Shock Capturing Techniques for hp -Adaptive Finite Elements

Alba Hierro^{a,b}, Santiago Badia^{a,b}, Pavel Kus^c

^a*Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.*

^b*Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Parc Mediterrani de la Tecnologia, UPC, Esteve Terradas 5, 08860 Castelldefels, Spain ([sbadia](mailto:sbadia@cimne.upc.edu),[ahierro](mailto:ahierro@cimne.upc.edu))@cimne.upc.edu).*

^c*Institute of Mathematics, Academy of Sciences of the Czech Republic (kus@math.cas.cz)*

Abstract

The aim of this work is to propose an hp -adaptive algorithm for discontinuous Galerkin methods that is capable to detect the discontinuities and sharp layers and avoid the spurious oscillation of the solution around them. In order to control the spurious oscillations, artificial viscosity is used with the particularity that it is only applied around the layers where the solution changes abruptly. To do so, a novel troubled-cell detector has been developed in order to mark the elements around those layers and to impose linear order in them. The detector takes advantage of the evolution of the value of the gradient through the adaptive process.

Keywords: hp -adaptivity, discontinuous Galerkin, shock capturing, troubled-cell detector, nonlinear stabilization, convection-diffusion, convection-dominated flows

1. Introduction

When working with convection-dominant convection-diffusion problems, the solution may contain steep layers where the solution changes abruptly. In the purely hyperbolic case, when the diffusion is zero, the solution may contain discontinuities. High resolution is needed around discontinuities and sharp layers, being very suitable to consider h -adapted meshes that lead to a quite uniform distribution of the error. Even better, using hp -adapted finite element (FE) spaces, one can improve accuracy by increasing the order of approximation (p) in smooth regions while reducing the element size (h) near discontinuities or sharp layers. This implies an extra effort that can be tackled in different ways (see Sect. 4.1). In order to select the FEs that need to be refined, an *a posteriori* error estimator has to be used. In hp -adaptivity, one also needs some kind of smooth indicator in order to decide whether to refine the mesh element or increase the order of approximation.

When working on adapted meshes with continuous FE spaces, one has to take into account how to adapt the mesh and preserve its regularity or how to deal with the hanging nodes that appear during the process. Some methods avoid the hanging nodes when adapting the mesh at the expense of increasing the aspect ratio (see, e.g., [34] for 2D meshes); the implementation of such methods for general 3D meshes is complicated. On the other hand, simple local adaptivity inevitably produces hanging nodes. Enforcing continuity (if required) implies an extra implementation work, especially when dealing

with arbitrary-level hanging nodes [29]. Discontinuous Galerkin (DG) methods can readily use non-conforming meshes, i.e., simple local adaptivity procedures, but they involve more degrees of freedom (DOFs) than continuous Galerkin ones for the same mesh. In [2], Badia and Baiges proposed a continuous-discontinuous formulation that allows one to use local adaptivity without the need to enforce continuity on hanging nodes. This approach combines the lower cost of continuous Galerkin methods and the capability of DG methods to deal with non-conforming meshes. In this work, we restrict ourselves to interior penalty DG methods and local adaptivity.

The numerical approximation of the problems considered above can lead to spurious oscillations around sharp layers or discontinuities, violating at the discrete level the monotonic properties of the continuous operators. Numerical techniques to avoid these oscillations are denoted as shock capturing techniques. We will use the common notion of shock capturing even though in this paper we will only deal with layers (for a discussion about the appropriateness of the name see [25]). Monotonic linear shock capturing can be at most first order accurate, due to the celebrated Godunov's theorem. In order to attain better accuracy, perturbations that depend on the solution itself must be considered.

There are several shock capturing strategies. Limiting strategies are typically applied to explicit time integration; a tentative solution is first obtained with the explicit time-marching scheme and next post-processed with the limiter to clean up oscillations. This approach is typical in fully explicit finite volume and DG codes for hyperbolic conservation laws [12]. A conceptually different family of methods are those based on adding artificial diffusion terms. The purpose of these methods is to perturb the problem at hand in such a way that the perturbed solution will be (almost) oscillation free. Since no post-process is needed and a viscous term is added to the formulation, this type of techniques are more suitable for implicit (or implicit-explicit) time integration and continuous FE spaces. This artificial diffusion terms can be understood as *nonlinear stabilization* methods. Only a few of the nonlinear stabilization schemes proposed in the literature have been proved to enjoy some variant of discrete maximum principle (DMP) [3, 4, 8, 11]. In most of these schemes, the nonlinear stabilization is only activated on the sharp layer/discontinuity, resorting to linear stabilization on smooth regions.

