

SMAI-JCM
SMAI JOURNAL OF
COMPUTATIONAL MATHEMATICS

The domain-of-dependence
stabilization for cut-cell meshes is
fully discretely stable

LOUIS PETRI, GUNNAR BIRKE, CHRISTIAN ENGWER & HENDRIK RANOCHA

Volume 12 (2026), p. 187-218.

<https://doi.org/10.5802/smai-jcm.147>

© The authors, 2026.



*The SMAI Journal of Computational Mathematics is a member
of the Centre Mersenne for Open Scientific Publishing*

<http://www.centre-mersenne.org/>

Submissions at <https://smai-jcm.centre-mersenne.org/ojs/submission>

e-ISSN: 2426-8399





The domain-of-dependence stabilization for cut-cell meshes is fully discretely stable

LOUIS PETRI¹
GUNNAR BIRKE²
CHRISTIAN ENGWER³
HENDRIK RANOCHA⁴

¹ Institute of Mathematics, Johannes Gutenberg University Mainz, Staudingerweg 9, 55128 Mainz, Germany

E-mail address: lpetri01@uni-mainz.de

² Applied Mathematics, University of Münster, Orleans-Ring 10, 48149 Münster, Germany

³ Applied Mathematics, University of Münster, Orleans-Ring 10, 48149 Münster, Germany

⁴ Institute of Mathematics, Johannes Gutenberg University Mainz, Staudingerweg 9, 55128 Mainz, Germany.

Abstract. We present a fully discrete stability analysis of the domain-of-dependence stabilization for hyperbolic problems. The method aims to address issues caused by small cut cells by redistributing mass around the neighborhood of a small cut cell at a semi-discrete level. Our analysis is conducted for the linear advection model problem in one spatial dimension. We demonstrate that fully discrete stability can be achieved under a time step restriction that does not depend on the arbitrarily small cells, using an operator norm estimate. Additionally, this analysis offers a detailed understanding of the stability mechanism and highlights some challenges associated with higher-order polynomials. We also propose a way to mitigate these issues to derive a feasible CFL-like condition. The analytical findings, as well as the proposed solution are verified numerically in one- and two-dimensional simulations.

2020 Mathematics Subject Classification. 65M12, 65M20, 65M70.

Keywords. cut-cell meshes, discontinuous Galerkin methods, domain-of-dependence stabilization, semi-bounded operators, energy stability, Runge–Kutta methods.

1. Introduction

Constructing a body-fitted mesh is time-consuming in cases where computational domains have irregular geometries. This can be avoided by using an efficient cut-cell mesh that extracts the computational domain from a Cartesian background mesh. This approach has become increasingly popular over the past decades, most recently for hyperbolic partial differential equations (PDEs). When employing discontinuous Galerkin (DG) methods in space, a cut-cell mesh poses several challenges: besides the arbitrarily shaped cells, the use of explicit solvers for time integration results in a CFL condition that requires a very small time step, which affects both stability and the correct domain of dependence.

In recent years, initial progress has been made in the field of hyperbolic cut-cell DG methods. In [21, 26, 37], an adaptation of the ghost penalty method for hyperbolic problems was examined, originating from elliptic cut-cell techniques. Another approach is the state redistribution method [23, 46], which was originally developed for finite volume schemes and has recently been extended to DG methods [5]. In this work, we analyze the domain-of-dependence (DoD) stabilization introduced in [19]. The core

The authors were supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the DFG priority program SPP 2410 (project number 526031774). LP and HR were also supported by the Daimler und Benz Stiftung (Daimler and Benz foundation, project number 32-10/22). GB and CE were also supported by the Germany's Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure.

<https://doi.org/10.5802/smai-jcm.147>

© The authors, 2026

TABLE 1.1. Summary of the proven properties of DoD-stabilized DG schemes. Depending on the case considered, the details of the DoD methods may differ from the version introduced in this article.

			advection	linear systems	scalar nonlinear eq.
monotonicity,	fully discrete,	P^0	1D [19]		1D [31]
TVD,	fully discrete,	P^0	1D [19]		
TVDM,	fully discrete,	P^1	1D [19]		
SBP,	semidiscrete,	P^k	1D [34]		
L^1 -stability,	fully discrete,	P^0	1D [19]		
L^2 -stability,	semidiscrete,	P^k	2D [39]	1D [34], 2D [9]	1D [31]
convergence,	fully discrete,	P^0	2D [8]		

idea of DoD is to add penalty terms into the semidiscretization that redistribute fluxes, adjusting the mass distribution within and between cells while restoring the domain of dependence for the downwind cell. Promising results have been observed for nonlinear hyperbolic equations in one spatial dimension [31] and for linear systems in two dimensions [8, 10]. Specifically, it has been shown in [10, 39] that the semidiscretization of DoD is stable for linear systems, and according to [31], its first-order case produces a monotone scheme. A list of all proven DoD properties up to this point is given in Table 1.1. Similar findings have been achieved for the state redistribution method [6, 46].

The novelty of this paper is, that we extend the semidiscrete stability analysis of the DoD method to a fully discrete setting. Specifically, we use an explicit Runge–Kutta method and examine whether the resulting fully discrete scheme is stable, meaning whether the numerical solution satisfies $\|u^{n+1}\| \leq \|u^n\|$, $n \geq 0$ under a CFL-like time step restriction that is independent of the arbitrarily small cut cells.

Our analysis focuses on the one-dimensional case to avoid irregularly shaped cells and specifically examine the stabilization mechanism for the linear advection equation. Building on studies that explore the strong stability of explicit Runge–Kutta (RK) methods [1, 36, 40, 41, 43], we utilize the semidiscrete stability of the DoD stabilization to choose high-order time-integration schemes and reduce the fully-discrete stability issue to a CFL-like condition $\Delta t \leq c\|L\|_M \leq C$, where $\|L\|_M$ is the discrete operator norm of the right-hand side L of the semidiscrete system $u'(t) = Lu(t)$. For the unstabilized scheme, this scales inversely with the smallest cell size. We show that applying DoD stabilization restores this scaling to that of the background cell by analyzing the involved mechanisms. To do so, we will derive the corresponding system of ordinary differential equations (ODEs) for the semidiscretization and estimate the operator norms of the submatrices in the sparse system. Using this approach, we can identify, explain, and improve additional mechanisms that impose further restrictions on the time step needed for stability.

This article is organized as follows. We start by introducing the problem setting, the method, and elaborating on our matrix notation of the semidiscrete system in Section 2. In Section 3, we provide necessary estimates for the involved quadrature nodes and weights, which we use in Section 4 for some initial partial operator norm estimates. In Section 5, we prove the main result, which states strong stability for an appropriate time step that is independent of arbitrarily small cells. An approach for improving other related problems is given in Section 6. Finally, we present some numerical results in 1D and 2D that confirm the theoretical observations.

2. Preliminaries

2.1. Problem setting

For the scope of this work, we consider the linear advection equation as a hyperbolic model problem in one spatial dimension

$$\partial_t u(t, x) + a \partial_x u(t, x) = 0, \quad u(0, x) = u_0(x),$$

on a domain $\Omega = (x_L, x_R)$ with periodic boundary conditions and $a > 0$. We focus on this simple hyperbolic model problem to investigate the effects of the DoD stabilization alone. To discretize the spatial domain, we split it into N cells $E_i = (x_{i-1}, x_i)$ for $i = 1, \dots, N$ with vertices $x_0 < x_1 < x_2 < \dots < x_{N-1} < x_N$. To simulate cut cells, we include a cut by the vertex x_c , such that we have the full set of vertices given by $x_0 < x_1 < x_2 < \dots < x_{c-1} < x_c < x_{c+1} < \dots < x_{N-1} < x_N$ with the cell sizes

$$\begin{aligned} |E_i| &= \Delta x_i = \Delta x, \quad i \in \{1, \dots, N\} \setminus \{c, c+1\}, \\ |E_c| &= \Delta x_c = \alpha \Delta x, \quad |E_{(c+1)}| = \Delta x_{c+1} = (1 - \alpha) \Delta x. \end{aligned}$$

Δx will be referred to as the background cell size. We assume that the $\alpha \in [0, 1/2]$, which we call the *cut-cell factor*, such that E_c may be an arbitrarily small cut cell, while $E_{(c+1)}$ is also a cut cell that is limited by the size $\Delta x/2$ from below. This spatial setting is visualized in Figure 2.1.

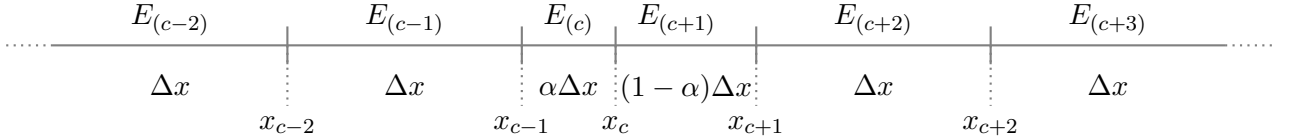


FIGURE 2.1. One-dimensional cut-cell setting.

2.2. Unstabilized scheme

We follow the method of lines approach and discretize in space using a DG method. Thus, we consider the polynomial space of degree p , denoted by \mathcal{P}^p , and the discrete function space

$$\mathcal{V}_h^p(\Omega) = \left\{ v_h \in L^2(\Omega) : v_h|_{E_i} \in \mathcal{P}^p(\Omega), \quad i = 1, \dots, N \right\}.$$

As local basis on each cell we use a Lagrange basis, either Gauss–Legendre (GL) or Gauss–Lobatto–Legendre (GLL). Since the elements of $\mathcal{V}_h^p(\Omega)$ are not well-defined at the vertices, we also introduce the jump

$$[[v_h]]_i = v_h(x_i)|_{E_i} - v_h(x_i)|_{E_{i+1}}.$$

Using a standard upwind flux [16, Section 3.2.1, eq. (3.8)], we obtain the semidiscrete scheme: find $u_h(t) \in \mathcal{V}_h^p$ such that

$$(\partial_t u_h(t), v_h)_{L^2(\Omega)} + a_h(u_h(t), v_h) = 0, \quad \forall v_h \in \mathcal{V}_h^p, \quad (2.1)$$

with the bilinear form

$$a_h(u_h, v_h) = - \sum_{i=1}^N a \int_{x_{i-1}}^{x_i} u_h \partial_x v_h \, dx + \sum_{i=1}^N a u_h|_{E_i}(x_i) [[v_h]]_i, \quad (2.2)$$

where the first sum consists of all volume terms and the second sum of all flux terms over the boundaries, resulting from integration by parts.

We will refer to this as the background method, which can be applied to both the equidistant background mesh without a cut cell and the cut-cell mesh. For time stepping, we utilize an explicit Runge–Kutta method.

2.3. DoD stabilization

In the one-dimensional setting, this scheme has to deal with two problems that arise because of an underlying cut-cell mesh ([38, Section 2.3], [19]):

- (1) Arbitrarily small cells
- (2) Incorrect domain of dependence of the outflow cell

Facing these, we want to avoid the background CFL condition $a\Delta t/(\alpha\Delta x) \leq C$, as α may become arbitrarily small, leading to unfeasible time-step conditions. To address this, we describe the DoD stabilization, applied as a correction term on the semidiscretization as introduced in [19].

Definition 2.1 (Extension operator). Consider $u_h \in \mathcal{V}_h^p(\Omega)$. The extension operator \mathcal{L}_E extends the polynomial function u_h from cell E to the whole domain Ω by

$$\mathcal{L}_E : \mathcal{V}_h^p(\Omega)|_E \rightarrow P^p(\Omega), \quad \text{where } \mathcal{L}_E(u_h) \in P^p(\Omega) \text{ with } \mathcal{L}_E(u_h)|_E = u_h|_E. \quad (2.3)$$

Note that this is just the evaluation of the uniquely defined polynomial $u_h|_E$ outside E .

The DoD stabilization will address the issues by treating the volume and flux terms of (2.2) separately. To manage the mass flowing into cell E_c from E_{c-1} , we introduce the stabilization term

$$J_h^{0,c}(u_h, v_h) = \eta_c a \left(\mathcal{L}_{E_{(c-1)}}(u_h)(x_c) - u_h|_{E_c}(x_c) \right) \llbracket v_h \rrbracket_c \quad (2.4)$$

for an $\eta_c > 0$, which will be specified soon. We can consider J^0 as a redistribution of fluxes, as it decreases the inflow into the small cell E_c from the inflow cell E_{c-1} and the outflow from E_c into the larger cut cell E_{c+1} .

The DoD stabilization uses an additional term

$$J_h^{1,c}(u_h, v_h) = \eta_c \int_{x_{c-1}}^{x_c} a \left(\mathcal{L}_{E_{(c-1)}}(u_h) - u_h \right) \cdot \partial_x \left(\mathcal{L}_{E_{(c-1)}}(v_h) - v_h \right) dx. \quad (2.5)$$

The term J^1 is intended to regulate the gradients within the cell E_c and is consequently represented by the volume term. Note that we use the extension operator for the cell $E_{(c-1)}$, as this is the only inflow cell of E_c in our specific case. Since we are only considering a single small cut cell, we omit the index c in (2.4) and (2.5) and combine them into the full DoD correction term $J_h = J_h^0 + J_h^1$. The stabilized semidiscretization now takes its final form

$$(\partial_t u_h(t), v_h)_{L^2(\Omega)} + a_h(u_h(t), v_h) + J_h(u_h(t), v_h) = 0, \quad \forall v_h \in \mathcal{V}_h^p. \quad (2.6)$$

Remark 2.2. In a setting with multiple cut cells that require stabilization, we need a correction term for each cut cell. As in the literature, we define a set \mathcal{S} of cells in need of stabilization (in our 1D case, this set would be defined $\mathcal{S} = \{E_j : |E_j| \leq \Delta x/2\}$) and then determining the stabilization terms by

$$J_h^0(u_h, v_h) = \sum_{j \in \mathcal{S}} J_h^{0,j}(u_h, v_h), \quad J_h^1(u_h, v_h) = \sum_{j \in \mathcal{S}} J_h^{1,j}(u_h, v_h),$$

as long as the cut cells are not adjacent. For this practically relevant case the stabilization needs to be extended by a weighted redistribution of similar DoD terms that gets applied to the whole neighborhood of the adjacent small cells. Here, we follow the existing literature and omit this case, as it would exceed the already technical machinery, while not contributing to the main goal of this work.

2.4. Stability properties of DoD

In [39], it was shown that the semidiscretization (2.6) is stable in a semidiscrete sense, i.e., the semidiscrete solution u_h satisfies

$$\|u_h(t)\|_{L^2(\Omega)} \leq \|u_h(0)\|_{L^2(\Omega)} \quad (2.7)$$

for any $t \geq 0$. In the following sections, we will prove that the DoD stabilization is stable in a fully discrete sense. Specifically, by selecting an explicit RK method to solve the semidiscrete system, there exists a time-step Δt that is independent of α , such that

$$\forall n \in \mathbb{N} : \|u^{n+1}\| \leq \|u^n\|.$$

2.5. Choice of η_c

The parameter $\eta_c \in [0, 1]$ determines “how much” we want to stabilize: The closer to 1 it is, the more mass gets redistributed directly to E_{c+1} from E_{c-1} . There are multiple ways to motivate the choice of this parameter.

The simplest is to just consider the cell fraction α : We stabilize by $\eta_c = 1 - \alpha$, such that the mass inflow from E_{c-1} into E_c is exactly given by the ratio of its relative volume α .

A second choice would take the actual domain of dependence into account: By setting $\eta_c = 1 - \min(1, \alpha/\nu)$, where $\nu = a\Delta t/\Delta x$ is the Courant number, we also take care on how many cells the information travels for a time-step. In the first-order case ($p = 0$, time stepping with the explicit Euler method), the fully discrete update for the outflow cell of the small cut cell will then receive the exact amount of information of the cells it depends on [38, Section 3.1.1].

The downside of the second choice is, that the resulting semi-discretization depends on the time step size, which requires to compute the semi-discretization for every step, even for linear systems. How to choose this parameter exactly remains an open question for now. Nevertheless, since also the analysis in [8] considers a time-step independent η_c , this indicates that such a choice would be favorable.

