Journal of Computational Physics 567 (2026) 115302

Contents lists available at ScienceDirect

Journal of Computational Physics

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

CutGDM: High-order cut Galerkin difference methods for the linear advection equation

Gunilla Kreiss [a], Peter Munch [b,*], Siyang Wang [c]

[a] Division of Scientific Computing, Department of Information Technology, Uppsala University, Sweden
[b] Institute of Mathematics, Technical University of Berlin, Germany
[c] Department of Mathematics and Mathematical Statistics, Umeå University, Sweden

ARTICLE INFO

Keywords:
Immersed methods
Galerkin difference method
Advection equation

ABSTRACT

The Galerkin difference method is a type of finite element methods where Lagrange shape functions are piecewise but span over multiple cells. With this, it allows to choose time-step sizes in explicit temporal methods independently of the polynomial order of the shape functions, which is a strong benefit compared to the standard finite element method, where the time-step sizes need to be reduced with increasing polynomial degree to maintain stability. This property has been proven to be valid for advection and wave equations. In this work, we extend the methodology for the linear advection equation on immersed domains, i.e., consider cut Galerkin difference methods. We analytically and numerically show that Galerkin difference methods are applicable to immersed domains and time-step sizes are comparable to the uncut case, i.e., are independent of the polynomial order.

1. Introduction

There are many wave phenomena that are of interest for science and technology and where there is a desire to be able to accurately predict wave propagation. In a number of important cases, waves can be modeled by first-order hyperbolic conservation laws. In most realistic cases, the model is nonlinear or linear with non-constant coefficients. In addition, there are spatial boundaries and interfaces of complex shape, where boundary and interface conditions restrict the solution and many interesting and important phenomena are generated. In the general setting, solutions can only be obtained through numerical techniques, most of which involve a computational grid with a spatial discretization parameter, here called h. In this work, we will consider a simple model equation and focus on a methodology for handling general geometric features.

In many physically relevant cases, it is often possible to derive energy estimates. Much work has been done to develop numerical methods that support corresponding estimates for the discrete approximations, see, for example, [1] for finite difference methods (FDM) and [2] for finite element methods (FEM). Energy estimates guarantee stability of a method, which is essential and, therefore, of central importance.

From the extensive research on numerical methods over the years, we know that high-order accuracy is efficient for hyperbolic problems [3,4], and, consequently, many high-order-accurate numerical methods exist. Explicit time-stepping is preferable at least if the time-step restriction is not more severe than $\Delta t \sim h$, which is acceptable due to the finite speed of propagation. For standard FDM, the time-step restriction is essentially independent of the spatial approximation order, while, for higher-order accuracy, the time-step

* Corresponding author.
E-mail addresses: gunilla.kreiss@it.uu.se (G. Kreiss), muench@math.tu-berlin.de (P. Munch), siyang.wang@umu.se (S. Wang).

https://doi.org/10.1016/j.jcp.2026.115302
Received 26 September 2025; Received in revised form 11 August 2026; Accepted 12 August 2026
Available online 17 August 2026
0021-9991/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

restriction for most Galerkin methods often becomes tighter with higher spatial order. For standard continuous and discontinuous Galerkin (DG) methods, with local polynomial order p, the time-step restriction is $\sim h/p^2$ in the limit of large p, see [5]. In the Galerkin difference method [6], a Galerkin approach is presented that has many similarities with the FDM. The degrees of freedom correspond to function values at equidistant points in space, the stiffness and mass matrices correspond to translation-invariant difference stencils, and time-step restrictions are similar to those of finite difference methods. It is important to note that this is true also for high order of accuracy. Related discontinuous Galerkin type methods have also been developed, see, for example, [7]. These methods also have more favorable time-step restrictions compared to the corresponding standard Lagrange-based DG methods.

A common way to handle geometric features is to align the computational mesh with the boundary or interface. In the FDM approach, this is achieved by a coordinate transformation. For complicated geometries, finding a smooth transformation for the whole domain becomes a major difficulty. The overlapping-grid technique, see [8], yields a stable and accurate approximation for important special cases. Another idea is to use several grid blocks, each with a local transformation [9]. There have also been efforts to develop immersed or embedded finite difference methods. Here, boundary/interface conditions are imposed without aligning to the grid. The inverse Lax-Wendroff procedure is an example of these efforts [10]. However, there are issues of efficiency, generality, and stability with these approaches, especially in high-order settings. In particular, there is no general way to formulate a methodology that supports a discrete energy estimate.

In FEM, unstructured discretization aligned with the geometric features is usually possible to achieve, but sometimes at the cost of poor mesh quality and/or large differences in element size. Efficiency is enhanced by using a regular and structured grid. A structured grid allows for efficient computations, and regular elements enhance accuracy and, thus, efficiency. For example, with Cartesian quadrilaterals, the meshing procedure is simple and always results in a high-quality mesh. In addition, the Cartesian mesh allows for tensor-product basis functions. By avoiding remeshing for dynamically evolving geometries or problems of geometric optimization, efficiency is further improved. These are strong arguments for using immersed methods where boundary/interface conditions can be imposed without aligning to the grid.

Immersed approaches have been developed in the finite-element community. In the CutFEM approach, boundary and interface conditions are usually imposed weakly. A regular computational mesh covers more than the domain of the problem, but integration is done only over the parts of elements that are within the domain. All basis functions that have some support within the domain are used in the approximation. There are, however, difficulties related to cut scenarios where some basis functions have very small support. This results in ill-conditioning of the mass matrix and a severe CFL restriction. These problems have been resolved to some extent by introducing jump stabilization [11] or the basis-function removal approach where the basis functions are based on B-splines [12]. The CutFEM methodology has been applied to quite general problem classes [13,14], and it has been shown that, with the stabilization, the time-step restriction of the CutFEM is similar to that of the corresponding aligned standard FEM for wave equations [15]. The Galerkin framework has allowed for deriving energy estimates. Immersed approaches also exist for other solution procedures, e.g., finite volume or discontinuous Galerkin methods; for these, other types of stabilization have been developed, e.g., the state redistribution algorithm [16,17], flux redistribution [18], and domain of dependence stabilization [19].

In this work, we will explore the combination of the Galerkin difference method on a Cartesian mesh and the CutFEM methodology applied to linear advection problems in one and two space dimensions. The aim is to develop a high-order immersed method that satisfies an energy estimate and has good time-step restriction for explicit time-stepping methods, similar to that of the standard Galerkin difference method. To mitigate the ill-conditioning of the mass matrix when small cut scenarios occur, we adopt stabilization techniques commonly used in CutFEM and integrate them into the proposed method. In addition, we want the methodology to allow for efficient implementation in a computational framework such as deal.II [20].

The remainder of this article is organized as follows. In Section 2, we provide an overview of the Galerkin difference method and their properties. Section 3 outlines the linear advection equation on immersed domains, and a semidiscrete weak formulation is derived. We prove stability and derive an a priori error estimate. The conditioning of the mass matrix and the spectrum of the discrete approximation of the differential operator are also discussed. In Section 4, we present numerical results supporting our theoretical findings. Finally, Section 5 summarizes our conclusions and points to future research directions.

2. The Galerkin difference method

The Galerkin difference method (GDM) builds upon the standard Galerkin framework and utilizes the weak form of the partial differential equations for spatial discretization. In this section, we describe the basis functions of GDM introduced in [6] and their important properties.

2.1. The basis function

The GDM utilizes non-standard nodal basis functions in the form of continuous piecewise polynomials of odd degree on a uniform grid. As an example, we consider domain $[0,1]$ discretized by a uniform grid $x_j, j = 0, 1, \cdots, N$ with grid size h, and let $\phi_j^{(p)}$ denote a degree p GDM basis function associated with node x_j in one dimension. In **Fig. 1**, we show a GDM basis function $\phi_j^{(p)}$ for polynomial degrees $p = 1, 3, 5$, where x_j is a node away from the boundaries. For $p = 1$, it is the same as the standard finite-element hat function. For larger p, however, the basis functions are different from the Lagrange polynomials and have a larger support. For $p = 3$ and 5, the basis functions have support in 4 and 6 intervals, respectively. In general, a GDM basis function of degree p is symmetric with respect to its node and has support in $p + 1$ intervals, i.e., $supp(\phi_j^{(p)}) = [x_{j-(p+1)/2}, x_{j+,(p+1)/2}]$. In addition, away from the boundaries, the basis

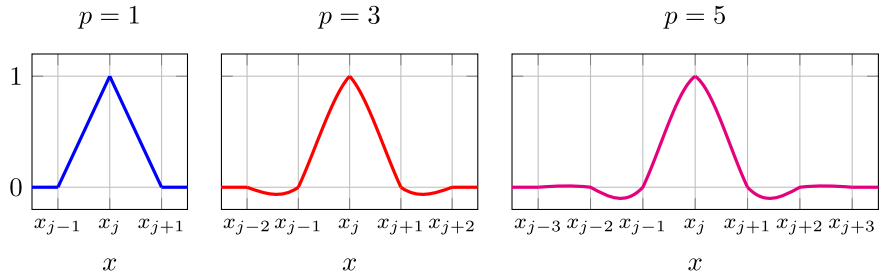

Fig. 1. GDM basis function associated with node j for degrees $p = 1, 3, 5$.

functions of the same degree but associated with different nodes are translations of each other, i.e., $\phi_k^{(p)}(x) = \phi_j^{(p)}(x - (k-j)h)$. This important relation gives rise to a finite-difference-type discretization matrix and is a property that the standard high-order Lagrange basis functions do not possess.