Up to our knowledge, all the shock capturing methods proposed so far that enjoy a DMP have been designed to work with linear (or at most quadratic) FEs [20]. In any case, the use of high-order FEs on steep layers or discontinuities is not wise, since there is no smoothness to be exploited in these regions if the mesh is not finer than the width of the physical layer. On the other hand, very refined meshes are needed to efficiently capture the layers, making adaptive mesh refinement mandatory. Based on these observations, the motivation of this work is *to combine nonlinear stabilization schemes satisfying the DMP (when applied to linear FEs) and hp-adaptivity*. The objective is to restrict the nonlinear stabilization around the sharp layers, where a very fine mesh with first order elements is to be automatically selected by the *hp*-adaptivity algorithm. On the other hand, far from those regions, where the solution can be smooth, the algorithm should increase the order and reduce element sizes.

Since we are interested in using *hp*-adaptive methods, we find it necessary not to apply the artificial viscosity everywhere but only on some troubled cells (some existing methods already scale the viscosity in such a way that it takes values very close to

zero in smooth regions [17, 3]). The nonlinear stabilization is used in smooth regions. There are various troubled-cell indicators; several of them have been compared in [37]. They use different criteria to identify the elements that may contain a sharp layer, discontinuity, or shock. For example, Cockburn and Shu [12] compare the gradients between neighboring elements. This was extended to higher orders with the analysis of the coefficients associated with different orders when using modal bases [28]. There are also some local discontinuity detectors which are based on the decay in the coefficients [6]. These troubled-cell detectors are designed for limiters in explicit simulations; once the solution is computed, troubled cells are detected and then properly modified (most of the time according to parameters obtained during the detection of the troubled cells). Herein, we are interested in troubled-cell indicators for implicit schemes that involve nonlinear stabilization methods (i.e., artificial viscosity). In a different setting, Vuik and Ryan were successful in detecting the non-smooth regions taking into account the properties associated with the multiwavelet coefficients [39].

Albeit they can be used in h -adapted meshes [40], the troubled-cell indicators mentioned above are designed for fixed meshes and they simply recalculate the troubled-cell indicator in each iteration instead of taking advantage of the solution history. In this paper we also aim *to develop a novel troubled-cell indicator and hp -adaptivity scheme that exploit the adaptivity history*. The idea is simple: if there is a discontinuity (or a steep layer with a width of an order much smaller than the characteristic length of the element), the gradient will be greater with each h -refinement of the elements, so we propose to compare the value of the gradient between parent and children elements in each adaptivity iteration in order to decide whether an element should be marked as suspicious of being in the neighborhood of a sharp layer. In the overall algorithm, linear order is imposed over the elements where the troubled-cell flag is activated.

The outline of this paper is the following. In Sect. 2 we will introduce the problem to solve as well as the notation and the interior penalty DG method that we will use. In Sect. 3 a new artificial viscosity method will be introduced; it is an extension of the one proposed in [4]. The next Sect. 4 will deal with the adaptivity strategy. Next, in Sect. 5 we will introduce a novel troubled-cell detector that will allow us to use the artificial viscosity presented before as well as to improve the hp -adaptivity process. Finally, in Sect. 6, some numerical experiments are reported to show the performance of the proposed method.

2. The interior penalty method for the convection-diffusion equation

In what follows we will introduce the convection-diffusion problem that we want to solve as well as the interior penalty discontinuous method that we will use to do so.