For the sake of our analysis, we will refer to a general η_c in the following by

$$\eta_c = 1 - \min\left\{1, \frac{\alpha}{\lambda_c}\right\}, \quad (2.8)$$

where $\lambda_c > 0$. The previous cases can be restored by setting $\lambda_c = 1$ or changing $\lambda_c = \nu$. In Section 6, we will present a different choice of λ_c , based on our upcoming analysis.

2.6. Achieving fully-discrete stability

For the remaining work, we will formulate the DoD stabilization in the specific version of the discontinuous Galerkin spectral element method (DGSEM) and display the resulting semi-discrete system in matrix formulation. We use this formulation to perform our fully discrete stability analysis. More precisely, we will use collocation with Gauss–Legendre (GL) and Gauss–Lobatto–Legendre (GLL) nodes and quadrature. We want to emphasize two important implications:

- Considering GL nodes for quadrature, the resulting scheme evaluates all integrals exactly for the linear problem, regardless of the choice of basis functions. Additionally, the solution is independent of a specific basis, as the polynomial on each cell is uniquely defined. The following analysis examines Lagrange basis functions at Gauss–Legendre nodes. However, since the operator norm does not depend on the chosen basis, the final result stated in Theorem 2.3 below still applies to the exact formulation (2.6) using exact integrals.
- Considering GLL nodes as quadrature and Lagrange basis nodes, we examine a discontinuous Galerkin spectral element method (DGSEM) with under-integration for constructing the

mass matrix. Therefore, the resulting system of ODEs will be an approximation to the exact semidiscrete system (2.6) for that specific choice. In this case, we will also rely on a semi-discrete stability argument that has not yet been explicitly formulated. However, following [10], the integrals of a_h involve polynomials of maximum degree $2p - 1$, making the GLL quadrature exact [29]. Furthermore, the volume terms are solely handled by integration by parts, which is also maintained by the summation-by-parts property [22] of the DGSEM.

Given a discrete norm $\|\cdot\|_M$ based on the specific DGSEM choice (details are provided in the next Section 2.7), we want to achieve the fully discrete stability, yielding

$$\forall n \in \mathbb{N} : \|u^{n+1}\|_M \leq \|u^n\|_M. \quad (2.9)$$

For that, we follow the techniques of [36, 43] to exploit the semidiscrete stability (2.7) of the DoD stabilization: Consider a spatial semidiscretization written in terms of the linear system $\partial_t u = Lu$. Given that this is stable, we know that the norm of the solution does not increase in time. We can derive

$$0 \geq \frac{d}{dt} \|u\|^2 = \left(\frac{d}{dt} u, u \right) + \left(u, \frac{d}{dt} u \right) = (Lu, u) + (u, Lu).$$

This implies $\forall u : (Lu, u) \leq 0$, i.e., the spatial operator L is semibounded. Applying a suitable explicit Runge–Kutta method for time-stepping, we can exploit the semiboundedness of L to obtain

$$\|u^{n+1}\|_M^2 - \|u^n\|_M^2 \quad (2.10)$$

non-positive under a condition $\Delta t \|L\|_M \leq C$, where C just depends on the chosen Runge–Kutta method. For example, we cite the values $C = 1$ for SSPRK(3, 3) and $C = 0.67493$ for SSPRK(10, 4) from [36].

In general, we obtain such a CFL-like condition for an explicit RK method of order p with p stages, if $p \equiv 3 \pmod{4}$ (while several more methods are leading to that), see [41, Theorem 4.2]. Moreover, Tadmor [44] proved stability in a weaker sense for a broader class of RK methods. These results are not directly applicable to this framework but hint that not only the strongly stable methods classified in [1, 41] ensure stability. Therefore, we reach an arbitrarily high order and all boils down to determine the operator norm $\|L\|_M$, which usually scales with $1/\min\{\Delta x_i : i = 1, \dots, N\}$. This is the common CFL-like condition. Thus, we will show that the stabilized scheme yields an operator norm that is independent of arbitrarily small cells with size $\alpha \Delta x$.

We anticipate the answer to this question and state our main result by the following theorem, while we provide its final proof in Section 5.

Theorem 2.3. *Consider the system of ODEs $\partial_t u = Lu$ that resembles the semidiscrete system (2.6), treated exactly or by under-integration with GLL quadrature. There is a constant C , just depending on the polynomial degree p and the chosen basis/nodes, such that*

$$\|Lu\|_M \leq \frac{Ca}{\Delta x} \|u\|_M,$$

i.e., the operator norm $\|L\|_M$ does not scale with the inverse α^{-1} of the small cut-cell size by applying DoD, as it would without that stabilization.

This implies fully-discrete stability under a cut-cell independent time step restriction, if we apply the DoD stabilization to a cut-cell grid.

2.7. Introducing notation and norms

Considering the semidiscrete system (2.1) or (2.6), we have N cells and due to the Lagrange basis functions of degree p , $u \in \mathbb{R}^{(p+1)N}$ will consist of the nodal values, and we can express it as

$$u = (u_1, u_2, \dots, u_N)^T, \quad u_i \in \mathbb{R}^{(p+1)}, \quad i \in \{1, \dots, N\},$$

where $u_i = (u_{i0}, \dots, u_{ip})^T \in \mathbb{R}^{p+1}$ are the values and $l_i = (l_{i0}, \dots, l_{ip})^T$ are the basis functions in cell E_i , such that $u_h|_{E_i} = \sum_{j=0}^p u_{ij} l_{ij}$. To simplify the notation later, we consider the reference element $R = [-1, 1]$ and an arbitrary cell $E_i = [x_{i-1}, x_i]$ with $\Delta x_i = x_i - x_{i-1}$. We can map the solution from R to E_i by

$$x(\gamma) = \frac{x_i + x_{i-1}}{2} + \gamma \frac{\Delta x_i}{2}.$$

Using the Lagrange basis $\tilde{l} = (\tilde{l}_0, \dots, \tilde{l}_p)^T$ at reference GL/GLL points in R , we have

$$u|_{E_i} = \sum_{k=0}^p u_{ik} l_{ik}(x(\gamma)) = \sum_{k=0}^p u_{ik} \tilde{l}_k(\gamma). \quad (2.11)$$

Denote the GL/GLL nodes $\{\tilde{x}_0, \dots, \tilde{x}_{Rp}\}$ and quadrature weights $\{\omega_0, \dots, \omega_p\}$ on the reference element R . Further, we consider the (GL or GLL) cut-cell nodes $\{\xi_{i0}, \dots, \xi_{ip}\} \in E_{i+1} = [x_i, x_i + \alpha \Delta x_i]$ and $\{\xi_0, \dots, \xi_p\}$ as the transformed nodes to the *reference cut cell* $[1, 2\alpha]$.

Considering polynomials $u_h, v_h \in \mathcal{V}_h^p$ in their Lagrange basis $u_h|_{E_i} = \sum_{j=0}^p u_{ij} l_{ij}$, $v_h|_{E_i} = \sum_{j=0}^p v_{ij} l_{ij}$, collocation leads to the discrete inner product

$$\begin{aligned} (u_h|_{E_i}, v_h|_{E_i})_{M_i} &= \sum_{k=0}^p \omega_k \Delta x_i / 2 \sum_{j=0}^p u_{ij} \tilde{l}_{ij}(\tilde{x}_k) \sum_{m=0}^p v_{im} \tilde{l}_{im}(\tilde{x}_k) \\ &= \sum_{k=0}^p \omega_k \Delta x_i / 2 u_{ik} v_{ik} = u_i^T M_i v_i \end{aligned}$$

for $i = 1, \dots, N$. Here, $\omega_k \Delta x_i / 2$ are the scaled quadrature weights for the cell E_i and the resulting mass matrix $M_i = \text{diag}(\omega_1, \dots, \omega_p) \Delta x_i / 2$ is diagonal. We can abbreviate this by writing $(u_i, v_i)_{M_i} := (u_h|_{E_i}, v_h|_{E_i})_{M_i}$. From this point, we can define the discrete norms

$$\|u_i\|_{M_i} := \sqrt{(u_i, v_i)_{M_i}}, \quad \|u\|_M := \sqrt{\sum_{i=1}^N \|u\|_{M_i}^2} = u^T M u,$$

where $M = \text{diag}(M_1, M_2, \dots, M_N)$ is the global mass matrix. Analogue, we can define the norm at the reference element R for $u \in \mathbb{R}^{p+1}$, that resembles the nodal values of a cell transformed to R :

$$\|u\|_{M_R} = u^T M_R u, \quad M = \text{diag}(\omega_1, \dots, \omega_p).$$

Naturally we can define the operator norm for every introduced vector norm. In the following, all the respective vector-induced operator norms are denoted by $\|A\|$, while its corresponding vector norm is uniquely defined by the dimension of $A \in \mathbb{R}^{n \times n}$: For $n = (p+1) \times N$, the vector norm is $\|\cdot\|_M$, while for $n = (p+1)$ it is $\|\cdot\|_{M_R}$ or $\|\cdot\|_{M_i}, i \in 1, \dots, N$. We can unify the latter cases because they induce the same operator norm: Let $u \in \mathbb{R}^{p+1}$, $A \in \mathbb{R}^{(p+1) \times (p+1)}$ and use for the cause of clarification $\|A\|_i$ as the induced operator norm of $\|u\|_{M_i}$, then

$$\|A\|_{M_i} = \max_{u \in \mathbb{R}^{p+1}} \frac{\|Au\|_{M_i}}{\|u\|_{M_i}} = \max_{u \in \mathbb{R}^{p+1}} \frac{\sqrt{\Delta x_i / 2} \|Au\|_{M_R}}{\sqrt{\Delta x_i / 2} \|u\|_{M_R}} = \max_{u \in \mathbb{R}^{p+1}} \frac{\sqrt{\Delta x_j / 2} \|Au\|_{M_j}}{\sqrt{\Delta x_j / 2} \|u\|_{M_j}} = \|A\|_{M_j}. \quad (2.12)$$

As usual, when we consider a polynomial space \mathcal{V}_h^p , the involved integrals account for polynomials up to degree $2p$. Therefore, we need to be mindful of the discrete inner product, which we use to

approximate the L^2 inner product. Since the Gauss–Legendre quadrature is exact for polynomials up to degree $2p + 1$, we have for this case

$$(u, v)_M = (u, v)_{L^2([-1,1])}, \quad u \in \mathcal{P}^p([-1, 1]) \quad (2.13)$$

In the case of Gauss–Lobatto–Legendre quadrature, we encounter the well-known issue that it is exact for polynomials up to degree $2p - 1$. Consequently, for arbitrary u and v of degree p , we cannot guarantee exactness. However, with respect to the discrete norm, we know that the discrepancy to the exact value of the involved integral is bounded by a factor $\sqrt{3}$, see [30], [13, (5.3.2)].

2.8. The matrix notation

To prove Theorem 2.3, we reformulate the semidiscrete system (2.6) as a system of ODEs

$$\partial_t u = Lu.$$

The exact shape of L will be derived in this section. We construct all matrices on the reference element, so we can explicitly highlight the dependence of the cell size. Therefore, we require some relations at first.

Definition 2.4. We define (see e.g. [28, eqs. (3.31), (3.47)])

$$\begin{aligned} D &= \left(\tilde{l}'_j(\tilde{x}_{Rl}) \right)_{l,j \in \{0, \dots, p\}}, & D_{E_i} &= \left(l'_{ij}(\xi_{il}) \right)_{l,j \in \{0, \dots, p\}}, & & \text{(derivative matrix),} \\ \hat{R} &= (\tilde{l}_0(1), \dots, \tilde{l}_p(1)), & \hat{R}_{E_i} &= (l_{i0}(x_i), \dots, l_{ip}(x_i)), & & \text{(right boundary extrapolation matrix),} \\ \hat{L} &= (\tilde{l}_0(-1), \dots, \tilde{l}_p(-1)), & \hat{L}_{E_i} &= (l_{i0}(x_{i-1}), \dots, l_{ip}(x_{i-1})), & & \text{(left boundary extrapolation matrix),} \\ \mathbb{I} &= \left(\tilde{l}(\xi_l) \right)_{l,j \in \{0, \dots, p\}}, & \mathbb{I}_{E_i} &= (l_{ji}(\xi_{il}))_{l,j \in \{0, \dots, p\}}, & & \text{(extrapolation matrix).} \end{aligned}$$

Lemma 2.5. *It holds*

$$\begin{aligned} (1) \quad D_{E_i} &= \frac{2}{\Delta x_i} D, \quad M_i = \frac{\Delta x_i}{2} M_R, \\ (2) \quad \hat{R}_{E_i} &= \hat{R}, \quad \hat{L}_{E_i} = \hat{L}, \quad \mathbb{I}_{E_i} = \mathbb{I}. \end{aligned}$$

Proof. For fixed l, j , we have by transformation $l_{ik}(x(\gamma)) = \tilde{l}_k(\gamma)$. Therefore,

$$\tilde{l}'_j(\tilde{x}_l) = (l_{ij}(x_{il}(\tilde{x}_l)))' = l'_{ij}(x_{il}) \cdot x'_{il}(\tilde{x}_l) = l'_{ij}(x_{il}) \frac{\Delta x_i}{2}.$$

The remaining statements follow just by the identity of transformation $l_{ik}(x(\gamma)) = \tilde{l}_k(\gamma)$ as in (2.11). \blacksquare

We proceed inserting all the basis functions l_1, \dots, l_N into the semidiscrete system to construct L . For $i, k = 1, \dots, N$, $j = 1, \dots, p$ with $k \neq i$, we have $(\partial_t u_h, l_{ij})_{M_k} = 0$, $(\partial_t u_h, l_{ij})_{M_i} = (\partial_t u_i, l_{ij})_{M_i}$ and $(u_h, \partial_x l_{ij})_{M_i} = (u_i, \partial_x l_{ij})_{M_R}$, because $l_{ij}|_{\Omega \setminus E_i} = 0$. We obtain for the respective terms of (2.1)

$$(\partial_t u_i, l_j)_{M_i} = \frac{\Delta x_i}{2} \omega_j \partial_t u_{ij}, \quad \Rightarrow (\partial_t u_i, l)_{M_i} = \frac{\Delta x_i}{2} M \partial_t u_i \quad (2.14)$$

$$(u_i, \partial_x l_j)_{M_i} = \sum_{k=0}^p u_k \frac{2}{\Delta x_i} D_{kj} \frac{\Delta x_i}{2} \omega_k = \sum_{k=0}^p D_{jk}^T \omega_k u_k, \quad \Rightarrow (u_i, \partial_x l)_{M_i} = D^T M u_i. \quad (2.15)$$

The remaining boundary terms also require the nodal values u_{i-1} of the inflow neighbor cell. As their values at x_{i-1} and x_i are generally (here: for GL nodes) not represented by the nodal values, we need

extrapolation to the vertices. Using Definition 2.4, the boundary terms of (2.2), with $v_h = l_{ij}$ are given by

$$\begin{aligned} u_i(x_i)l_{ij}(x_i) - u_{i-1}(x_{i-1})l_{ij}(x_{i-1}) &= l_{ij}(x_i)\widehat{R}_{E_i}u_i - l_{ij}(x_{i-1})\widehat{R}_{E_{i-1}}u_{i-1}, \quad j \in \{0, \dots, p\} \\ \Rightarrow u_i(x_i)l_i(x_i) - u_{i-1}(x_{i-1})l_i(x_{i-1}) &= \widehat{R}_{E_i}^T \widehat{R}_{E_i}u_i - \widehat{L}_{E_i}^T \widehat{R}_{E_{i-1}}u_{i-1} = \widehat{R}^T \widehat{R}u_i - \widehat{L}^T \widehat{R}u_{i-1} \end{aligned} \quad (2.16)$$

Finally combining (2.14), (2.15) and (2.16) transforming to an arbitrary cell E_i , $i \in \{1, \dots, N\}$, we obtain

$$M\partial_t u_i = \frac{2a}{\Delta x} \left(D^T M u_i - \widehat{R}^T \widehat{R} u_i + \widehat{L}^T \widehat{R} u_{i-1} \right)$$

for the background method.