Explicit formulas of $\phi_j^{(p)}$ are constructed such that

$$\phi_j^{(p)}(x) = \begin{cases} l_i(x), & x \in [x_{j+i-(p+1)/2}, x_{j+i-(p-1)/2}], \quad i = 0, 1, \cdots p, \\ 0, & \text{elsewhere.} \end{cases}$$

Here, each l_i is a Lagrange polynomial of degree p, which interpolates at nodes $x_{j-p+i}, x_{j-p+i+1}, \cdots, x_{j+i}$, taking nodal value 1 at x_j and zero elsewhere. We note that $[x_{j+i-(p+1)/2}, x_{j+i-(p-1)/2}]$ is symmetrically centered within $[x_{j-p+i}, x_{j+i}]$. As an example, the explicit formula of $\phi_j^{(3)}$ is:

$$\phi_j^{(3)}(x) = \begin{cases} +\frac{1}{6h^3}(x - x_{j-3})(x - x_{j-2})(x - x_{j-1}), & x_{j-2} < x \le x_{j-1}, \\ -\frac{1}{2h^3}(x - x_{j-2})(x - x_{j-1})(x - x_{j+1}), & x_{j-1} < x \le x_j, \\ +\frac{1}{2h^3}(x - x_{j-1})(x - x_{j+1})(x - x_{j+2}), & x_j < x \le x_{j+1}, \\ -\frac{1}{6h^3}(x - x_{j+1})(x - x_{j+2})(x - x_{j+3}), & x_{j+1} < x \le x_{j+2}, \\ 0 & \text{elsewhere.} \end{cases}$$

2.2. Higher-order derivatives

The second derivative of the continuous piecewise polynomial $\phi_j^{(3)}(x)$ is continuous, but the first and third derivatives are discontinuous on the nodes $x_{j-2}, \cdots, x_{j+2}$. Moreover, the quotient of the jump in the first and third derivatives is the same on these nodes,

$$\left[\!\left[\frac{d^3}{dx^3}\phi_j^{(3)}(x) \right]\!\right] = -\frac{6}{h^2}\left[\!\left[\frac{d}{dx}\phi_j^{(3)}(x) \right]\!\right], \text{ for } x = x_{j-2}, \cdots, x_{j+2},$$

where the jump of a function $f(x)$ at $x = x_j$ is defined as $[\![f(x_j)]\!] := \lim_{x \to x_j^+} f(x) - \lim_{x \to x_j^-} f(x)$. In other words, the jump in the third derivative is equal to the jump in the first derivative multiplied with constant $-6/h^2$. This property also holds for basis functions of higher degrees. In general, the basis function $\phi_j^{(p)}$ satisfies

$$\left[\!\left[\frac{d^{2k+1}}{dx^{2k+1}}\phi_j^{(p)}(x) \right]\!\right] = \frac{c_k}{h^{2k}}\left[\!\left[\frac{d}{dx}\phi_j^{(p)}(x) \right]\!\right], \text{ for } x = x_{j-(p+1)/2}, \cdots, x_{j+(p+1)/2}, \tag{1}$$

for $k = 1, \cdots, (p-1)/2$ and constant c_k. Even derivatives of $\phi_j^{(p)}$ are continuous.

2.3. Basis functions near boundaries

Since the basis functions extend across multiple intervals, extra attention is required near boundaries. As an example, consider the boundary at $x = x_0$ in the left panel of Fig. 2. There are $(p-1)/2$ basis functions that are associated with nodes outside the domain but have support inside the domain. We refer to these nodes as ghost points. The straightforward approach is to include all basis functions that have support in the domain. In this case, the finite dimensional function space is constructed as the span of basis functions with support inside the domain, i.e., $\phi_j^{(p)}$, where $j = -(p-1)/2, \cdots, N + (p-1)/2$. In Fig. 2 (left panel), we plot the basis functions for $p = 3$ near the boundary at $x = 0$, with one basis function $\phi_{-1}^{(3)}$ associated with ghost point x_{-1}. In this case, all basis functions are horizontal translations of each other.

The second approach is to modify the basis functions near boundaries by extrapolating the $(p-1)/2$ basis functions associated with the ghost points. To preserve the high-order accuracy, the extrapolation has to be at least $p+1$ order accurate. As an example,

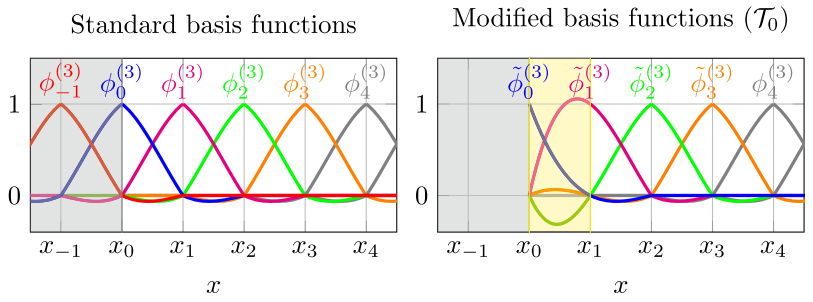

Fig. 2. Basis functions for $p = 3$ near the boundary x_0. The exterior domain is indicated in gray. The boundary cell containing modified basis functions, in contrast to interior cells, is highlighted in yellow. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

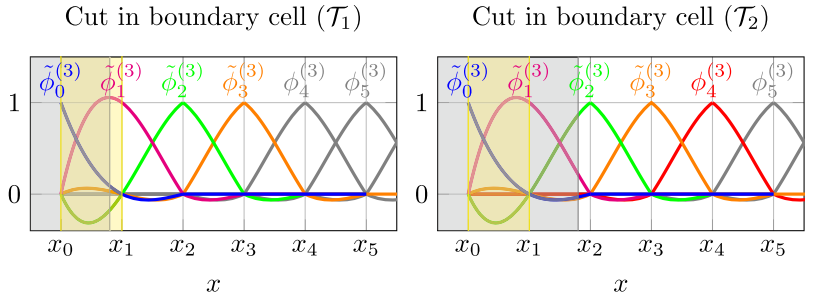

Fig. 3. Active basis functions for $p = 3$ near the boundary: case ($\mathcal{T}_1$, left) with the first boundary cell cut by the boundary at $x = x_0 - \alpha h$. If the cut is in an interior cell, as, for example, in the case in the right panel ($\mathcal{T}_2$), the stabilization couples the DoF related to $\tilde{\phi}_0^{(3)}$, which has very small support within the domain, to a DoF related to basis functions with larger support, $\tilde{\phi}_3^{(3)}$. .

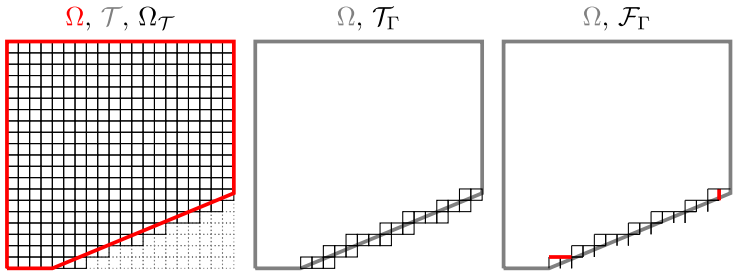

Fig. 4. A ramp domain Ω is immersed in a Cartesian background mesh $\mathcal{T}$, and $\Omega_{\mathcal{T}}$ consists of all cells overlapping with Ω. The cells $\mathcal{T}_\Gamma$ are intersected and, at faces $\mathcal{F}_\Gamma$, ghost penalty is applied. Red lines indicate faces where stabilization has no effect for the case when $p = 3$, in contrast to the standard CutFEM case.

the modified basis functions for $p = 3$ are plotted in the right panel of Fig. 2. Here, the standard fourth-order-accurate extrapolation stencil $u_{-1} = 4u_0 - 6u_1 + 4u_2 - u_3$ results in four modified basis functions near the boundary,

$$\tilde{\phi}_0^{(3)} = \phi_0^{(3)} + 4\phi_{-1}^{(3)},\ \tilde{\phi}_1^{(3)} = \phi_1^{(3)} - 6\phi_{-1}^{(3)},\ \tilde{\phi}_2^{(3)} = \phi_2^{(3)} + 4\phi_{-1}^{(3)},\ \tilde{\phi}_3^{(3)} = \phi_3^{(3)} - \phi_{-1}^{(3)}.$$

If a cell contains part of the basis function that is modified by the above procedure, then we call it a boundary cell. Otherwise, we call it an interior cell. We observe that the modified basis functions are smooth in the first two intervals between x_0 and x_2. Since the basis functions are modified only by $\phi_{-1}^{(3)}$, the local basis functions are only changed in $[x_0, x_1]$.

Here, the finite dimensional function space is the span of the modified basis functions near the boundaries and of the standard basis functions in the interior.

2.4. Extension to higher dimensions

For a given multidimensional domain $\Omega \in \mathbb{R}^d$, see an example in Fig. 4, we construct a Cartesian mesh $\mathcal{T}$ on a domain $\tilde{\Omega}$ where $\tilde{\Omega}$ is a rectangular-shaped domain such that Ω is immersed in it. The basis functions on $\mathcal{T}$ are the tensor product of the one-dimensional

basis functions that are modified near boundaries, i.e., the second approach explained in Section 2.3. However, only the basis functions with support in Ω are used in the computation. This means that basis functions with support cut by $\partial\Omega$ can be either of standard type or modified type. We refer to the nodes outside Ω as ghost points as well. This situation is visualized for 1D in **Fig. 3**. In 1D, aligning the boundaries is straightforward. In 2D, however, this is only the case if the boundaries of the domain and the computational domain overlap. In general, cells that are cut by the boundary may contain only modified basis functions (in the region where the boundary of the domain $\partial\Omega$ is close to the boundary of the background mesh $\partial\tilde{\Omega}$), only standard basis functions (in the region where $\partial\Omega$ is sufficiently far away from $\partial\tilde{\Omega}$), or a mix of standard and modified basis functions.

Consider cells $T \in \mathcal{T}$ that have some part inside Ω, i.e., $T \cap \Omega \neq \emptyset$, and define $\Omega_{\mathcal{T}}$ as the domain that corresponds to $\mathcal{T}$, that is

$$\Omega_{\mathcal{T}} = \bigcup_{T\in\mathcal{T}} T, \quad T \cap \Omega \neq \emptyset. \tag{2}$$

Let $\mathcal{T}_\Gamma$ denote the set of elements intersected by $\partial\Omega$,

$$\mathcal{T}_\Gamma = \{T \in \mathcal{T} \ : \ T \cap \partial\Omega \neq \emptyset\}.$$

We also define $\mathcal{F}_\Gamma$:

$$\mathcal{F}_\Gamma = \{F = \bar{T}_1 \cap \bar{T}_2 \ : \ T_1 \in \mathcal{T}_\Gamma \text{ or } T_2 \in \mathcal{T}_\Gamma,\ T_1, T_2 \in \mathcal{T}_\Gamma\}.$$

We assume that the background mesh is sufficiently fine so that the immersed geometry is well resolved. We denote h the mesh size and V_h^p the space of functions spanned by the basis functions with support in Ω.

