ICCFD13-2026-332

Arbitrary high-order embedded methods for hyperbolic PDEs: unified theory, stability, and validation in 1D and 2D

Mirco Ciallella*
Corresponding author: mirco.ciallella@u-paris.fr
* Université Paris Cité and Sorbonne Université, CNRS, LJLL, F-75005 Paris, France

Abstract: Embedded, or immersed, approaches have the goal of reducing to the minimum the computational costs associated with the generation of body-fitted meshes by only employing fixed, possibly Cartesian, meshes over which complex boundaries can move freely. However, this boundary treatment introduces a geometrical error of the order of the mesh size that, if not treated properly, can spoil the global accuracy of a high-order discretization, herein based on discontinuous Galerkin. The Shifted Boundary method and the Reconstruction for Off-site Data method are recently developed arbitrary high-order embedded approaches that compensate for this geometrical error. The former is based on Taylor expansions, while the latter relies on the solution of a constrained minimization problem. Both have the goal of correcting boundary conditions imposed on a computational boundary, which does not match the physical one. In this work, a novel unified theoretical framework is established showing that these two promising embedded boundary techniques can both be expressed in terms of polynomial corrections, thereby revealing a shared fundamental structure despite the seemingly different approaches. This unified perspective yields two major advances: first, it provides a significant algorithmic simplification, replacing both Taylor expansions and minimization problems with straightforward polynomial evaluations; second, it enables a fully-discrete stability analysis. Leveraging this framework, a thorough stability study of high-order DG methods coupled with polynomial correction-based embedded boundary treatments is conducted. Through a detailed eigenspectrum analysis of the discretized operators for the linear advection equation for polynomial degrees up to six, the impact of the boundary treatment on the global stability of the fully-discrete scheme is investigated. While the analysis is conducted for the one-dimensional linear advection equation and homogeneous Dirichlet boundary conditions, the theoretical findings are corroborated by numerical experiments with non-homogeneous boundary conditions for both linear and nonlinear hyperbolic equations, i.e. the Euler equations for compressible gas dynamics, in both one and two dimensions.

Keywords: Stability analysis, discontinuous Galerkin, high-order embedded boundary methods, hyperbolic equations.

1. Introduction

High-order methods have the potential to achieve very accurate solutions with reduced computational cost with respect to low order methods [1]. However, ensuring consistency and stability for high-order boundary conditions remains a very challenging task, especially when dealing with complex curved geometries. When dealing with non-trivial boundaries, accurately representing geometric features of the computational domain is as important as the numerical method itself [2]. As a matter of fact, the global accuracy of the numerical solution is given by both the accuracy of the numerical method and the precision used for the geometry representation.

A widespread approach in high-order numerical simulations is the isoparametric approach, whose goal is to use high-order polynomial reconstructions to approximate the given geometry, with the same precision of the numerical method used to approximate the problem [2]. This method achieves optimal convergence rates and improved accuracy in the presence of curved boundaries, and allows one to discretize complex geometries with very coarse meshes. However, the generation of high-order curvilinear meshes remains a real challenge: nonlinear mappings, complex quadrature rules, and a higher number of required degrees of freedom are just a few of the computational challenges that one needs to address. Furthermore, additional difficulties related to mesh generation [3] arise, which needs advanced techniques to ensure mesh validity and quality. Despite recent progress, dealing with curvilinear meshes remains a multifaceted task for realistic applications [4].

The main goal of embedded boundary methods is to reduce the effort related to mesh generation and mesh motion by only employing simple, fixed, easily generated meshes, over which embedded boundaries can move freely. However, contrary to body-fitted meshes, this treatment introduces a geometrical error proportional to the mesh size h. In this case, the numerical challenge is no longer related to mesh generation but to the development of accurate and stable embedded boundary conditions. The main idea behind these methods was introduced in the pioneering work of Peskin [5]. In the last fifty years, there has been a huge effort to develop unfitted methods with the objective of minimizing computational costs related to mesh generation and motion. Some of these works are found in [6, 7, 8, 9, 10], and references therein. Although most of these methods are first and second order accurate, there has been a recent interest in developing high-order embedded boundary methods, to be coupled with arbitrary high-order methods used to discretize the physical problem. Recent works in this direction can be found in [11, 12, 13].

The Shifted Boundary (SB) method was introduced in [14] as a Taylor-based embedded boundary method, which allows one to achieve high-order accuracy by imposing the boundary condition at the computational boundary, and correcting the geometrical error through a Taylor expansion. In [15], it was shown that it is possible to rewrite the truncated Taylor expansion of the SB method as a polynomial correction, given by the evaluation of the internal polynomial on the real boundary point. This simplifies the development, which no longer needs the computation of the high-order derivatives terms coming from the Taylor expansion. Some applications of the SB method can be found in [14, 16, 17, 18]. The Reconstruction for Off-site Data (ROD) method was developed before in finite volume [19, 20] and later in DG [21, 22]. In the context of DG, the ROD method consists of imposing consistent embedded boundary conditions by building a modified polynomial that is as close as possible to the internal one, with respect to the Euclidean norm, and satisfies exactly the exact boundary condition on a set of real boundary points. The coefficients of the ROD polynomial are simply defined as the outcome of a constrained minimization problem. With the new polynomial, the boundary condition is imposed at the computational boundary as for the SB method.

Despite their seemingly different foundations, Taylor expansion for SB and constrained minimization for ROD, here we show that both methods share a common structure: they can be written as polynomial corrections differing only by a scaling factor. The key point to achieve this result was studying the ROD method for a one-dimensional configuration, where the minimization problem can be explicitly solved, and the resulting polynomial can be written in a closed form. Several advantages come from this unified framework, which allows us to study the properties of both methods, compare them systematically, and derive new embedded approaches based on similar principles. A direct consequence of this unified view is a novel way to extend the method to multiple dimensions, bypassing the limitations of the original ROD formulation. Although the standard ROD and the novel polynomial corrections are equivalent in the one-dimensional case, their extension to multiple dimensions becomes different. Contrary to the original multi-dimensional ROD formulation, which required the inversion of the linear system coming from the minimization problem with multiple constraints, the one-dimensional polynomial correction can be extended straightforwardly to multiple dimensions without the need of any matrix inversion because it can be applied locally at each boundary point. On one side, this unified framework clearly reduces the implementation complexity of both families of methods; on the other, it allows us to write both methods in matrix form, which is a fundamental step to analyze their stability properties through the eigenspectrum visualization of the discretized operators.

This unified view enables, for the first time, a systematic stability analysis of SB and ROD for hyperbolic problems. The ROD method, as it was introduced in [22, 21], considers the squared Euclidean norm in the minimization problem; herein, we show that, by using the squared L^2 norm, it is possible to achieve a scheme that has improved stability properties. Following the same proof that allows us to recast the ROD method as a polynomial correction, we show that also the approach using the L^2 norm can be recast as a polynomial correction, which differs once again only in the definition of the scaling factor. The results of the analysis show that, for polynomials of degree $p \leq 4$, it is possible to derive a ROD method that is stable for explicit time integration, under a standard CFL condition. This is an important improvement when compared to the SB method, which shows a restriction on the CFL also for lower order polynomials. In the second part of the analysis, it is investigated whether implicit time integration can improve the stability properties of the embedded methods considered herein. Although the stability regions are enlarged for implicit time integration, we observe that for very high-order methods all considered approaches present lower bound conditions on the CFL, which needs to be high enough to ensure unconditional stability. All results obtained through the proposed analysis are validated through a targeted set of numerical experiments where several boundary configurations are considered, and with accuracy varying between second and seventh order.

The rest of the paper is organized as follows. In section 2 the high-order DG discretization for hyperbolic equations is presented. In section 3 the notation of the embedded boundary methods is introduced. Section 4 describes the high-order Taylor-based embedded boundary treatment, namely the SB polynomial correction. Section 5 presents the standard ROD method in multiple dimensions, based on the minimization of the squared Euclidean norm with multiple constraints to impose boundary conditions. In section 6, we show that several variations of minimization-based methods can be recast as polynomial corrections. The stability analysis for all embedded boundary methods are presented in section 7. Numerical tests that validate all the theoretical results are presented in section 8, and some conclusions are drawn in 9.

2. High-order discontinuous Galerkin for hyperbolic equations

2.1 Problem setting and notation

Herein, we consider high-order discretizations of the hyperbolic equation with a source term $s(x)$ in one space dimension, given by

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = s(x), \tag{1}$$

where $f(u)$ is the conservative flux function. For the linear advection equation, the flux function is simply taken $f(u) = u$. The generalization of DG methods to systems of hyperbolic equations and multiple dimensions is straightforward for Cartesian grids and omitted here for brevity. Below, the notation (x,t) will be dropped for compactness.

Let us consider a one-dimensional computational domain Ω discretized into N_e non-overlapping elements Ω_i, such that $\Omega = \bigcup_{i=1}^{N_e} \Omega_i$. Without loss of generality, the cell $\Omega_i = [x_{i-1/2}, x_{i+1/2}]$ has a fixed size of Δx. The numerical solution u is approximated by u_h, which belongs to the global discrete space V_h defined as: $V_h := \bigoplus_{i=1}^{N_e} V_i, \quad \text{where} \quad V_i := \mathbb{P}^p(\Omega_i)$. V_i is the local approximation space on element Ω_i, given by the space of polynomials of degree $p \geq 0$. Hence, the numerical solution is a piece-wise polynomial that is discontinuous across element interfaces. Inside each element Ω_i, the approximation u_h is expressed as a linear combination of basis functions $\{\varphi_n^i\}_{n=0}^p$ of the space V_i:

$$u_h(x,t)|_i = \sum_{n=0}^{p} \varphi_n^i(x) u_{i,n}(t), \tag{2}$$

where $\{u_{i,n}(t)\}_{n=0}^p$ are the degrees of freedom of cell Ω_i.

The elemental semi-discrete DG weak formulation is written by projecting (1) and integrating by parts,

$$\int_{\Omega_i} \varphi_m^i \frac{\mathrm{d}u_h}{\mathrm{d}t}\,\mathrm{d}x - \int_{\Omega_i} \partial_x \varphi_m^i f(u_h)\,\mathrm{d}x + \int_{\partial\Omega_i} \varphi_m^i \hat{f}(u_h^-, u_h^+)\, n\,\mathrm{d}\Gamma = \int_{\Omega_i} \varphi_m^i s(x)\,\mathrm{d}x, \quad \forall \varphi_m^i \in V_i. \tag{3}$$

n is the outward normal, and $\hat{f}(u_h^-, u_h^+)$ is a numerical flux that depends on the left $(\cdot)^-$ and right $(\cdot)^+$ reconstructed interface values. The numerical flux at the element interfaces is computed using a local Lax-Friedrichs flux:

$$\hat{f}(u_h^-, u_h^+) = \frac{1}{2}(f(u_h^-) + f(u_h^+)) - \frac{\rho}{2}(u_h^+ - u_h^-),$$

where $\rho = \max(|f'(u_h^-)|, |f'(u_h^+)|)$. This numerical flux reduces to the upwind flux $\hat{f}(u_h^-, u_h^+) = u_h^-$ for a linear advection equation with positive advection speed equal to one.

2.2 Matrix form and eigenvalue problem

In this section, the matrix form of the semi-discrete problem is derived, which is a fundamental step to analyze the stability properties of the considered embedded boundary methods through the eigenspectrum visualization of the discretized operators. For the linear advection equation, the semi-discrete problem for a generic element Ω_i can be written in matrix form as follows:

$$M_i \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} - K_i^s \mathbf{u}_i + K_i^R \mathbf{u}_i - K_i^L \mathbf{u}_{i-1} = \mathbf{s}_i, \tag{4}$$

where $\mathbf{u}_i = [u_{i,0}, \dots, u_{i,p}]^T$ is the vector of degrees of freedom belonging to element Ω_i. The mass and stiffness matrices are defined as,

$$M_i = \int_{\Omega_i} \varphi_m^i \varphi_n^i\,\mathrm{d}x, \qquad K_i^s = \int_{\Omega_i} \partial_x \varphi_m^i \varphi_n^i\,\mathrm{d}x, \quad \forall \varphi_m^i, \varphi_n^i \in V_i, \tag{5}$$

while interface matrices are defined as,

$$K_i^R = \varphi_m^i(x_{i+1/2})\varphi_n^i(x_{i+1/2}), \qquad K_i^L = \varphi_m^i(x_{i-1/2})\varphi_n^{i-1}(x_{i+1/2}) \tag{6}$$

where $\varphi_m^i(x_{i+1/2})$ represents the m-th basis function of the element Ω_i evaluated in $x_{i+1/2}$.

To trace the stability region of the methods from the eigenspectrum, it is necessary to write the system in a global matrix form of the kind

$$\mathbf{M}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = \mathbf{K}\mathbf{U}, \tag{7}$$

where $\mathbf{M}$ and $\mathbf{K}$ represent the global mass and stiffness matrices, respectively. $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{N_e}]^T$ is the global vector of degrees of freedom.