The same procedure can be applied to the stabilization terms (2.4) and (2.5). Using the notation introduced in 2.1, we have

$$\begin{aligned} J_h^{0,c}(u_h, l_c) &= a\eta_c l_c(x_c) \left(\mathcal{L}_{E_{(c-1)}}(u_h)(x_c) - u_h|_{E_c}(x_c) \right), \\ J_h^{0,c}(u_h, l_{(c+1)}) &= -a\eta_c l_{(c+1)}(x_c) \left(\mathcal{L}_{E_{(c-1)}}(u_h)(x_c) - u_h|_{E_c}(x_c) \right), \\ J_h^{0,c}(u_h, l_i) &= \mathbf{0}, \quad i \notin \{c, (c+1)\}, \quad \text{with } \mathbf{0} = (0, \dots, 0)^T \in \mathbb{R}^{(p+1)}. \end{aligned}$$

Now, we want to specify all the appearing terms further. Using Lemma 2.5, we can write $l_c(x_c) = (l_{c0}(x_c), \dots, l_{cp}(x_c))^T = \widehat{R}^T$ and $l_{c+1}(x_c) = \widehat{L}^T$. With the same arguments, the extrapolation B_{J_0} from $L_{(c-1)}$ to the outflow boundary x_c , that is given by $B_{J_0}u_{(c-1)} = \mathcal{L}_{E_{(c-1)}}(u_h)(x_c)$, we observe that B_J is just the extrapolation from the basis functions $l_{(c-1)}$ of the inflow cell to the outflow boundary of the cut cell x_c , i.e., $B_J = \sum_{k=0}^p l_{k(c-1)}(x_c)$. The last term is just the already treated extrapolation to the outflow boundary and therefore given by $u_h|_{E_c}(x_c) = \widehat{R}u_c$. So we can conclude the matrix-reformulation of (2.4) to

$$J_h^{0,c} \left(u_h, \begin{pmatrix} l_c \\ l_{(c+1)} \end{pmatrix} \right) = a\eta_c \begin{pmatrix} \widehat{R}^T B_{J_0} & -\widehat{R}^T \widehat{R} \\ -\widehat{L}^T B_{J_0} & \widehat{L}^T \widehat{R} \end{pmatrix} \begin{pmatrix} u_{(c-1)} \\ u_c \end{pmatrix}, \quad J_h^{0,c}(u_h, l_i) = \mathbf{0}, \quad i \notin \{c, (c+1)\}.$$

We abbreviate $\widehat{B}_R = \widehat{R}^T \widehat{R}$, $\widehat{B}_L = \widehat{L}^T \widehat{R}$, $\widehat{B}_{J_1} = \widehat{R}^T B_{J_0}$ and $\widehat{B}_{J_2} = \widehat{L}^T B_{J_0}$ in the remaining work.

Continuing with $J_h^{1,c}$, we just have a few non-zero terms again because of the integral over E_c :

$$\begin{aligned} J_h^{1,c}(u_h, l_{(c-1)}) &= a\eta_c \int_{x_{c-1}}^{x_c} \left(\mathcal{L}_{E_{(c-1)}}(u_h) - u_h \right) \cdot \partial_x \mathcal{L}_{E_{(c-1)}}(l_{(c-1)}) \, dx, \\ J_h^{1,c}(u_h, l_c) &= -a\eta_c \int_{x_{c-1}}^{x_c} \left(\mathcal{L}_{E_{(c-1)}}(u_h) - u_h \right) \cdot \partial_x l_c \, dx, \\ J_h^{1,c}(u_h, l_i) &= \mathbf{0}, \quad i \notin \{(c-1), c\}, \quad \text{with } \mathbf{0} = (0, \dots, 0)^T \in \mathbb{R}^{(p+1)}. \end{aligned}$$

In these equations, we have to evaluate $\mathcal{L}_{E_{(c-1)}}(u_h)$ inside the cell E_i , i.e., at its quadrature nodes. Therefore, we can write this again as the extrapolation matrix $\mathbb{I} = \mathbb{I}_{E_c}$, such that $\mathbb{I}u_{(c-1)} = \mathcal{L}_{E_{(c-1)}}(u_h)|_{E_c}$. From this point, we can proceed as for (2.15):

$$\begin{aligned} J_h^{1,c}(u_h, l_{ci}) &= -a\eta_c \sum_{j=0}^p \left((\mathbb{I}u_{(c-1)})_j - u_{c_j} \right) \cdot \frac{2}{\alpha \Delta x} D_{ji} \frac{\alpha \Delta x}{2} \omega_j = -a\eta_c \left(\sum_{j=0}^p D_{ij}^T \omega_j (\mathbb{I}u_{(c-1)} - u_c)_j \right) \\ \Rightarrow J_h^{1,c}(u_h, l_c) &= -a\eta_c D^T M (\mathbb{I}u_{(c-1)} - u_c) = a\eta_c \begin{pmatrix} -D^T M \mathbb{I} & D^T M \end{pmatrix} \begin{pmatrix} u_{(c-1)} \\ u_c \end{pmatrix}. \end{aligned}$$

To rewrite the last remaining term $J_h^{1,c}(u_h, l_c)$, we first have to focus on $\partial_x \mathcal{L}_{E_{in}}(l_{(c-1)j}(x))$ in detail.

and the blocks affected by the DoD stabilization take the form

$$\begin{aligned}
 L_{(c-1)} &= S_{c-1}M^{-1}\left(D^T M - \widehat{B}_R - \eta_c \mathbb{I}^T D^T M \mathbb{I}\right), \\
 L_{(c-1)R} &= S_{c-1}M^{-1}\left(\eta_c \mathbb{I}^T D^T M\right), \\
 L_{cL} &= S_c M^{-1}\left(\eta_c D^T M \mathbb{I} + \widehat{B}_L - \eta_c \widehat{B}_{J_1^0}\right), \\
 L_c &= S_c M^{-1}\left(D^T M - \eta_c D^T M - \widehat{B}_R + \eta_c \widehat{B}_R\right), \\
 L_{(c+1)LL} &= S_{c+1}M^{-1}\eta_c \widehat{B}_{J_2^0}, \\
 L_{(c+1)L} &= S_{c+1}M^{-1}\left(\widehat{B}_L - \eta_c \widehat{B}_L\right),
 \end{aligned} \tag{2.18}$$

with the scaling coefficients

$$S_i = \frac{2a}{\Delta x_i}.$$

We will use this construction to prove the main result of this work in Section 5, stated by Theorem 2.3.

3. Interpolation and quadrature properties

For the estimates of partial operator norms in Section 4, we require certain properties regarding interpolation nodes and quadrature weights that depend on the polynomial degree and the type of nodes. Specifically, these include the minimal distance between two nodes and the largest ratio of two quadrature weights. Since we cannot calculate the nodes and weights exactly for Gauss–Legendre quadrature or for Gauss–Lobatto–Legendre, we will utilize known estimates and adapt them in this section to suit our requirements.

Since we are focusing on the reference element $[-1, 1]$ in this section and have to consider the details of the different node types, we specify our notation by denoting the Gauss–Legendre nodes as $\widehat{x}_0, \dots, \widehat{x}_p \in (-1, 1)$ and the Gauss–Lobatto–Legendre nodes as $\bar{x}_0, \dots, \bar{x}_p \in [-1, 1]$. Similarly, we denote the respective quadrature weights as $\widehat{\omega}_0, \dots, \widehat{\omega}_p$ and $\bar{\omega}_0, \dots, \bar{\omega}_p$. As this specification will not be required in the subsequent sections, we will shift then back to the notation $\tilde{x}_0, \dots, \tilde{x}_p$, that represents both cases again.

The following estimates are based on known estimates for the Gauss–Legendre and Gauss–Lobatto–Legendre nodes and weights: For the half of the nodes inside $(-1, 0)$, we can write

$$\widehat{x}_{j-1} = -\cos(\widehat{\theta}_j), \quad j \in \left\{1, \dots, \left\lfloor \frac{p+1}{2} \right\rfloor\right\}, \tag{3.1}$$

$$\bar{x}_j = -\cos(\bar{\theta}_j), \quad j \in \left\{1, \dots, \left\lfloor \frac{p-1}{2} \right\rfloor\right\}, \tag{3.2}$$

for some $\widehat{\theta}, \bar{\theta} \in (0, \frac{\pi}{2})$. Note that we have $\bar{x}_0 = -1$ for every p and $\bar{x}_{\frac{p}{2}} = \widehat{x}_{\frac{p}{2}} = 0$, if p is even. Then, we have from [13, eqs. (2.3.14) and (2.3.15)] that

$$\left(j - \frac{1}{2}\right) \frac{\pi}{p+1} < \widehat{\theta}_j < j \frac{\pi}{p+2}, \quad j \in \left\{1, \dots, \left\lfloor \frac{p+1}{2} \right\rfloor\right\}, \tag{3.3}$$

$$j \frac{\pi}{p} < \bar{\theta}_j < (j+1) \frac{\pi}{p+1}, \quad j \in \left\{1, \dots, \left\lfloor \frac{p-1}{2} \right\rfloor\right\}. \tag{3.4}$$

We receive the other half of the nodes by symmetry to the origin with the same properties. Further, detailed information about these estimates can be found in [42] and [32]. As the proofs of Lemmas 3.1 and 3.2 are rather technical, we provide them in the Appendices A.1 and A.2.

Lemma 3.1 (Minimal weight quotient estimate). *There are constants $\widehat{K}_1, \overline{K}_1$, such that for all $i, j \in \{0, \dots, p\}$*

$$\frac{\widehat{\omega}_i}{\widehat{\omega}_j} \leq \widehat{K}_1 p, \quad \frac{\overline{\omega}_i}{\overline{\omega}_j} \leq \overline{K}_1 p,$$

i.e., the quotient of any weights increases at worst linearly in p .

Lemma 3.2 (Minimal node distance estimate). *There are constants $\widehat{K}_2 > 0$ and $\overline{K}_2 > 0$, such that for all $i, j \in \{0, \dots, p\}$ with $i \neq j$,*

$$|\widehat{x}_i - \widehat{x}_j| \geq \frac{\widehat{K}_2}{p^3}, \quad |\overline{x}_i - \overline{x}_j| \geq \frac{\overline{K}_2}{p^3}.$$

Remark 3.3. We note that the cubic dependency on p for the GLL case yields closer nodes for higher orders. Numerical tests indicate that a bound $\overline{x}_i - \overline{x}_j \geq C/p^2$ (as we technically achieved inside the proof of Lemma 3.2 for the Gauss–Legendre nodes) seems more valid. Using other boundary estimates for θ_j , as in [32, Lemma 2.4] might be sufficient for that.

4. Preparation of the proof

In this section, we analyze the operator norm of the various forms of the extension operator $\mathcal{L}_{E_{in}}$ that is given by \mathbb{I} in the matrix notation. Because of Lemma 2.5, we can consider the reference element, where \mathbb{I} interpolates the solution from $[-1, 1]$ to $[1, 1 + 2\alpha]$.

Lemma 4.1. *There is a constant C depending on the chosen type of basis (but not on p), such that*

$$\|\mathbb{I}u\|_{M_R} \leq Cp^{\frac{3}{2}} \left((2 + 2\lambda_c)p^3 \right)^p \|u\|_{M_R},$$

i.e., the operator norm of \mathbb{I} scales (at most) exponentially in the polynomial degree p .

Proof. The interpolation matrix is defined by the standard Lagrange interpolation polynomials \widetilde{l} defined at $\widetilde{x}_0, \dots, \widetilde{x}_p$ and evaluated at ξ_0, \dots, ξ_p . Thus, for $u \in \mathbb{R}^{p+1}$, we have

$$\begin{aligned} \|\mathbb{I}u\|_{M_R}^2 &= \left\| \left(\sum_{i=0}^p \widetilde{l}_i(\xi_1)u_i, \sum_{i=0}^p \widetilde{l}_i(\xi_2)u_i, \dots, \sum_{i=0}^p \widetilde{l}_i(\xi_p)u_i \right)^T \right\|_{M_R}^2 \\ &= \sum_{j=0}^p \omega_j \left(\sum_{i=0}^p \widetilde{l}_i(\xi_j)u_i \right)^2 = \sum_{j=0}^p \omega_j \left(\sum_{i=0}^p \widetilde{l}_i^2(\xi_j)u_i^2 + 2 \sum_{\substack{i,k=0 \\ i < k}}^p \widetilde{l}_i(\xi_j)u_i \widetilde{l}_k(\xi_j)u_k \right) \\ &\leq \sum_{j=0}^p \omega_j \left(\sum_{i=0}^p \widetilde{l}_i^2(\xi_j)u_i^2 + 2 \sum_{\substack{i,k=0 \\ i < k}}^p \frac{\widetilde{l}_i^2(\xi_j)u_i^2}{2} + \frac{\widetilde{l}_k^2(\xi_j)u_k^2}{2} \right) = p \sum_{j=0}^p \omega_j \sum_{i=0}^p \widetilde{l}_i(\xi_j)^2 u_i^2 \end{aligned}$$

by applying Young’s inequality in the estimate. We have to change the coefficients of the weights in front of every u_i . Therefore, we use Lemma 3.1 to obtain $K_1 p \geq \max\{\frac{\omega_i}{\omega_j}, i, j \in \{0, \dots, p\}\}$ for both

types of nodes, such that we can redistribute and unite all $u_i \in \mathbb{R}$ to

$$\begin{aligned}
 \|\mathbb{I}u\|_{M_R}^2 &\leq K_1 p^2 \sum_{j=0}^p \left(\sum_{i=0}^p \tilde{l}_j(\xi_i)^2 \right) \omega_j u_j^2 \\
 &\leq K_1 p^2 \left(\max_{j \in \{0, \dots, p\}} \sum_{i=0}^p \tilde{l}_j(\xi_i)^2 \right) \|u\|_{M_R}^2 \\
 &\leq K_1 p^3 \left(\max_{j \in \{0, \dots, p\}} \tilde{l}_j(\xi_p)^2 \right) \|u\|_{M_R}^2 \\
 &= K_1 p^3 \left(\max_{j \in \{0, \dots, p\}} \prod_{\substack{m=0 \\ m \neq j}}^p \frac{1 + \alpha + \alpha \tilde{x}_p - \tilde{x}_m}{\tilde{x}_j - \tilde{x}_m} \right)^2 \|u\|_{M_R}^2 \\
 &\leq K_1 p^3 \left(\prod_{\substack{m=0 \\ m \neq j}}^p \frac{2 + 2\lambda_c}{\tilde{x}_j - \tilde{x}_m} \right)^2 \|u\|_{M_R}^2,
 \end{aligned}$$

if we stabilize just for $\alpha \leq \lambda_c$.

As shown in Lemma 3.2, there is a constant K_2 , such that the distance $\tilde{x}_i - \tilde{x}_{i-1}$ of every node pair is larger than K_2/p^3 for every i . Therefore, we can estimate the product further to

$$\|\mathbb{I}u\|_{M_R}^2 \leq K_1 p^3 (2 + 2\lambda_c)^{2p} K_2 (p^3)^{2p} \|u\|_{M_R}^2. \quad \blacksquare$$

The following statement about the derivative of the interpolation is similar to the previous one; however, we require and indeed obtain a proportionality to α for its operator norm. The remaining factor will again exhibit an exponential scaling in the magnitude p^p , which also applies to problems with larger cut cells.

Lemma 4.2. *There exists a $\mathcal{C}(p)$, that may depend exponentially on p , such that*

$$\|D\mathbb{I}u\|_{M_R} \leq \alpha \mathcal{C}(p) \|u\|_{M_R}.$$

This holds for all cut cell factors α .

The proof can be found in Appendix A.3 and is mainly based on the properties of the differentiation matrix D and the exponential scaling as in Lemma 4.1.

To cover all cases that will arise, we need another similar statement about the extension operator that gets evaluated at the inflow boundary of the outflow cell of the cut cell. The required arguments were partially previously used in Lemma 4.1.