2.1. The convection-diffusion problem

We will consider the convection-diffusion problem with Dirichlet boundary conditions:

$$\begin{cases} Lu := -\nabla \cdot (\mu \nabla u) + \nabla \cdot (\beta u) &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega \end{cases} \quad (1)$$

where Ω is an open, bounded, connected subset of $\mathbb{R}^d$ with a Lipschitz boundary $\partial\Omega$, d is the space dimension (it is assumed to be arbitrary although in Sects. 3 and 5 the analysis

is restricted to the 2-dimensional case), $f \in L^2(\Omega)$, and $g \in H^{\frac{1}{2}}(\partial\Omega)$. We assume that $\mu > 0$ is constant and $\beta \in (H^1(\Omega) \cap C^0(\Omega))^d$ is solenoidal (i.e., $\nabla \cdot \beta = 0$). We denote by $\partial\Omega^-$ the inflow boundary defined as $\partial\Omega^- = \{x \in \partial\Omega | \beta(x) \cdot n_{\partial\Omega}(x) < 0\}$ for $n_{\partial\Omega}$ the outside normal to Ω . It is well known that the operator L associated to problem (1) enjoys the maximum principle (for proofs on maximum principles for elliptic problems see [19]). In particular, it means that no interior oscillations can appear inside the domain when $f = 0$. But it is also well known that, when the problem is convection-dominant, the solution may have sharp interior and boundary layers. It is in those regions where the discrete solution may exhibit spurious oscillations, and where nonlinear stabilization terms aim to regularize the problem. We note that the nonlinear stabilization schemes used in this work can also be applied to the purely hyperbolic problem [4], i.e., for $\mu = 0$, even though we have not considered this case in Sect. 6 because the *a posteriori* error estimator used in Sect. 4.1 do not apply under such circumstances.

2.2. Notation

We consider that there are a total of $M \leq \bar{M}$ adaptive iterations. For each adaptive step m we will need to consider the partition $\mathcal{T}_m = \{K_i\}_{i=1}^{N_m^{\text{el}}}$ of Ω by N_m^{el} elements of characteristic length $h_{K_i}^m$. We consider quadrilateral (hexahedral in 3 dimensions) elements. Since we will use isotropic refinement, we can assume that the length of the edges of the element in any dimension is of the order of the characteristic length of the element. Each mesh $\mathcal{T}_m$ corresponds to a mesh adaptation of the previous partition of the domain $\mathcal{T}_{m-1}$. This implies that every element $K_i \in \mathcal{T}_m$ is either equivalent to a certain $K_j \in \mathcal{T}_{m-1}$ or it corresponds to a transformation of an element or a set of them from $\mathcal{T}_{m-1}$. This transformation is either a isotropic refinement of an element (K_i is one of the children of an element $K_j \in \mathcal{T}_{m-1}$) or a coarsening of a set of elements (K_i is the parent of the set of elements). Isotropic refinement divides each element into $n_c = 2^d$ identical sub-elements. The coarsening, at the same time, can only be performed with a set of n_c elements that already come from a previous refinement of an element. In order to ease the transfer of data between consecutive meshes, we will establish a relation between the indices of the elements in $\mathcal{T}_m$ and the immediate previous mesh $\mathcal{T}_{m-1}$. To this end we define the function $F_m : \{1, \dots, N_m^{\text{el}}\} \rightarrow \{1, \dots, N_{m-1}^{\text{el}}\}$ in the following way: $K_{F_m(i)} \in \mathcal{T}_{m-1}$ is either the corresponding equivalent element in $\mathcal{T}_{m-1}$, the parent of K_i or the first of the children it comes from (we can assume without loss of generality that the children of an element are consecutively listed). In the case of a coarsened element $K_i \in \mathcal{T}_m$, its corresponding elements in $\mathcal{T}_{m-1}$ are $K_{F_m(i)+c}$ for $c = 0, \dots, n_c - 1$.

Given $p \in \mathbb{N}$, let $\mathbb{P}_p(\omega)$ be the space of polynomials up to order p on the domain $\omega \subset \Omega$. Given $\{p_m^j\}_{j=1}^{N_m^{\text{el}}}$ the array with the approximation order for every element in $\mathcal{T}_m$, we define g_m as the piece-wise polynomial function on $\partial\Omega$ such that $g_m|_{\partial K_i \cap \partial\Omega}$ is the polynomial of order p_m^i obtained by the L^2 -projection of $g|_{\partial K_i \cap \partial\Omega}$ onto $\mathbb{P}_{p_m^i}(\partial K_i \cap \partial\Omega)$ for any $K_i \in \mathcal{T}_m$ containing a boundary facet. Now we can define the discrete space considered henceforth as the discontinuous space of piece-wise polynomial functions $V_m = \{v_m | v_m|_{K_i} \in \mathbb{P}_{p_m^i}(K_i) \ \forall K_i \in \mathcal{T}_m\}$. For any $v_m \in V_m$, we notate as $v_m^{K_i} \in \mathbb{P}_{p_m^i}(\bar{K}_i)$ and $\nabla v_m^{K_i} \in (\mathbb{P}_{p_m^i-1}(\bar{K}_i))^d$ the functions that correspond to the restriction of v_m and its gradient to K_i . We will denote by $\|v\|_\omega := (\int_\omega v^2)^{\frac{1}{2}}$ the $L^2(\omega)$ norm of $v|_\omega$ for any