To derive the eigenvalue problem for the explicit time integration of the semi-discrete system (7), we consider the amplification factor of arbitrary high-order Deferred Correction (DeC) time integration schemes [23] of order $p+1$, which gives

$$\mathbf{U}^{n+1} = \left(\sum_{k=0}^{p+1} \frac{\mu^k}{k!}\right)\mathbf{U}^n, \tag{8}$$

where $\mathbf{U}^n = \mathbf{U}(t^n)$ and $\mu = \lambda\Delta t$, with Δt the time step and λ the eigenvalue of semi-discrete operator $\mathbf{M}^{-1}\mathbf{K}$. In this work the polynomial degree for the space discretization is taken up to $p = 6$ and it is coupled with DeC up to order 7. Therefore, the stability region of the considered fully discrete problem is $|z(\mu)| \leq 1$, where $z(\mu) = \sum_{k=0}^{p+1} \frac{\mu^k}{k!}$.

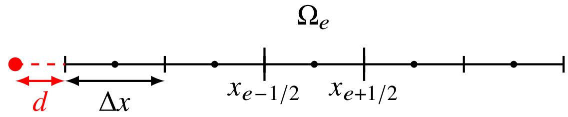

Figure 1: Mesh configuration: tessellation of one-dimensional elements with a boundary condition (red circle) imposed at a distance d from the left-most interface of the domain, which is considered as the computational boundary of the problem.

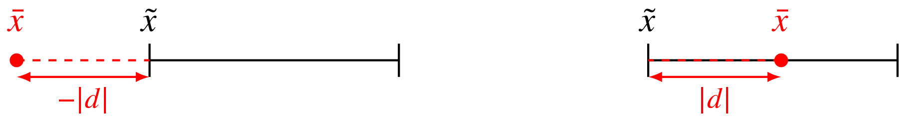

Figure 2: Boundary position: the real boundary position can be either outside the first mesh element (left), or inside the first mesh element (right). In the first case, the distance will be taken as negative $d < 0$, while for the second one the distance is taken as positive $d > 0$.

3. Notation of embedded boundary methods

The aim of this work is to explore the impact of high-order embedded boundary methods when the distance d from the real boundary to the computational one is of the order of the mesh size Δx. This is a classical configuration that can occur when dealing with numerical simulations with boundaries treated as *immersed* on a Cartesian mesh (see figure 1 for more details). In multi-dimensional configurations, the mapping between the real boundary and the computational boundary is not unique and can be prescribed using different approaches: closest-point, level-set, projections. However, in one space dimension, the mapping between the two is uniquely identified, which simplifies the analysis.

Consider $\bar{x}$ the position of a point on the real boundary, while $\tilde{x}$ the position of the corresponding computational boundary point. The mapping between the two is given by $\bar{x} = \tilde{x} + d$, where d is the distance between them. Thus, the distance is allowed to vary within the range $d \in [-\Delta x, \Delta x]$: meaning that the real boundary can be either outside the first mesh element (when $d < 0$), or inside the first mesh element (when $d > 0$), as shown in figure 2.

As a standard interface, the boundary condition is imposed at the computational boundary through a numerical flux. For the linear advection and an upwind flux, the left boundary flux of the domain is given by $\hat{f}(v_h, u_h^+) = v_h$, where v_h corresponds to the considered high-order embedded boundary reconstruction.

4. Shifted boundary polynomial correction

In this section, the main notation of the SB polynomial correction to impose embedded boundary conditions is recalled. Due to its simpler shape, it will be then straightforward to make a link with the novel polynomial corrections developed in section 6. The idea of the considered SB treatment consists in imposing boundary conditions on the computational boundary rather than the real one, while retaining the high-order accuracy thanks to a truncated Taylor expansion. Let u_D be the prescribed Dirichlet boundary condition of the advected variable on the real boundary $\bar{x}$. A modified boundary condition v_h for the computational boundary $\tilde{x}$ can be developed starting from the following Taylor expansion:

$$u(\bar{x}) = u(\tilde{x} + d) = u(\tilde{x}) + \sum_{m=1}^{p} \partial_x^{(m)} u(\tilde{x}) \frac{d^m}{m!}. \tag{9}$$

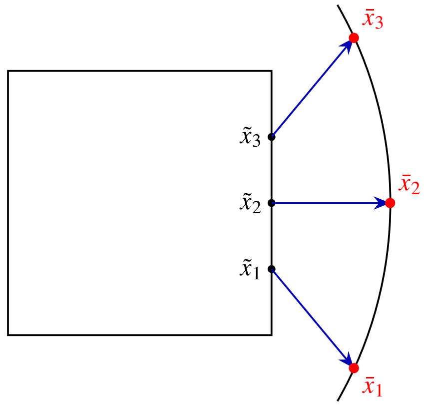

Figure 3: Mapping between computational and real boundary points in 2D.

Thus, a modified boundary condition can be defined including all additional corrective terms and by imposing the real boundary condition $u(\bar{x}) = u_D$:

$$v_h(\tilde{x}) = u_D - \sum_{m=1}^{p} \partial_x^{(m)} u_h(\tilde{x}) \frac{d^m}{m!}. \tag{10}$$

As shown in [15], the SB polynomial correction can be extracted naturally from equation (9), meaning that the whole set of high-order correction terms can be recovered completely by simply evaluating the polynomial of the boundary cell in the point $\bar{x} = \tilde{x} + d$ and by making the difference with the solution evaluated in $\tilde{x}$. Finally the modified Dirichlet boundary condition v_h on the computational boundary can be reformulated under the following correction:

$$v_h(\tilde{x}) = u_D - (u_h(\bar{x}) - u_h(\tilde{x})) = [\boldsymbol{\varphi}(\tilde{x}) - \boldsymbol{\varphi}(\bar{x})]^T \mathbf{u} + u_D, \tag{11}$$

where $\mathbf{u}$ are the degrees of freedom of the boundary cell, and $\boldsymbol{\varphi} = [\varphi_0, \ldots, \varphi_p]^T$ are the basis functions.

It should be noticed that equation (11) can be easily extended to multiple dimensions by enforcing the modified boundary condition with the polynomial correction in each quadrature point of the boundary integral performed on the computational boundary as shown in Figure 3 (see [15] for further details). Indeed, in multiple space dimensions, we would simply have

$$v_h(\tilde{\mathbf{x}}) = [\boldsymbol{\varphi}(\tilde{\mathbf{x}}) - \boldsymbol{\varphi}(\bar{\mathbf{x}})]^T \mathbf{u} + u_D,$$

where $\mathbf{x} = (x, y)$ are the two-dimensional coordinates and $\boldsymbol{\varphi}(\mathbf{x})$ are the multi-dimensional basis functions.

5. Standard ROD method with multiple constraints

When working with DG methods, the standard ROD in multiple dimensions [21, 22] is based on the solution of a constrained minimization problem, where the modified polynomial v_h is built by minimizing the distance to the internal polynomial u_h while satisfying the boundary condition on a set of real boundary points, which are typically the quadrature points on the computational boundary mapped onto the real boundary.

The minimization problem can be written as follows. Let u_h be the polynomial approximation of degree p in the boundary element Ω_b. A modified polynomial v_h is defined as $v_h(\mathbf{x}) = \boldsymbol{\varphi}^T(\mathbf{x})\mathbf{v}$, where the coefficients $\mathbf{v}$ are determined by minimizing the distance to $\mathbf{u}$ subject to the boundary constraints $v_h(\bar{\mathbf{x}}_k) = u_D(\bar{\mathbf{x}}_k)$ for $k = 1, \ldots, K$. Let us take $\mathbf{u}_D = [u_D(\bar{\mathbf{x}}_1), \ldots, u_D(\bar{\mathbf{x}}_K)]^T$ where K is the number of constraints in each boundary cell, in general equal to the number of quadrature points used for the boundary integration. Following the same notation as above, $\bar{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ refer to points on the real and computational boundary, respectively.

When considering multiple constraints, the functional of the ROD method becomes:

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{v} - \mathbf{u}\|_2^2 + \sum_{k=1}^{K} \lambda_k \left(v_h(\bar{\mathbf{x}}_k) - u_D(\bar{\mathbf{x}}_k) \right) \tag{12}$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^T$ is the vector of Lagrange multipliers. In this case, the optimality conditions give:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = \mathbf{v} - \mathbf{u} + \Phi(\bar{\mathbf{x}})\boldsymbol{\lambda} = 0, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \Phi^T(\bar{\mathbf{x}})\mathbf{v} - \mathbf{u}_D = 0 \tag{13}$$

where $[\Phi(\bar{\mathbf{x}})]_{jk} = \varphi_j(\bar{\mathbf{x}}_k)$.

In earlier works related to the development of ROD for discontinuous Galerkin [21, 22], the following linear system is solved to obtain the coefficients $\mathbf{v}$:

$$\begin{bmatrix} I & \Phi(\bar{\mathbf{x}}) \\ \Phi^T(\bar{\mathbf{x}}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathbf{u}_D \end{bmatrix}, \tag{14}$$

Then, the coefficients $\mathbf{v}$ can be used to compute the ROD boundary condition as $v_h(\tilde{\mathbf{x}}) = \boldsymbol{\varphi}^T(\tilde{\mathbf{x}})\mathbf{v}$.

Remark 5.1 (SB and ROD differences in multiple dimensions)**.** *Contrary to the SB method, where each quadrature point on the computational boundary is updated independently, the ROD method computes one modified polynomial for each boundary cell that is aware of all the boundary conditions imposed on* $\{\bar{\mathbf{x}}_k\}$ *(see Figure 3). Then, the modified polynomial is evaluated in the corresponding quadrature points* $\{\tilde{\mathbf{x}}_k\}$ *on the computational boundary to compute the modified boundary condition.*

6. Minimization-based embedded methods as polynomial corrections

In this section, we show that, in a one-dimensional setting, the ROD method reduces to a polynomial correction. Remarkably, this correction shares the same structure as the SB polynomial correction, differing only by a scaling factor.

Let u_h be the polynomial approximation of degree p in the boundary element. Following the same approach as in the previous section, we define a modified polynomial $v_h = \boldsymbol{\varphi}^T(x)\mathbf{v}$ where the coefficients $\mathbf{v}$ are determined by minimizing the distance to $\mathbf{u}$ subject to the boundary constraint $v_h(\bar{x}) = u_D$. Notice that in this case, only one constraint is imposed on the polynomial v_h.

It can be noticed that the distance $\mathcal{D}$ can be chosen in different ways as long as it stays a convex function. The simplest choice, as also followed in [21, 22], consists in taking the squared Euclidean distance $\|\cdot\|_2^2$. The polynomial correction found using the Euclidean norm will be referred to as ROD-E.

Herein, we also show a possible alternative to the ROD-E, which is obtained by taking the distance based on the squared L^2 norm $\|\cdot\|_{L^2}^2$. For this formulation, the method label will be ROD-L^2.

In the following, we show how it is possible to write both minimization-based approaches as polynomial corrections. The key point is to move to a one-dimensional setting, where only one constraint is imposed, which allows us to explicitly solve the minimization problem and to write the resulting polynomial in a closed form. Then, the resulting polynomial correction can be extended to multiple dimensions by applying the same correction locally at each boundary point, without the need of solving a global minimization problem with multiple constraints as in the original ROD formulation. Applying a local correction at each boundary point would not be equivalent to the original ROD formulation, which was computing one modified polynomial for each boundary cell. Instead, the local correction would mean that a different modified polynomial is computed for each boundary point. Herein, we prove that these local corrections are also consistent and arbitrary high-order accurate.

To further unify the different formulations, at the end of the section, we introduce a general weighted ROD method, denoted ROD-W, where the distance is measured via a weighted squared Euclidean norm $\|\cdot\|_W^2$, with W a positive definite matrix. This single framework encompasses ROD-E and ROD-L^2 as special cases: choosing $W = I$ recovers ROD-E, while $W = M$ yields ROD-L^2. Moreover, the SB method also emerges as a particular instance of ROD-W with a specific choice of the weight matrix W.

6.1 Euclidean norm as cost function

By using the Euclidean norm, the constrained minimization problem of ROD-E can be solved by introducing a Lagrange multiplier λ and by considering the following functional:

$$\mathcal{L}(\mathbf{v}, \lambda) = \frac{1}{2}\|\mathbf{v} - \mathbf{u}\|_2^2 + \lambda(v_h(\bar{x}) - u_D). \tag{15}$$

The optimality conditions for this problem, in vector form, can be written as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = \mathbf{v} - \mathbf{u} + \lambda \boldsymbol{\varphi}(\bar{x}) = \mathbf{0}, \qquad \frac{\partial \mathcal{L}}{\partial \lambda} = \boldsymbol{\varphi}^T(\bar{x})\mathbf{v} - u_D = 0 \tag{16}$$

If one wanted to solve this in a system form, the following linear system would need to be inverted:

$$\begin{bmatrix} I & \boldsymbol{\varphi}(\bar{x}) \\ \boldsymbol{\varphi}^T(\bar{x}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ u_D \end{bmatrix} \tag{17}$$

By solving this system, the polynomial coefficients $\mathbf{v}$ are obtained and $v_h(\tilde{x})$ can be directly computed.