3. The cut Galerkin difference method

Consider the advection equation

$$\dot{u} + \beta \cdot \nabla u = 0 \qquad \text{in } \Omega \times (0,T), \tag{3a}$$

$$u = u^+ \qquad \text{on } \Gamma^{\text{in}} \times (0,T), \tag{3b}$$

$$u = u_0 \qquad \text{on } \Omega \times \{t = 0\}, \tag{3c}$$

posed on a given domain Ω with boundary $\partial\Omega$. The inflow boundary is $\Gamma^{\text{in}} = \{x \in \partial\Omega \,|\, \beta \cdot n < 0\}$ with outward unit normal n, and the outflow boundary is $\Gamma^{\text{out}} = \partial\Omega/\Gamma^{\text{in}}$. The advection velocity β is a smooth function in Ω. In particular, we consider the case with divergence-free velocity field, i.e., $\nabla \cdot \beta = 0$.

3.1. The cut Galerkin difference method for linear advection

We multiply Eq. (3a) by a test function $v \in V_h^p$ and use Nitsche's method to impose the inflow boundary condition (3b). After using Green's identity, we obtain, for the weak solution u_h,

$$\begin{aligned} 0 &= (\dot{u}_h, v)_\Omega + (\beta \cdot \nabla u_h, v)_\Omega + \tau(u_h - u^+, \beta \cdot n v)_{\Gamma^{\text{in}}} \\ &= (\dot{u}_h, v)_\Omega + \frac{1}{2}(\beta \cdot \nabla u_h, v)_\Omega - \frac{1}{2}(u_h, \beta \cdot \nabla v)_\Omega + \frac{1}{2}(u_h, \beta \cdot n v)_\Gamma \\ &\quad + \tau(u_h - u^+, \beta \cdot n v)_{\Gamma^{\text{in}}}, \end{aligned}$$

where the penalty parameter τ is determined through stability analysis. The semidiscretization reads: for each fixed $t \in (0, t_f]$, find $u_h \in V_h^p$ such that

$$m(\dot{u}_h, v) + a(u_h, v) = L(v), \quad \forall v \in V_h^p, \tag{4}$$

where

$$\begin{aligned} m(u_h, v) &= (u_h, v)_\Omega, \\ a(u_h, v) &= \frac{1}{2}(\beta \cdot \nabla u_h, v)_\Omega - \frac{1}{2}(u_h, \beta \cdot \nabla v)_\Omega \\ &\quad + \frac{1}{2}(u_h, \beta \cdot n v)_{\Gamma^{\text{out}}} + \frac{1}{2}(1 + 2\tau)(u_h, \beta \cdot n v)_{\Gamma^{\text{in}}}, \\ L(v) &= \tau(u^+, \beta \cdot n v)_{\Gamma^{\text{in}}}. \end{aligned} \tag{5}$$

It is well-known that, in (4), the mass matrix can be ill-conditioned when small cuts occur in the mesh, and (4) needs to be stabilized [15]. In this work, we use the ghost-penalty technique to stabilize both bilinear forms,

$$\begin{aligned} M(u_h, v) &= m(u_h, v) + \gamma_M j_1(u_h, v), \\ A(u_h, v) &= a(u_h, v) + \gamma_A j_0(u_h, v), \end{aligned}$$

where

$$j_s(u_h, v) = \sum_{F \in \mathcal{F}_\Gamma} h^{2+s} \left(\left[\partial_n u_h\right], \left[\partial_n v\right] \right)_{\mathcal{F}_\Gamma}. \tag{6}$$

In the high-order CutFEM for the wave equation, based on degree p Lagrange basis functions [15], the stabilization utilizes jumps in derivatives of order up to $p-1$. However, in CutGDM, it is sufficient to include only jumps in the first derivatives in the stabilization because higher-order derivatives can be expressed as a constant that multiplies the jump in the first derivative, as shown in (1). The parameters γ_M and γ_A control the amount of stabilization in the mass matrix and the stiffness matrix, respectively.

In [15], an upper bound of the condition number of the stabilized mass matrix of CutFEM is derived by utilizing the jumps in derivatives up to p, and this upper bound is independent of the cut size. Following the same approach, the condition number of the stabilized mass matrix of the proposed CutGDM is also bounded independently of the cut size.

The stabilized semidiscretization reads: for each fixed $t \in (0, t_f]$, find $u_h \in V_h^p$ such that

$$M(\dot{u}_h, v) + A(u_h, v) = L(v), \quad \forall v \in V_h^p. \tag{7}$$

3.2. Stability analysis

In the following, we derive an energy estimate for the stabilized semidiscretization to prove stability.

Theorem 1 (Stability). *Assume the parameters $\gamma_M \geq 0, \gamma_A \geq 0$ and $\tau \leq 1/(\epsilon-2) < -1/2$ for some $0 < \epsilon < 2$ in (7), then the semidiscretization satisfies the energy estimate*

$$\frac{d}{dt} M(u_h, u_h) < -\frac{\tau}{2\epsilon} \|u^+\|^2_{\Gamma^{in}_\beta}. \tag{8}$$

The β-weighted norm on Γ^{in} is defined as $\|u^+\|^2_{\Gamma^{in}_\beta} = -(u^+, \beta \cdot n u^+)_{\Gamma^{in}}$, which is always nonnegative because $\beta \cdot n < 0$ on Γ^{in}. Note also that the right-hand side of (8) is nonnegative because τ is negative and ϵ is positive.

Proof. Choosing the test function v to be u_h in (7), we have

$$M(\dot{u}_h, u_h) + \frac{1}{2}(u_h, \beta \cdot n u_h)_{\Gamma^{\text{out}}} + \frac{1}{2}(1+2\tau)(u_h, \beta \cdot n u_h)_{\Gamma^{\text{in}}} + \gamma_A j_0(u_h, u_h) = L(u_h).$$

By using $\beta \cdot n > 0$ on Γ^{out} and $\gamma_A j_0(u_h, u_h) \geq 0$, we obtain

$$\begin{aligned} & M(\dot{u}_h, u_h) \\ =& -\frac{1}{2}(u_h, \beta \cdot n u_h)_{\Gamma^{\text{out}}} - \frac{1}{2}(1+2\tau)(u_h, \beta \cdot n u_h)_{\Gamma^{\text{in}}} - \gamma_A j_0(u_h, u_h) + \tau(u^+, \beta \cdot n u_h)_{\Gamma^{\text{in}}} \\ \leq& -\frac{1}{2}(1+2\tau)(u_h, \beta \cdot n u_h)_{\Gamma^{\text{in}}} + \tau(u^+, \beta \cdot n u_h)_{\Gamma^{\text{in}}}. \end{aligned}$$

We apply Young's inequality to the second term on the right-hand side,

$$M(\dot{u}_h, u_h) \leq \frac{1}{2}(1+2\tau)\|u_h\|^2_{\Gamma^{\text{in}}_\beta} - \tau\left(\frac{1}{2\epsilon}\|u^+\|^2_{\Gamma^{\text{in}}_\beta} + \frac{\epsilon}{2}\|u_h\|^2_{\Gamma^{\text{in}}_\beta}\right).$$

To bound the terms with $\|u_h\|^2_{\Gamma^{\text{in}}_\beta}$, it is clear that, at least, we need $1+2\tau < 0$. We require

$$\frac{1}{2}(1+2\tau) - \frac{\tau\epsilon}{2} \leq 0 \Rightarrow \tau \leq \frac{1}{\epsilon-2} < -\frac{1}{2} \text{ when } 0 < \epsilon < 2.$$

Then, we have

$$M(\dot{u}_h, u_h) \leq -\frac{\tau}{2\epsilon}\|u^+\|^2_{\Gamma^{\text{in}}_\beta},$$

and the desired estimate follows. □

We note that the stability estimate holds even for the non-stabilized case $\gamma_M = \gamma_A = 0$, but $M(u_h, u_h)$ may approach zero when the cut size goes to zero and the mass matrix loses positive definiteness. Therefore, it is important to choose positive values for these two parameters.

3.3. A priori error estimation

In the following, we derive a standard a priori error estimate for the stabilized semidiscretization.

Theorem 2. *Consider the semidiscretization (7). The error $e = u - u_h$ satisfies*

$$M(e, e) \leq C h^{2p}\left(\|u\|^2_{H^{q+1}(\Omega)} + \|\dot{u}\|^2_{H^{q+1}(\Omega)}\right)$$

if $\tau < -1/2$ and $\gamma_M \geq 0, \gamma_A \geq 0$.

Proof. Let $\pi_h : L^2(\Omega) \to V_h^p$ be the unfitted L^2 projection operator built upon the standard L^2 projection operator $\tilde{\pi}_h : L^2(\Omega_\mathcal{T}) \to V_h^p$ and an extension operator [21]. More precisely, for $u \in L^2(\Omega)$, we define

$$\pi_h u := \tilde{\pi}_h \mathcal{E} u,$$

where $\mathcal{E} : L^2(\Omega) \to L^2(\Omega_\mathcal{T})$ is a bounded extension operator satisfying stability estimate $\|\mathcal{E}u\|_{L^2(\Omega_\mathcal{T})} \le C\|u\|_{L^2(\Omega)}$, see [22].

We split e into two parts, the projection error $\eta = u - \pi_h u$ and the remaining part $\theta = \pi_h u - u_h \in V_h^p$ such that $e = \eta + \theta$. We note that u and η are defined on physical domain Ω, and u_h and $\pi_h u$ on the larger domain $\Omega_\mathcal{T}$. When using the standard inverse inequality to estimate η on Γ, its bound contains terms defined on $\mathcal{T}_\Gamma$ that is partially outside Ω but inside $\Omega_\mathcal{T}$. To this end, we need special projection error estimates adapted from [21],

$$\|\eta\|_\Omega \le \|\eta\|_{\Omega_\mathcal{T}} \le Ch^{p+1}\|u\|_{H^{p+1}(\Omega)}, \tag{9}$$

$$\|\eta\|_\Gamma \le Ch^{p+1/2}\|u\|_{H^{p+1}(\Omega)}, \tag{10}$$

$$\|\eta\|_{\mathcal{F}_\Gamma} \le Ch^{p+1/2}\|u\|_{H^{p+1}(\Omega)}. \tag{11}$$

We note that, in (9), the term $\|\eta\|_{\Omega_\mathcal{T}}$ shall be interpreted with a function u being extended from Ω to $\Omega_\mathcal{T}$ by the stable extension operator such that $\|u\|_{H^r(\Omega_\mathcal{T})} \le C\|u\|_{H^r(\Omega)}$ for $r = 0, 1, \cdots, p+1$. For the sake of simplified notation, we do not specify the extension operator in the formulas.

In the following, we derive an estimate for θ in the energy norm, and the error estimate for e follows from the triangle inequality.