function $v \in L^2(\Omega)$. Similarly, $\|v\|_{\infty, \omega} := \max_{x \in \omega} v(x)$ will denote the $L^\infty(\omega)$ norm. Also we

define the measure of ω as $|w| := \int_{\omega} 1$.

Given the mesh $\mathcal{T}_m$, the intersection $F = \partial K_i \cap \partial K_j$ of two neighboring elements $K_i, K_j \in \mathcal{T}_m$ is called an interior facet of $\mathcal{T}_m$ if it is a set of dimension $d - 1$. The set of all the interior facets is denoted by $\mathcal{E}_m^0$. On the other hand, the non-empty intersection $F = \partial K_i \cap \partial \Omega$ of an element $K_i \in \mathcal{T}_m$ with the boundary of the domain, is called a boundary facet. The set of all the facets is denoted by $\mathcal{E}_m$. We denote by $\mathcal{E}_m(K_i)$ the set of faces $F \in \mathcal{E}_m$ that are contained on ∂K_i . For any facet $F \in \mathcal{E}_m^0$ we will denote K_F^+ and K_F^- the only two neighboring elements such that $\partial K_F^+ \cap \partial K_F^- = F$. In addition, we can name n_F^+ and n_F^- the unitary normal to face F outside K_F^+ and K_F^- , respectively. For the facets on the boundary only K_F^+ and n_F^+ are defined. Given a facet F , we define the characteristic face length as

$$h_F = \begin{cases} \min\{h_{K_F^+}, h_{K_F^-}\} & \text{if } F \in \mathcal{E}_m^0 \\ h_{K_F^+} & \text{if } F \in \partial \Omega. \end{cases}$$

Given $v_m \in V_m$, we can define the common concepts of average $\{\!\{ \cdot \}\!\}$ and jump $\llbracket \cdot \rrbracket$ on an interior point x of a facet $F \in \mathcal{E}_m^0$ as follows:

$$\{\!\{v_m\}\!\}(x) = \frac{1}{2}(v_m^{K_F^+}(x) + v_m^{K_F^-}(x)), \quad \llbracket v_m \rrbracket(x) = v_m^{K_F^+}(x)n_F^+ + v_m^{K_F^-}(x)n_F^-.$$

We also define the harmonic average of q on x as

$$\langle v_m \rangle(x) = \frac{2v_m^{K_F^+}(x)v_m^{K_F^-}(x)}{v_m^{K_F^+}(x) + v_m^{K_F^-}(x)}.$$

On boundary facet points $x \in F$, $F \subset \partial \Omega$, we define $\{\!\{v_m\}\!\}(x) = v_m^{K_F^+}(x)$, $\llbracket v_m \rrbracket(x) = v_m^{K_F^+}(x)n_F^+(x)$ and $\langle q \rangle(x) = v_m^{K_F^+}(x)$.

2.3. Weak form and the interior penalty discontinuous Galerkin approximation

For each adaptive iteration m , the weighted incomplete interior penalty discontinuous Galerkin (wIPDG) method with nonlinear artificial viscosity stabilization reads as

$$\text{Find } u_m \in V_m \text{ such that } a_m(u_m, v_m) = l(v_m) \quad \forall v_m \in V_m, \quad (2)$$