Of course, the solution of the system (17) can be found in a closed form, which allows us to write the ROD-E modified boundary condition as a polynomial correction, as shown in the following proposition.

Proposition 6.1. *Computing the ROD-E modified boundary condition by inverting the linear system* (17) *and evaluating the polynomial v_h in $\tilde{x}$ is equivalent to the following polynomial correction:*

$$v_h(\tilde{x}) = [\boldsymbol{\varphi}(\tilde{x}) - \alpha^{\text{ROD-E}}(\tilde{x})\boldsymbol{\varphi}(\bar{x})]^T \mathbf{u} + \alpha^{\text{ROD-E}}(\tilde{x}) u_D, \quad \text{where} \quad \alpha^{\text{ROD-E}}(\tilde{x}) = \frac{\boldsymbol{\varphi}(\tilde{x})^T \boldsymbol{\varphi}(\bar{x})}{\|\boldsymbol{\varphi}(\bar{x})\|_2^2}. \tag{18}$$

Proof. The solution of this minimization problem can be found starting from the optimality conditions that give:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = \mathbf{v} - \mathbf{u} + \lambda \boldsymbol{\varphi}(\bar{x}) = \mathbf{0}, \qquad \frac{\partial \mathcal{L}}{\partial \lambda} = \boldsymbol{\varphi}^T(\bar{x})\mathbf{v} - u_D = 0. \tag{19}$$

By substituting the first expression into the second, we compute λ and the modified degrees of freedom of the ROD-E corrected polynomial read:

$$\mathbf{v} = \mathbf{u} - \left(\frac{\boldsymbol{\varphi}(\bar{x})^T \mathbf{u} - u_D}{\boldsymbol{\varphi}(\bar{x})^T \boldsymbol{\varphi}(\bar{x})} \right) \boldsymbol{\varphi}(\bar{x}) \tag{20}$$

In order to write ROD-E as a polynomial correction, it is necessary to consider the evaluation of the v_h in the computational boundary point $\tilde{x}$, which gives

$$v_h(\tilde{x}) = \boldsymbol{\varphi}(\tilde{x})^T \mathbf{v} = \boldsymbol{\varphi}(\tilde{x})^T \mathbf{u} - \left(\frac{\boldsymbol{\varphi}(\bar{x})^T \mathbf{u} - u_D}{\|\boldsymbol{\varphi}(\bar{x})\|_2^2} \right) \boldsymbol{\varphi}(\tilde{x})^T \boldsymbol{\varphi}(\bar{x}). \tag{21}$$

It is now easy to infer that the ROD-E modified boundary condition can be recast as the polynomial correction (18). □

Remark 6.1 (Similarities with the SB method)**.** *As a matter of fact, by writing ROD-E as in equation* (18)*, it is possible to make a direct link with the SB correction* (11)*. The SB polynomial correction is* *retrieved if the factor $\alpha^{ROD\text{-}E}(\tilde{x})$ happens to be equal to 1.*

6.2 L^2 norm as cost function

For the ROD-L^2 method, the constrained minimization problem consists in minimizing the following functional:

$$\mathcal{L}(\mathbf{v},\lambda) = \frac{1}{2}\int_{\Omega_b} (v_h(x) - u_h(x))^2 \mathrm{d}x + \lambda(v_h(\tilde{x}) - u_D)$$

Substituting the polynomial expansions, the mass matrix appears in the first optimality condition. The problem to solve can be rewritten in the following matrix form:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = M(\mathbf{v} - \mathbf{u}) + \lambda \boldsymbol{\varphi}(\tilde{x}) = \mathbf{0}, \qquad \frac{\partial \mathcal{L}}{\partial \lambda} = \boldsymbol{\varphi}(\tilde{x})^T \mathbf{v} - u_D = 0.$$

Again, by solving the following linear system,

$$\begin{bmatrix} M & \boldsymbol{\varphi}(\tilde{x}) \\ \boldsymbol{\varphi}^T(\tilde{x}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \lambda \end{bmatrix} = \begin{bmatrix} M\mathbf{u} \\ u_D \end{bmatrix} \tag{22}$$

the coefficients $\mathbf{v}$ are obtained, and the evaluated polynomial $v_h(\tilde{x})$ can be computed.

Proposition 6.2. *Computing the ROD-L^2 modified boundary condition by inverting the linear system* (22) *and evaluating the polynomial v_h in $\tilde{x}$ is equivalent to the following polynomial correction:*

$$v_h(\tilde{x}) = [\boldsymbol{\varphi}(\tilde{x}) - \alpha^{ROD\text{-}L^2}(\tilde{x})\boldsymbol{\varphi}(\tilde{x})]^T \mathbf{u} + \alpha^{ROD\text{-}L^2}(\tilde{x}) u_D, \quad \textit{where} \quad \alpha^{ROD\text{-}L^2}(\tilde{x}) = \frac{\boldsymbol{\varphi}(\tilde{x})^T M^{-1} \boldsymbol{\varphi}(\tilde{x})}{\|\boldsymbol{\varphi}(\tilde{x})\|^2_{M^{-1}}} \tag{23}$$

Proof. From the optimality conditions, we obtain the following relation for $\mathbf{v}$ and λ:

$$\mathbf{v} = \mathbf{u} - \lambda M^{-1}\boldsymbol{\varphi}(\tilde{x}), \qquad \lambda = \frac{\boldsymbol{\varphi}(\tilde{x})^T \mathbf{u} - u_D}{\boldsymbol{\varphi}(\tilde{x})^T M^{-1} \boldsymbol{\varphi}(\tilde{x})}$$

In this case, the modified degrees of freedom are:

$$\mathbf{v} = \mathbf{u} - \left(\frac{\boldsymbol{\varphi}(\tilde{x})^T \mathbf{u} - u_D}{\boldsymbol{\varphi}(\tilde{x})^T M^{-1} \boldsymbol{\varphi}(\tilde{x})} \right) M^{-1} \boldsymbol{\varphi}(\tilde{x})$$

As it was done above, in order to write the ROD-L^2 as a polynomial correction, it is necessary to evaluate v_h at the computational boundary $\tilde{x}$:

$$v_h(\tilde{x}) = \boldsymbol{\varphi}(\tilde{x})^T \mathbf{v} = \boldsymbol{\varphi}(\tilde{x})^T \mathbf{u} - \left(\frac{\boldsymbol{\varphi}(\tilde{x})^T \mathbf{u} - u_D}{\|\boldsymbol{\varphi}(\tilde{x})\|^2_{M^{-1}}} \right) \boldsymbol{\varphi}(\tilde{x})^T M^{-1} \boldsymbol{\varphi}(\tilde{x}). \tag{24}$$

From equation (24), it is direct to recast the ROD-L^2 method as the polynomial correction (23).

□

6.3 General weight matrix in cost function

With the goal of building a general framework that encompasses all the different methods presented above, it is possible to consider a more general cost function in the ROD formulation by introducing a general symmetric positive definite weight matrix W.

The minimization-based approach with a general weight matrix W, which we here call the ROD-W method, reconstructs a modified polynomial v_h that minimizes a weighted distance with the internal polynomial u_h, while satisfying the Dirichlet condition on the real boundary.

The constrained minimization problem is solved using the Lagrangian:

$$\mathcal{L}(\mathbf{v}, \lambda) = \frac{1}{2}(\mathbf{v} - \mathbf{u})^T W(\mathbf{v} - \mathbf{u}) + \lambda(v_h(\bar{x}) - u_D)$$

Following the same steps as above, the ROD-W corrected boundary value can be expressed as:

$$v_h(\tilde{x}) = [\boldsymbol{\varphi}(\tilde{x}) - \alpha^{\text{ROD-W}}(\tilde{x})\boldsymbol{\varphi}(\bar{x})]^T \mathbf{u} + \alpha^{\text{ROD-W}}(\tilde{x}) u_D, \quad \text{where} \quad \alpha^{\text{ROD-W}}(\tilde{x}) = \frac{\boldsymbol{\varphi}(\tilde{x})^T W^{-1} \boldsymbol{\varphi}(\bar{x})}{\|\boldsymbol{\varphi}(\bar{x})\|^2_{W^{-1}}}.$$

It is straightforward to show that, when $W = I$ (identity matrix), we recover the ROD-E method. Instead, when $W = M$ (mass matrix), we recover the ROD-L^2 method.

In this general framework, the SB correction can be interpreted as a ROD method with a particular weight matrix W_{SB} that satisfies:

$$\frac{\boldsymbol{\varphi}(\tilde{x})^T W_{\text{SB}}^{-1} \boldsymbol{\varphi}(\bar{x})}{\boldsymbol{\varphi}(\bar{x})^T W_{\text{SB}}^{-1} \boldsymbol{\varphi}(\bar{x})} = 1 \quad \Longleftrightarrow \quad \boldsymbol{\delta}^T W_{\text{SB}}^{-1} \boldsymbol{\varphi}(\bar{x}) = 0, \quad \text{where} \quad \boldsymbol{\delta} = \boldsymbol{\varphi}(\tilde{x}) - \boldsymbol{\varphi}(\bar{x}).$$

Although rather unlikely in actual problems, if W_{SB}^{-1} had by chance the shape: $W_{\text{SB}}^{-1} = I - \frac{\delta\delta^T}{\delta^T\delta}$, the ROD method would coincide with the SB method. As a matter of fact, after simple computations, this matrix gives

$$\boldsymbol{\delta}^T \left(I - \frac{\boldsymbol{\delta}\boldsymbol{\delta}^T}{\boldsymbol{\delta}^T\boldsymbol{\delta}} \right) \boldsymbol{\varphi}(\bar{x}) = \boldsymbol{\delta}^T \boldsymbol{\varphi}(\bar{x}) - \boldsymbol{\delta}^T \frac{\boldsymbol{\delta}\boldsymbol{\delta}^T}{\boldsymbol{\delta}^T\boldsymbol{\delta}} \boldsymbol{\varphi}(\bar{x}) = 0.$$

7. Stability analysis

To simplify the presentation of this section, the analysis will be performed with a minimal system configuration. We consider two elements containing $p + 1$ degrees of freedom each. In contrast to the von Neumann analysis, which assumes an infinite number of cells, the presence of boundary conditions makes the analysis dependent on the finite number of cells in the domain. Nevertheless, the qualitative behavior of the results remains unaffected by the specific number of cells. As a matter of fact, we show that only the eigenvalues of the boundary cell are the ones affected by the embedded boundary treatment, while adding more internal cells only adds more eigenvalues that do not affect the stability of the scheme. In the context of DG methods, it is trivial to show that thanks to the fact that the global matrices have a block structure. For the analysis of the embedded boundary methods, the CFL number is normalized by the maximum CFL number of the internal DG scheme, computed assuming a periodic domain. For simplicity, the analysis is performed taking $\Delta x = 1$. Concerning implicit integration, only implicit Euler will be considered to show in practice the beneficial effect of implicit time integration.

7.1 Standard periodic boundary conditions and explicit time integration

The DG method, following the notation of equation (4), for a system of two cells with periodic boundary conditions imposed at the left (of the first cell $\Omega_1 = [1/2, 3/2]$) and at the right (of the second cell $\Omega_2 = [3/2, 5/2]$) reads

$$\begin{aligned} M_1 \frac{\mathrm{d}\mathbf{u}_1}{\mathrm{d}t} &= K_1^s \mathbf{u}_1 - (K_1^R \mathbf{u}_1 - K_1^L \mathbf{u}_2), \\ M_2 \frac{\mathrm{d}\mathbf{u}_2}{\mathrm{d}t} &= K_2^s \mathbf{u}_2 - (K_2^R \mathbf{u}_2 - K_2^L \mathbf{u}_1). \end{aligned} \tag{25}$$

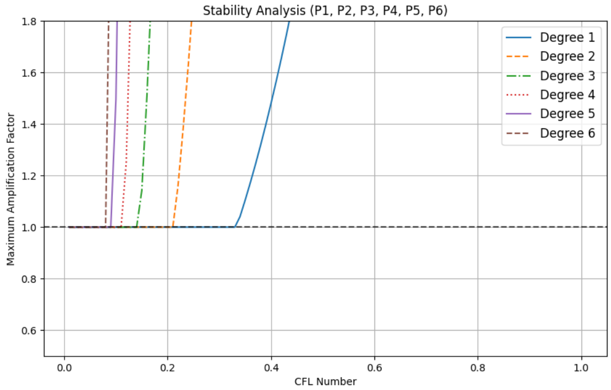

Figure 4: Amplification factor: periodic boundary conditions for the explicit discontinuous Galerkin method. When the amplification factor becomes greater than 1, the scheme is unstable.