The exact solution u satisfies (7), and the corresponding error equation is

$$m(\dot{e}, v) + a(e, v) + \gamma_M j_1(\dot{e}, v) + \gamma_A j_0(e, v) = 0, \quad \forall v \in V_h^p.$$

Substituting $e = \eta + \theta$ into the above equation, we obtain

$$\begin{aligned}
&m(\dot{\theta}, v) + a(\theta, v) + \gamma_M j_1(\dot{\theta}, v) + \gamma_A j_0(\theta, v) \\
=& - m(\dot{\eta}, v) - a(\eta, v) - \gamma_M j_1(\dot{\eta}, v) - \gamma_A j_0(\eta, v) \\
=& - m(\dot{\eta}, v) - a(\eta, v) + \gamma_M j_1(\pi_h \dot{u}, v) + \gamma_A j_0(\pi_h u, v).
\end{aligned} \tag{12}$$

In the last equality, we have used that $j_s(u, v) = 0$ for all $v \in V_h^p$. Next, we choose the test function v to be θ for any fixed t. Then, by using the expression of the bilinear form (5), the left-hand side of (12) becomes,

$$\begin{aligned}
\mathrm{LHS}_{(12)} &= m(\dot{\theta}, \theta) + a(\theta, \theta) + \gamma_M j_1(\dot{\theta}, \theta) + \gamma_A j_0(\theta, \theta) \\
&= \frac{1}{2}\frac{d}{dt}\left[m(\theta, \theta) + \gamma_M j_1(\theta, \theta)\right] + a(\theta, \theta) + \gamma_A j_0(\theta, \theta) \\
&= \frac{1}{2}\frac{d}{dt}\left[m(\theta, \theta) + \gamma_M j_1(\theta, \theta)\right] + \gamma_A j_0(\theta, \theta) \\
&\quad + \frac{1}{2}(\theta, \beta \cdot n\theta)_{\Gamma^{\mathrm{out}}} + \frac{1}{2}(1 + 2\tau)(\theta, \beta \cdot n\theta)_{\Gamma^{\mathrm{in}}}.
\end{aligned}$$

In the above equation, the first term is the energy change rate of θ, which is shown to be bounded by the other terms together with the right-hand side of (12).

On the right-hand side of (12), we have

$$\mathrm{RHS}_{(12)} = - m(\dot{\eta}, \theta) - a(\eta, \theta) + \gamma_M j_1(\pi_h \dot{u}, \theta) + \gamma_A j_0(\pi_h u, \theta).$$

Using again (5) and Green's identity to $a(\eta, \theta)$, yields

$$\begin{aligned}
\mathrm{RHS}_{(12)} =& - m(\dot{\eta}, \theta) + \gamma_M j_1(\pi_h \dot{u}, \theta) + \gamma_A j_0(\pi_h u, \theta) \\
&+ (\eta, \beta \cdot \nabla\theta) - (\eta, \beta \cdot n\theta)_{\Gamma^{\mathrm{out}}} - (1 + \tau)(\eta, \beta \cdot n\theta)_{\Gamma^{\mathrm{in}}}.
\end{aligned}$$

Combining $\mathrm{LHS}_{(12)}$ and $\mathrm{RHS}_{(12)}$ and using Young's inequality, we obtain

$$\begin{aligned}
&\frac{d}{dt}\left[m(\theta, \theta) + \gamma_M j_1(\theta, \theta)\right] \\
\le& - 2\gamma_A j_0(\theta, \theta) - (\theta, \beta \cdot n\theta)_{\Gamma^{\mathrm{out}}} - (1 + 2\tau)(\theta, \beta \cdot n\theta)_{\Gamma^{\mathrm{in}}} \\
&+ \|\dot{\eta}\|_\Omega^2 + \|\theta\|_\Omega^2 + \gamma_M j_1(\pi_h \dot{u}, \pi_h \dot{u}) + \gamma_M j_1(\theta, \theta) \\
&+ \gamma_A j_0(\pi_h u, \pi_h u) + \gamma_A j_0(\theta, \theta) + h^{-2}\|\eta\|_\Omega^2 + h^2\|\beta \cdot \nabla\theta\|_\Omega^2 \\
&+ (\eta, \beta \cdot n\eta)_{\Gamma^{\mathrm{out}}} + (\theta, \beta \cdot n\theta)_{\Gamma^{\mathrm{out}}} - 2(1 + \tau)(\eta, \beta \cdot n\theta)_{\Gamma^{\mathrm{in}}} \\
\le& - (1 + 2\tau)(\theta, \beta \cdot n\theta)_{\Gamma^{\mathrm{in}}} + \|\dot{\eta}\|_\Omega^2 + \|\theta\|_\Omega^2 + \gamma_M j_1(\pi_h \dot{u}, \pi_h \dot{u}) + \gamma_M j_1(\theta, \theta) \\
&+ \gamma_A j_0(\pi_h u, \pi_h u) + h^{-2}\|\eta\|_\Omega^2 + h^2\|\beta \cdot \nabla\theta\|_\Omega^2 \\
&+ (\eta, \beta \cdot n\eta)_{\Gamma^{\mathrm{out}}} - 2(1 + \tau)(\eta, \beta \cdot n\theta)_{\Gamma^{\mathrm{in}}}.
\end{aligned} \tag{13}$$

All terms containing η can be bounded by using the projection error estimate and the trace inequality [6],

$$\|\dot{\eta}\|_\Omega^2 \le Ch^{2p+2}\|\dot{u}\|_{H^{p+1}(\Omega)}^2, \tag{14}$$

$$h^{-2}\|\eta\|_\Omega^2 \le Ch^{2p}\|u\|_{H^{p+1}(\Omega)}^2, \tag{15}$$

$$(\eta, \beta \cdot n\eta)_{\Gamma^{\mathrm{out}}} \le Ch^{2p+1}\|u\|_{H^{p+1}(\mathcal{T}_{\Gamma^{\mathrm{out}}})}^2, \tag{16}$$

$$h^{-1}(\eta, \beta \cdot n\eta)_{\Gamma^{\mathrm{in}}} \le Ch^{2p}\|u\|_{H^{p+1}(\mathcal{T}_{\Gamma^{\mathrm{in}}})}^2. \tag{17}$$

For the following terms containing θ, we use the inverse inequality and the trace inequality in order to obtain

$$h^2\|\beta\cdot\nabla\theta\|^2_{\Omega}\le C\|\theta\|^2_{\Omega}, \tag{18}$$

$$h(\theta,\beta\cdot n\theta)_{\Gamma^{\text{in}}}\le C\|\theta\|^2_{\mathcal{T}_{\Gamma^{\text{in}}}}\le C\|\theta\|^2_{\Omega}. \tag{19}$$

Next, we consider the term

$$\gamma_M j_1(\pi_h\dot u,\pi_h\dot u)=\gamma_M j_1(\dot u-\pi_h\dot u,\dot u-\pi_h\dot u)=\gamma_M h^3\|\partial_n(\dot u-\pi_h\dot u)\|^2_{\mathcal{T}_{\Gamma^{\text{in}}}}.$$

By using the trace equality and the interpolation error estimate, we have

$$\begin{aligned}\|\partial_n(\dot u-\pi_h\dot u)\|^2_{\mathcal{T}_{\Gamma^{\text{in}}}}&\le Ch^{-1}\|\partial_n(\dot u-\pi_h\dot u)\|^2_{\mathcal{T}_{\Gamma^{\text{in}}}}\\&\le Ch^{2p-1}\|\dot u\|^2_{H^{p+1}(\mathcal{T}_{\Gamma^{\text{in}}})},\end{aligned}$$

which leads to

$$\gamma_M j_1(\pi_h\dot u,\pi_h\dot u)\le Ch^{2p+2}\|\dot u\|^2_{H^{p+1}(\mathcal{T}_{\Gamma^{\text{in}}})}. \tag{20}$$

Similarly, we have

$$\gamma_A j_0(\pi_h u,\pi_h u)\le Ch^{2p+1}\|u\|^2_{H^{p+1}(\mathcal{T}_{\Gamma^{\text{in}}})}. \tag{21}$$

Substituting estimates (14)–(21) into (13), we obtain

$$\frac{d}{dt}\left[m(\theta,\theta)+\gamma_M j_1(\theta,\theta)\right]\le C_1[m(\theta,\theta)+\gamma_M j_1(\theta,\theta)]+C_2h^{2p}.$$

After integration in time, we have the final error estimate

$$m(\theta,\theta)+\gamma_M j_1(\theta,\theta)\le Ch^{2p}\Big(\|u\|^2_{H^{p+2}(\Omega)}+\|\dot u\|^2_{H^{p+1}(\Omega)}\Big).$$

The desired estimate for e follows from the triangle inequality. □

We note that the error estimate corresponds to pth order convergence rate, but we observe an optimal rate of $p+1$ in the L^2 norm in numerical experiments.

3.4. Condition number of the mass matrix and spectrum of the semidiscretization

Let $\mathcal{M}$ and $\mathcal{S}$ denote the mass and stiffness matrices of the semidiscretization (7). In the following, we use the 1D advection equation with wave speed equal to one as an example to investigate the condition number of $\mathcal{M}$ and the spectrum of $\mathcal{M}^{-1}\mathcal{S}$ for different cases.

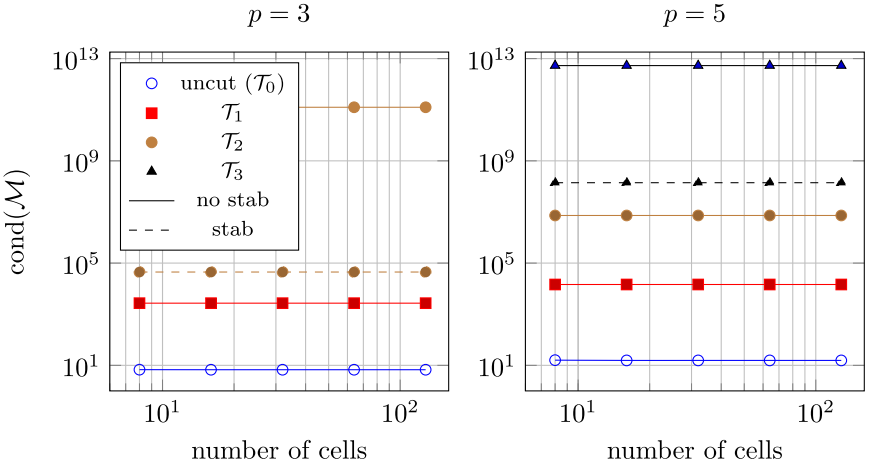

Fig. 5. Condition number of the 1D mass matrix for $p=3$ and 5. The value of $\alpha=0.001$ for all cases with cut. We show results from different cut scenarios, including a case when the ghost-penalty stabilization is disabled. Here, the mesh $\mathcal{T}_0$ aligns with the boundary. When $p=3$, there is only one boundary cell. In this case, when the mesh $\mathcal{T}_1$ is used, the boundary cuts through this boundary cell, while for the mesh $\mathcal{T}_2$, the first interior cell is cut. When $p=5$, there are two boundary cells. In this case, the meshes $\mathcal{T}_1$ and $\mathcal{T}_2$ are cut in the first and second boundary cell, respectively, and the mesh $\mathcal{T}_3$ is cut in the first interior cell.