where

$$\begin{aligned} a_m(u_m, v_m) = & \sum_{K_i \in \mathcal{T}_m} \int_{K_i} ((\mu + \varepsilon_{K_i}(u_m)) \nabla u_m \nabla v_m - u_m \beta \cdot \nabla v_m) \\ & - \sum_{F \in \mathcal{E}_m} \int_F \mu(\xi \llbracket u_m \rrbracket \{\!\{ \nabla v_m \}\!\} + \{\!\{ \nabla u_m \}\!\} \llbracket v_m \rrbracket) + \sum_{F \in \mathcal{E}_m} \int_F c^{\text{ip}} h_F^{-1} \langle \mu + \varepsilon_m(u_m) \rangle \llbracket u_m \rrbracket \llbracket v_m \rrbracket \\ & + \sum_{F \in \mathcal{E}_m \setminus \partial \Omega^-} \int_F \{\!\{ \beta u_m \}\!\} \llbracket v_m \rrbracket + c^{\text{bms}} \sum_{F \in \mathcal{E}_m^0} \int_F |\beta| \llbracket u_m \rrbracket \llbracket v_m \rrbracket \end{aligned} \quad (3)$$

and

$$\begin{aligned}
l(v_m) = & \sum_{K_i \in \mathcal{T}_m} (f_m, v_m)_K - \sum_{F \in \partial\Omega^-} \int_F \beta \cdot n_{\partial\Omega} g_m v_m \\
& + \sum_{F \in \partial\Omega} \int_F \xi \mu g_m \{\!\!\{ \nabla v_m \}\!\!\} \cdot n_{\partial\Omega} - \sum_{F \in \partial\Omega} \int_F c^{\text{ip}} h_F^{-1} \langle \mu + \varepsilon_m(u_m) \rangle g_m v_m,
\end{aligned} \tag{4}$$

where f_m (similarly to g_m) denotes the piece-wise polynomial approximations in $\mathcal{T}_m$ given by the L^2 -projections of f onto V_m . The parameter c^{ip} is a constant that scales the diffusion interior penalty term while c^{bms} is the constant that stabilizes the convection term as it is explained in [7]. The values that we have used in our computations are specified in Sect. 6. The value of ξ is discussed in Remark 2.

Remark 1. *The function $\varepsilon_m(u_m)$ corresponds to a nonlinear piece-wise artificial viscosity that introduces extra stability to the method. This artificial viscosity can be activated or not and its value may rely on different parameters of the problem and the numerical solution u_m . Its restriction to element K_i is denoted by $\varepsilon_{K_i}(u_m)$ and its computation will be defined in Sect. 3.*

Remark 2. *The choice of ξ leads to different interior penalty methods: symmetric ($\xi = 1$), incomplete ($\xi = 0$), or anti-symmetric ($\xi = -1$). The original interior penalty method is the symmetric one but some nice monotonicity properties have been shown in [4] when we take $\xi = 0$. Our approach is to have a dynamic ξ depending on u_m in such a way that it takes values in $[0, 1]$ depending on the properties of u_m in each element or facet. The specific implementation of such a feature will be introduced in the experimental Sect. 6.*

Remark 3. *Since the main motivation of this article is to accurately represent discontinuities or sharp layers, we aim to keep also the sharp layers on boundaries in our numerical experiments, to check the quality of our approach. As a result, boundary conditions are enforced strongly, which is not common in DG methods. In this case, the test space is replaced by $V_m^{\text{trial}} = \{v_m \in V_m : v_m|_{\partial\Omega} = g_m\}$ and the trial space by $V_m^{\text{test}} = \{v_m \in V_m : v_m|_{\partial\Omega} = 0\}$. Thus, the terms including g in the right-hand side will eventually disappear.*

3. Nonlinear stabilization: The shock capturing

Artificial viscosity shock capturing or nonlinear stabilization schemes basically consist in adding an extra non consistent viscosity term ε to the original problem. Its value is solution-dependent and, in each element K_i , it is usually of the order $h_{K_i} \|\beta\|_{\infty, K_i}$ scaled by a constant and shock detector S_{K_i} . There are different features that can be taken into consideration to compute S_{K_i} such that the elemental residual of the equation [13, 30, 25, 26], the entropy energy of the solution [21], or the decaying of the high order coefficients [27, 36], among others. In this work, as in [9, 8], we will focus on the variation of the gradient of the solution between elements.

