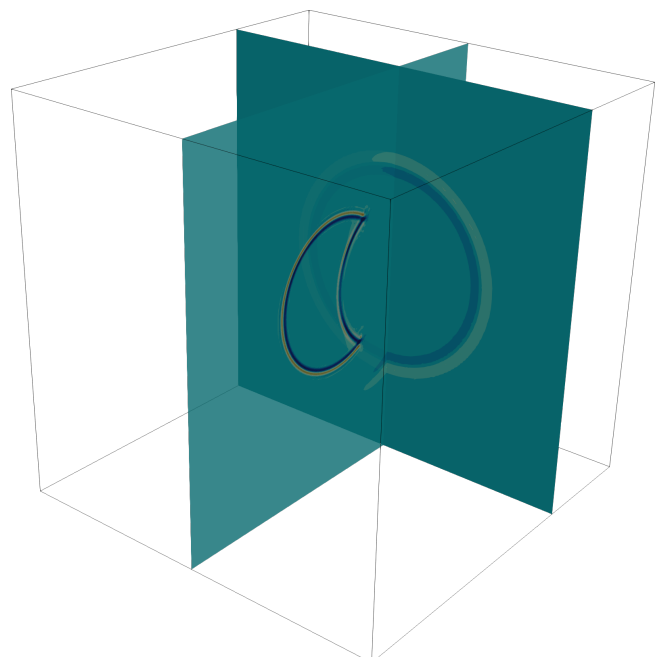## Acknowledgements
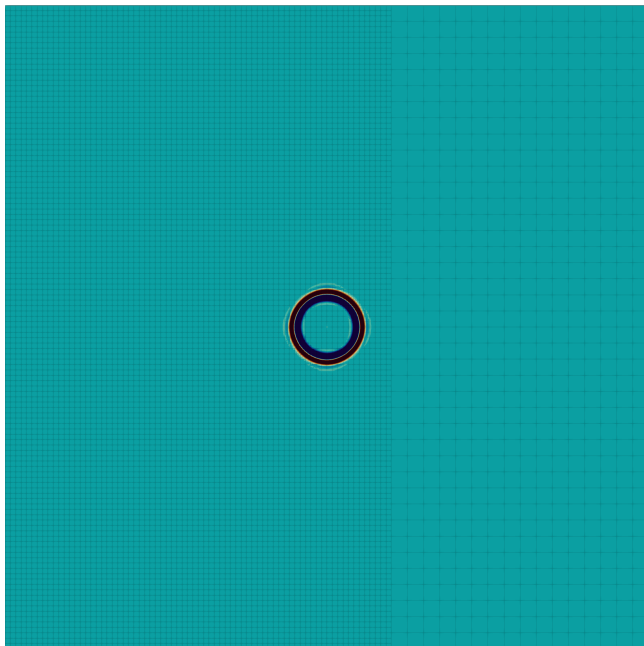
(a) x_1-x_2 plane, $t = 0.1$ s.

(b) x_1-x_2 plane, $t = 0.2$ s.
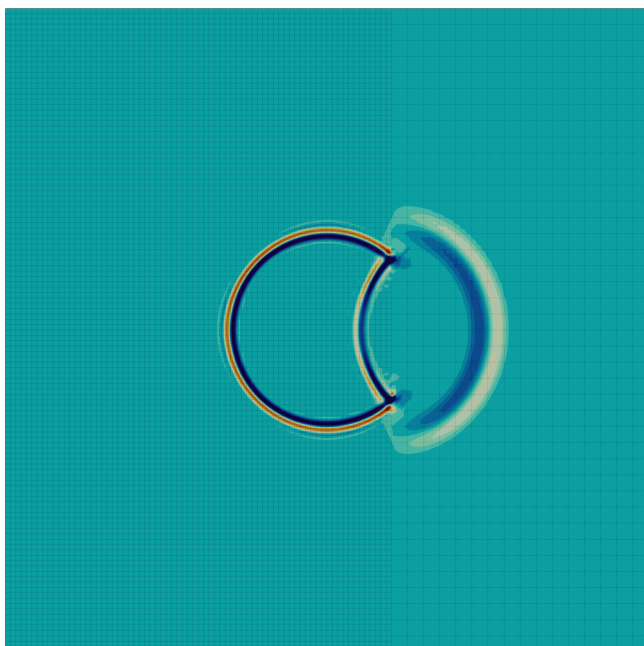
(c) x_1-x_2 plane, $t = 0.3$ s.