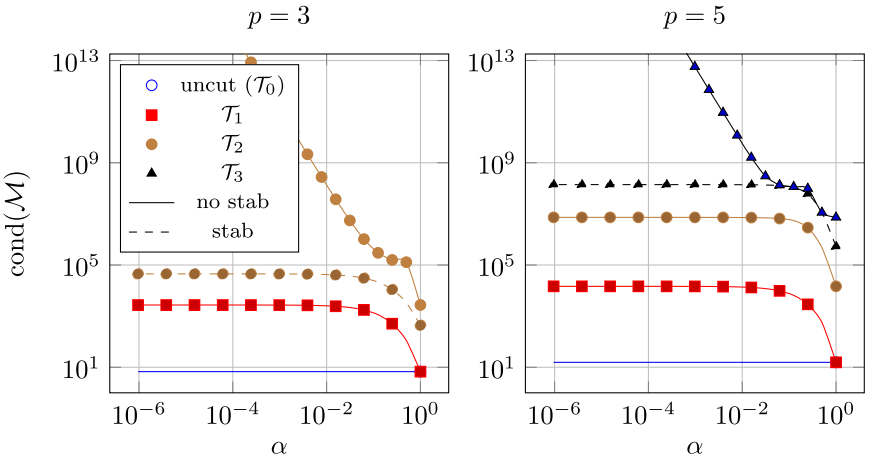

Fig. 6. Condition number of the 1D mass matrix for $p = 3$ and 5, 64 cells and different values of α. We show different numbers of boundary cells and the effect of disabling the ghost-penalty stabilization. The legend follows the same convention as in Fig. 5.

Let $0 \leq \alpha \ll 1$ and consider the domain $\Omega = [-1 - \alpha h, 1]$ and the background meshes $\mathcal{T}_i$ with nodes $x_j = jh, j = -i, \ldots, 0, 1, 2, \ldots$, where $h = 1/N$ for some integer N. We focus on the left boundary, which is cut at $x = -1 - \alpha h$ either in a boundary cell or in an interior cell depending on how the background domain $\tilde{\Omega} = [x_L, 1]$ is constructed. For $p = 3$, there is one boundary cell. If $x_L = -1 - h$, the cut is in this boundary cell and the modified basis is used in computation, see Fig. 3. If $x_L \leq -1 - 2h$, the cut is in an interior cell and only basis functions of the standard type are used. When $p = 5$, there are two boundary cells. When $x_L = -1 - h$ or $-1 - 2h$, the cut is in the first or second boundary cell, respectively. If $x_L \leq -1 - 3h$, the cut is in an interior cell. We note that stabilization has no effect at cell faces where the extrapolation procedure has removed the non-smooth features of basis functions. For comparison, we also consider the uncut case with $\alpha = 0$ and $x_L = -1$, using the modified basis. **Fig. 5** shows the numerically determined condition number of $\mathcal{M}$ with $\alpha = 0.001$ for the cases with cut. It is obvious that the condition number is independent of h and cuts lead to worse condition numbers. Moreover, one can see that the condition number grows as the cut moves to a cell further away: $6.7 \to 2.7 \cdot 10^3 \to 1.2 \cdot 10^{11}$ for $p = 3$ and $15.6 \to 1.4 \cdot 10^4 \to 7.3 \cdot 10^6 \to 5.3 \cdot 10^{12}$ for $p = 5$. For cuts in interior cells, stabilization significantly reduces the value of the condition number: $1.2 \cdot 10^{11} \to 4.5 \cdot 10^4$ for $p = 3$ and $5.3 \cdot 10^{12} \to 1.4 \cdot 10^8$ for $p = 5$. However, the values are still higher than in the case that the cut is in boundary cells.

Fig. 6 shows the dependency of the conditioning of the mass matrix on the cut position α. It is clear that, if the cut is in interior cells, stabilization is essential, expressed as increasing condition number with decreasing α. In all other cases, the condition number is larger than in the uncut case, but is bounded independently of the cut position α ($p = 3$: $6.7 \to 2.7 \cdot 10^3$, $4.4 \cdot 10^2 \to 4.5 \cdot 10^5$; $p = 5$: $15.6 \to 1.5 \cdot 10^4$, $1.5 \cdot 10^4 \to 7.3 \cdot 10^6$, $5.3 \cdot 10^5 \to 1.4 \cdot 10^8$). To address the challenge of high condition numbers in the magnitude of 10^8, we solve the corresponding system of linear equations, using the conjugate gradient method with a preconditioner via the incomplete LU factorization. In practical simulations, however, we did not observe negative impact on the stability due to finite precision (see Section 4).

Next, we investigate the spectrum of the semidiscretization matrix $\mathcal{M}^{-1}\mathcal{S}$, which is important for step-size restriction in explicit time-stepping schemes. **Fig. 7** shows the spectrum of $\mathcal{M}^{-1}\mathcal{S}$ for different cases. We only consider the case that we do not have a cut and the case that the cut is inside an interior cell, and we apply stabilization. In addition, we plot the stability regions of the classical Runge–Kutta method (RK4) and RK6 (see Appendix A.1), which are the time-stepping methods we use in our numerical experiments for $p = 3$ and $p = 5$, respectively. The results indicate that the cut affects the spectrum but does not modify the spectral radius, allowing time steps similar to the uncut case. Furthermore, we can determine time-step sizes that should suffice for stability: the critical CFL number ($\mathrm{CFL} = \frac{\Delta t}{hv}$) is $\mathrm{CFL}_{crit} = 1.3$ for $p = 3$ and $\mathrm{CFL}_{crit} = 0.6$ for $p = 5$. For 2D, the values are reduced by a factor $\sqrt{2}$, leading to $\mathrm{CFL}_{crit} = 0.9$ for $p = 3$ and $\mathrm{CFL}_{crit} = 0.4$ for $p = 5$. Throughout the experiments, we apply $\mathrm{CFL} = 0.4$. Computations show that using a Courant number close to the limit yields stable computations, but we get smaller errors when using a somewhat lower Courant number.

4. Numerical results

In this section, we present numerical results for the linear advection equation in 2D, starting with a ramp domain. We then consider a domain with curved boundaries and conclude with a problem with a composite domain where the advection speed changes abruptly at the domain interface.

For all experiments, we use $\mathrm{CFL} = 0.4$, $\gamma_A = \gamma_M = 0.5$ and $\tau = -1$. The linear system is solved with a conjugate gradient solver that is preconditioned with ILU. We use small benchmark codes, which are based on the open-source deal.II finite-element library [20]. The

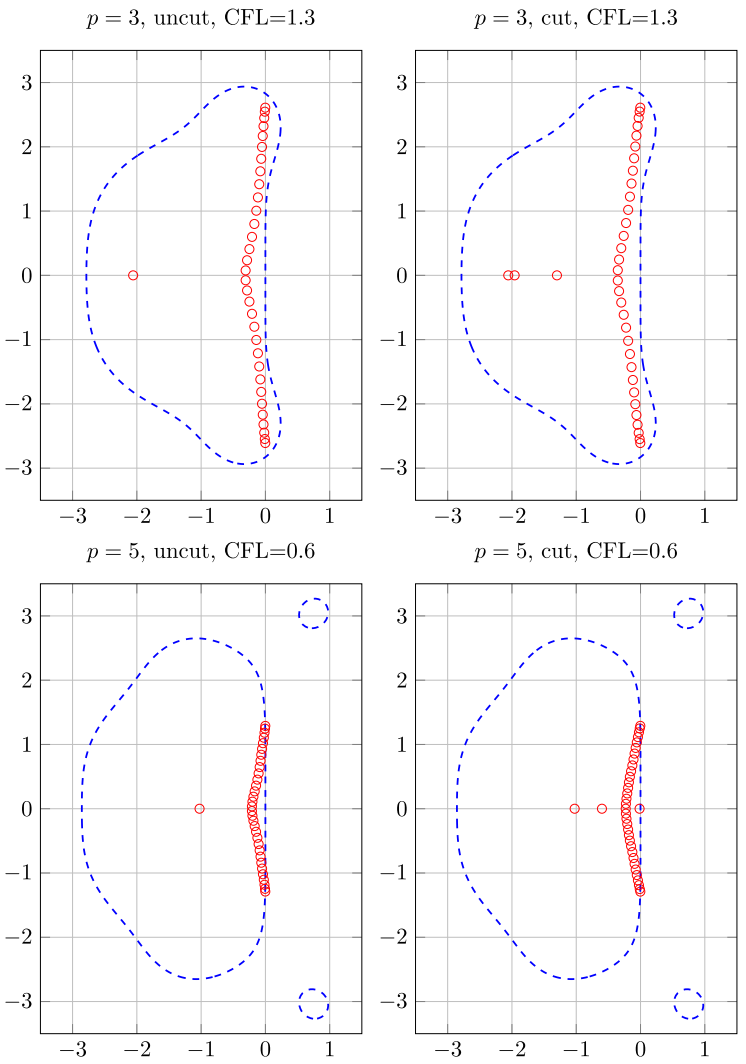

Fig. 7. Spectrum of $\mathcal{M}^{-1}S$ for $p = \{3, 5\}$ and 32 cells. The eigenvalues are scaled by CFL $\cdot\, \Delta x/v$, with CFL selected so that all eigenvalues are within the stability region. Dashed lines indicate the stability region of RK4 and RK6, respectively.

benchmark codes are publicly available on GitHub.[1] For further implementation details, in particular regarding the basis functions, definition of the computational domain, and quadrature generation, see Appendix A.2.