Therefore, the global matrices for this system are

$$\mathbf{M} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} K_1^s - K_1^R & K_1^L \\ K_2^L & K_2^s - K_2^R \end{bmatrix},$$

where, in this simplified analysis based on uniform meshes, $M_1 = M_2$ and $K_1^s = K_2^s$ and they are equal to the definitions given in equation (5). The global vector of degrees of freedom is simply $\mathbf{U}(t) = [\mathbf{u}_1(t)\ \mathbf{u}_2(t)]^T$. Following the notation given in equation (6), interface matrices are defined as

$$K_1^R = \varphi_i^1(x_{3/2})\varphi_j^1(x_{3/2}), \qquad K_1^L = \varphi_i^1(x_{1/2})\varphi_j^2(x_{5/2}),$$

and

$$K_2^R = \varphi_i^2(x_{5/2})\varphi_j^2(x_{5/2}), \qquad K_2^L = \varphi_i^2(x_{3/2})\varphi_j^1(x_{3/2}).$$

It can be noticed that, with periodic boundary conditions, the standard CFL restriction for DG is obtained: $\mathrm{CFL}_{\max}^p \approx \frac{1}{2p+1}$. The maximum amplification factor for periodic boundary conditions is presented in figure 4. In the following sections, a normalized CFL $\in [0, 1]$ is considered by using the reference CFL values obtained in this analysis, i.e. $\frac{\Delta t}{\Delta x} = \mathrm{CFL}_{\max}^p \mathrm{CFL}$. This is done to simplify the visualization of the stability regions for the embedded boundary methods.

7.2 Shifted boundary treatment and explicit time integration

In this section, the homogeneous Dirichlet boundary condition is considered to be applied on a point that does not coincide with the left interface of the first (boundary) element. Since an upwind flux is considered, there is no need to impose any boundary condition of the right interface of the second element. Recall that the analysis only includes two elements to simplify the eigenspectrum visualization.

Following the notation in section 4, the analysis will consider both situations shown in figure 2 where the boundary condition is *shifted* to the left and to the right of the left interface of the first cell (i.e. $d < 0$ and $d > 0$, with $d \in [-\Delta x, \Delta x]$). Since the analysis is performed with $\Delta x = 1$, thus the distance will be varied considering $d \in [-1, 1]$. Having either $d < 0$ or $d > 0$ is a choice that can be taken when meshing a domain and embedding an internal boundary: $d < 0$ means that the elements crossed by the embedded boundary are not considered as active elements, while $d > 0$ means that the elements crossed by the embedded boundary are kept as active elements of the discretization. Herein, as well as in the following sections, we study both configurations to illustrate both possibilities. Due to the upwind flux, the left interface receives the flux given by the SB polynomial correction (11):

$$v_h(x_{1/2}) = u(x_{1/2}) - u(x_{1/2}+d) + u_D = \sum_{j=0}^{p}(\varphi_j^1(x_{1/2}) - \varphi_j^1(x_{1/2}+d))u_{1,j}, \tag{26}$$

where a homogeneous Dirichlet condition $u_D = 0$ is imposed, and $u(x_{1/2}+d)$ is the value of the solution of the boundary cell on the real boundary $\bar{x}$.

In this case, the global system is

$$\begin{aligned} M_1\frac{\mathrm{d}\mathbf{u}_1}{\mathrm{d}t} &= K_1^s\mathbf{u}_1 - (K_1^R\mathbf{u}_1 - K_1^{\mathrm{SB}}\mathbf{u}_1), \\ M_2\frac{\mathrm{d}\mathbf{u}_2}{\mathrm{d}t} &= K_2^s\mathbf{u}_2 - (K_2^R\mathbf{u}_2 - K_2^L\mathbf{u}_1), \end{aligned} \tag{27}$$

where the left interface matrix that introduces the SB polynomial correction is defined as

$$K_1^{\mathrm{SB}} = \varphi_i^1(x_{1/2})(\varphi_j^1(x_{1/2}) - \varphi_j^1(x_{1/2}+d)).$$

Therefore, the global matrices for this system are

$$\mathbf{M} = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} K_1^s - K_1^R + K_1^{\mathrm{SB}} & 0 \\ K_2^L & K_2^s - K_2^R \end{bmatrix}, \tag{28}$$

where the other local matrices are defined as in section 7.1. The stability region for explicit discontinuous Galerkin schemes with SB polynomial correction is shown in figure 5. As in the previous section, the stability region is given by the amplification factor computed from the eigenvalues of the discretized operators, which are amplified with the corresponding high-order time integration method (8). It should be noticed that the stability region is not symmetric with respect to the value of $d \in [-1,1]$. When $d \in [-1,0]$, there are stable areas with a constraint on the CFL number that becomes stricter as $d \to -1$ and p increases. While for $d \in [0,1]$ the stability region of the method is not constrained by the CFL number, but the method becomes unstable after a certain threshold $d > d_{max}$, which depends on the polynomial degree. In particular, when $d = -1$, the maximum admissible time step is dictated by a CFL ≤ 0.6 for $\mathbb{P}^1$, CFL ≤ 0.2 for $\mathbb{P}^2$, CFL ≤ 0.05 for $\mathbb{P}^3$, CFL ≤ 0.008 for $\mathbb{P}^4$. For $\mathbb{P}^5$, the furthest distance is $d = -0.25$ with an associated CFL ≤ 0.08. For $\mathbb{P}^6$, the furthest distance is $d = -0.07$ with an associated CFL ≤ 0.3. This means that for $\mathbb{P}^5$ and $\mathbb{P}^6$, also for $d \in [-1,0]$, the SB method becomes unconditionally unstable after a certain distance threshold. Since, for $d \in [0,1]$, the scheme becomes unconditionally unstable after a certain distance threshold even for lower order polynomials, the configuration $d > 0$ is in general not recommended.

It is also interesting to look at the characteristic polynomial and explicit eigenvalues of the semi-discrete system. Indeed, thanks to the characteristic polynomial, it is possible to prove that only the eigenvalues of the boundary cell are affected by the SB polynomial correction, i.e. they depend on the distance d, while the eigenvalues of the internal cells (no matter how many there are) do not depend on d. For a two cells system with a $\mathbb{P}^1$ method, the characteristic polynomial reads

$$\mathcal{P}_{\mathbb{P}^1}(\mu) = \frac{\left(\mu^2 + 4\mu + 6\right)\left(\mu^2 - 6\mu d + 4\mu + 6\right)}{12^2}.$$

By increasing the number of cells, for example to three cells, the characteristic polynomial becomes

$$\mathcal{P}_{\mathbb{P}^1}(\mu) = \frac{\left(\mu^2 + 4\mu + 6\right)^2\left(\mu^2 - 6\mu d + 4\mu + 6\right)}{12^3},$$

and so on for more cells.

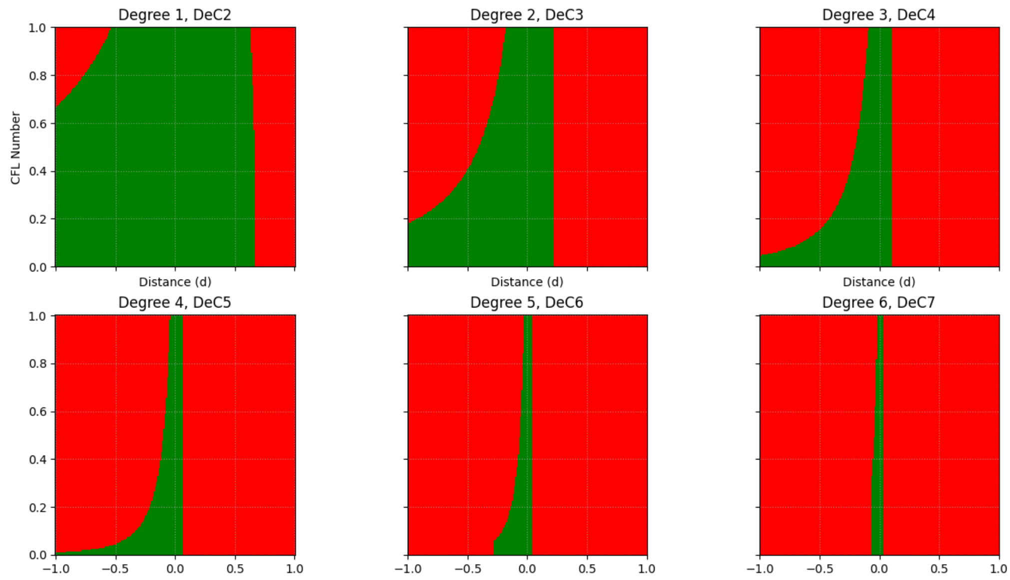

Figure 5: Stability region: SB polynomial correction of the homogeneous Dirichlet condition for the explicit discontinuous Galerkin method. Green areas are stable, while red areas are unstable.

The same behaviour is also observed for $\mathbb{P}^2$. While, for $\mathbb{P}^3$ and higher order polynomials, the characteristic polynomial becomes extremely cumbersome to be reported here.

For $\mathbb{P}^1$, analytical eigenvalues can be easily computed by hand, and read as

$$\{-2 \pm \sqrt{2}\mathbb{i},\ 3d \pm \sqrt{9d^2 - 12d - 2} - 2\}, \tag{29}$$

where the first two are the eigenvalues of the cell far from the boundary, while the last two are the ones of the cell affected by the SB polynomial correction. The fact that only the eigenvalues of the boundary cell depend on the distance d is a direct consequence of the block structure of the global matrices in DG methods, which allows to factorize the characteristic polynomial as shown above. As expected, some of the eigenvalues depend on the distance d of the immersed boundary from the left edge of the cell, which may bring either a positive or a negative real part. In particular, the real part of the last two eigenvalues is negative if $d < 2/3 \approx 0.66$ which is the distance threshold after which the $\mathbb{P}^1$ scheme becomes unconditionally unstable. For $\mathbb{P}^2$ and higher order polynomials, computing symbolic eigenvalues analytically becomes intractable, thus only the numerical eigenvalues are trusted.

7.3 ROD treatment and explicit time integration

Following the same configuration considered above for the SB treatment. The analysis for the ROD-E and ROD-L^2 boundary treatment can be performed thanks to the reformulations carried out in section 6, which allows one to write easily the ROD correction in matrix form.

In this case, the global system for two cells is

$$\begin{aligned} M_1 \frac{\mathrm{d}\mathbf{u}_1}{\mathrm{d}t} &= K_1^s \mathbf{u}_1 - (K_1^R \mathbf{u}_1 - K_1^{\mathrm{ROD}} \mathbf{u}_1), \\ M_2 \frac{\mathrm{d}\mathbf{u}_2}{\mathrm{d}t} &= K_2^s \mathbf{u}_2 - (K_2^R \mathbf{u}_2 - K_2^L \mathbf{u}_1), \end{aligned} \tag{30}$$

where the left interface matrix that introduces the ROD polynomial correction is defined as

$$K_1^{\mathrm{ROD}} = \varphi_i^1(x_{1/2})(\varphi_j^1(x_{1/2}) - \alpha^{\mathrm{ROD}}\, \varphi_j^1(x_{1/2} + d)).$$

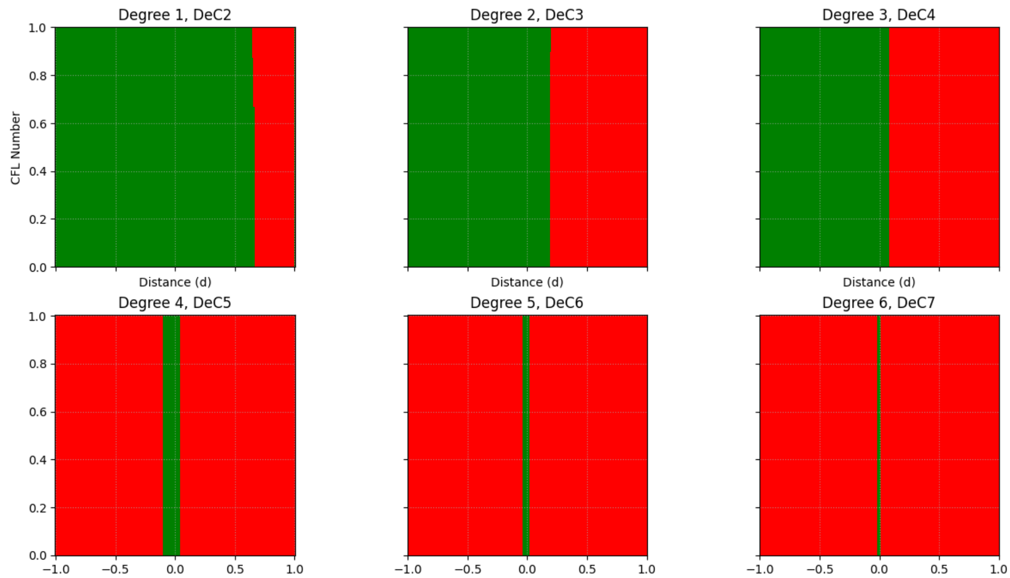

Figure 6: Stability region: ROD-E correction of the homogeneous Dirichlet condition for the explicit discontinuous Galerkin method. Green areas are stable, while red areas are unstable.