(d) 3D view, $t = 0.38$ s.

Figure 9: Application: Snapshot of acoustic pressure at different times. At $t = 0.3$ s the transmitted, the reflected and the Huygens wave can be seen.

References

[1] Quiroz L, Beckers P. Non-conforming mesh gluing in the finite elements method. *International Journal for Numerical Methods in Engineering* 1995; 38(13): 2165–2184. doi: 10.1002/nme.1620381303

[2] Kaltenbacher M. *Numerical simulation of mechatronic sensors and actuators: Finite elements for computational multiphysics.* Heidelberg, New York, Dordrecht, London: Springer. 3rd ed. 2015

[3] Flemisch B, Kaltenbacher M, Wohlmuth BI. Elasto-acoustic and acoustic-acoustic coupling on non-matching grids. *International Journal for Numerical Methods in Engineering* 2006; 67(13): 1791–1810. doi: 10.1002/nme.1669

[4] Benek J, Buning P, Steger J. In: .

[5] Schoder S, Junger C, Kaltenbacher M. Computational aeroacoustics of the EAA benchmark case of an axial fan. *Acta Acustica* 2020; 4(5): 22. doi: 10.1051/aacus/2020021

[6] Pezzano S, Duvigneau R. A fully-conservative sliding grid algorithm for compressible flows using an isogeometric discontinuous Galerkin scheme. *Computer Methods in Applied Mechanics and Engineering* 2022; 395: 115000. doi: 10.1016/j.cma.2022.115000

[7] Bernardi C, Debit N, Maday Y. Coupling finite element and spectral methods: First results. *Mathematics of Computation* 1990; 54(189): 21–39. doi: 10.1090/s0025-5718-1990-0995205-7

[8] Deparis S, Forti D, Gervasio P, Quarteroni A. INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces. *Computers & Fluids* 2016; 141: 22–41. doi: 10.1016/j.compfluid.2016.03.033

[9] Nitsche J. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 1971; 36(1): 9–15. doi: https://doi.org/10.1007/BF02995904

[10] Arnold D, Brezzi F, Cockburn B, Marini L. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2002; 39(5): 1749-1779. doi: 10.1137/S0036142901384162

[11] Hermann V, Käser M, Castro CE. Non-conforming hybrid meshes for efficient 2-D wave propagation using the discontinuous Galerkin method. *Geophysical Journal International* 2010; 184(2): 746–758. doi: 10.1111/j.1365-246x.2010.04858.x

[12] Laughton E, Tabor G, Moxey D. A comparison of interpolation techniques for non-conformal high-order discontinuous Galerkin methods. *Computer Methods in Applied Mechanics and Engineering* 2021; 381: 113820. doi: 10.1016/j.cma.2021.113820

[13] Kaltenbacher M, Floss S. Nonconforming finite elements based on Nitsche-type mortaring for inhomogeneous wave equation. *Journal of Theoretical and Computational Acoustics* 2018; 26(03): 1850028. doi: 10.1142/s2591728518500287

[14] Roppert K, Schoder S, Toth F, Kaltenbacher M. Non-conforming Nitsche interfaces for edge elements in curl–curl-type problems. *IEEE Transactions on Magnetics* 2020; 56(5): 1–7. doi: 10.1109/TMAG.2020.2980477

[15] Cohen GC. *Higher-Order Numerical Methods for Transient Wave Equations.* Berlin, Heidelberg: Springer. 1st ed. 2002

[16] Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods.* New York, NY: Springer . 2008

[17] Heinz J, Kaltenbacher M. A nonconforming DG formulation for the acoustic conservation laws. In: Leistner P. , ed. *Fortschritte der Akustik - DAGA 2022*, PA: DEGA; 2022: 142 – 145.

[18] Bassi F, Crivellini A, Pietro DD, Rebay S. An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations. *Journal of Computational Physics* 2006; 218(2): 794 - 815. doi: https://doi.org/10.1016/j.jcp.2006.03.006

[19] Bassi F, Crivellini A, Pietro DAD, Rebay S. An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Computers & Fluids* 2007; 36(10): 1529 - 1546. doi: https://doi.org/10.1016/j.compfluid.2007.03.012

[20] Engquist B, Majda A. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation* 1977; 31(139): 629–651. doi: 10.1090/s0025-5718-1977-0436612-4

[21] Hüppe A. *Spectral Finite Elements for Acoustic Field Computation.* Alpen-Adria-Universität Klagenfurt, Austria: Phd Thesis . 2012.