4.1. Ramp

In the first experiment, we use higher-order polynomials, $p = 3$ and 5, for the ramp setup for different angles γ, shown in **Fig. 8** and adopted from [19]. We choose $\Omega = (0, 1)^2$ and start the ramp at $x = 0.2001$. For the definition of the initial data, we use a rotated and shifted coordinate system $(\hat{x}, \hat{y})$ that we derive from the standard Cartesian system (x, y) by

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} x - 0.2001 \\ y \end{pmatrix}.$$

The above-described coordinate system is defined in such a way that the $\hat{x}$-direction is parallel to the ramp. In this coordinate system, the smooth initial data is given by

$$u_0(\hat{x}, \hat{y}) = \sin\left(\frac{\sqrt{2}\pi\hat{x}}{1 - 0.2001}\right).$$

[1] https://github.com/peterrum/dealii-galerkin-difference-methods.

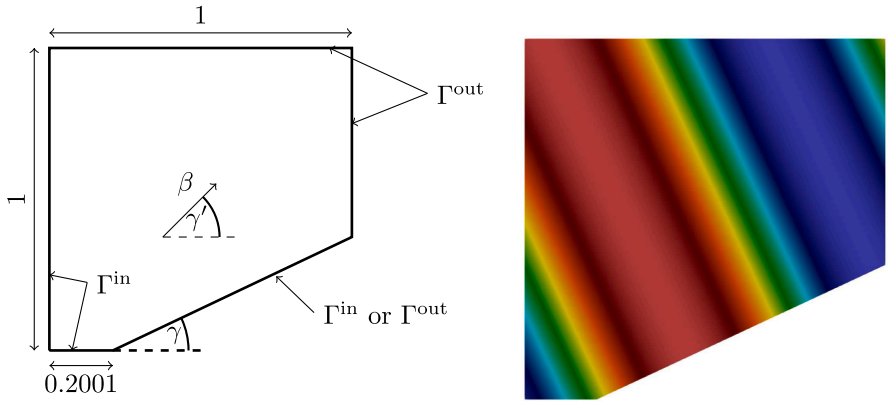

Fig. 8. Left: setup of the ramp example. The ramp has a slope γ and the velocity γ', with $5° \le \gamma \le 45°$ and $0° \le \gamma' \le 90°$. If $\gamma < \gamma'$, the ramp has an inflow boundary condition. If $\gamma > \gamma'$, the ramp has an outflow boundary condition. In the case that $\gamma = \gamma'$, the direction is parallel to the ramp, corresponding to a "do-nothing" (outflow) boundary condition. Right: initial condition for $\gamma = 25°$. For the visualization of a potential mesh, see Fig. 4.

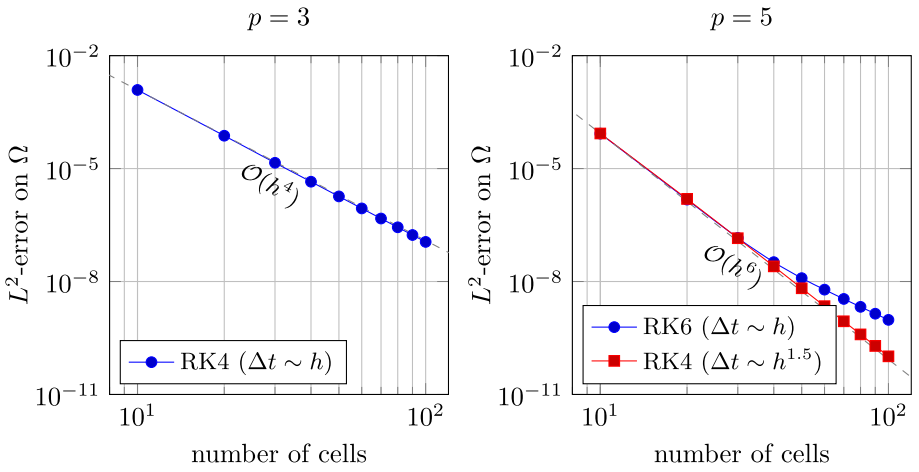

Fig. 9. Convergence study for parallel ramp ($\gamma = \gamma' = 25°$): L^2-error on Ω.

The transport velocity β is defined in a coordinate system$(\hat{x}, \hat{y})$ that we derive from the standard Cartesian system by rotation by γ':

$$\beta(\hat{x}, \hat{y}) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

The above-described coordinate system is defined in such a way that the $\hat{x}$-direction is either parallel to the ramp ($\gamma' = \gamma$) or not ($\gamma' \neq \gamma$). To discretize in time, we use RK4 and RK6 for $p = 3$ and $p = 5$ respectively with the analytical boundary condition

$$g(t, \hat{x}, \hat{y}) = u_0(\hat{x} - |\beta| t, \hat{y}).$$

According to [23], the inflow information g in the RK stages needs to be imposed by running RK updates for g as well. This procedure is essential to reduce temporal errors introduced by the time derivatives of the boundary data, which might dominate in the GDM case, where large time steps can be used.

4.1.1. Parallel advection

We start with a transport direction that is parallel to the ramp ($\gamma = \gamma'$) and vary the slope of the ramp. **Figs. 9** and **10** present L^2-errors on Ω and Γ for $p \in \{3, 5\}$, $\gamma = 25°$, and different CFL numbers. We observe a perfect convergence for $p = 3$. For $p = 5$, the simulation is stable, however, not the optimal convergence rate is reached for L^2-error on Ω for CFL 0.4. Running RK4 with $\Delta t \sim h^{1.5}$ allows us to recover the optimal convergence. In order to show that this observation is unrelated to the cut, we provide results from uncut simulations (the same setup but without ramp) in Appendix A.3. Here, we also observe a suboptimal convergence indicating that it is caused by the temporal discretization error.

Asymptotically, we expect to see a deterioration of convergence when using RK6 due to the fact that there are negative coefficients in all Runge–Kutta methods of order higher than 4 [24], leading to cancellation errors. From the plots, we see that the deterioration is

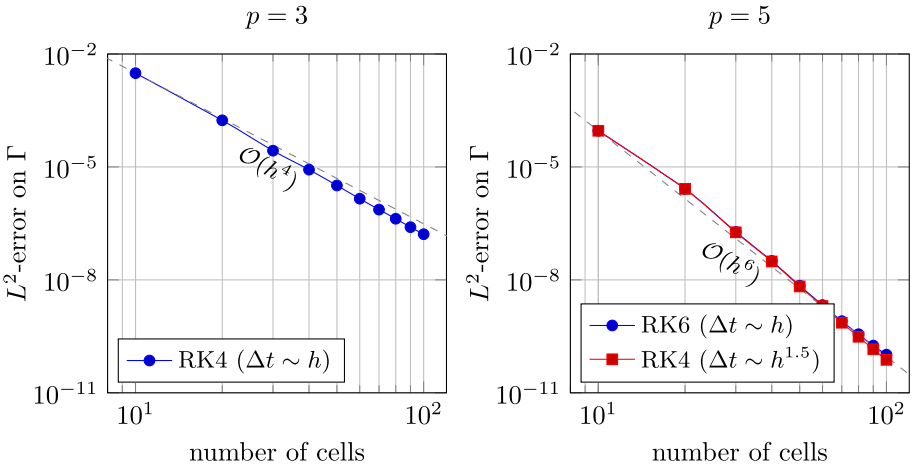

Fig. 10. Convergence study for parallel ramp ($\gamma = \gamma' = 25°$): L^2-error on Γ.

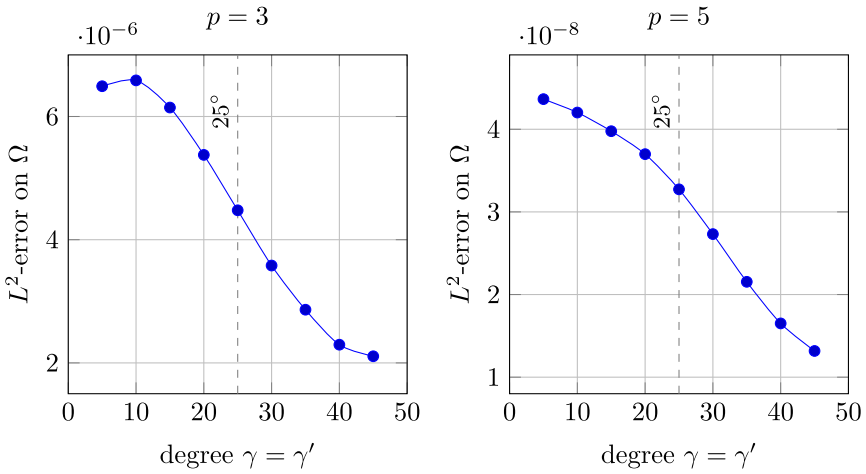

Fig. 11. Parallel ramp ($5° \leq \gamma \leq 45°$) with 40 cells in each direction and fixed CFL (0.4 for $p = 3$ and 0.1 for $p = 5$).

only observed for extremely fine grids and small time steps, where the error level is very low, substantiating our assumption. Overall, RK6 gives significant better results than RK4 for the same time-step size (maximal factor: 17).

In Fig. 11, we vary the slope of the ramp while keeping the number of cells and the CFL number constant. One can see that the simulations are stable independently of the chosen slope. With increasing slope, the error reduces.

For reference, Fig. 12 provides results obtained with FEM on structured body-fitted meshes (see Fig. 13). The results indicate that FEM and CutGDM yield comparable errors for the same number of DoFs. However, FEM needs significantly smaller time-steps (in the figure: $1/(2p+1)$), as in the non-cut case [3]. This observation indicates the potential for a fast (Cut)GDM solver once an efficient linear solver has been developed.

4.1.2. Non-parallel advection

In the following simulations, we fix the slope of the ramp to $\gamma = 25°$ and vary the transport-direction slope, $0 \leq \gamma' \leq 90$, while keeping the number of cells and the CFL number constant. This means that either an inflow or an outflow boundary condition is applied at the ramp. All simulations are stable and errors decrease with increasing slope, as indicated in Fig. 14.