By varying α^{ROD}, different ROD approaches can be analyzed.

For the Euclidean distance used for the minimization problem,

$$\alpha^{\text{ROD-E}} = \frac{[\boldsymbol{\varphi}^1(x_{1/2})]^T[\boldsymbol{\varphi}^1(x_{1/2}+d)]}{\|\boldsymbol{\varphi}^1(x_{1/2}+d)\|_2^2}$$

and the stability regions are given in figure 6. Contrary to the SB correction, the ROD-E method with polynomial degree $p = 1,2,3$ shows stable areas for all values of $d \in [-1,0]$, with no constraint on the CFL. Instead, for $d \in [0,1]$, the method becomes unstable after a certain threshold $d > d_{max}$, which depends on the polynomial degree used to build the scheme, with the same d_{max} observed for the SB method. Although the ROD-E method shows better stability properties for $p = 1,2,3$ and distance $d \in [-1,0]$ than the SB method, for higher order polynomials $p = 4,5,6$, also the ROD-E method becomes unconditionally unstable after a certain threshold. Contrary to the SB method, the ROD-E method never shows a constraint on the CFL number, it is either stable under a classical CFL condition or it becomes unconditionally unstable after a certain distance threshold.

As done in section 4, we can analytically compute the eigenvalues of the ROD-E boundary discretized operator. The analytical eigenvalues associated with the boundary element with the $\mathbb{P}^1$ method are

$$\mu_{1,2}^{\text{ROD-E}} = \frac{-3d^2+5d-2 \pm \sqrt{9d^4-18d^3+13d^2-2d-2}}{2d^2-2d+1}.$$

Also in this case, the real part of the eigenvalues is negative when $d < 2/3$, which is the distance threshold after which the scheme becomes unconditionally unstable.

When taking the L^2 distance for the ROD minimization problem,

$$\alpha^{\text{ROD-}L^2} = \frac{[\boldsymbol{\varphi}^1(x_{1/2})]^T M^{-1}[\boldsymbol{\varphi}^1(x_{1/2}+d)]}{\|\boldsymbol{\varphi}^1(x_{1/2}+d)\|_{M^{-1}}^2},$$

and the stability regions are given in figure 7.

Contrary to the ROD-E method, the ROD-L^2 shows stable areas for all values of $d \in [-1,0]$, with no

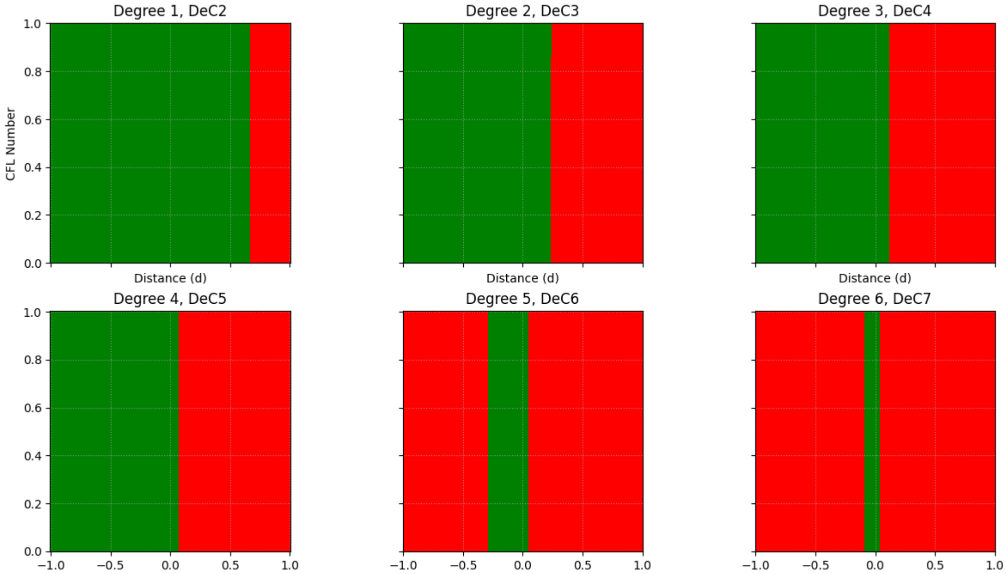

Figure 7: Stability region: ROD-L^2 correction of the homogeneous Dirichlet condition for the explicit discontinuous Galerkin method. Green areas are stable, while red areas are unstable.

constraint on the CFL with $p = 1, 2, 3, 4$, while for $p = 5, 6$ a constraint on the distance threshold appears. Instead, for $d \in [0, 1]$, the method becomes unstable after a certain threshold $d > d_{max}$, which depends on the polynomial degree used to build the scheme, as also observed for ROD-E. It is interesting to notice that by changing the distance function used in the ROD minimization problem, the stability region can be improved. As a matter of fact, for all values of $d \in [-1, 0]$, ROD-E was stable only for $p = 1, 2, 3$, while ROD-L^2 is also stable for $p = 4$.

The analytical eigenvalues of the ROD-L^2 $\mathbb{P}^1$ method are

$$\mu_{1,2}^{\text{ROD-}L^2} = \frac{-9d^2 + 12d - 4 \pm \sqrt{81d^4 - 108d^3 + 36d^2 + 12d - 8}}{2\left(3d^2 - 3d + 1\right)}$$

Again, the real part of the eigenvalues is negative when $d < 2/3$.

7.4 Shifted boundary treatment and implicit time integration

Due to the stability issues observed for embedded boundary methods with explicit time integration, it is interesting to analyze the effect of implicit time integration on the stability of the scheme. To give a practical example of the effect of implicit time integration, we consider the implicit Euler method. If we consider the implicit Euler time integration of the discontinuous Galerkin method with the SB polynomial correction, the global system reads

$$\begin{aligned} M_1 \frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\Delta t} &= K_1^s \mathbf{u}_1^{n+1} - (K_1^R \mathbf{u}_1^{n+1} - K_1^{SB} \mathbf{u}_1^{n+1}), \\ M_2 \frac{\mathbf{u}_2^{n+1} - \mathbf{u}_2^n}{\Delta t} &= K_2^s \mathbf{u}_2^{n+1} - (K_2^R \mathbf{u}_2^{n+1} - K_2^L \mathbf{u}_1^{n+1}), \end{aligned} \tag{31}$$

In this case, the system reduces to finding the eigenvalues of the following amplification matrix:

$$\begin{bmatrix} \mathbf{u}_1^{n+1} \\ \mathbf{u}_2^{n+1} \end{bmatrix} = \left[I - \Delta t \mathbf{M}^{-1} \mathbf{K}\right]^{-1} \begin{bmatrix} \mathbf{u}_1^n \\ \mathbf{u}_2^n \end{bmatrix} \tag{32}$$

where the global matrices are those defined in equation (28).

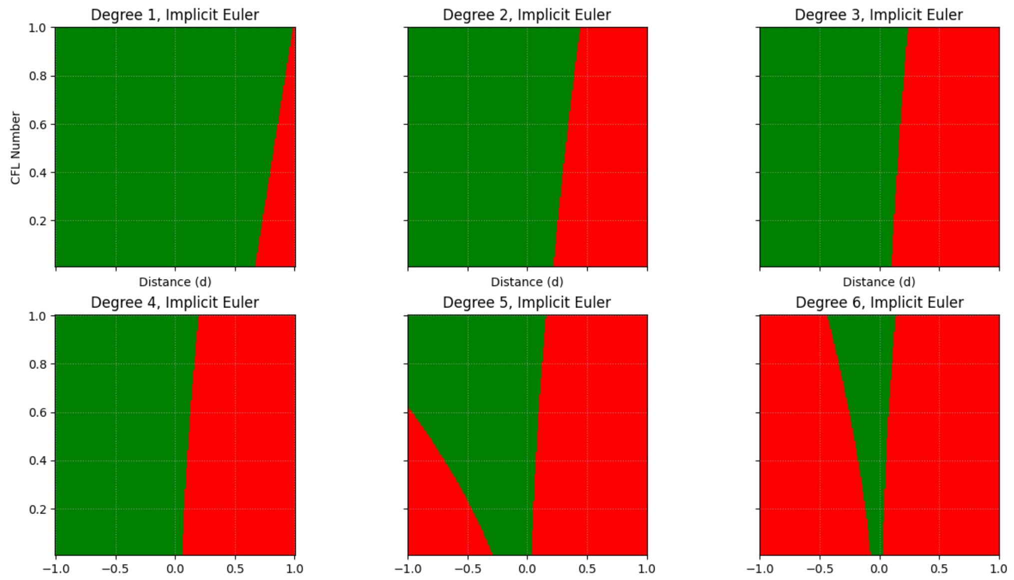

Figure 8: Stability region: SB correction of the homogeneous Dirichlet condition for the implicit discontinuous Galerkin method and CFL $\in [0, 1]$. Green areas are stable, while red areas are unstable.

The stability regions for a CFL $\in [0, 1]$ are shown in figure 8. It is interesting to notice that the implicit time integration is able to improve the stability regions of the SB method, at least for values of $d \in [-1, 0]$ and $p = 1, 2, 3, 4$. However, when going very high-order for $p = 5, 6$, instabilities start to arise even for implicit time integration and a CFL greater than a certain threshold is necessary to stabilize. In particular, stability for higher order cases imposes a lower bound on the CFL condition: CFL ≥ 0.7 for $\mathbb{P}^5$, and CFL ≥ 2.5 for $\mathbb{P}^6$. This is shown in figure 9, where the stability regions for CFL $\in [0, 10]$ are plotted.

7.5 ROD treatment and implicit time integration

The same procedure can be applied to study the effect of implicit time integration when dealing with ROD treatment. If we consider the implicit Euler time integration of the discontinuous Galerkin method with the ROD (either ROD-E or ROD-L^2) polynomial correction, the global system reads

$$\begin{aligned} M_1 \frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\Delta t} &= K_1^s \mathbf{u}_1^{n+1} - (K_1^R \mathbf{u}_1^{n+1} - K_1^{\mathrm{ROD}} \mathbf{u}_1^{n+1}), \\ M_2 \frac{\mathbf{u}_2^{n+1} - \mathbf{u}_2^n}{\Delta t} &= K_2^s \mathbf{u}_2^{n+1} - (K_2^R \mathbf{u}_2^{n+1} - K_2^L \mathbf{u}_1^{n+1}), \end{aligned} \tag{33}$$

In figure 10, the stability regions for ROD-E are shown considering CFL $\in [0, 10]$. Since, for values of $d \in [-1, 0]$, the ROD-E with $p = 1, 2, 3$ polynomials and explicit time integration is stable under standard CFL condition, we are more interested about the impact that the implicit time integration may have on higher order methods, i.e. with $p = 4, 5, 6$. However, it is interesting to note that with implicit time integration, the ROD-E method with $p = 4, 5, 6$ is stable only for a value of CFL greater than a certain threshold, which becomes higher when using higher polynomials. In particular, we need CFL ≥ 3 for $\mathbb{P}^4$, CFL ≥ 6 for $\mathbb{P}^5$, and CFL ≥ 9 for $\mathbb{P}^6$. This means that, although implicit time integration is able to improve the stability of the ROD-E method, it is not able to make it unconditionally stable for all values of $d \in [-1, 0]$ and for all polynomial degrees.