Lemma 4.3. *There is a constant C , such that*

$$\|\widehat{B}_{J_2} u\|_{M_R} \leq \left(C p^{\frac{3}{2}} ((2 + 2\lambda_c) p^3)^p \right) \|u\|_{M_R}.$$

The proof can be found in Appendix A.4 and uses a similar idea as in Lemma 4.1.

Remark 4.4. Lemmas 4.1, 4.2, and 4.3 indicate that the operator norms $\|\mathbb{I}\|$, $\|D\mathbb{I}\|$, and $\|\widehat{B}_{J_2}\|$ may significantly increase with larger cut cells and higher-order spatial discretizations. The unstable extrapolation is the underlying cause of this effect. Although these estimates may not be sharp, we provide a numerical verification of the scalings in Figure 4.1, depicted for $\|\mathbb{I}\|$.

On the left side, we observe that all orders form a straight line, confirming that for a fixed p , the operator norm scales approximately with α^p (where we can assume $\alpha = \lambda_c$ by considering Lemma 4.1). Similarly, we see in the top right that $\|\mathbb{I}\|$ is indeed exponential in p . However, taking the p -th root

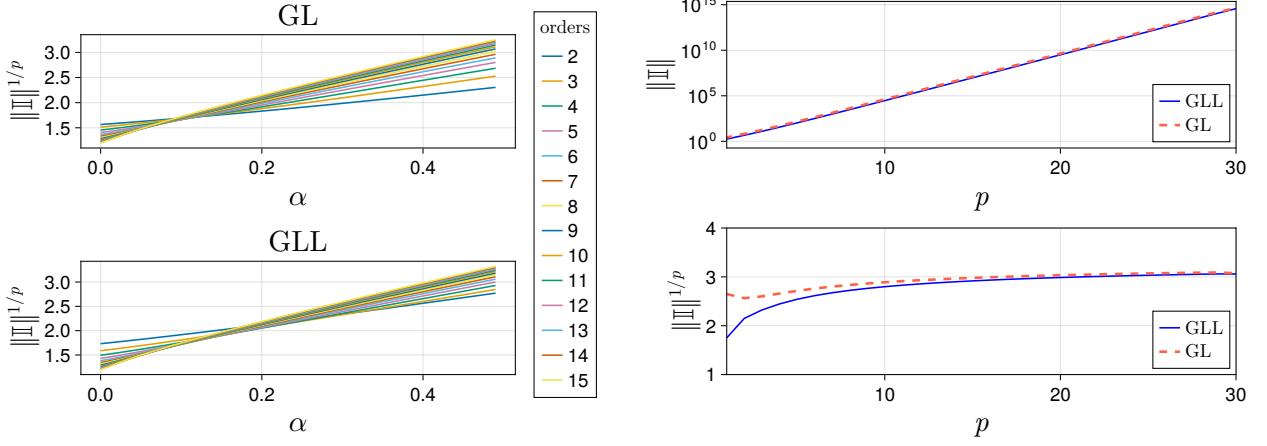


FIGURE 4.1. Operator norm of $\|II\|$ (top right) and $\|II\|^{1/p}$ (else) from the reference element $[-1, 1]$ to the cut cell $[1, 1 + 2\alpha]$. For the plots on the right, we use the fixed value $\alpha = 0.4$.

for every value results in a progression of p^τ for $\tau < 1$. Thus, the scaling factor $(p)^{3p+\frac{3}{2}}$ in Lemma 4.1 is a slight overestimation. A noteworthy misleading estimate lies in $1 + \alpha(1 + \tilde{x}_p) - \tilde{x}_m \leq 2 + 2\lambda_c$, which leads to an amplifying factor, while its actual value may be less than 1 and therefore primarily dampens the norm for small α .

Lemma 4.5. Consider the mass matrix $M_R \in \mathbb{R}^{(p+1) \times (p+1)}$ and an arbitrary $A \in \mathbb{R}^{(p+1) \times (p+1)}$. For the operator norm $\|\cdot\|$ induced by the vector norm given by $\|u\|_{M_R}^2 = u^T M_R u$, we have

$$\|M_R^{-1} A^T M_R\| = \|A\|.$$

Proof. $\|A\|$ corresponds to the largest singular value of A . Therefore, we have $\|A^*\| = \|A\|$, where the adjoint A^* is defined by the property $\langle v, Au \rangle_{M_R} = \langle A^* v, u \rangle_{M_R}$. Therefore, it remains to show that $A^* = M_R^{-1} A^T M_R$. We conclude this by

$$\langle u, M_R^{-1} A^T M_R v \rangle_{M_R} = u^T M_R M_R^{-1} A^T M_R v = u^T A^T M_R v = \langle A^T u, v \rangle_{M_R}. \quad \blacksquare$$

5. Proof of the main result

We have worked out all the tools required for Theorem 2.3 until now. In this section, we will provide proof and elaborate on the important terms occurring in the final estimate.

Proof of Theorem 2.3. We will follow a straightforward estimate by using our previously derived results. Consider L , as it is set up in (2.17). For $u = (u_1, u_2, \dots, u_N)^T \in \mathbb{R}^{(p+1)N}$, $u_n \in \mathbb{R}^{(p+1)}$ for

$1 \leq n \leq N$, we have

$$\begin{aligned} \|Lu\|_M^2 &= \left\| \begin{pmatrix} L_1u_1 + L_1Lu_N \\ L_2Lu_1 + L_2u_2 \\ \vdots \\ L_{(c-1)L}u_{(c-2)} + L_{(c-1)}u_{(c-1)} + L_{(c-1)R}u_c \\ L_{cL}u_{(c-1)} + L_cu_c \\ L_{(c+1)LL}u_{(c-1)} + L_{(c+1)L}u_c + L_{(c+1)}u_{(c+1)} \\ L_{(c+2)L}u_{(c+1)} + L_{(c+2)}u_{(c+2)} \\ \vdots \\ L_{(N-1)L}u_{(N-2)} + L_{(N-1)}u_{(N-1)} \\ L_{NL}u_{N-1} + L_Nu_N \end{pmatrix} \right\|_M^2 \\ &= \|L_1u_1 + L_1Lu_N\|_{M_1}^2 + \|L_2Lu_1 + L_2u_2\|_{M_2}^2 + \cdots \\ &\quad + \|L_{(c-1)L}u_{(c-2)} + L_{(c-1)}u_{(c-1)} + L_{(c-1)R}u_c\|_{M_{(c-1)}}^2 + \|L_{cL}u_{(c-1)} + L_cu_c\|_{M_c}^2 \\ &\quad + \|L_{(c+1)LL}u_{(c-1)} + L_{(c+1)L}u_c + L_{(c+1)}u_{(c+1)}\|_{M_{(c+1)}}^2 + \|L_{(c+2)L}u_{(c+1)} + L_{(c+2)}u_{(c+2)}\|_{M_{(c+2)}}^2 \\ &\quad + \cdots + \|L_{(N-1)L}u_{(N-2)} + L_{(N-1)}u_{(N-1)}\|_{M_{(n-1)}}^2 + \|L_{NL}u_{N-1} + L_Nu_N\|_{M_N}^2, \end{aligned}$$

where $\|\cdot\|_{M_i}^2 = \Delta x_i \|\cdot\|_{M_R}^2$ yields the discrete norm in cell i . By applying the triangle inequality to each vector that appears, we also obtain mixed terms, which we further separate into their natural components using Young's inequality, e.g.,

$$\begin{aligned} \|L_{nL}u_{n-1} + L_nu_n\|_{M_n}^2 &\leq \|L_{nL}u_{n-1}\|_{M_n}^2 + 2\|L_{nL}u_{n-1}\|_{M_n} \|L_nu_n\|_{M_n} + \|L_nu_n\|_{M_n}^2 \\ &\leq 2\left(\|L_{nL}u_{n-1}\|_{M_n}^2 + \|L_nu_n\|_{M_n}^2\right). \end{aligned}$$

Therefore, we can estimate all terms as

$$\begin{aligned} \|Lu\|_M^2 &\leq 2\left(\|L_1u_1\|_{M_1}^2 + \|L_1Lu_N\|_{M_1}^2\right) + 2\left(\|L_2Lu_1\|_{M_2}^2 + \|L_2u_2\|_{M_2}^2\right) + \cdots \\ &\quad + 3\left(\|L_{(c-1)L}u_{c-2}\|_{M_{(c-1)}}^2 + \|L_{(c-1)}u_{(c-1)}\|_{M_{(c-1)}}^2 + \|L_{(c-1)R}u_c\|_{M_{(c-1)}}^2\right) \\ &\quad + 2\left(\|L_{cL}u_{c-1}\|_{M_c}^2 + \|L_cu_c\|_{M_c}^2\right) \\ &\quad + 3\left(\|L_{(c+1)LL}u_{(c-1)}\|_{M_{(c+1)}}^2 + \|L_{(c+1)L}u_c\|_{M_{(c+1)}}^2 + \|L_{(c+1)}u_{(c+1)}\|_{M_{(c+1)}}^2\right) \\ &\quad + 2\left(\|L_{(c+2)L}u_{(c+1)}\|_{M_{(c+2)}}^2 + \|L_{(c+2)}u_{(c+2)}\|_{M_{(c+2)}}^2\right) + \cdots \\ &\quad + 2\left(\|L_{NL}u_{N-1}\|_{M_N}^2 + \|L_Nu_N\|_{M_N}^2\right) \end{aligned}$$

and then apply the compatibility of the operator norm and collect all terms to achieve

$$\begin{aligned} &\|Lu\|_M^2 \\ &\leq 2\left(\sum_{n \notin \{(c-1), c, (c+1)\}} \left(\|L_n\|_{M_n}^2 + \|L_{(n+1)L}\|_{M_n}^2\right) \|u_i\|_{M_n}^2\right) \\ &\quad + 3\left(\|L_{(c-1)}\|_{M_{(c-1)}} \|u_{(c-1)}\|_{M_{(c-1)}}^2 + \|L_{(c-1)R}\|_{M_{(c-1)}} \|u_c\|_{M_{(c-1)}}^2 + \|L_{(c-1)L}\|_{M_{(c-1)}} \|u_{(c-2)}\|_{M_{(c-1)}}^2\right) \\ &\quad + 2\left(\|L_c\|_{M_c} \|u_c\|_{M_c}^2 + \|L_{cL}\|_{M_c} \|u_{(c-1)}\|_{M_c}^2\right) \\ &\quad + 3\left(\|L_{(c+1)LL}\|_{M_{(c+1)}} \|u_{(c-1)}\|_{M_{(c+1)}}^2 + \|L_{(c+1)L}\|_{M_{(c+1)}} \|u_c\|_{M_{(c+1)}}^2 + \|L_{(c+1)}\|_{M_{(c+1)}} \|u_{(c+1)}\|_{M_{(c+1)}}^2\right) \\ &\quad + 2\|L_{(c+2)L}\|_{M_{(c+2)}} \|u_{(c+1)}\|_{M_{(c+1)}}^2 \end{aligned}$$

$$\begin{aligned}
 &= 2 \left(\sum_{n \notin \{(c-1), c, (c+1)\}} \left(\|L_n\|_{M_n}^2 + \|L_{(n+1)L}\|_{M_n}^2 \right) \|u_i\|_{M_n}^2 \right) + \|L_{(c-1)L}\|_{M_{(c-1)}}^2 \|u_{(c-2)}\|_{M_{(c-2)}}^2 \\
 &\quad + \left(3\|L_{(c-1)}\|_{M_{(c-1)}}^2 + 2\alpha\|L_{cL}\|_{M_c}^2 + 3(1-\alpha)\|L_{(c+1)LL}\|_{M_{(c+1)}}^2 \right) \|u_{(c-1)}\|_{M_{(c-1)}}^2 \\
 &\quad + \left(\frac{3}{\alpha}\|L_{(c-1)R}\|_{M_{(c-1)}}^2 + 2\|L_c\|_{M_c}^2 + 3\frac{1-\alpha}{\alpha}\|L_{(c+1)L}\|_{M_{(c+1)}}^2 \right) \|u_c\|_{M_c}^2 \\
 &\quad + \left(3\|L_{(c+1)}\|_{M_{(c+1)}}^2 + \frac{2}{1-\alpha}\|L_{(c+2)L}\|_{M_{(c+2)}}^2 \right) \|u_{(c+1)}\|_{M_{(c+1)}}^2 \\
 &\leq \|u\|_M^2 \max \begin{cases} 2\|L_n\|_{M_n}^2 + 3\|L_{(n+1)L}\|_{M_n}^2, & (n \notin \{(c-1), c, (c+1)\}) \\ \|L_{(c-1)}\|_{M_{(c-1)}}^2 + 2\alpha\|L_{cL}\|_{M_c}^2 + 3(1-\alpha)\|L_{(c+1)LL}\|_{M_{(c+1)}}^2, \\ \frac{3}{\alpha}\|L_{(c-1)R}\|_{M_{(c-1)}}^2 + 2\|L_c\|_{M_c}^2 + 3\frac{1-\alpha}{\alpha}\|L_{(c+1)L}\|_{M_{(c+1)}}^2, \\ \|L_{(c+1)}\|_{M_{(c+1)}}^2 + \frac{2}{1-\alpha}\|L_{(c+2)L}\|_{M_{(c+2)}}^2. \end{cases}
 \end{aligned}$$

We now apply the specific terms of the block matrices, as shown in (2.18). To do this, we omit the indices of the operator, as they all coincide, as noted in (2.12).

$2\|L_n\|^2 + 3\|L_{(n+1)L}\|^2$ originate (for $n \notin \{(c-1), c, (c+1)\}$) from the background method. Therefore, these terms will not be responsible for any problems and we estimate in that case

$$\begin{aligned}
 2\|L_n\|_{M_n}^2 + 3\|L_{(n+1)L}\|^2 &\leq 3\left(\|L_n\|^2 + \|L_{(n+1)L}\|^2\right) \\
 &\leq 3S_n^2\left(2\|M^{-1}\widehat{B}_R\|^2 + 2\|M^{-1}D^T M\|^2 + \|M^{-1}\widehat{B}_L\|\right) \\
 &= 3S_n^2\left(2\|M^{-1}\widehat{B}_R\|^2 + 2\|D\|^2 + \|M^{-1}\widehat{B}_L\|\right),
 \end{aligned}$$

again by using the Cauchy–Schwarz and Young’s inequalities and applying Lemma 4.5 afterwards, as we also will do in the following to split the operator norms.