In general, these techniques take advantages of the good smooth properties of the Laplacian operator. In particular, a key feature to avoid oscillations is to inherit the

Figure 1: Solutions to the discontinuity propagation problem of 6.2 with 20×20 meshes

maximum principle of the continuum equation by the numerical scheme, leading to the so called discrete maximum principle (DMP); see [8, 10]. Unfortunately, even for the Laplacian operator itself, monotonicity has only been proved on acute simplicial meshes, which are hard to get in practice. A monotonicity-preserving (for acute meshes) nonlinear stabilization for continuous FEs has been proposed in [3] for the multi-dimensional case. For DG methods the situation is worse, because a DMP for the Laplacian has only been proved, as far as we know, for 1D problems and incomplete IPDG schemes [23]. A nonlinear stabilization for DG approximations of transport problems that enjoys a DMP under the same assumptions has been proved in [4]. Even though none of these methods can be, by construction, monotonically-preserving in general, to define nonlinear stabilization schemes that keep the DMP properties of the underlying Laplacian operator has been proved to be a good approach when designing such algorithms. We refer to [3, 4] for some tests with DMP-preserving methods on non-acute meshes.

Restricting the problem to 2D, the nodal Gradient Jump Viscosity (nGJV) introduced in [4] was designed for triangular structured meshes. The element-wise constant artificial diffusion was scaled by the shock detector, which took into account the variation in any direction of the gradient of the function in each node of the element. Since we are interested in working with adaptive meshes of quadrilaterals (or hexahedra), we need to fit the method to this new paradigm. The first new property to take into account is that the gradient of bi-linear solutions is not constant in each element anymore. The natural extension is to take the value of the gradient of the elemental solution at the nodal point. We checked the behavior of this definition with a coarse mesh of 20×20 quads and solving the discontinuity propagation problem defined in Sect. 6.2. The results are plotted in Fig. 1(a) and Fig. 1(b) and show that, even though the DMP is still attained, the regularity of the problem near the exponential layers on the boundary is lost. The second hurdle is to deal with the hanging nodes that appear in the adaptive meshes. It is not clear whether the computation of the viscosity should take into account the value of the shock detector in those points and, if so, how to consider the parent neighbor.

Since, in fact, the DMP is not proved for nGJV for dimensions greater than one, another option is to look for an alternative extension of the DMP-satisfying artificial viscosity for 1D that allows us to circumvent such problems. A possible approach is to consider an extension of the boundary gradient jump viscosity (bGJV) introduced in [3]

to the discontinuous scheme. The constant nonlinear viscosity in K_i reads as follow:

$$\varepsilon(u_m)|_{K_i} = c_{\text{giv}} h_{K_i} \|\beta\|_{\infty, K_i} \max_{F \in \mathcal{E}_m(K_i)} S_F^{\Gamma_{K_i}(F)}(u_m), \quad (5)$$

where $\Gamma_{K_i} : \mathcal{E}_m(K_i) \rightarrow \{-, +\}$ is a function such that $K_i = K_F^{\Gamma_{K_i}(F)}$ for each $F \in \mathcal{E}_m(K_i)$, c_{giv} is set to be 0.5 and S_F^α is defined as

$$S_F^\alpha(u_m) = \frac{|\nabla u_m(\bar{x}) \cdot n|}{|\nabla u_m(\bar{x})|} \max_{x \in F} (s_F^\alpha(x))^q, \quad (6)$$

where S_F is the shock detector defined for every $x \in F$ by

$$s_F^\alpha(x) := \frac{\left| \nabla u_m^{K_F^\alpha}(x) \cdot n_F^\alpha h_{K_F^\alpha} + \nabla u_m^{K_F^{-\alpha}}(x) \cdot n_F^{-\alpha} h_{K_F^{-\alpha}} + 2\llbracket u_m \rrbracket(x) \cdot n_F^\alpha \right|}{\left| \nabla u_m^{K_F^\alpha}(x) \cdot n_F^\alpha h_{K_F^\alpha} \right| + \left| \nabla u_m^{K_F^{-\alpha}}(x) \cdot n_F^{-\alpha} h_{K_F^{-\alpha}} + \llbracket u_m \rrbracket(x) \cdot n_F^\alpha \right| + |\llbracket u_m \rrbracket(x) \cdot n_F^\alpha|}. \quad (7)$$

In this definition α stands for $-/+$ (therefore $-\alpha$ is the opposite sign), $\bar{x} = \arg \max_{x \in F} (s_F^\alpha(x))$. When the denominator in (7) vanishes, we simply take $s_F^\alpha = 0 \forall x \in F$. The term $\frac{|\nabla u_m(x) \cdot n|}{|\nabla u_m(x)|}$ in (6) corresponds to a correction term to avoid excessive smearing. In practice, when computing these terms, the maximum is approximated by the maximum at the Gauss points.