Similarly, also for the ROD-L^2 method with implicit time integration, the main focus of the analysis is to study whether it is possible to stabilize the method for very high-order polynomials. As a matter of fact, the ROD-L^2 method is already stable for all values of $d \in [-1, 0]$ and $p = 1, 2, 3, 4$, with explicit time integration. However, similar to the ROD-E method, even implicit time integration does not provide an

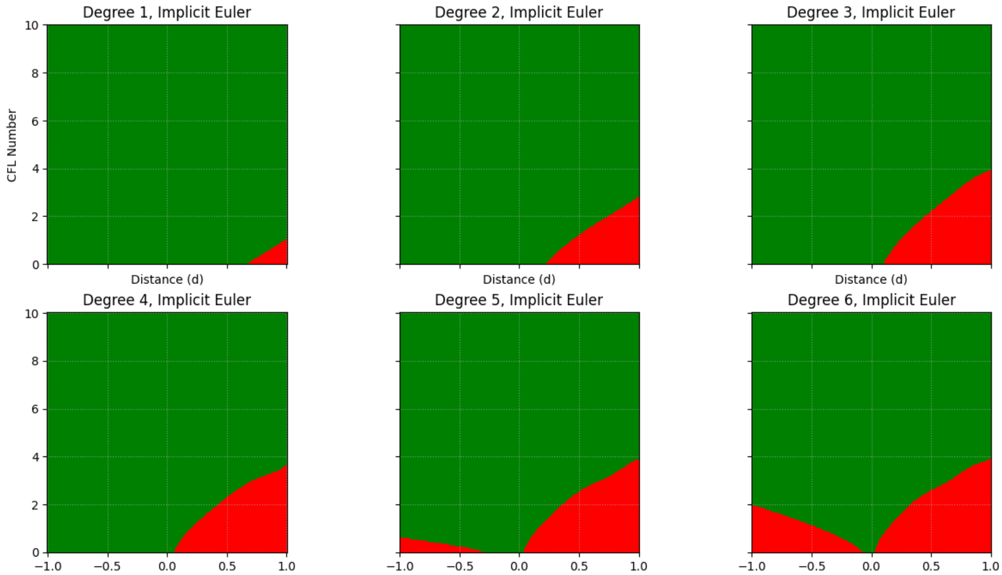

Figure 9: Stability region: SB correction of the homogeneous Dirichlet condition for the implicit discontinuous Galerkin method and CFL $\in [0, 10]$. Green areas are stable, while red areas are unstable.

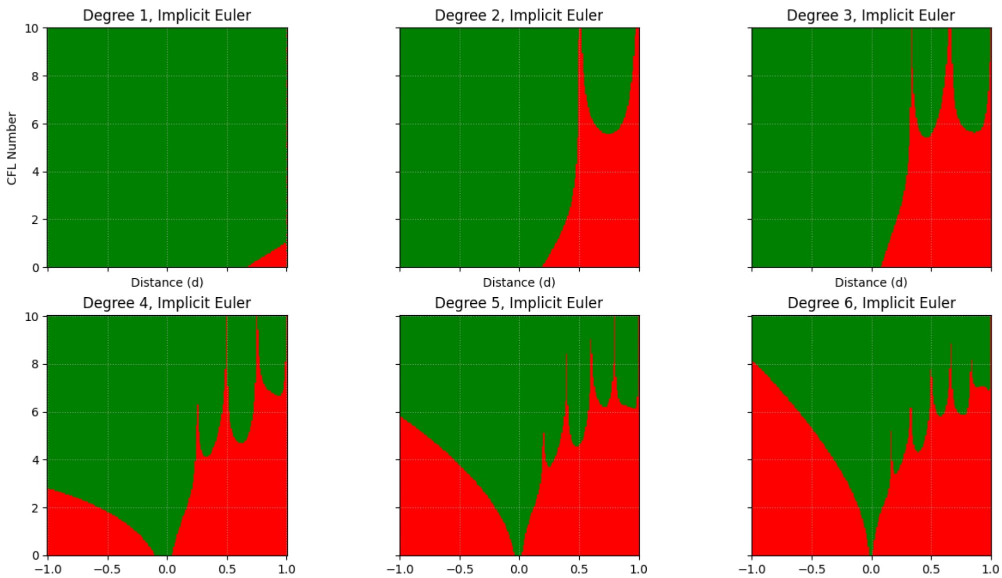

Figure 10: Stability region: ROD-E correction of the homogeneous Dirichlet condition for the implicit discontinuous Galerkin method and CFL $\in [0, 10]$. Green areas are stable, while red areas are unstable.

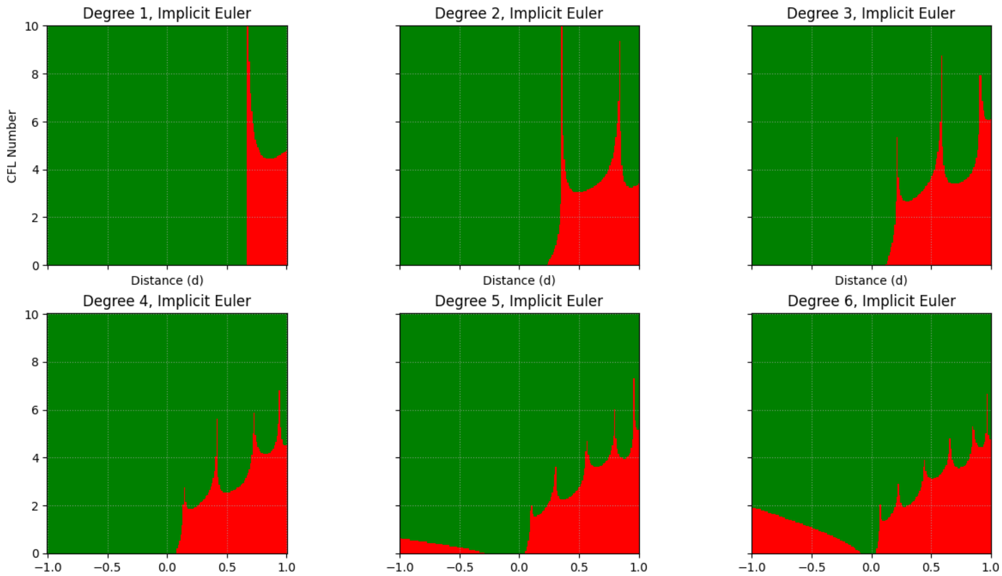

Figure 11: Stability region: ROD-L^2 correction of the homogeneous Dirichlet condition for the implicit discontinuous Galerkin method and CFL $\in [0, 10]$. Green areas are stable, while red areas are unstable.

unconditionally stable scheme for all values of CFL $\in [0, 1]$. On the contrary, as shown in figure 11, for $p = 5, 6$ a minimum CFL is always needed to have a stable method. Although better than ROD-E, we still need CFL ≥ 0.7 for $\mathbb{P}^5$, and CFL ≥ 2 for $\mathbb{P}^6$ to have a stable scheme.

8. Numerical experiments

In this section, we focus on three sets of numerical experiments to validate the theoretical findings of section 7. In section 8.1, the one-dimensional linear advection equation is considered to validate the stability and accuracy of the method for different embedded configurations. We test all the methods considered in the analysis, i.e. SB, ROD-E and ROD-L^2 corrections, with explicit and implicit time integration. In section 8.2, the two-dimensional linear advection equation is studied to show the stability of the method also in this more general case. In particular, to validate that the one-dimensional analysis also holds in 2D, we consider that the same distance value d is used for all the quadrature points of the edge integral, and we show that the stability regions are perfectly in line with those found in the 1D analysis. Finally, in section 8.3, the nonlinear system of Euler equations with embedded boundary is considered to show that all the CFL conditions found in the analysis also hold for this more complex case. Since only stationary solutions are considered, all numerical simulations are run until when the time residual reaches machine precision, which is set around 10^{-13} in all the tests.

8.1 Linear advection equation in 1D

For the numerical experiments, the linear advection equation with source term is set up to preserve a stationary manufactured solution given by the following exact solution $u_{ex}(x) = 0.1\sin(\pi x)$, which is also taken as initial condition. The source term is computed as

$$\partial_t u_{ex}(x) + \partial_x u_{ex}(x) = s(x), \quad \text{that gives} \quad s(x) = 0.1\pi\cos(\pi x).$$

The computational domain is $\Omega = [0, 2]$ with a Dirichlet boundary condition imposed at the left boundary. No conditions are imposed at the right boundary due to the pure upwind flux. A set of uniform meshes with N_e elements is considered.

The SB, ROD-E and ROD-L^2 polynomial corrections are imposed considering different distance values

Table 1: Linear advection equation: L_2 errors $\|u - u_{ex}\|_2$ and estimated order of accuracy (EOA) with SB polynomial correction and explicit time integration for polynomials $\mathbb{P}^p$. CFL and d are chosen to consider the worst case scenario, i.e. the furthest possible $d \in [-1, 0]$ and the associated maximum CFL.

	$\mathbb{P}^1$, CFL=6E-1, $d = -1.00$		$\mathbb{P}^2$, CFL=2E-1, $d = -1.00$		$\mathbb{P}^3$, CFL=5E-2, $d = -1.00$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
20	5.98E-04	–	2.68E-03	–	2.59E-05	–
40	1.27E-04	2.23	3.37E-04	3.00	8.12E-07	4.99
80	3.02E-05	2.07	4.22E-05	3.00	2.54E-08	5.00
160	7.45E-06	2.01	5.28E-06	3.00	7.65E-10	5.04
	$\mathbb{P}^4$, CFL=8E-3, $d = -1.00$		$\mathbb{P}^5$, CFL=8E-2, $d = -0.25$		$\mathbb{P}^6$, CFL=3E-1, $d = -0.07$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5	2.58E-02	–	4.80E-05	–	4.95E-07	–
10	8.89E-04	4.86	3.96E-07	6.92	4.52E-09	6.77
20	2.84E-05	4.96	3.13E-09	6.98	3.67E-11	6.94
40	8.94E-07	4.99	2.46E-11	6.99	2.91E-13	6.97

Table 2: Linear advection equation: L_2 errors $\|u - u_{ex}\|_2$ and estimated order of accuracy (EOA) with the ROD-E correction and explicit time integration for polynomials $\mathbb{P}^p$. CFL and d are chosen to consider the worst case scenario, i.e. the furthest possible $d \in [-1, 0]$ and the associated maximum CFL.

	$\mathbb{P}^1$, CFL=1.0, $d = -1.00$		$\mathbb{P}^2$, CFL=1.0, $d = -1.00$		$\mathbb{P}^3$, CFL=1.0, $d = -1.00$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
20	5.08E-04	–	1.45E-03	–	2.28E-05	–
40	1.21E-04	2.07	1.81E-04	2.99	7.16E-07	4.99
80	2.98E-05	2.01	2.26E-05	2.99	2.23E-08	5.00
160	7.43E-06	2.00	2.83E-06	2.99	6.28E-10	5.14
	$\mathbb{P}^4$, CFL=1.0, $d = -0.10$		$\mathbb{P}^5$, CFL=1.0, $d = -0.04$		$\mathbb{P}^6$, CFL=1.0, $d = -0.015$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5	4.43E-05	–	6.72E-07	–	4.88E-08	–
10	1.73E-06	4.67	6.07E-09	6.78	4.55E-10	6.74
20	5.69E-08	4.92	6.11E-11	6.63	3.70E-12	6.94
40	1.80E-09	4.98	7.59E-13	6.33	7.47E-15	8.95

d of the boundary from the left interface of the first cell. When $d = 0$, the boundary coincides with the left interface of the first cell, and therefore the correction is not applied. This is the case of standard fitted boundary conditions. Since all studied methods present more instabilities when $d \in [0, \Delta x]$, the distances considered in the numerical experiments will be taken as $d \in [-\Delta x, 0]$, meaning with the real boundary point collocated outside the first mesh element. For simplicity, in convergence tables, the distance is reported in non-dimensional form, i.e. $d/\Delta x$. Stability results obtained through the analysis are perfectly corroborated by the numerical results for all configurations of d, and for both explicit and implicit time integration.

In table 1, we present the numerical results obtained with the SB correction and explicit time integration. We consider the maximum distance that makes the scheme stable under the associated maximum CFL value. As also mentioned above, for $d < 0$, it is possible to reach the maximum distance of $d = -1$ for polynomials $p = 1, 2, 3, 4$ with a strict CFL condition that becomes stricter as p increases. For $p = 5, 6$, the scheme is unconditionally unstable for $d = -1$, and the maximum distance value for such high-order polynomials is much lower: $d = -0.25$ for $\mathbb{P}^5$, and $d = -0.07$ for $\mathbb{P}^6$.

In table 2, we present the numerical results obtained with the ROD-E correction and explicit time integration. Again, we consider the greater distance that makes the scheme stable under the associated maximum CFL value. As also mentioned above, for $d < 0$, it is possible reach the maximum distance of $d = -1$ for polynomials $p = 1, 2, 3$ with no condition on the CFL. For $p = 4, 5, 6$, the scheme is unconditionally unstable for $d = -1$, and the maximum distance value for such polynomials is much lower: $d = -0.10$ for $\mathbb{P}^4$, $d = -0.04$ for $\mathbb{P}^5$, and $d = -0.015$ for $\mathbb{P}^6$. Contrary to the SB method, there are no cases where it is possible to stabilize the ROD-E method by lowering the CFL. ROD-E is either stable under a classical CFL condition or unconditionally unstable, and this is what happens for higher order polynomials after a certain distance threshold.

Table 3: Linear advection equation: L_2 errors $||u - u_{ex}||_2$ and estimated order of accuracy (EOA) with the ROD-L^2 correction and explicit time integration for polynomials $\mathbb{P}^p$. CFL and d are chosen to consider the worst case scenario, i.e. the furthest possible $d \in [-1, 0]$ and the associated maximum CFL.