Applying Lemma 4.2 and the property $(1 - \eta_c)/\alpha = \lambda_c^{-1}$, we continue with

$$\begin{aligned}
 &\|L_{(c-1)}\|^2 + 2\alpha\|L_{cL}\|^2 + 3(1-\alpha)\|L_{(c+1)LL}\|^2 \\
 &\quad \leq 6\left(\|S_{c-1}M^{-1}D^T M\|^2 + \|S_{c-1}M^{-1}\widehat{B}_R\|^2 + \|\eta_c S_{c-1}M^{-1}\mathbb{I}^T D^T M\mathbb{I}\|^2\right) \\
 &\quad \quad + 4\alpha\left(\|S_c(1-\eta_c)M^{-1}\widehat{B}_L\|^2 + \|S_c\eta_c M^{-1}MD\mathbb{I}\|^2\right) + 6(1-\alpha)\|\eta_c S_{c+1}M^{-1}\widehat{B}_{J_0^2}\|^2 \\
 &\quad \leq S_n^2\left(6\left(\|D\|^2 + \|M^{-1}\widehat{B}_R\|^2 + \eta_c^2\alpha^2\mathcal{C}(p)^2\|\mathbb{I}\|^2\right)\right. \\
 &\quad \quad \left. + 4\alpha\left(\frac{1}{\lambda_c^2}\|M^{-1}\widehat{B}_L\|^2 + \eta_c^2\mathcal{C}(p)^2\right) + 6\frac{\eta_c^2}{1-\alpha}\|M^{-1}\widehat{B}_{J_0^2}\|^2\right).
 \end{aligned}$$

Next, we examine the terms originating from the cut cell itself:

$$\begin{aligned}
 &\frac{3}{\alpha}\|L_{(c-1)R}\|^2 + 2\|L_c\|^2 + 3\frac{1-\alpha}{\alpha}\|L_{(c+1)L}\|^2 \\
 &\quad \leq \frac{3}{\alpha}\|S_{c-1}\eta_c M^{-1}\mathbb{I}^T D^T M\|^2 \\
 &\quad \quad + 4\left(\|S_c(1-\eta_c)M^{-1}\widehat{B}_R\|^2 + \|S_c(1-\eta_c)M^{-1}D^T M\|^2\right) + 3\frac{1-\alpha}{\alpha}\|S_{c+1}(1-\eta_c)M^{-1}\widehat{B}_L\|^2 \\
 &\quad \leq S_n^2\left(3\eta_c^2\alpha\mathcal{C}(p)^2 + \frac{4}{\lambda_c^2}\left(\|M^{-1}\widehat{B}_R\|^2 + \|D\|^2\right) + 3\frac{\alpha}{\lambda_c^2(1-\alpha)}\|M^{-1}\widehat{B}_L\|^2\right).
 \end{aligned}$$

The final term we need to look at can be estimated similar to the terms of the background method, resulting in

$$\|L_{(c+1)}\|^2 + \frac{2}{1-\alpha}\|L_{(c+2)L}\|^2 = S_n^2 \left(\frac{6}{(1-\alpha)^2} (\|D\|^2 + \|M^{-1}\widehat{B}_R\|^2) + \frac{4}{1-\alpha} \|M^{-1}\widehat{B}_L\|^2 \right).$$

We can now combine these results to receive an estimate for the full operator norm:

$$\|Lu\|_M^2 \leq \frac{4a^2\|u\|_M^2}{\Delta x^2} \max \begin{cases} 3 \left(2\|M^{-1}\widehat{B}_R\|^2 + 2\|D\|^2 + \|M^{-1}\widehat{B}_L\| \right), \\ 6 \left(\|D\|^2 + \|M^{-1}\widehat{B}_R\|^2 + \eta_c^2 \alpha^2 \mathcal{C}(p)^2 \|\mathbb{I}\|^2 \right) \\ + 4\alpha \left(\frac{1}{\lambda_c^2} \|M^{-1}\widehat{B}_L\|^2 + \eta_c^2 \mathcal{C}(p)^2 \right) + 6 \frac{\eta_c^2}{1-\alpha} \|M^{-1}\widehat{B}_{J_0^2}\|, \\ 3\eta_c^2 \alpha \mathcal{C}(p)^2 + \frac{4}{\lambda_c^2} \left(\|M^{-1}\widehat{B}_R\|^2 + \|D\|^2 \right) + 3 \frac{\alpha}{\lambda_c^2(1-\alpha)} \|M^{-1}\widehat{B}_L\|^2, \\ \frac{6}{(1-\alpha)^2} (\|D\|^2 + \|M^{-1}\widehat{B}_R\|^2) + \frac{4}{1-\alpha} \|M^{-1}\widehat{B}_L\|^2. \end{cases} \quad (5.1)$$

None of the remaining terms depend on α^{-1} : The remaining α -dependencies can be estimated by $\alpha \leq 0.5$. Besides the operator norms that do not originate from the background method, the remaining ones were estimated in Lemmas 4.1, 4.2, and 4.3, where we also use $\alpha \leq 0.5$ to find a C just depending on p and the chosen nodes, such that

$$\|Lu\|_M \leq \frac{Ca}{\Delta x} \|u\|_M. \quad \blacksquare$$

Remark 5.1 (Multiple cut cells in need of stabilization). Note that this model case effectively addresses a single small cut cell. Extending the analysis to multiple cut cells leads to certain consequences, depending on their distance: Multiple small cut cells with

- a minimum of zero cells in between: As elaborated in Remark 2.2, this formulation of DoD is not applicable to these type of problems yet, so we exclude this case here.
- a minimum of one cell in between: The outflow cell of the first small cell is also the inflow cell of the second small cell. In this case, this cell experiences also both of the effects coming from the DoD stabilization, being an inflow and outflow cell. In terms of the operator norm, the submatrices affecting $u_{(c_1+1)} = u_{(c_2-1)}$ are scaled by $(1 - \alpha_1)^{-2}$ but also consist of the additional terms for stabilizing u_{c_2} , which leads to an even bigger operator norm. Nevertheless, these effects are locally additive contributions. As the full operator norm is determined by the largest local contribution, the final operator norm will remain in the same order of magnitude. This can be verified though simulations by the need of a smaller time step size.
- two or more cells in between: In this case, the effects described above vanish, such that the stabilizations are independent. Therefore, this does not influence our estimate. In reality, one of these two stabilizations determines the sharp operator norm. Since the operator norm in dependence of the cut-cell factor alpha is not monotone, further determinations of the cell dominating the operator norm can not be made in the general case.

As implied by Theorem 2.3, the DoD stabilization addresses the small cell problem for arbitrary small cut cells successfully. However, we can extract some more crucial information out of the operator norm estimate. As stated in Lemmas 4.1, 4.2, and 4.3 and numerically observed (and moderated in its impact) in Remark 4.4, the respective operator norms have an exponential scaling in p and tend to increase for larger cut cells, roughly about $(1 + \alpha)^p$. We observe in simulations, that this correlates to our CFL restrictions: Small cut cells can be stabilized perfectly in the sense that we obtain a similar CFL condition as for Figure 5.1, where we display this behavior for Gauss-Lobatto nodes. $\eta_c = 1$ displays the case where $1 - \eta_c$ aligns with the cut-cell size. Because the other choice of

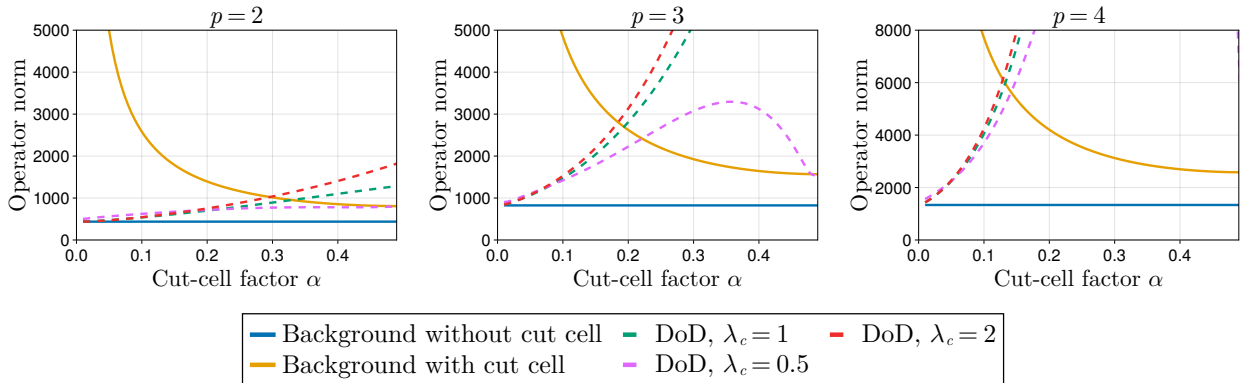


FIGURE 5.1. Operator norms for GLL nodes and different orders with $\Delta x = 1/50$.

$\eta_c = 1 - \min\{1, \alpha\Delta x/(a\Delta t)\}$ varies by Δt and the transport speed a , we show some variations of λ_c . We observe that smaller Courant numbers tend to increase the operator norm, while larger ones seem to decrease it. We will next investigate the latter phenomenon.

Remark 5.2. Other works such as [47, eqs. (3.2a/b)] that examine fully discrete stability use so-called inverse inequalities that estimate the volume and boundary terms and relate them to the mesh size. By using the matrix-type framework as in this article, this is equivalent to use the homogeneity of the operator norms to extract the scaling on the cell, e.g., $\|S_i Du\| = S_i \|Du\| \leq S_i \|D\| \|u\|$ where $\|D\|$ does not depend on the spatial step size and the inverse dependency on Δx_i originates from S_i .

6. A pragmatic correction approach

The undesired side effect of an increasing operator norm for larger cut cells, when considering higher-order spatial discretizations in combination with the DoD stabilization, limits its application to cut-cell meshes. However, as shown in Figure 5.1 for $p = 3$, specific choices of λ_c can mitigate the strong increase of the operator norms.

Considering (5.1), there are particularly two mechanisms involved. On one hand, the terms that depend on $\mathcal{C}(p)$ and \mathbb{I} are responsible for the exponential increase in p of $\|Lu\|_M$. Each of them has a coefficient of at least η_c . On the other hand, we have the term $4\alpha/\lambda_c^2 \|M^{-1}\hat{B}_L\|^2$, which originates from $\|L_{cL}\|$ and describes the reduced inflow into the cut cell. Smaller values for λ_c lead to η_c decreasing more rapidly according to (2.8) for increasing α , reaching $\eta_c = 0$ for $\alpha = \lambda_c$. Thus, we can control the terms involving $\mathcal{C}(p)$ and \mathbb{I} through the coefficient η_c . However, reducing λ_c obviously means decreasing the amount of stabilization. Therefore, we pay a price elsewhere. Generally, this corresponds in (5.1) to all terms with λ_c in the denominator. A numerical analysis of the operator norms of the involved block matrices L_c, L_{cL}, \dots suggests that the term $4\alpha/\lambda_c^2 \|M^{-1}\hat{B}_L\|^2$ dominates at some point. Therefore, we need to find an appropriate λ_c , that stabilizes not too much and not too little, ensuring a balance between extrapolation and minor cell problems. This motivates us to find optimized values for λ_c .

6.1. Determination of the optimized λ_c

First, we aim to provide explicit values for the choices of λ_c that minimize the operator norm. We calculate these by evaluating the operator norm of the right-hand side $L(\lambda_c, \alpha)$ for $\alpha \in [0, 0.5]$, $\lambda_c \in$

TABLE 6.1. Optimized values for λ_c to obtain a minimal operator norm by (6.1).

Poly.Deg.		Gauss–Lobatto–Legendre		Gauss–Legendre	
0	6	1.00000	0.12877	1.00000	0.11699
1	7	0.87665	0.10218	0.78913	0.09643
2	8	0.53986	0.0862	0.44159	0.07978
3	9	0.32132	0.07213	0.27871	0.06808
4	10	0.22302	0.06070	0.19529	0.05909
5	11	0.16104	0.05375	0.14927	0.05063

$[0, 1]$, and extracting λ_c such that

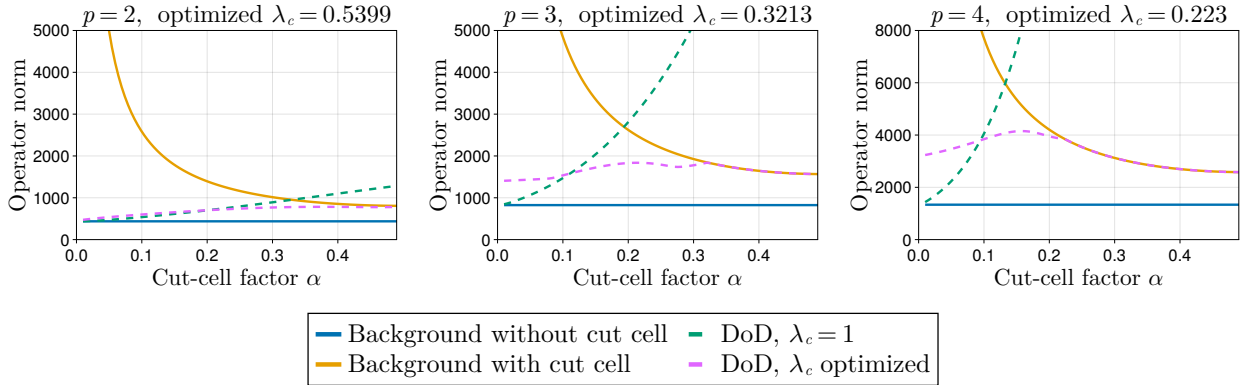
$$\min_{\lambda_c \in \mathcal{S}_{\lambda_c}} \max_{\alpha \in \mathcal{S}_\alpha} \{ \|L(\lambda_c, \alpha)\|_M \}, \quad \mathcal{S}_{\lambda_c} = [0.001, 0.499], \quad \mathcal{S}_\alpha = [0.01, 1], \quad (6.1)$$

for a fixed order p , choice of nodes, and a grid of 51 cells. We include the cut between cells 25 and 26 (the position did not significantly impact any experiments) and took 51 equidistant discrete values, including the boundaries of $\mathcal{S}_{\lambda_c}, \mathcal{S}_\alpha$, to evaluate expression (6.1). We present the resulting values for λ_c in Table 6.1 and denote these as optimized λ_c in the following. The origin of this effect, along with experiments, suggests that these values are independent of the number of cells and the present cut cells, assuming we do not need to stabilize multiple neighboring cut cells.

Applying these values to the DoD scheme, we will denote this as the optimized scheme. Note that this is solely an optimization to balance the operator norm for this specific model case.

6.2. Exemplary discussion of the optimized λ_c

In Figure 6.1, we display the evolution of some operator norms for an optimized value of λ_c , depending on the chosen basis.


 FIGURE 6.1. Operator norms for GLB nodes and different orders with $\Delta x = 1/50$.

The concave parts clearly correspond to the damping of the extrapolation terms, while we observe otherwise an increasing trend in α . This also suggests that $4\alpha/\lambda_c^2 \|M^{-1}\widehat{B}_L\|^2$ is responsible for these parts, as the other terms in (5.1) that depend on $1/\lambda_c^2$ are constant in α (and retain their dependency on it according to previous estimates). Note that this optimized choice solely aims to reduce the operator norm and neglects the intention behind the exact physical domain of dependence. Nevertheless, we still contend with the increased stencil for the outflow cell, albeit with a weighting that does not depend

on the physical domain of dependence. Additionally, in the optimized cases, the DoD operator norm docks onto the unstabilized background norm, which corresponds to turning off the stabilization, i.e., $\eta_c = 0$, as intended. This indicates that we have effectively addressed the extrapolation aspect, which was never meant to be governed by the minimum function in the definition of η_c .

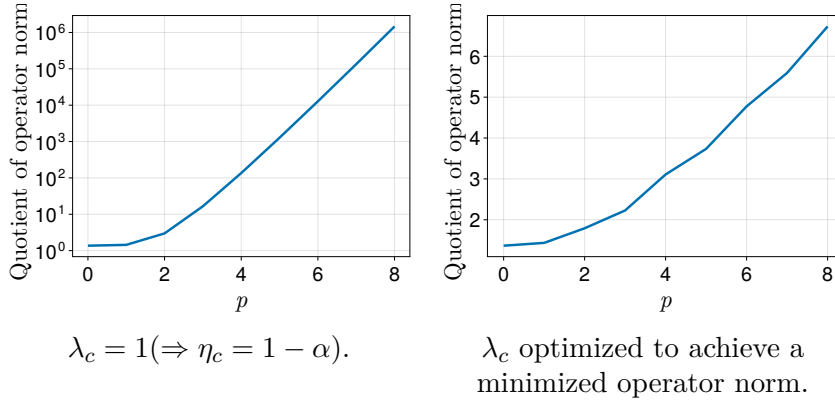


FIGURE 6.2. Quotients of $\|L_{\text{DoD}}\|/\|L_{\text{background}}\|$ for different choices of λ_c , using GLL nodes. Note the different scalings of the y -axes (logarithmic on the left, linear on the right).

In Figure 6.2, we show the quotient of $\min_{\lambda_c} \|Lu\|$ and the background operator norm for GLL nodes. We observe that the discrepancy between these norms increases with higher orders, leading to more restrictive time steps as p increases. However, we can still limit it to some extent compared to previous choices of η_c .