[22] Heinz J, Kaltenbacher M. Comparison of discontinuous Galerkin formulations for the acoustic conservation laws. In: Waubke H, Balazs P. , eds. *Fortschritte der Akustik - DAGA 2021*, PA: DEGA; 2021: 904 – 907.

[23] Hochbruck M, Pažur T, Schulz A, Thawinan E, Wieners C. Efficient time integration for discontinuous Galerkin approximations of linear wave equations. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 2014; 95(3): 237–259. doi: 10.1002/zamm.201300306

[24] Nguyen N, Peraire J, Cockburn B. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics* 2011; 230(10): 3695 - 3718. doi: https://doi.org/10.1016/j.jcp.2011.01.035

[25] Kronbichler M, Schoeder S, Müller C, Wall WA. Comparison of implicit and explicit hybridizable discontinuous Galerkin methods for the acoustic wave equation. *International Journal for Numerical Methods in Engineering* 2016; 106(9): 712-739. doi: 10.1002/nme.5137

[26] Wang H, Sihar I, Muñoz RP, Hornikx M. Room acoustics modelling in the time-domain with the nodal discontinuous Galerkin method. *The Journal of the Acoustical Society of America* 2019; 145(4): 2650–2663. doi: 10.1121/1.5096154

[27] LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems.* Cambridge: Cambridge University Press . 2002

[28] Arndt D, Fehn N, Kanschat G, et al. ExaDG: High-order discontinuous Galerkin for the exa-scale. In: Bungartz HJ, Reiz S, Uekermann B, Neumann P, Nagel WE., eds. *Software for Exascale Computing - SPPEXA 2016-2019*, PA: Springer International Publishing; 2020; Cham: 189–224.

[29] Arndt D, Bangerth W, Feder M, et al. The `deal.II` library, version 9.4. *Journal of Numerical Mathematics* 2022. doi: 10.1515/jnma-2022-0054

[30] The CGAL Project . *CGAL User and Reference Manual.* CGAL Editorial Board. 5.5 ed. 2022.

[31] Schoeder S, Wall W, Kronbichler M. ExWave: A high performance discontinuous Galerkin solver for the acoustic wave equation. *SoftwareX* 2019; 9: 49 - 54. doi: https://doi.org/10.1016/j.softx.2019.01.001

[32] Toulorge T, Desmet W. Optimal Runge–Kutta schemes for discontinuous Galerkin space discretizations applied to wave propagation problems. *Journal of Computational Physics* 2012; 231(4): 2067–2091. doi: 10.1016/j.jcp.2011.11.024

[33] Fehn N, Wall WA, Kronbichler M. Robust and efficient discontinuous Galerkin methods for under-resolved turbulent incompressible flows. *Journal of Computational Physics* 2018; 372: 667 - 693. doi: 10.1016/j.jcp.2018.06.037

[34] Toulorge T, Desmet W. CFL Conditions for Runge–Kutta discontinuous Galerkin methods on triangular grids. *Journal of Computational Physics* 2011; 230(12): 4657–4678. doi: 10.1016/j.jcp.2011.02.040

[35] Ainsworth M. Discrete dispersion relation for hp-version finite element approximation at high wave number. *SIAM Journal on Numerical Analysis* 2004; 42(2): 553–575. doi: 10.1137/s0036142903423460

[36] Dürrwächter J, Kurz M, Kopper P, Kempf D, Munz CD, Beck A. An efficient sliding mesh interface method for high-order discontinuous Galerkin schemes. *Computers & Fluids* 2021; 217: 104825. doi: 10.1016/j.compfluid.2020.104825

[37] Bangerth W, Geiger M, Rannacher R. Adaptive Galerkin finite element methods for the wave equation. *Computational Methods in Applied Mathematics* 2010; 10(1): 3–48. doi: 10.2478/cmam-2010-0001

[38] Köcher U, Bause M. Variational space–time methods for the wave equation. *Journal of Scientific Computing* 2014; 61(2): 424–453. doi: 10.1007/s10915-014-9831-3

[39] Perugia I, Schöberl J, Stocker P, Wintersteiger C. Tent pitching and Trefftz–DG method for the acoustic wave equation. *Computers & Mathematics with Applications* 2020; 79(10): 2987–3000. doi: 10.1016/j.camwa.2020.01.006