4.2. Curved pipe

The aim of this experiment is to demonstrate that the methodology works for curved boundaries. We compute the solution to a rotational flow transport with angular velocity $\omega = 1$ around a quarter circle with inner radius $r_i = 1$, outer radius $r_o = 1.43$, and divergence-free velocity field $\beta = [-y, x]^\top$. The background domain has the size $[0, 1.5]^2$ and is discretized with 50 cells in each direction. The initial and boundary conditions are given by the superposition of Gaussian functions $f_j(\vec{x}) = \exp\left(-(||\vec{x} - \vec{x}_j||/r_0)^2/2\right)$

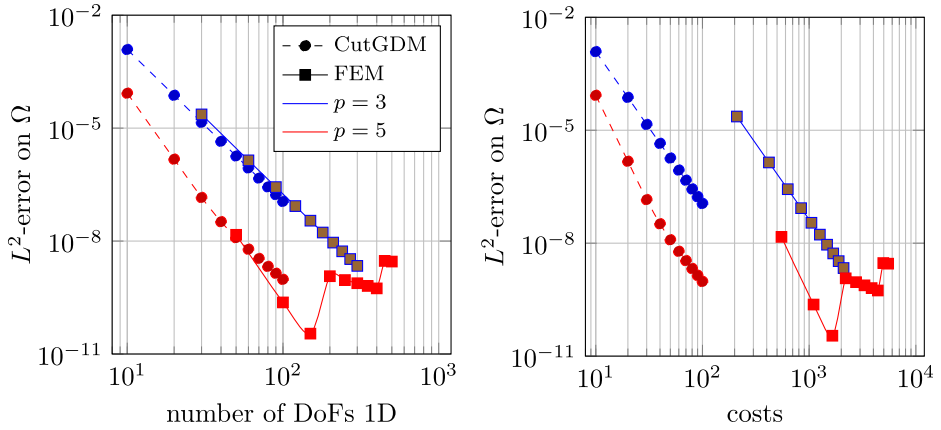

Fig. 12. Comparison of FEM and CutGDM ($\gamma = 0°$, $\gamma' = 25°$). The costs of CutGDM are given as numbers of DoFs in 1D and the costs of FEM are given as numbers of DoFs in 1D multiplied by the factor $2p + 1$ corresponding to the time-step restriction of FEM.

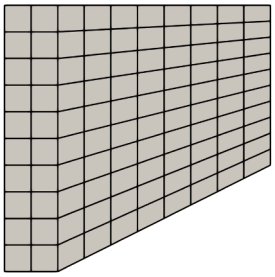

Fig. 13. Structured body-fitted mesh used for FEM.

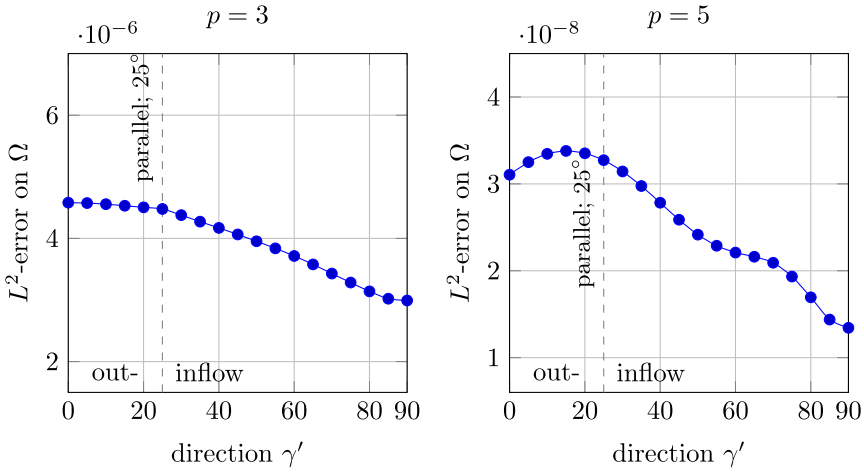

Fig. 14. Non-parallel ramp ($\gamma = 25°$, $0° \leq \gamma' \leq 90°$). For details on the setup, see Fig. 11.

with the centers positioned every 45° either on r_i or on r_o and $r_0 = 0.2$. **Fig. 15** shows the mesh, the initial condition, and the computed solution at different time steps. We observe that the wave structures are preserved very well at the cut cells.

4.3. Composite domain: discontinuous velocity

The aim of this experiment is to show that our methodology can be adapted to a problem with an internal interface where the coefficient of the problem, the transport direction, changes abruptly. This would, for example, correspond to a problem with a material interface. We have straightforwardly applied the conservative coupling technique developed for discontinuous Galerkin methods in

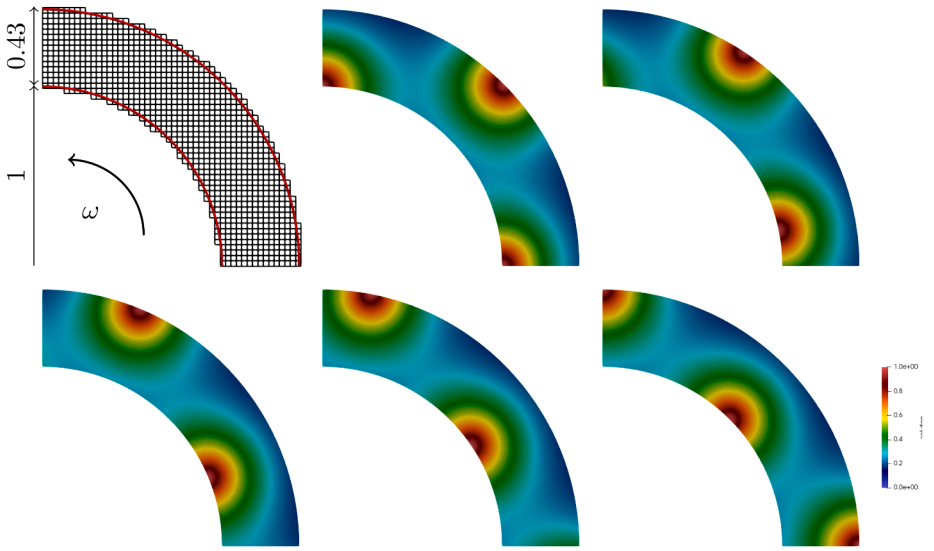

Fig. 15. Pipe domain: $t \in \{0, 0.20, , 0.39, 0.60, 0.79\}$.

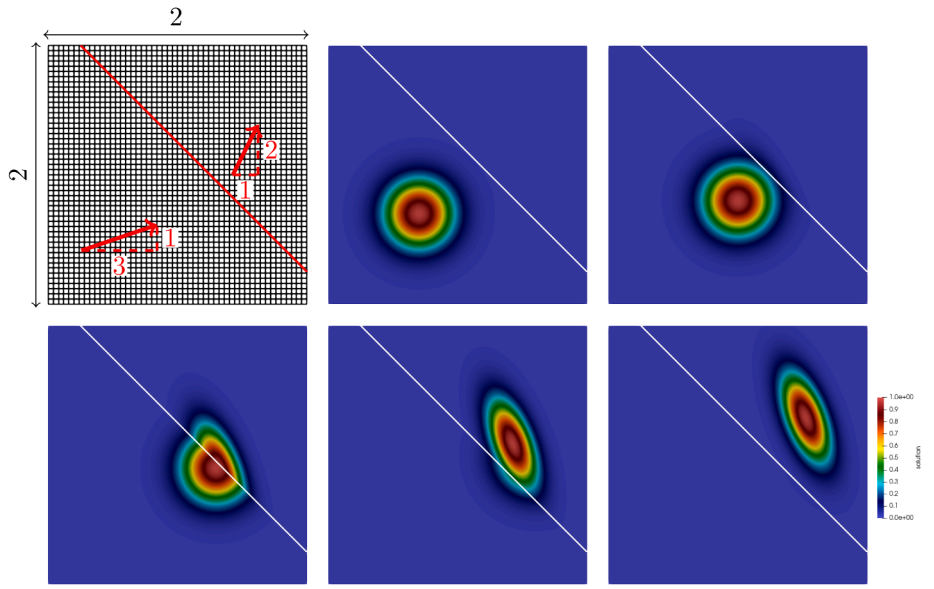

Fig. 16. Composite domain: $t \in \{0, 0.1, 0.2, 0.3, 0.4\}$.

[25], which leads, in our case, to weak coupling via upwind fluxes. We consider the same setting as the 2D example in that paper, but with a smooth pulse as initial condition: $u_0(x, y) = \exp\left(-16||\vec{x} - \vec{x}_0||^2\right)$ with $\vec{x}_0 = [-0.3, -0.3]^\top$. The domain is $[-1, 1]^2$ and has 50 cells in each direction. The interface is a 135-degree diagonal through $[0.25, 0.0]^\top$. The transport velocity changes from $[3, 1]^\top$ to $[1, 2]^\top$. **Fig. 16** shows the mesh, the initial condition and the computed solution at different time steps.

5. Conclusions and outlook

We have developed a cut Galerkin difference method for the linear advection equation on a Cartesian background mesh with non-aligned boundaries and material interfaces. Our implementation of the method can yield up to sixth-order accuracy. The semidiscretization satisfies a discrete energy estimate to ensure stability, and we have derived an a priori error estimate. To address challenges associated with the cut cells, we employ the jump stabilization technique that significantly improves the spectrum property of the spatial discretization and the conditioning of the mass matrix. We have found that it suffices to include only the jump in the first derivative in the stabilization term. A significant advantage of our approach is that the spectrum of the CutGDM is nearly identical to the corresponding uncut case, for which it was shown in [6] that the CFL time-step restriction is independent of the order of the

method. For the conditioning of the mass matrix, we have observed a growth in the condition number compared to the uncut case. It is important to note that the condition number remains bounded independently of how the boundary cuts the mesh. We have carried out a series of numerical tests for the 2D linear advection equation along a ramp, in a pipe-shaped domain or in piecewise constant media to verify the accuracy property and the robustness of the developed method. We believe the technique can be extended to more complicated problems, including nonlinear conservation laws and wave equations in second-order form. Additionally, further improvement on the conditioning of the mass matrix can be explored by considering other types of stabilization techniques.

CRediT authorship contribution statement

Gunilla Kreiss: Writing – original draft, Validation, Methodology, Investigation, Conceptualization; **Peter Munch:** Writing – original draft, Validation, Software, Investigation, Conceptualization; **Siyang Wang:** Writing – original draft, Validation, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge discussions with Pei Fu, Felix Langer, Sandra May, and Simon Sticko as well as the `deal.II` community.

Data availability

Data will be made available on request.

Appendix A.