To provide a deeper practical understanding of the dominating terms that determine the choice of η_c , we display in Figure 6.3 the operator norms of the involved block matrices for the discussed choices of λ_c in the specific cases of GL nodes for $p = 4, 5$. Note that we additionally show the operator norm of the full right-hand side matrix L and $(\star) = S_{c-1}M^{-1}\eta_c\mathbb{I}^T D^T M\mathbb{I}$. We observe that for $\lambda_c = 1$, the terms that contain the extrapolation operator have a significantly higher norm for larger cut cells than the remaining ones. We want to emphasize that the norm of (\star) , which applies the extrapolation to the test function and the solution, dominates $L_{(c-1)}$ and all the other terms. This aligns with our theoretical analysis. By using the optimized values for λ_c , we observe that these terms are not responsible for the major contribution to the norm of L , but still represent the maximum value. Further, we can assume that the terms L_c and $L_{(c-1)}$ contribute the most to $\|L\|_M$, as their operator norms are of the largest magnitude. This also aligns with our theory, as this optimization corresponds to reduced stabilization of the small cut cell.

7. Numerical tests

In this section, we present numerical tests of the linear transport equation to validate our theoretical findings. We begin with the one-dimensional linear advection equation as in the previous analysis and then proceed with some applications in 2D.

The 1D numerical methods are implemented in Julia [7] based on the infrastructure of `SummationByPartsOperators.jl` [35].

The 2D tests are implemented using the C++ PDE framework DUNE [2, 3, 4, 11], including the dune-functions module for functions space bases [17, 18]. We further rely on the TPMC [20] library to calculate the cut cell geometries. The presented 2D data was generated with the help of GNU Parallel [45].

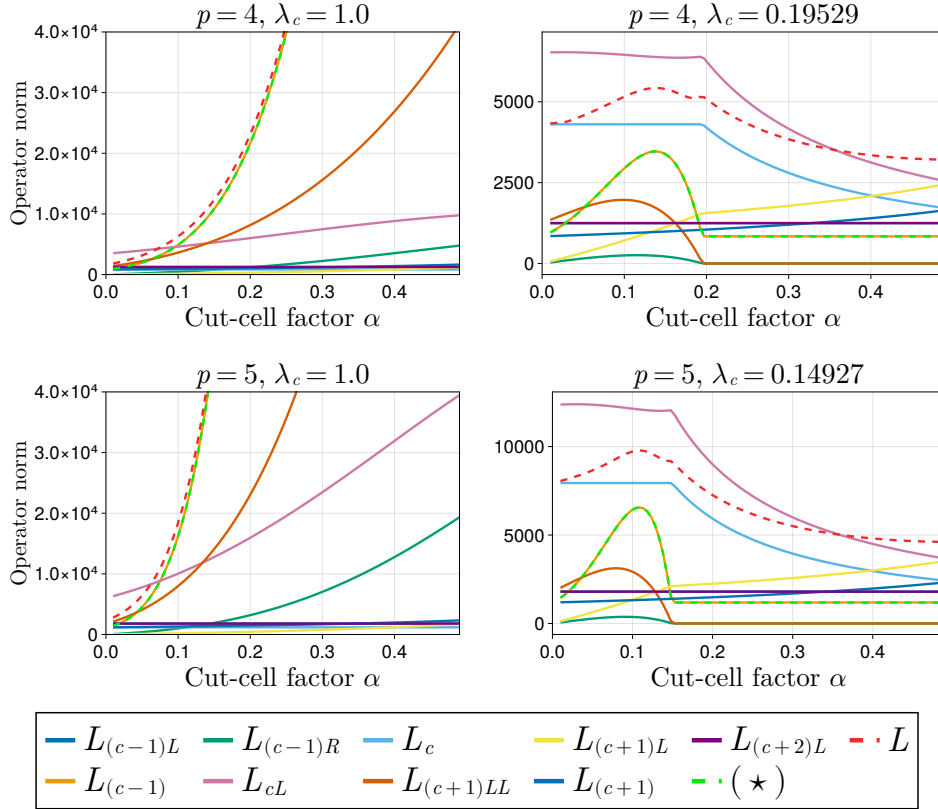


FIGURE 6.3. Operator norms of the involved block matrices in (2.17), including L and the partial term $(\star) = S_{c-1}M^{-1}\eta_c\mathbb{I}^T D^T M\mathbb{I}$ of $L_{(c-1)}$. Using Gauss–Legendre nodes and $\Delta x = 1/50$.

We used Makie.jl [14] to generate the plots. All code and data required to reproduce the numerical results presented in this article are available in our reproducibility repository [33].

7.1. One-dimensional linear advection

To investigate this optimization in practice, we apply the fully discrete scheme to the linear advection equation and determine the sharp CFL condition by adjusting the time step under which the method remains stable in long-term simulation. We will compare this to the classic case with $\lambda_c = 1$ and the background method, i.e., the DGSEM without any cut cells in the mesh. We use the same settings for the semidiscretization as above and again vary α in $[0.001, 0.49]$. We choose $a = 1$ and $\sin(2\pi x)$ as the initial condition over the domain $[0, 1]$. For time-stepping, which affects the explicit results, we apply appropriate explicit RK methods that correspond to the order of the semidiscretization, specifically SSPRK(2,2), SSPRK(3,3) [25], and SSPRK(10,4) [27]. For the first order, we tested the explicit Euler method and obtained $\lambda_c = 1$; there are clearly no differences compared to DoD with $\eta_c = \eta_c(\Delta t)$. In each case, we obtain the sharp CFL restriction of 1 (independent of α), which is also recognized from the background method. Consequently, we do not present a plot for this case. The remaining results, up to order 4, are illustrated in Figure 7.1.

We observe a similar behavior for both types of nodes: For the second order methods, we still see barely a difference between the two choices. Considering the higher orders, the time step dependent DoD required a slightly loose CFL condition for small cut cells, as it strongly shifts for the larger cut

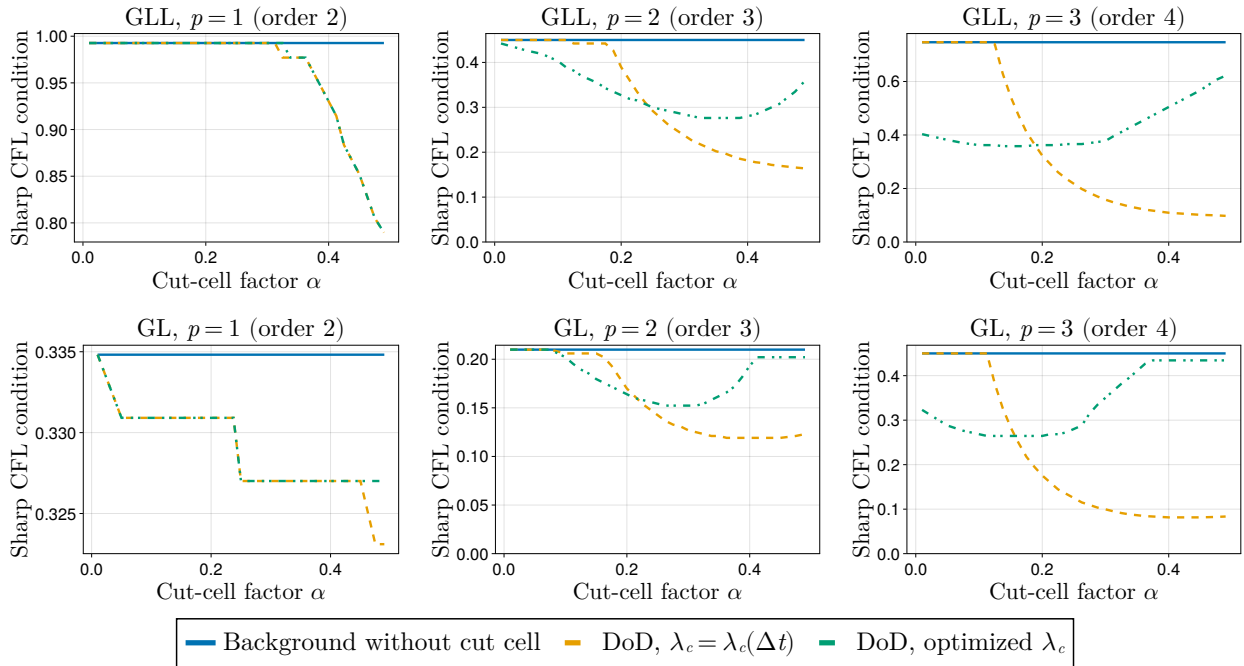


FIGURE 7.1. Sharp CFL restriction for long time stability, depending on the cut cell size.

cells. We also observe, that this gets more crucial, the higher the order of the scheme. This is exactly what we would expect, as small values for λ_c reduce the amount of stabilization, that is necessary for small cut cells and tends to be problematic for larger ones. As in “practice”, there are cut cells of arbitrary sizes. Therefore, the question of which parameters to choose reduces also to the highest possible CFL, that does not depend on a specific cut cell size, i.e. we aim to choose the lowest possible CFL value for a specific choice of λ_c . In that case, our optimal choice seems to be more suitable and will be even more for higher order methods for this model problem. Please note that the overall increased CFL values for the fourth-order case compared to the third-order case are due to the usage of the SSPRK(10,4), which uses more stages than necessary to achieve fourth-order accuracy to improve stability properties.

Next, we want to investigate these methods for numerical convergence with the sharp CFL conditions. Specifically, we select the largest sharp CFL, such that all considered cases (the background method and the two DoD variants) for fixed nodes and polynomial degree are stable and refine the mesh by applying multiple cut cells, ranging from $\alpha = 10^{-3}$ to $\alpha = 0.49$. We also apply a safety factor of 0.95 to the respective sharp CFL value to ensure that minor calculation errors do not impact stability. As shown in Figure 7.2, convergence is achieved as expected. In Figure 7.3, we illustrate the error in relation to the number of time steps leading up to the final time of $T = 1$, which correlates with the efficiency of the different schemes. For this case, we select the corresponding maximal sharp CFL number for which the method is stable and apply a safety factor of 0.99. Here, we observe that the optimization results in at least the same and often smaller errors than the “classic” method, indicating that this operator norm-based choice not only contributes to a more favorable CFL condition but also impacts the actual accuracy. Once again, please note that these results pertain to the simple model problem for now.

DOMAIN-OF-DEPENDENCE STABILIZATION FOR CUT-CELL MESHES

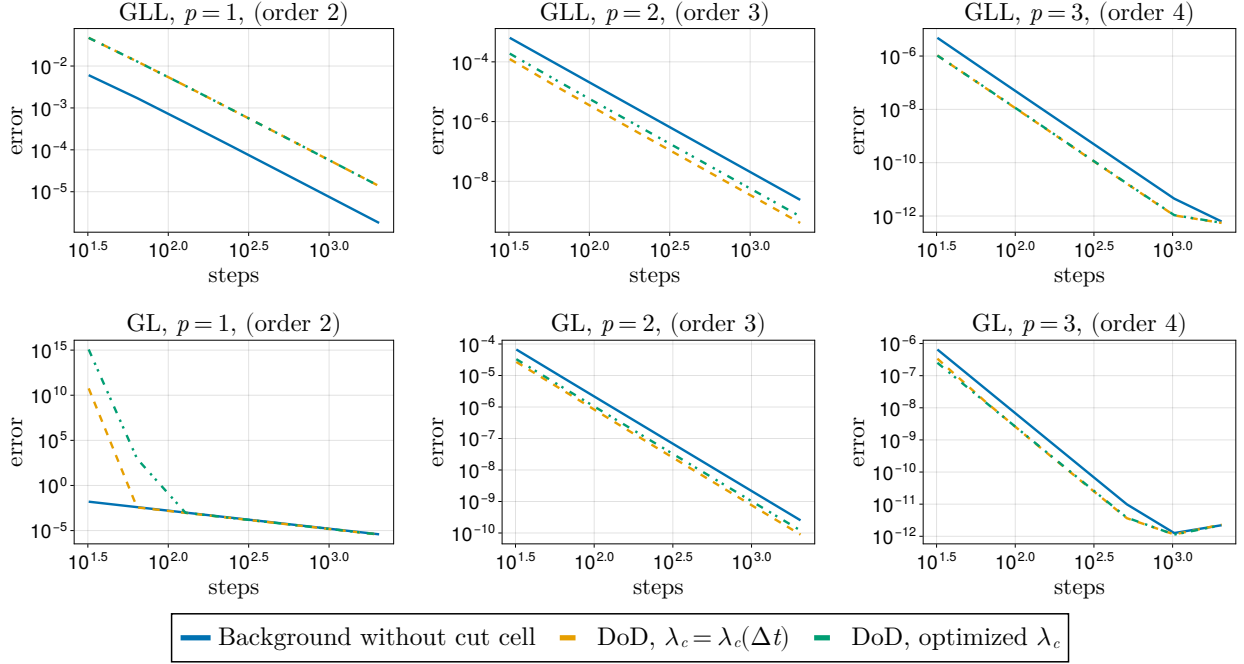


FIGURE 7.2. Convergence errors at the time $T = 1$ with respect to the norm $\| \cdot \|_M$.

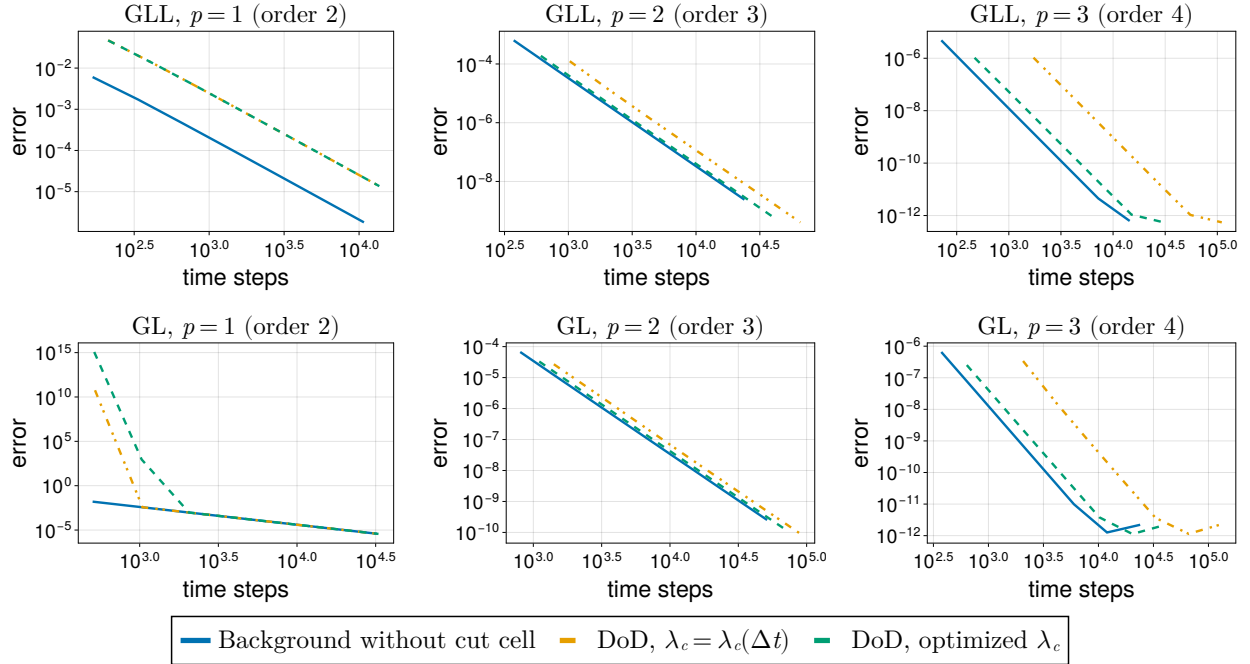
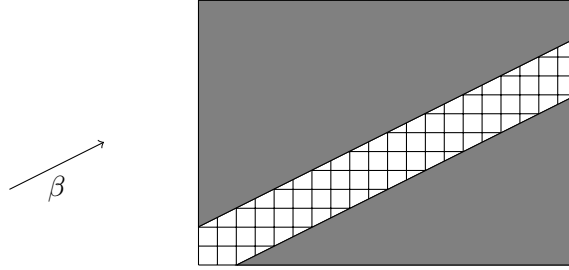


FIGURE 7.3. Work precision diagram with errors at the time $T = 1$ with respect to the norm $\| \cdot \|_M$.