The behavior of this new method was tested in a 20×20 quads mesh. It needed few nonlinear iterations to converge and, as it is shown in Fig. 3, the result obtained is much smoother than with the previous one. We will call this method the *discontinuous boundary Gradient Jump Viscosity* (dbGJV) and is the one we will use hereinafter.

Remark 4. *When we restrict the definition in (6) to the piece-wise linear continuous solutions in triangular homogeneous meshes the values of this new definition and the original S_F associated to the boundary Gradient Jump Viscosity method in [3] are equivalent. We recall the definition was given by*

$$S_F(u_m) = \frac{\int_F |\nabla u_m \cdot n| d\sigma}{\int_F |\nabla u_m| d\sigma} \left(\frac{\int_F \|\nabla u_m\| d\sigma}{2 \int_F \|\nabla u_m \cdot n\| d\sigma} \right)^q. \quad (8)$$

But in that case the value of ∇u_m is constant in each element, $\llbracket u \rrbracket$ is zero since it is continuous and the values of h_{K_i} nullify, so (6) and (8) read the same.

4. *hp*-Adaptivity

The interest in *hp*-adaptive FE methods goes back to classical results published, e.g., in [22], where the authors showed the possibility of attaining exponential convergence. Since then, a lot of computer implementations of the method have been realized, despite its rather large technical complexity (see, e.g., [1, 14, 5, 29, 32], where we list only few references with some other significance for this work). Different implementations differ greatly in their purpose, generality, types of elements used, and many other aspects. In

this contribution, we use quadrilateral elements and isotropic refinements. The choice of DG methods makes the use of hanging nodes easier, since there is no need to enforce continuity in hanging nodes. Nevertheless, we will allow only first-level hanging nodes, meaning that one element and its neighbors refinement can only be one level away. The experience shows that this restriction does not lead to significant slow-down of the convergence due to extra refinement and is used by most modern codes.

4.1. The *a posteriori* error estimator and the adaptivity algorithm

For the *hp*-adaptive paradigm, one not only needs to determine which elements need to be refined but choose between *h*- or *p*-refinement. This is one of the big challenges of these methods and it has been tackled in different ways. Some approaches use an *a priori* knowledge of trouble-spotting elements, setting them to only be *h*-refined [1, 38]. Demkowicz and co-workers [16, 14, 15] project the solution in different mesh adaptations and make the adaptation evolve by choosing the best option according to a suitable *a posteriori* error estimator. Some approaches analyze the decay of the Legendre polynomials coefficients [24, 31]. These techniques and some more are compared in the survey performed by Mitchell and McClain [35]. In that survey the winning strategy seems to be the smooth predictor proposed by Melenk and Wohlmuth in [33] so this is the one we will use in our numerical experiments.

The smooth predictor basically consists of predicting the error assuming that the solution is smooth. Knowing that, in the smooth case, the order of convergence of the L^2 -error in an element K_i is $h_{K_i}^{p_m^i+1}$, one can predict the evolution of the error when the *h*- or *p*-refinement takes place. An *a posteriori* error estimator that decays as the L^2 -error for the Laplacian problem is presented in [33]. When an element is marked for refinement, the local *a posteriori* error estimator is compared with the predicted error (computed in the last adaptivity step involving that element). If the error is greater than predicted, the solution is assumed to be non-smooth and the element is marked for *h*-refinement. Otherwise it is marked for *p*-refinement. We will basically use this scheme but, since we are interested in enforcing linear order (and thus, force the *h*-refinement) in elements where we want to apply the shock capturing, it has been slightly modified (see algorithm in Sect. 5.3 for an explicit definition of the *hp*-adaptivity algorithm).