A.1. Runge–Kutta method

The Butcher tableaus of RK4 and RK6 are given as:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & & \frac{1}{2} & & \\ 1 & & & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}, \qquad \begin{array}{c|ccccccc} 0 & & & & & & & \\ \frac{1}{3} & \frac{1}{3} & & & & & & \\ \frac{2}{3} & 0 & \frac{2}{3} & & & & & \\ \frac{1}{3} & \frac{1}{12} & \frac{1}{3} & -\frac{1}{12} & & & & \\ \frac{1}{2} & -\frac{1}{16} & \frac{9}{8} & -\frac{3}{16} & -\frac{3}{8} & & & \\ \frac{1}{2} & 0 & \frac{9}{8} & -\frac{3}{8} & -\frac{3}{4} & \frac{1}{2} & & \\ 1 & \frac{9}{44} & -\frac{9}{11} & \frac{63}{44} & \frac{18}{11} & 0 & -\frac{16}{11} & \\ \hline & \frac{11}{120} & 0 & \frac{27}{40} & \frac{27}{40} & -\frac{4}{15} & -\frac{4}{15} & \frac{11}{120} \end{array}.$$

The corresponding amplification factors are:

$$G_4(h) = 1 + h + h^2/2 + h^3/6 + h^4/24, \text{ for RK4},$$

$$G_6(h) = G_4(h) + h^5/120 + h^6/720 - h^7/2160, \text{ for RK6}.$$

The resulting stability regions are given as $||G_i|| \leq 1$, which are shown in Fig. 7 for $i = 4$ and 6.

A.2. Implementation

All numerical experiments in this publication have been performed with programs based on the open-source library deal.II [20]. As the focus of this library is finite-element computation, its application programming interface (API) is centered around looping over cells/elements and performing operations on these. The latter includes, e.g., reading/writing from/to matrices/vectors, for which the indices of all degrees of freedom of a cell/element are needed, and the evaluation of cell integrals. This cell-centric view is in conflict with the vertex-centric view common in the GDM context. However, the vertex-centric view can be recast into a cell-centric one for structured meshes. For this purpose, all shape functions with non-zero support are collected on a cell and processed together. **Fig. A.1** shows the basis functions on boundary cells and an interior cell for $p = 3$. This integrates very well with the *hp*-adaptivity supported by `deal.II`.

During cell loops, non-local degrees of freedom, i.e., degrees of freedom not owned by the current cells but by the neighbors (of the neighbors) of the current cell, must be accessed. **Fig. A.2** visualizes this, implying that all DoFs on a patch around the cell need to be queried. Since `deal.II` does not support this kind of access pattern, we were required to modify all instances where global matrices

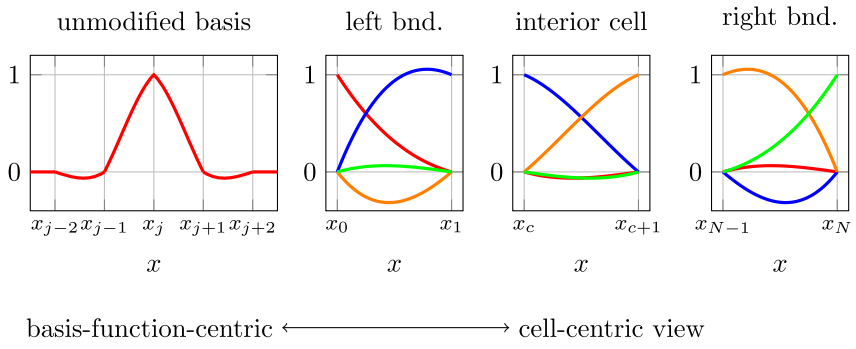

Fig. A.1. Basis functions with non-zero support on boundary cells and an interior cell for $p = 3$.

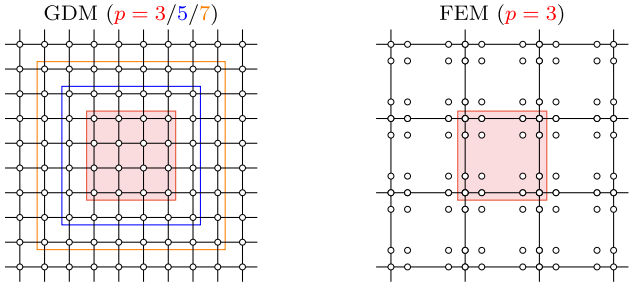

Fig. A.2. Visualization of access to global vectors and matrices during cell loops for GDM and FEM.

and vectors are accessed (in particular during memory allocation, assembly, and postprocessing). However, it is straightforward to determine the indices of the relevant basis functions based on the cell indices in the case of GDM and lexicographic numbering of the basis functions and the cells.

Recasting the vertex-centric view into a cell-centric one allows us to reuse large parts of the deal.II infrastructure for the computation of contributions to the system matrix and the right-hand-side vector. Here, we need to evaluate integrals on (part of) cells, faces, and interfaces. Such integrals are supported by the CutFEM infrastructure of `deal.II`, which generates the required (high-order) quadratures based on the algorithm of [26] and a given distance function. Geometrical features of boundaries and interfaces are presented by level-set functions.

Furthermore, we heavily rely on `deal.II`'s linear-algebra and parallelization routines. We parallelize the loops over cells by assigning subsets of cells to different MPI processes.

A.3. Additional numerical results for parallel ramp

Fig. A. 3 presents GDM results from uncut simulations (without ramp).

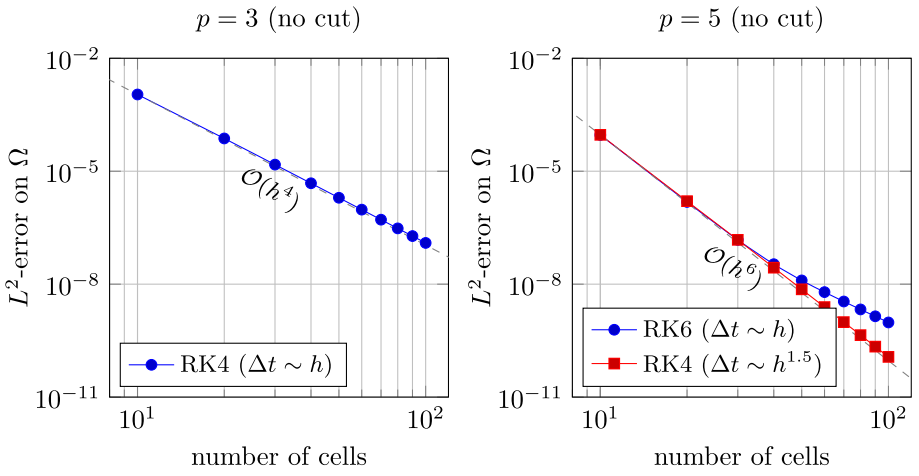

Fig. A. 3. Convergence study on a domain that is *not cut* ($\gamma = 0°$, $\gamma' = 25°$).

References

[1] B. Gustafsson, H.O. Kreiss, J. Oliger, Time–Dependent problems and difference methods, John Wiley & Sons, 2013.
[2] M.G. Larson, F. Bengzon, The finite element method: Theory, implementation, and applications, Springer, 2010.
[3] T. Hagstrom, G. Hagstrom, Grid stabilization of high–order one–sided differencing II: second–order wave equations, J. Comput. Phys. 231 (2012) 7907–7931.
[4] H.O. Kreiss, J. Oliger, Comparison of accurate methods for the integration of hyperbolic equations, Tellus 24 (1972) 199–215.
[5] G.E. Karniadakis, S.J. Sherwin, Spectral/hp element methods for computational fluid dynamics, Oxford University Press, 1999.
[6] J.W. Banks, T. Hagstrom, On Galerkin difference methods, J. Comput. Phys. 313 (2016) 310–327.
[7] J.W. Banks, B.B. Buckner, T. Hagstrom, K. Juhnke, Discontinuous Galerkin Galerkin differences for the wave equation in second-order form, SIAM J. Sci. Comput. 43 (2021) A1497–A1526.
[8] G. Chesshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1990) 1–64.
[9] M.H. Carpenter, J. Nordström, D. Gottlieb, A stable and conservative interface treatment of arbitrary spatial accuracy, J. Comput. Phys. 148 (2) (1999) 341–365.
[10] S. Tan, C.W. Shu, Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws, J. Comput. Phys. 229 (2010) 8144–8166.
[11] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, Appl. Numer. Math. 62 (2012) 328–341.
[12] D. Elfverson, M.G. Larson, K. Larsson, CutIGA with basis function removal, Adv. Model Simul. Eng. 5 (2018) 1–19.
[13] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: discretizing geometry and partial differential equations, Int. J. Numer. Meth. Engng. 104 (2015) 472–501.
[14] E. Burman, P. Hansbo, M.G. Larson, S. Zahedi, Cut finite element methods, Acta Numer. (2025) 1–121.
[15] S. Sticko, G. Kreiss, Higher order cut finite elements for the wave equation, J. Sci. Comput. 80 (2019) 1867–1887.
[16] M. Berger, A. Giuliani, A state redistribution algorithm for finite volume schemes on cut cell meshes, J. Comput. Phys. 428 (2021) 109820.
[17] A. Giuliani, A two-dimensional stabilized discontinuous Galerkin method on curvilinear embedded boundary grids, SIAM J. Sci. Comput. 44 (1) (2022) A389–A415.
[18] P. Colella, D.T. Graves, B.J. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, J. Comput. Phys. 211 (2006) 347–366.
[19] F. Streitbürger, G. Birke, C. Engwer, S. May, DoD stabilization for higher-order advection in two dimensions, in: Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2020 + 1: Selected Papers from the ICOSAHOM Conference, Vienna, Austria, July 12-16, 2021, Springer, 2022, pp. 495–508.
[20] P.C. Africa, D. Arndt, W. Bangerth, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, S. Kinnewig, M. Kronbichler, M. Maier, P. Munch, M. Schreter-Fleischhacker, J.P. Thiele, B. Turcksin, D. Wells, V. Yushutin, The deal.II library, version 9.6, J. Numer. Math. 32 (4) (2024) 369–380.
[21] C. Gürkan, S. Sticko, A. Massing, Stabilized cut discontinuous Galerkin methods for advection-reaction problems, SIAM J. Sci. Comput. 42 (2020) A2620–A2654.
[22] E. Stein, Singular integrals and differentiability properties of functions, Princeton University Press, 1970.
[23] M.H. Carpenter, D. Gottlieb, S. Abarbanel, W. Don, The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error, SIAM J. Sci. Comput. 16 (6) (1995) 1241–1252.
[24] J.C. Butcher, Numerical Methods For Ordinary Differential Equations, John Wiley & Sons, 2016.
[25] P. Fu, T. Frachon, G. Kreiss, S. Zahedi, High order discontinuous cut finite element methods for linear hyperbolic conservation laws with an interface, J. Sci. Comput. 90 (3) (2022) 84.
[26] R.I. Saye, High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles, SIAM J. Sci. Comput. 37 (2) (2015) A993–A1019.