FIGURE 7.4. Channel geometry with cut-cell mesh and advection velocity β .

7.2. Two-dimensional linear advection in a channel

We consider the linear advection equation

$$\begin{aligned} \partial_t u(t, x) + \beta \cdot \nabla_x u(t, x) &= 0 && \text{in } \Omega \subset \mathbb{R}^2, \\ u(0, x) &= u_0(x) && \text{in } \Omega, \\ u(t, x) &= 0 && \text{on } \partial\Omega, \end{aligned}$$

with homogeneous Dirichlet data in a channel domain with parallel walls, see Figure 7.4 for a graphical representation. The lower, respectively the upper wall, will start at a position $(x_0, 0)^t$, respectively $(0, x_0)^t$ with an offset $x_0 \in \mathbb{R}$. The geometry is completely defined by the choice of x_0 and the angle γ between the walls and the x -axis. The flow direction β is a constant unit vector parallel to the channel walls.

First, we consider the choice $\gamma = 45^\circ$. This creates a geometric setup where all small cut cells in need of stabilization share the same size and shape, similar to the one-dimensional case. We can define the size of these small cells, i.e., the cut-cell factor α , by adjusting the offset x_0 , which allows us to run numerical tests with a single size factor.

We partially replicate the 1D analysis of the operator norm given in (6.1) in 2D by estimating an optimal value for the parameter λ_c from (2.8) for different polynomial degrees. The results are presented in Figure 7.6, where we plotted different choices of λ_c over a variety of cut-cell factors. In particular, we plotted a choice of λ_c which is close to the optimal factors presented in the previous section for the one-dimensional case. As seen in the plots these values are not optimal anymore for the two-dimensional channel geometry. However, we were able to obtain new optimal values for λ_c , which we included in the plots. We note that the function graphs of these new values possess a shape with two peaks similar to the one-dimensional case, highlighting the optimization of different contributions of the scheme.

As a first step towards a truly multidimensional setting, we consider additional angles of γ for the channel geometry. Once we move away from the case $\gamma = 45^\circ$ the cut-cell structure of the geometry becomes more rich and direct control over cell volume fractions is lost. Therefore, the previous optimization procedure is not directly applicable anymore. Based on geometrical considerations (see Figure 7.5 for an explanation) we instead suggest the general choice

$$\lambda_c^\gamma = \lambda_c^{45^\circ} \tan(\gamma) \tag{7.1}$$

for $\gamma \in (0^\circ, 45^\circ]$ and $\lambda_c^{45^\circ}$ the optimal value already obtained and displayed in Figure 7.6. Note that the limit case of $\gamma = 0^\circ$ is just a Cartesian grid and requires no stabilization.

We can still compare this choice with an optimal value for λ_c obtained over the same offsets x_0 that we used for the $\gamma = 45^\circ$ channel (even though we want to stress here again, that now these offsets in general do not correspond to cell volume fractions). The results of this comparison are compiled in Table 7.1. In general, we can see good agreement between the suggested choice and the optimal value.

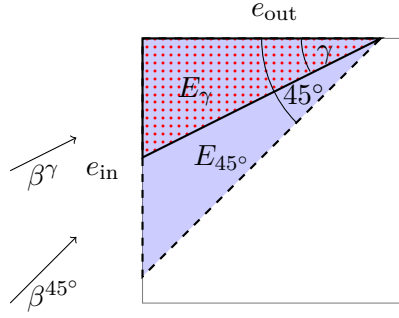


FIGURE 7.5. Comparison of two triangular cut cells, E_{45° in blue and E_γ with red dots, with different cut angles and flow directions (that are always parallel to the cut). The distance in the normal direction of the inflow face e_{in} stays the same for both cut cells, no matter the cut angle, so we would expect that in both cases the same amount of stabilization is necessary, at least if the cut interface is a no-flow face (which we assume to be the case here). Note that we have $|E_\gamma| = \tan(\gamma)|E_{45^\circ}|$ and thus $\alpha_\gamma = \tan(\gamma)\alpha_{45^\circ}$ where α_γ and α_{45° are the cut-cell factors of E_γ and E_{45° , respectively.

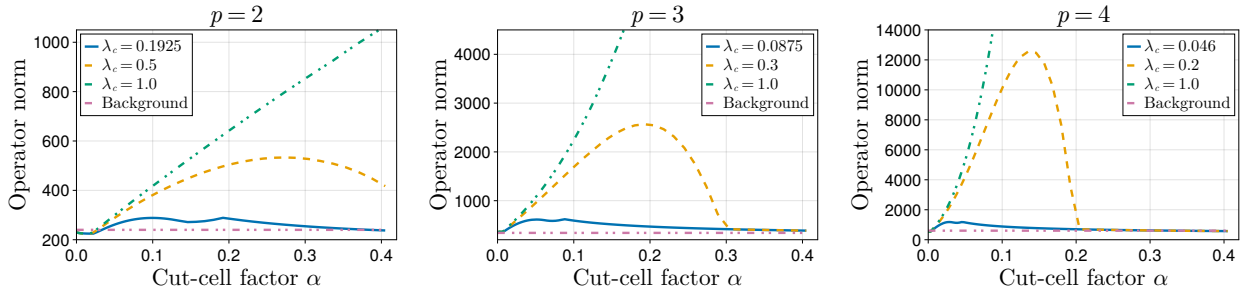


FIGURE 7.6. Operator norms for polynomial degrees $p = 2, 3, 4$ and different choices of λ_c on the channel geometry.

TABLE 7.1. Comparison of the choice of λ_c for different angles γ .

	$p = 2$				$p = 3$				$p = 4$			
	$\lambda_c \tan(\gamma)$	$\ L\ $	opt.	$\ L\ $	$\lambda_c \tan(\gamma)$	$\ L\ $	opt.	$\ L\ $	$\lambda_c \tan(\gamma)$	$\ L\ $	opt.	$\ L\ $
35°	0.135	316	0.161	299	0.061	683	0.067	667	0.032	1355	0.033	1338
25°	0.090	342	0.107	322	0.041	761	0.045	734	0.021	1496	0.023	1460
15°	0.052	360	0.062	339	0.024	798	0.027	787	0.012	1586	0.012	1586
5°	0.017	368	0.020	347	0.008	815	0.009	803	0.004	1588	0.004	1588

Equipped with a good λ_c for different polynomial degrees and channel angles we can give numerical estimates of CFL numbers. These are presented in Table 7.2 and correspond to typical CFL numbers based on the background mesh for DG schemes in combination with our chosen time integration methods.

8. Summary and conclusions

In this work, we have given a proof that the domain-of-dependence (DoD) stabilization is fully-discrete stable for a specific class of explicit and arbitrarily high-order Runge–Kutta methods. To perform this

TABLE 7.2. CFL numbers achieved for different channel geometries and polynomial degrees p with the optimal parameter λ_c given by (7.1).

	$p = 1$	$p = 2$	$p = 3$
45°	0.35	0.24	0.47
35°	0.38	0.24	0.52
25°	0.37	0.23	0.47
15°	0.37	0.21	0.45
5°	0.37	0.21	0.43

analysis, we rewrote the semidiscretization as a system of ODEs within the DGSEM framework. In this setting, we also managed to formulate this result for the case of under-integration with Gauss–Lobatto–Legendre nodes and quadrature. In particular, the stability question reduced to an operator norm estimate, and we demonstrated that this operator norm, and therefore the resulting CFL-like condition, scale without a problematic dependence on the arbitrarily small cut cells, as intended.

We discovered further CFL restrictions that appear for larger cut cells treated with the stabilization, especially in higher-order cases. Our analysis shows that these restrictions originate from the extension operator, which projects polynomials outside their assigned cell. We also quantified this issue with an operator norm estimate. Because this unintended behavior bypassed the implemented stabilization cutoff, we suggested a practical fix by selecting the penalty parameter to minimize the operator norm across all potential cut-cell sizes. This effectively results in an optimal CFL-like condition and restores the useful cutoff. Numerical examples are provided to illustrate and support our theoretical findings.

As an outlook, we also provide a numerical extension to the case of advection in two spatial dimensions, where we can observe the same phenomenon in a similar form. Because of the arbitrary shape of the cut cells, this issue becomes more intricate. Nevertheless, these first results indicate, that by a suitable choice of parameters, the time-step restriction remains in the same order of magnitude as in a respective background scheme. A solution that combines the approach of the 1D case with geometric ideas leads to a promising initial result.

Appendix A. Postponed proofs

In this appendix, we provide the more technical proofs to prepare Theorem 2.3.

A.1. Proof of Lemma 3.1

Proof. Inequality [13, eq. (2.3.16)] yields the weight estimates

$$c_1 p^{-1} \sqrt{1 - \hat{x}_j^2} \leq \hat{\omega}_j \leq c_2 p^{-1} \sqrt{1 - \hat{x}_j^2}, \tag{A.1}$$

for every j , where c_1 and c_2 do not depend on j and p . The inequality (A.1) also can be formulated for Gauss–Lobatto–Legendre nodes, i.e., we just have to substitute $\hat{x}_j \rightarrow \bar{x}_j$ and $\hat{\omega}_j \rightarrow \bar{\omega}_j$ everywhere. Therefore, we get by using (3.1)

$$\frac{\hat{\omega}_i}{\hat{\omega}_j} \leq \frac{c_2 p^{-1} \sqrt{1 - \hat{x}_i^2}}{c_1 p^{-1} \sqrt{1 - \hat{x}_j^2}} = \frac{c_2 \sin(\hat{\theta}_i)}{c_1 \sin(\hat{\theta}_j)}.$$

Considering the Gauss–Legendre nodes, we can use $2\theta/\pi \leq \sin(\theta) \leq 1$ for $\theta \in [0, \frac{\pi}{2}]$ combined with (3.3) for $j = 1$ to obtain

$$\frac{\hat{\omega}_i}{\hat{\omega}_j} \leq \frac{c_2}{c_1 \frac{1}{p+1}} = \frac{c_2}{c_1} (p+1).$$

As the case $p = 0$ is trivial, we assume that $p > 0$ and can conclude the assumption with $\widehat{K}_1 := 2c_2/c_1$ for the Gauss–Legendre case.

For the Gauss–Lobatto–Legendre weights, we can perform similar estimates with (3.2) and (3.4), but notice that we have to treat the lower bound of weight $\bar{\omega}_0$ differently. In that specific case, we can easily exploit a Legendre polynomial-related formula for the weights, that yields (see e.g. [13, eq. (2.3.11)], [15, eq. (2.7.1.12)])

$$\bar{\omega}_j = \frac{2}{p(p+1)} \frac{1}{L_p(\bar{x}_j)^2},$$

where L_p is the Legendre polynomial of degree p . The normalization property of L_p yields $L_p(1) = 1$ and by the symmetry/antisymmetry we can deduce that $L_p(\bar{x}_0)^2 = 1$. Therefore, we receive an explicit expression of $\bar{\omega}_0$ and can estimate

$$\frac{\bar{\omega}_i}{\bar{\omega}_j} \leq \frac{c_{GLL2} p^{-1} \sin(\bar{\theta}_i)}{\bar{\omega}_j} \leq \frac{c_{GLL2}}{2} \leq \bar{K}_1 p$$

with a similar, appropriate choice of \bar{K}_1 as in the previous case. \blacksquare

A.2. Proof of Lemma 3.2

Proof. Using a trigonometric identity [12, eq. (2.5.2.1.3)], we can derive for $1 \leq j \leq \lfloor (p+1)/2 \rfloor$, that

$$\begin{aligned} |\hat{x}_j - \hat{x}_{j-1}| &= \left| \cos(\hat{\theta}_j) - \cos(\hat{\theta}_{j+1}) \right| = \left| 2 \sin\left(\frac{\hat{\theta}_{j+1} - \hat{\theta}_j}{2}\right) \sin\left(\frac{\hat{\theta}_{j+1} + \hat{\theta}_j}{2}\right) \right| \\ &\geq \left| \frac{2}{\pi^2} (\hat{\theta}_{j+1}^2 - \hat{\theta}_j^2) \right|, \end{aligned} \quad (\text{A.2})$$

where the last estimate follows from $\sin(x) \geq 2x/\pi \geq 0$ for $x \in [0, \pi/2]$. Using (3.3), we obtain

$$\begin{aligned} |\hat{x}_j - \hat{x}_{j-1}| &\geq \left| \frac{2}{\pi^2} \left(\frac{\pi^2(j + \frac{1}{2})^2}{(p+1)^2} - \frac{\pi^2 j^2}{(p+2)^2} \right) \right| \\ &= \left| 2 \left(\frac{(j + \frac{1}{2})^2 (p+2)^2 - j^2 (p+1)^2}{(p+1)^2 (p+2)^2} \right) \right| \\ &\geq 2 \frac{(j + \frac{1}{2})^2 - j^2}{(p+2)^2} \geq \frac{5}{2} \frac{1}{(p+2)^2}. \end{aligned} \quad (\text{A.3})$$

As this estimate holds for all $\hat{x}_j \in (-1, 0)$, we now want to include the nodes in $[0, 1)$. Because of symmetry, it suffices to consider the distance between $\hat{x}_{\lfloor (p+1)/2 \rfloor}$ and $\hat{x}_{\lfloor (p+1)/2 \rfloor + 1} \geq 0$. With the previous derivations in (A.2), we can deduce in the same way

$$\begin{aligned} \left| \hat{x}_{\lfloor \frac{p+1}{2} \rfloor + 1} - \hat{x}_{\lfloor \frac{p+1}{2} \rfloor} \right| &\geq \hat{x}_{\lfloor \frac{p+1}{2} \rfloor} - 0 \geq \frac{2}{\pi^2} \left(\left(\frac{\pi}{2} \right)^2 - \hat{\theta}_{\lfloor \frac{p+1}{2} \rfloor}^2 \right) \geq \frac{1}{2} - \frac{2}{\pi^2} \left(\frac{p\pi}{p+2} \right)^2 \\ &= \frac{1}{2} \left(1 - \frac{p^2}{(p+2)^2} \right) = \frac{2p+2}{(p+2)^2} \\ &\geq \frac{p+2}{(p+2)^2} = \frac{1}{p+2}. \end{aligned}$$

Note that $1/(p+2) \geq 2.5/(p+2)^2$ for $p > 0$ and therefore the estimate in (A.3) dominates. Therefore, we can conclude the case of Gauss–Legendre nodes by choosing $\widehat{K}_2 := 5/18$, so that for $p \geq 1$,

$$\frac{\widehat{K}_2}{p^3} = \frac{5}{2 \cdot 9p^3} \leq \frac{5p^3}{2p^3(p+2)^2} \leq \hat{x}_i - \hat{x}_j.$$

Considering now the Gauss–Lobatto–Legendre nodes \bar{x}_j , we proceed as for the Gauss–Legendre nodes using (3.4), resulting in

$$\begin{aligned} |\bar{x}_j - \bar{x}_{j-1}| &\geq \frac{2}{\pi^2} (\bar{\theta}_{j+1}^2 - \bar{\theta}_j^2) \\ &\geq 2(j+1)^2 \left(\frac{1}{p^2} - \frac{1}{(p+1)^2} \right) \\ &\geq 8 \frac{2p+1}{p^2(p+1)^2} \geq \frac{16}{(p+1)^3}. \end{aligned} \tag{A.4}$$

For the difference to an $\bar{x}_i \geq 0$, we again have

$$\bar{x}_{\lfloor \frac{p-1}{2} \rfloor} - 0 \geq \frac{2}{\pi^2} \left(\left(\frac{\pi}{2} \right)^2 - \left(\frac{\pi}{p} \right)^2 \right) = \frac{p^2 - 4}{2p^2},$$

which does not exceed the estimate of two other consecutive nodes in (A.4) (for $p \geq 3$, as this estimate is just there required). For the Gauss–Lobatto–Legendre nodes, we also have to consider the distance to the nodes at the interval borders $\bar{x}_0 = -1$ and $\bar{x}_p = 1$. There, we have

$$\bar{x}_1 - \bar{x}_0 = 2 \sin(\bar{\theta}_1) \geq \frac{2}{\pi^2} \bar{\theta}_1^2 = \frac{2}{p^2},$$

which does dominate the estimate for $p \leq 4$, compared to (A.4). Therefore, we find a constant C_{GLL} , such that for all orders of p it holds

$$|\bar{x}_i - \bar{x}_j| \geq \frac{C_{GLL}}{(p+1)^3}.$$

With $\widehat{K}_2 := C_{GLL}/8$, we can estimate analogue as for the Gauss–Legendre nodes to conclude the statement. ■