	$\mathbb{P}^1$, CFL=1.0, $d = -1.00$		$\mathbb{P}^2$, CFL=1.0, $d = -1.00$		$\mathbb{P}^3$, CFL=1.0, $d = -1.00$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
20	5.57E-04	–	1.92E-03	–	1.19E-05	–
40	1.24E-04	2.16	2.40E-04	2.99	3.74E-07	4.99
80	3.00E-05	2.04	3.01E-05	2.99	1.17E-08	4.99
160	7.44E-06	2.01	3.76E-06	2.99	3.68E-10	4.99
	$\mathbb{P}^4$, CFL=1.0, $d = -1.00$		$\mathbb{P}^5$, CFL=1.0, $d = -0.25$		$\mathbb{P}^6$, CFL=1.0, $d = -0.05$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5	1.93E-02	–	2.42E-05	–	1.40E-07	–
10	6.46E-04	4.90	1.98E-07	6.93	1.27E-09	6.78
20	2.05E-05	4.97	1.57E-09	6.98	1.03E-11	6.94
40	6.44E-07	4.99	1.23E-11	6.99	8.26E-14	6.95

Table 4: Linear advection equation: L_2 errors $||u - u_{ex}||_2$ and estimated order of accuracy (EOA) with SB polynomial correction and implicit time integration for polynomials $\mathbb{P}^p$. It is always considered the case where $d = -1.0$ with the smallest CFL ≥ 1 value to make the scheme stable.

	$\mathbb{P}^1$, CFL=1.0		$\mathbb{P}^2$, CFL=1.0		$\mathbb{P}^3$, CFL=1.0	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
20	5.98E-04	–	2.68E-03	–	2.59E-05	–
40	1.27E-04	2.23	3.37E-04	3.00	8.12E-07	4.99
80	3.02E-05	2.07	4.22E-05	3.00	2.54E-08	5.00
160	7.45E-06	2.01	5.28E-06	3.00	7.65E-10	5.04
	$\mathbb{P}^4$, CFL=1.0		$\mathbb{P}^5$, CFL=1.0		$\mathbb{P}^6$, CFL=2.5	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5	2.58E-02	–	3.48E-03	–	2.04E-03	–
10	8.89E-04	4.86	2.85E-05	6.93	1.76E-05	6.85
20	2.84E-05	4.96	2.26E-07	6.98	1.41E-07	6.96
40	8.94E-07	4.99	1.77E-09	6.99	1.18E-09	6.90

In table 3, we present the numerical results obtained with the ROD-L^2 correction and explicit time integration. Again, we consider the greater distance that makes the scheme stable under the associated maximum CFL value. As also mentioned above, for $d < 0$, it is possible reach the maximum distance of $d = -1$ for polynomials $p = 1, 2, 3, 4$ with no condition on the CFL, which means that we have the method $p = 4$ that is stable contrary to ROD-E. For $p = 5, 6$, the scheme is unconditionally unstable for $d = -1$, and the maximum distance value for such polynomials is much lower: $d = -0.25$ for $\mathbb{P}^5$, and $d = -0.05$ for $\mathbb{P}^6$. Also in this case, there are no cases where it is possible to stabilize the ROD-L^2 method by lowering the CFL. ROD-L^2 is either stable under a classical CFL condition or unconditionally unstable, and this is what happens for higher order polynomials after a certain distance threshold, similarly to what was observed for ROD-E.

In table 4, we present the numerical results obtained with the SB correction and implicit time integration. For implicit time integration, it is always possible to obtain a stable scheme for $d = -1$. However, as it was studied above, for some cases it necessary to have a CFL higher than a certain threshold. For the SB method with $p = 1, 2, 3, 4$ there are no lower bound conditions on the CFL, and it is therefore possible to use any CFL value and the scheme will stays stable. However, for $p = 5$ it is necessary to have CFL ≥ 0.6, while for $p = 6$ we need a CFL ≥ 2.5. This means that even with implicit time integration, the higher order methods still have some parasitic modes that need higher CFL values to be damped out.

In table 5, we present the numerical results obtained with the ROD-E correction and implicit time integration. As for the SB method, for implicit time integration, it is always possible to obtain a stable scheme for $d = -1$ but for some cases the considered method has a lower bound condition on the CFL. For the ROD-E method with $p = 1, 2, 3$ there are no conditions on the CFL, and it is therefore possible to use any CFL value and the scheme will stays stable. This was expected since these schemes were also stable under standard CFL condition with explicit time integration. However, for $p = 4, 5, 6$ a lower

Table 5: Linear advection equation: L_2 errors $||u - u_{ex}||_2$ and estimated order of accuracy (EOA) with ROD-E correction and implicit time integration for polynomials $\mathbb{P}^p$. It is always considered the case where $d = -1.0$ with the smallest CFL ≥ 1 value to make the scheme stable.

	$\mathbb{P}^1$, CFL=1.0		$\mathbb{P}^2$, CFL=1.0		$\mathbb{P}^3$, CFL=1.0	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
20	5.08E-04	–	1.45E-03	–	2.28E-05	–
40	1.21E-04	2.07	1.81E-04	2.99	7.16E-07	4.99
80	2.98E-05	2.01	2.26E-05	2.99	2.23E-08	5.00
160	7.43E-06	2.00	2.83E-06	2.99	6.28E-10	5.14
	$\mathbb{P}^4$, CFL=3.0		$\mathbb{P}^5$, CFL=6.0		$\mathbb{P}^6$, CFL=9.0	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5	2.77E-02	–	2.50E-02	–	2.33E-02	–
10	1.10E-03	4.65	2.08E-04	6.91	2.17E-04	6.74
20	3.62E-05	4.92	1.65E-06	6.97	1.77E-06	6.93
40	1.15E-06	4.98	1.29E-08	6.99	1.44E-08	6.93

Table 6: Linear advection equation: L_2 errors $||u - u_{ex}||_2$ and estimated order of accuracy (EOA) with ROD-L^2 correction and implicit time integration for polynomials $\mathbb{P}^p$. It is always considered the case where $d = -1.0$ with the smallest CFL ≥ 1 value to make the scheme stable.

	$\mathbb{P}^1$, CFL=1.0		$\mathbb{P}^2$, CFL=1.0		$\mathbb{P}^3$, CFL=1.0	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
20	5.57E-04	–	1.92E-03	–	1.19E-05	–
40	1.24E-04	2.16	2.40E-04	2.99	3.74E-07	4.99
80	3.00E-05	2.04	3.01E-05	2.99	1.17E-08	4.99
160	7.44E-06	2.01	3.76E-06	2.99	3.68E-10	4.99
	$\mathbb{P}^4$, CFL=1.0		$\mathbb{P}^5$, CFL=1.0		$\mathbb{P}^6$, CFL=2.0	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5	1.93E-02	–	2.06E-03	–	1.53E-03	–
10	6.46E-04	4.90	1.67E-05	6.94	1.29E-05	6.88
20	2.05E-05	4.97	1.32E-07	6.98	1.03E-07	6.97
40	6.44E-07	4.99	1.03E-09	6.99	8.09E-10	6.99

bound on the CFL is necessary to ensure stability: for $p = 4$ it is necessary to have CFL ≥ 3, for $p = 5$ we need CFL ≥ 6, while for $p = 6$ we have CFL ≥ 9.

In table 6, we present the numerical results obtained with the ROD-L^2 correction and implicit time integration. Also in this case, it is always possible to obtain a stable scheme for $d = -1$ but for some cases the considered method has a lower bound condition on the CFL. For the ROD-L^2 method with $p = 1, 2, 3, 4$ there a no conditions on the CFL, and it is therefore possible to use any CFL value and the scheme will stays stable. Once again, if we study the coupling of higher order methods with implicit time integration, we show that for $p = 5$ it is necessary to have CFL ≥ 0.7, while for $p = 6$ we need CFL ≥ 2. Although less strong than in the ROD-E case, also the ROD-L^2 method, for very high-order polynomials, shows some parasitic modes that need higher CFL values to be stabilized.

It can be noticed that for all three methods, the expected order of accuracy is reached for all configurations of d and for both explicit and implicit time integration, which means that the stability issues do not affect the accuracy of the method when the scheme is stable. Moreover, all methods experience a superconvergent behaviour for $p = 3, 5$, achieving one additional order of accuracy. The same is also observed for implicit time integration in the following. Probably this superconvergence is related to the time-independent nature of boundary conditions, and it is indeed no longer present when considering test cases with time-dependent conditions, which are omitted for brevity.

8.2 Linear advection equation in 2D

In order to further investigate the stability properties of the methods, we also consider the linear advection equation in two dimensions. We consider a square domain $[0, 2]^2$ with a uniform Cartesian grid and positive advection velocities in both directions, i.e. $\partial_t u + c_x \partial_x u + c_y \partial_y u = s(x, y)$ with $c_x = c_y = 1$. Similar to the 1D case, we consider a manufactured solution of the form $u_{ex}(x, y) = 0.1 \sin(x + y)$

Table 7: Linear advection equation in two dimensions: L_2 errors $||u - u_{ex}||_2$ and estimated order of accuracy (EOA) with SB, ROD-E, and ROD-L^2 polynomial correction and explicit time integration for polynomials $\mathbb{P}^p$. CFL and d are chosen to consider the worst case scenario, i.e. $d = -1$ and the associated maximum CFL.

SB	$\mathbb{P}^1$, CFL=6E-1, $d = -1$		$\mathbb{P}^2$, CFL=1.5E-1, $d = -1$		$\mathbb{P}^3$, CFL=4E-2, $d = -1$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5×5	1.89E-02	–	6.92E-03	–	5.67E-04	–
10×10	5.06E-03	1.90	8.18E-04	3.08	4.08E-05	3.80
20×20	1.36E-03	1.89	9.92E-05	3.04	2.70E-06	3.91
40×40	3.54E-04	1.94	1.22E-05	3.02	1.73E-07	3.96
80×80	9.04E-05	1.97	1.51E-06	3.01	1.01E-08	4.09

ROD-E	$\mathbb{P}^1$, CFL=1.0, $d = -1$		$\mathbb{P}^2$, CFL=1.0, $d = -1$		$\mathbb{P}^3$, CFL=1.0, $d = -1$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5×5	1.36E-02	–	3.85E-03	–	7.69E-05	–
10×10	3.90E-03	1.80	4.52E-04	3.09	2.16E-06	5.15
20×20	1.06E-03	1.87	5.41E-05	3.06	5.91E-08	5.19
40×40	2.78E-04	1.93	6.60E-06	3.03	3.76E-09	3.97
80×80	7.10E-05	1.96	8.14E-07	3.01	–	–

ROD-L^2	$\mathbb{P}^1$, CFL=1.0, $d = -1$		$\mathbb{P}^2$, CFL=1.0, $d = -1$		$\mathbb{P}^3$, CFL=1.0, $d = -1$		$\mathbb{P}^4$, CFL=1.0, $d = -1$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
5×5	1.26E-02	–	5.01E-03	–	4.01E-04	–	8.47E-05	–
10×10	3.67E-03	1.77	5.91E-04	3.08	2.88E-05	3.80	2.53E-06	5.06
20×20	1.01E-03	1.86	7.12E-05	3.05	1.92E-06	3.90	7.68E-08	5.04
40×40	2.63E-04	1.93	8.72E-06	3.02	1.24E-07	3.94	2.36E-09	5.02
80×80	6.72E-05	1.96	1.08E-06	3.01	8.30E-09	3.90	–	–

with the associated source term $s(x, y)$. We consider embedded Dirichlet boundary conditions on the left and bottom boundaries, while we consider fitted Neumann boundary conditions on the right and top boundaries. Since the goal of this section is to prove that the one-dimensional stability results are also valid in multiple dimensions, we consider that the embedded boundaries have no curvature, and are located at the same distance from the left and bottom boundaries in both directions, i.e. $d_x = d_y = d$. The algorithmic implementation of the methods in multiple dimensions is carried out by performing a tensor product on both basis functions and quadrature points used in the 1D algorithm, and it is therefore straightforward for Cartesian grids. The time step is computed based on the CFL condition, which considers the advection velocity in both directions: $\Delta t = CFL^p_{\max}\, CFL/(\frac{c_x}{\Delta x} + \frac{c_y}{\Delta y})$. In one dimension ROD-E coincides with the original ROD method, but in multiple dimensions it does not. There, ROD-E and ROD-L^2 do not couple all boundary conditions of a boundary cell; rather, they apply local corrections at each quadrature point. Therefore, this section is also useful to show that the stability properties and accuracy of ROD-E and ROD-L^2 are not affected by the coupling of multiple boundary conditions in multiple dimensions, and that the stability results obtained in one dimension are still valid in multiple dimensions. As shown in table 7, the same stability constraints are observed for the three methods, and the expected order of accuracy is reached for all configurations. In this case, for conciseness we only focus on the worst case scenario, i.e. $d = -1$ and the associated maximum CFL value. For the SB method, we consider only $p = 1, 2, 3$ since for higher order polynomials the CFL restriction becomes too severe. For ROD-E and ROD-L^2 we consider the polynomials for which it is possible to run simulations with $d = -1$ without additional CFL restrictions ($CFL \leq 1$). Hence, for ROD-E we consider $p = 1, 2, 3$, while for ROD-L^2 we consider $p = 1, 2, 3, 4$. For higher order polynomials the scheme is unconditionally unstable for $d = -1$ and therefore it is not possible to run simulations with explicit time integration.

8.3 Euler equations for compressible gasdynamics in 1D

To further validate the stability analysis performed in this work, a set of numerical experiments for the one-dimensional Euler equations for compressible gasdynamics is carried out. In particular, we also show here that the stability restrictions found for the linear advection equation also hold for this nonlinear system of conservation laws. For simplicity, we discuss here all polynomial corrections with explicit time integration to corroborate that, in the nonlinear case, the same stability restrictions observed for the linear advection equation hold.

Table 8: Euler equations for compressible gasdynamics in 1D: L_2 errors for the momentum $\|(\rho u) - (\rho u)_{ex}\|_2$ and estimated order of accuracy (EOA) with SB, ROD-E, and ROD-L^2 polynomial corrections and explicit time integration by varying the distance d for polynomials $\mathbb{P}^p$. CFL and d are chosen to consider the worst case scenario, i.e. $d = -1$ and the associated maximum CFL.

SB	$\mathbb{P}^1$, CFL=6E-1, $d = -1$		$\mathbb{P}^2$, CFL=1.5E-1, $d = -1$		$\mathbb{P}^3$, CFL=4E-2, $d = -1$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
10	3.10E-03	–	1.95E-02	–	7.25E-04	–
20	5.22E-04	2.57	2.46E-03	2.98	1.87E-05	5.27
40	1.12E-04	2.21	3.08E-04	2.99	3.67E-07	5.67
80	2.72E-05	2.04	3.85E-05	3.00	1.09E-08	5.06

ROD-E	$\mathbb{P}^1$, CFL=1.0, $d = -1$		$\mathbb{P}^2$, CFL=1.0, $d = -1$		$\mathbb{P}^3$, CFL=1.0, $d = -1$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
10	2.12E-03	–	1.03E-02	–	5.14E-04	–
20	4.63E-04	2.19	1.30E-03	2.98	6.75E-06	6.25
40	1.10E-04	2.06	1.63E-04	2.99	3.57E-07	4.24
80	2.73E-05	2.01	2.04E-05	3.00	8.06E-08	2.14

ROD-L^2	$\mathbb{P}^1$, CFL=1.0, $d = -1$		$\mathbb{P}^2$, CFL=1.0, $d = -1$		$\mathbb{P}^3$, CFL=1.0, $d = -1$		$\mathbb{P}^4$, CFL=0.95, $d = -1$	
N_e,	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA	L_2 error	EOA
10	2.62E-03	–	1.38E-02	–	3.66E-04	–	5.76E-04	–
20	5.00E-04	2.38	1.74E-03	2.98	1.14E-05	5.00	1.83E-05	4.97
40	1.13E-04	2.15	2.18E-04	2.99	3.62E-07	4.97	5.82E-07	4.97
80	2.74E-05	2.04	2.73E-05	3.00	1.56E-08	4.53	1.84E-08	4.98

The Euler equations can be written in the conservative form $\partial_t \mathbf{q} + \partial_x \mathbf{f}(\mathbf{q}) = \mathbf{s}(x,t)$ with the following definitions: $\mathbf{q} = [\rho, \rho u, \rho E]^T$, $\mathbf{f}(\mathbf{q}) = [\rho u, \rho u^2 + p, \rho H u]^T$, where ρ is the density, u is the velocity, p is the pressure and $E = e + u^2/2$ is the specific total energy, e being the specific internal energy. Finally, the specific total enthalpy is $H = h + p/\rho$, where $h = e + p/\rho$ is the specific enthalpy. We also define ρu as the momentum. The system is closed by the equation of state for an ideal gas $p = (\gamma - 1)\rho e$, where $\gamma = 1.4$ is the ratio of specific heats. As in the previous section, we consider a stationary manufactured solution on the domain $\Omega = [0, 2]$, given by the following exact solution in terms of the primitive variables $\mathbf{v} = [\rho, u, p]^T$: $\mathbf{v}_{ex}(x) = [1.0, 1.0 + 0.1\sin(\pi x), 1.0]^T$, which is also taken as initial condition. The source term associated to this manufactured solution is computed and added to the right-hand side of the Euler equations.

In table 8, the numerical results obtained with SB, ROD-E, and ROD-L^2 polynomial corrections and explicit time integration are shown. The embedded boundary condition is imposed at the left boundary, while a classical fitted condition for the right boundary is considered. We present in table 8 the convergence studies with the worst case scenario for embedded configurations, i.e. $d = -1$ and the associated maximum CFL, to show that almost the same stability restrictions are observed for this nonlinear system of conservation laws. For simplicity, for the SB correction, we only show the results for $p = 1, 2, 3$, since for higher order polynomials the CFL condition is intractable with explicit time integration, and it is not possible to reach the maximum distance. Notice that also in the nonlinear case, the accuracy of the method is always preserved when the scheme is stable under the same CFL condition as for the linear advection equation. Concerning the ROD-E and ROD-L^2 corrections, we show the results for the configurations where it is possible to reach the maximum distance with explicit time integration, which means that for ROD-E we show the results for $p = 1, 2, 3$, while for ROD-L^2 we show the results for $p = 1, 2, 3, 4$. We can notice that also in the nonlinear case, the ROD-L^2 is stable even for $p = 4$ with explicit time integration, while the ROD-E is not stable for $p = 4$ and higher order polynomials, as it was observed for the linear advection equation. Only for a few configurations the nonlinear system appears to have a slightly more restrictive stability condition than the linear advection equation, which is reasonable due to the more complex nature of the Euler equations. It can be noticed that the same superconvergence behavior observed for the linear advection equation with $\mathbb{P}^3$ is also present here for certain embedded configurations, probably due to the time-independent nature of the manufactured solution.

9. Conclusions

In this paper, we have presented a unified framework to perform a stability analysis for embedded boundary treatments coupled with DG methods for hyperbolic equations. The numerical study was performed by visualizing the eigenspectrum of the discretized operators for a simplified system, and has provided critical insights into the stability properties of the considered geometrically unfitted approaches. To perform this study, it was necessary to write all embedded methods in matrix form, to allow for a straightforward integration into the global discretized operators. We showed for the first time that the ROD method can be recast as a polynomial correction that only depends on the basis function values on the real boundary and on the computational one, similarly to the one obtained for the SB method [15]. For all considered embedded treatments, it is interesting to notice that the stability region is not symmetric when considering external ($d < 0$) or internal ($d > 0$) embedded boundaries. The findings of the analysis show that the ROD polynomial correction, when discarding the element crossed by the embedded boundary (i.e. for $d < 0$ configurations), does not introduce any restriction on the CFL for polynomial degree $p \leq 3$ for the ROD-E version, and $p \leq 4$ for the ROD-L^2 version. This is an advantage with respect to the SB method, which shows a restriction on the CFL that becomes stricter as the polynomial degree increases, even for low polynomial degrees. Instead, for $d > 0$, all considered methods become unconditionally unstable beyond a critical distance $d > d_{max}(p)$, where d_{max} decreases when increasing the polynomial degree. In the second part of the analysis, we studied the impact of implicit time integration to understand whether it is possible to overcome the instabilities that arise with explicit methods. In all cases, we observed improvements in the stability properties of the embedded methods with implicit time integration. Moreover, for several configurations it is possible to obtain an unconditionally stable scheme. However for very high-order methods, several configurations present lower bound conditions on the CFL: i.e. the CFL number should be high enough to stabilize the method. This is probably related to the fact that, even for implicit time integration, there are still some parasitic modes that make the method unstable and need larger CFL values to be stabilized. The high-order convergence slopes were recovered for all stable configurations (d, CFL), and instabilities were observed precisely where the analysis predicted them. Furthermore, the same stability regions found for the one-dimensional linear advection equation are also valid for two-dimensional problems and for the nonlinear Euler equations for compressible gasdynamics, which means that the stability analysis performed in this work is not only valid for linear scalar equations but also for nonlinear systems of conservation laws. Future work will extend this stability analysis to hyperbolic equations discretized by stabilized continuous Galerkin to investigate the impact of these embedded methods on a different finite element framework. Further studies concerning the impact of curvature on the stability of embedded methods are also planned.

References

[1] Zhijian J Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, et al. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[2] Francesco Bassi and Stefano Rebay. High-order accurate discontinuous finite element solution of the 2d euler equations. *Journal of computational physics*, 138(2):251–285, 1997.

[3] Per-Olof Persson and Jaime Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 949, 2009.

[4] Alexander Coppeans, Krzysztof Fidkowski, and Joaquim R Martins. Aerodynamic shape optimization with curved mesh adaptation. In *AIAA SCITECH 2025 Forum*, page 0782, 2025.

[5] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[6] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*,

37(1):239–261, 2005.

[7] Anita Hansbo and Peter Hansbo. An unfitted finite element method, based on nitsche's method, for elliptic interface problems. *Computer methods in applied mechanics and engineering*, 191(47-48):5537–5552, 2002.

[8] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2):457–492, 1999.

[9] Marsha Berger and Michael Aftosmis. Progress towards a cartesian cut-cell method for viscous compressible flow. In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1301, 2012.

[10] Armando Coco and Giovanni Russo. Finite-difference ghost-point multigrid methods on cartesian grids for elliptic problems in arbitrary domains. *Journal of Computational Physics*, 241:464–501, 2013.

[11] Sirui Tan and Chi-Wang Shu. Inverse lax-wendroff procedure for numerical boundary conditions of conservation laws. *Journal of Computational Physics*, 229(21):8144–8166, 2010.

[12] Nabil M Atallah, Claudio Canuto, and Guglielmo Scovazzi. The high-order shifted boundary method and its analysis. *Computer Methods in Applied Mechanics and Engineering*, 394:114885, 2022.

[13] Andrew Giuliani. A two-dimensional stabilized discontinuous galerkin method on curvilinear embedded boundary grids. *SIAM Journal on Scientific Computing*, 44(1):A389–A415, 2022.

[14] Alex Main and Guglielmo Scovazzi. The shifted boundary method for embedded domain computations. part i: Poisson and stokes problems. *Journal of Computational Physics*, 372:972–995, 2018.

[15] Mirco Ciallella, Elena Gaburro, Marco Lorini, and Mario Ricchiuto. Shifted boundary polynomial corrections for compressible flows: high order on curved domains using linear meshes. *Applied Mathematics and Computation*, 441:127698, 2023.

[16] Jens Visbech, Allan P Engsig-Karup, and Mario Ricchiuto. A spectral element solution of the poisson equation with shifted boundary polynomial corrections: influence of the surrogate to true boundary mapping and an asymptotically preserving robin formulation. *Journal of Scientific Computing*, 102(1):11, 2025.

[17] Mirco Ciallella, Mario Ricchiuto, Renato Paciorri, and Aldo Bonfiglioli. Extrapolated discontinuity tracking for complex 2d shock interactions. *Computer Methods in Applied Mechanics and Engineering*, 391:114543, 2022.

[18] Walter Boscheri and Mirco Ciallella. High order treatment of moving curved boundaries: Arbitrary-lagrangian-eulerian methods with a shifted boundary polynomial correction. *Journal of Computational Physics*, 539:114215, 2025.

[19] Ricardo Costa, Stéphane Clain, Raphaël Loubère, and Gaspar J Machado. Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection–diffusion equation with dirichlet condition. *Applied Mathematical Modelling*, 54:752–767, 2018.

[20] Ricardo Costa, João M Nóbrega, Stéphane Clain, Gaspar J Machado, and Raphaël Loubère. Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries. *International Journal for Numerical Methods in Engineering*, 117(2):188–220, 2019.

[21] Milene Santos, Adérito Araújo, Sílvia Barbeiro, Stéphane Clain, Ricardo Costa, and Gaspar J Machado. Very high-order accurate discontinuous galerkin method for curved boundaries with polygonal meshes. *Journal of Scientific Computing*, 100(3):66, 2024.

[22] Mirco Ciallella, Stephane Clain, Elena Gaburro, and Mario Ricchiuto. Very high order treatment of embedded curved boundaries in compressible flows: Ader discontinuous galerkin with a space-time reconstruction for off-site data. *Computers & Mathematics with Applications*, 175:1–18, 2024.

[23] Alok Dutt, Leslie Greengard, and Vladimir Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.