A.3. Proof of Lemma 4.2

Proof. We use the barycentric form of the derivative matrix D , which is given by [28, eq. (3.48)]

$$\begin{aligned} D_{ij} &= \frac{\prod_{m=0, m \neq i}^N \tilde{x}_i - \tilde{x}_m}{\prod_{m=0, m \neq j}^N \tilde{x}_j - \tilde{x}_m} \left(\frac{1}{\tilde{x}_i - \tilde{x}_j} \right), \quad i \neq j, \\ D_{ii} &= - \sum_{\substack{m=0 \\ m \neq i}}^M D_{im}. \end{aligned}$$

By applying the negative sum property of the diagonal element, we can expand

$$\begin{aligned} (D\mathbb{I})_{ik} &= \sum_{j=0}^p D_{ij} \tilde{l}_k(\xi_j) = \sum_{\substack{j=0 \\ j \neq i}}^p D_{ij} (\tilde{l}_k(\xi_j) - \tilde{l}_k(\xi_i)) \\ &= \sum_{\substack{j=0 \\ j \neq i}}^p D_{ij} \left(\prod_{\substack{m=0 \\ m \neq k}}^p \frac{1 - \tilde{x}_m + \alpha(1 + \tilde{x}_j)}{\tilde{x}_k - \tilde{x}_m} - \prod_{\substack{m=0 \\ m \neq k}}^p \frac{1 - \tilde{x}_m + \alpha(1 + \tilde{x}_i)}{\tilde{x}_k - \tilde{x}_m} \right). \end{aligned}$$

In the last line, we see that the products just differ by the part that depends on α . We can therefore extract this term, such that

$$\prod_{\substack{m=0 \\ m \neq k}}^p \frac{1 - \tilde{x}_m + \alpha(1 + \tilde{x}_i)}{\tilde{x}_k - \tilde{x}_m} = \frac{\left(\prod_{\substack{m=0 \\ m \neq k}}^p 1 - \tilde{x}_m \right) + \alpha(1 + \tilde{x}_i) \left(\sum_{\substack{m_1=0 \\ m_1 \neq k}}^p \left(\prod_{\substack{m=0 \\ m \neq m_1, k}}^p 1 - \tilde{x}_m \right) \right)}{\prod_{\substack{m=0 \\ m \neq k}}^p (\tilde{x}_k - \tilde{x}_m)} \\ + \frac{\alpha^2(1 + \tilde{x}_i)^2 \left(\sum_{\substack{m_1, m_2=0 \\ m_1, m_2 \neq k}}^p \left(\prod_{\substack{m=0 \\ m \neq m_1, m_2, k}}^p 1 - \tilde{x}_m \right) \right) + \dots}{\prod_{\substack{m=0 \\ m \neq k}}^p (\tilde{x}_k - \tilde{x}_m)}.$$

Therefore, we can deduce that

$$(D\mathbb{I})_{ik} = \sum_{\substack{j=0 \\ j \neq i}}^p D_{ij} \left(\prod_{\substack{m=0 \\ m \neq k}}^p \frac{1 - \tilde{x}_m}{\tilde{x}_k - \tilde{x}_m} - \prod_{\substack{m=0 \\ m \neq k}}^p \frac{1 - \tilde{x}_m}{\tilde{x}_k - \tilde{x}_m} + \alpha \mathcal{P}(\alpha) \right) = \alpha \sum_{\substack{j=0 \\ j \neq i}}^p D_{ij} \mathcal{P}(\alpha),$$

where $\mathcal{P}(\alpha)$ is a polynomial in α . Knowing this notation about $(D\mathbb{I})_{ik}$, we can do similar estimates about $D_{jk}\mathcal{P}(\alpha)$ as in Lemma 4.1 (which we will not explore again in detail here), that will again lead to an exponential scaling in p for larger cut cells. As $\mathcal{P}(\alpha)$ is bounded for all $\alpha \in [0, 0.5]$, we can omit this dependency. Nevertheless, we remain with the exponential scaling in p , and using entry-based matrix norms (see, e.g., [24, eq. (2.3.8)]) we can estimate the whole expression by a function $\mathcal{C}(p)$. ■

A.4. Proof of Lemma 4.3

Proof. In terms of interpolation and extrapolation operators, we can write $\widehat{B}_{J_2^0} = \widehat{L}^T B_J$, where

$$B_J = \left(\tilde{l}_1(1 + 2\alpha), \tilde{l}_2(1 + 2\alpha), \dots, \tilde{l}_{p+1}(1 + 2\alpha) \right)$$

is the interpolation matrix from the respective nodes of the reference element $[-1, 1]$ to the reference outflow cell boundary $1 + 2\alpha$ (that is limited by $1 + 2\lambda_c$). Therefore, we receive with the extended outflow boundary value $\bar{u} := B_J u$

$$\begin{aligned} \|\widehat{B}_{J_2^0} u\|_{M_R} &= u^T B_J^T L M_R L^T B_J u \\ &= \bar{u} L M_R L^T \bar{u} = \bar{u}^2 \sum_{j=0}^{p+1} \omega_j \widehat{L}_j^2 = \frac{C_2}{4} \bar{u}^2, \end{aligned}$$

where ω_j denotes the respective GL or GLL weights. To estimate \bar{u} , which is exactly the extrapolation to the outflow boundary of the cut cell, we follow the same arguments as in the proof of Lemma 4.1 under consideration that an additional factor $1/\max_{i=1, \dots, p+1}(\omega_i)$ is required that we include in C . This proves the statement. ■

References

- [1] Franz Achleitner, Anton Arnold, and Ansgar Jüngel. Necessary and Sufficient Conditions for Strong Stability of Explicit Runge-Kutta Methods. In *From Particle Systems to Partial Differential Equations. PSPDE X, Braga, Portugal, June 2022*, volume 465 of *Proceedings in Mathematics & Statistics*, pages 1–21. Springer, 2024.

- [2] Peter Bastian, Markus Blatt, Andreas Dedner, Nils-Arne Dreier, Christian Engwer, René Fritze, Carsten Gräser, Christoph Grüninger, Domonic Kempf, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. The Dune framework: Basic concepts and recent developments. *Comput. Math. Appl.*, 81:75–112, 2021.
- [3] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Ralf Kornhuber, Mario Ohlberger, and Oliver Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part II: Implementation and Tests in DUNE. *Computing*, 82(2–3):121–138, 2008.
- [4] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework. *Computing*, 82(2–3):103–119, 2008.
- [5] Marsha Berger and Andrew Giuliani. A state redistribution algorithm for finite volume schemes on cut cell meshes. *J. Comput. Phys.*, 428: article no. 109820 (20 pages), 2021.
- [6] Marsha Berger and Andrew Giuliani. A New Provably Stable Weighted State Redistribution Algorithm. *SIAM J. Sci. Comput.*, 46(5):A2848–A2873, 2024.
- [7] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.*, 59(1):65–98, 2017.
- [8] Gunnar Birke, Christian Engwer, Jan Giesselmann, and Sandra May. Error Analysis of a First-Order DoD Cut Cell Method for 2D Unsteady Advection. *J. Sci. Comput.*, 106(1): article no. 1 (33 pages), 2025.
- [9] Gunnar Birke, Christian Engwer, Sandra May, Louis Petri, and Hendrik Ranocha. An Energy-Preserving Domain of Dependence Stabilization for the Linear Wave Equation on Cut-Cell Meshes. <https://arxiv.org/abs/2601.19877>, under review, 2026.
- [10] Gunnar Birke, Christian Engwer, Sandra May, and Florian Streitbürger. DoD Stabilization of linear hyperbolic PDEs on general cut-cell meshes. *Proc. Appl. Math. Mech.*, 23(1): article no. e202200198, 2023.
- [11] Markus Blatt, Samuel Burbulla, Ansgar Burchardt, Andreas Dedner, Christian Engwer, Carsten Gräser, Christoph Grüninger, Robert Klöfkorn, Timo Koch, Santiago Ospina De Los Rios, Simon Praetorius, and Oliver Sander. The Distributed and Unified Numerics Environment (DUNE), Version 2.10. <https://arxiv.org/abs/2506.23558>, 2025.
- [12] Ilya N. Bronstein. *Taschenbuch der Mathematik*. Harri Deutsch, 1987.
- [13] Claudio Canuto, Mohammed Hussaini, Alfio Quarteroni, and Thomas Zang. *Spectral Methods: Fundamentals in Single Domains*, volume 23 of *Scientific Computation*. Springer, 2006.
- [14] Simon Danisch and Julius Krumbiegel. Makie.jl: Flexible high-performance data visualization for Julia. *J. Open Source Softw.*, 6(65): article no. 3349 (5 pages), 2021.
- [15] Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Computer Science and Applied Mathematics. Academic Press Inc., 2014.
- [16] Daniele Di Pietro and Alexandre Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*, volume 69 of *Mathématiques & Applications*. Springer, 2012.
- [17] Christian Engwer, Carsten Gräser, Steffen Müthing, Simon Praetorius, and Oliver Sander. Concepts for Composing Finite Element Function Space Bases. <https://arxiv.org/abs/2508.10125>, under review, 2025.
- [18] Christian Engwer, Carsten Gräser, Steffen Müthing, and Oliver Sander. The interface for functions in the dune-functions module. In *Proceedings of the 3rd Dune User Meeting (2015)*, Archive of Numerical Software, pages 95–110. heiBOOKS, 2017.
- [19] Christian Engwer, Sandra May, Andreas Nüßing, and Florian Streitbürger. A stabilized DG cut cell method for discretizing the linear transport equation. *SIAM J. Sci. Comput.*, 42(6):A3677–A3703, 2020.
- [20] Christian Engwer and Andreas Nüßing. Geometric Reconstruction of Implicitly Defined Surfaces and Domains with Topological Guarantees. *ACM Trans. Math. Softw.*, 44(2): article no. 14 (20 pages), 2017.

- [21] Pei Fu and Gunilla Kreiss. High Order Cut Discontinuous Galerkin Methods for Hyperbolic Conservation Laws in One Space Dimension. *SIAM J. Sci. Comput.*, 43(4):A2404–A2424, 2021.
- [22] Gregor J. Gassner. A Skew-Symmetric Discontinuous Galerkin Spectral Element Discretization and Its Relation to SBP-SAT Finite Difference Methods. *SIAM J. Sci. Comput.*, 35(3):A1233–A1253, 2013.
- [23] Andrew Giuliani. A Two-Dimensional Stabilized Discontinuous Galerkin Method on Curvilinear Embedded Boundary Grids. *SIAM J. Sci. Comput.*, 44(1):A389–A415, 2022.
- [24] Gene H. Golub and Charles F. Van Loan. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, 2013.
- [25] Sigal Gottlieb and Chi-Wang Shu. Total Variation Diminishing Runge-Kutta Schemes. *Math. Comput.*, 67:73–85, 1996.
- [26] Ceren Gürkan, Simon Stickle, and André Massing. Stabilized Cut Discontinuous Galerkin Methods for Advection-Reaction Problems. *SIAM J. Sci. Comput.*, 42(5):A2620–A2654, 2020.
- [27] David I. Ketcheson. Highly Efficient Strong Stability-Preserving Runge–Kutta Methods with Low-Storage Implementations. *SIAM J. Sci. Comput.*, 30(4):2113–2136, 2008.
- [28] David A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Springer, 2009.
- [29] David A. Kopriva and Gregor J. Gassner. On the Quadrature and Weak Form Choices in Collocation Type Discontinuous Galerkin Spectral Element Methods. *J. Sci. Comput.*, 44(2):136–155, 2010.
- [30] David A. Kopriva, Jan Nordström, and Gregor J. Gassner. Error Boundedness of Discontinuous Galerkin Spectral Element Approximations of Hyperbolic Problems. *J. Sci. Comput.*, 72(1):314–330, 2017.
- [31] Sandra May and Florian Streitbürger. DoD stabilization for non-linear hyperbolic conservation laws on cut cell meshes in one dimension. *Appl. Math. Comput.*, 419: article no. 126854 (17 pages), 2022.
- [32] Seymour V. Parter. On the Legendre–Gauss–Lobatto Points and Weights. *J. Sci. Comput.*, 14(4):347–355, 1999.
- [33] Louis Petri, Gunnar Birke, Christian Engwer, and Hendrik Ranocha. Reproducibility repository for “The domain-of-dependence stabilization for cut-cell meshes is fully discretely stable”. https://github.com/louispetri/2025_dod_linear_stability, 2025.
- [34] Louis Petri, Sigrun Ortleb, Gunnar Birke, Christian Engwer, and Hendrik Ranocha. Domain-of-dependence-stabilized cut-cell discretizations of linear kinetic models with summation-by-parts properties, 2026. under review.
- [35] Hendrik Ranocha. SummationByPartsOperators.jl: A Julia library of provably stable semidiscretization techniques with mimetic properties. *J. Open Source Softw.*, 6(64): article no. 3454 (5 pages), 2021.
- [36] Hendrik Ranocha and Philipp Öffner. L_2 Stability of Explicit Runge-Kutta Schemes. *J. Sci. Comput.*, 75(2):1040–1056, 2018.
- [37] Simon Stickle and Gunilla Kreiss. Higher Order Cut Finite Elements for the Wave Equation. *J. Sci. Comput.*, 80(3):1867–1887, 2019.
- [38] Florian Streitbürger. *Stabilized discontinuous Galerkin methods for solving hyperbolic conservation laws on grids with embedded objects*. PhD thesis, Technische Universität Dortmund, 2023.
- [39] Florian Streitbürger, Gunnar Birke, Christian Engwer, and Sandra May. DoD Stabilization for Higher-Order Advection in Two Dimensions. In *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2020+1*, pages 495–508. Springer, 2023.
- [40] Zheng Sun and Chi-Wang Shu. Stability of the fourth order Runge-Kutta method for time-dependent partial differential equations. *Ann. Math. Sci. Appl.*, 2(2):255–284, 2017.
- [41] Zheng Sun and Chi-Wang Shu. Strong Stability of Explicit Runge-Kutta Time Discretizations. *SIAM J. Numer. Anal.*, 57(3):1158–1182, 2019.

- [42] Gabor Szegő. *Orthogonal Polynomials*, volume 23 of *Colloquium Publications*. American Mathematical Society, 1975.
- [43] Eitan Tadmor. From Semidiscrete to Fully Discrete: Stability of Runge-Kutta Schemes by the Energy Method II. In *Collected Lectures on the Preservation of Stability under Discretization*, volume 109 of *Proceedings in Applied Mathematics*, pages 25–49. Society for Industrial and Applied Mathematics, 2002.
- [44] Eitan Tadmor. On the stability of Runge–Kutta methods for arbitrarily large systems of ODEs. *Commun. Pure Appl. Math.*, 78(4):821–855, 2025.
- [45] Ole Tange. Gnu Parallel 20250422 ('Tariffs'), 2025. GNU Parallel is a general parallelizer to run multiple serial command line programs in parallel without changing them.
- [46] Christina G. Taylor, Lucas C. Wilcox, and Jesse Chan. An energy stable high-order cut cell discontinuous Galerkin method with state redistribution for wave propagation. *J. Comput. Phys.*, 521: article no. 113528 (23 pages), 2025.
- [47] Yuan Xu, Qiang Zhang, Chi-Wang Shu, and Haijin Wang. The L2-norm Stability Analysis of Runge–Kutta Discontinuous Galerkin Methods for Linear Hyperbolic Equations. *SIAM J. Numer. Anal.*, 57(4):1574–1601, 2019.