To be able to benefit from the smooth convergence *hp*-adaptivity technique, an *a posteriori* error estimator that converges as the L^2 -norm of the actual error is needed. To this end, the error estimator proposed in [18] has been implemented. For each element K_i , it consists of three parts:

$$(\eta_m^i)^2 = \eta_{m,R_i}^2 + \eta_{m,E_i}^2 + \eta_{m,J_i}^2, \quad (9)$$

where η_{R_i} is the interior residual term, while η_{E_i} and η_{J_i} correspond to the edge residual and the error in the jumps on the faces, respectively. The original definition of the terms will be adapted to the case of isotropic refinement of quadrilateral meshes. Also, since strong Dirichlet boundary conditions are used, we have removed the boundary terms. It was noticed that, when the boundary conditions were not polynomial (and thus could not be exactly captured by the solution), the value of the approximation error on the boundary led to overestimate the error on the boundary leading to an over-refinement of

the boundary elements (even when the solution was smooth). So, the definitions of the local *a posteriori* error estimator terms read like

$$\eta_{m,R_i}^2 = \mu^{-1} (p_m^i)^{-2} h_{K_i}^2 \|f_m + \mu \Delta u_m - \beta \nabla u_m\|_{K_i}^2, \quad (10)$$

$$\eta_{m,E_i}^2 = \frac{1}{2} \sum_{F \in \mathcal{E}_m(K_i) \setminus \partial\Omega} \frac{h_F}{\mu p_F} \|\llbracket \mu \nabla u_m \rrbracket\|_F^2, \quad (11)$$

$$\eta_{m,J_i}^2 = \frac{1}{2} \sum_{F \in \mathcal{E}_m(K_i) \setminus \partial\Omega} \left(\frac{\mu (c^{\text{ip}})^2 p_F^2}{h_{K_i}} + \frac{\mu p_F^2}{h_{K_i}} + \frac{h_F}{\mu p_F} \right) \|\llbracket u_m \rrbracket\|_F^2. \quad (12)$$

The value of p_F is $p_F := \max_{K_i \supset F} p_m^i$. It is proved in [18] that there exists a norm with respect to which this error estimator is *almost* asymptotically equivalent.¹ This norm cannot be proved to be equivalent to the L^2 -error but, in practice, the numerical results show agreement between the *a posteriori* estimator and the weighted L^2 -error of the approximation defined by $\mu^{-\frac{1}{2}} \|u - u_m\|_{L^2(\Omega)}$.

Following this scheme, a predicted error array E_m is constructed at the beginning of each adaptivity iteration m . For each element, we will define the elemental predicted value E_m^i based on the value of the *a posteriori* error estimator $\eta_{m-1}^{Fm(i)}$ of the corresponding element in $\mathcal{T}_{m-1}$. When the numerical solution u_m is obtained, the actual *a posteriori* error estimator will be compared to this predicted value E_m^i to decide whether to h - or p -refine.

In practice, we have developed a variation of the mesh adaptivition proposed by Melenk and Wohlmuth in [33]. The first change is in the computation of the predicted errors: the computation of the size h_{K_i} of an element K_i has been added in order to better compute the predicted error when it is p -refined. Also, we have included the cases of coarsening and, moreover, we have changed the prediction for not refined elements. The explicit computation of the predicted error is shown in Alg. 1. Notice that if the element has not been modified (refined or coarsened), neither are the predicted values. For the coarsened elements, we will set the predicted error in such a way that we reinforce the p -refinement. If an element K_i is p -coarsened, the predicted error will be set to $E_m^i = \infty$ so it is p -refined in the next mesh adaptation. On the other hand, when a set of elements is h -coarsened we will take an error proportional to the greatest error of the set. This is due to the fact that the troubled-cell detector already reinforces the h -refinement and we want to recover the order of convergence of the method in the smooth regions that have been unnecessarily h -refined.

The selection of elements to refine and how to refine them has also been slightly modified and it relies on the troubled-cell detector introduced in the next section. It will be specified in Subsect. 5.3.

¹*Almost* meaning that it is sub-optimal with respect to the polynomial degree in the upper bound and, moreover, the bounds are up to an approximation error of the force term and the convective term that explode with $\mu \rightarrow 0$

Algorithm 1: Set the predicted error array E_m

```

for  $i = 1 \dots N_m^{\text{el}}$  do
  if  $K_i$  is a child of  $K_{Fm(i)} \in \mathcal{T}_{m-1}$  then
     $E_m^i = 2^{-p_m^i - 1} \gamma_p^{\frac{1}{2}(p_m^i - p_{m-1}^{Fm(i)})} \gamma_h^{\frac{1}{2}} \eta_{m-1}^{Fm(i)}$ 
  else if  $K_i$  corresponds to  $K_{Fm(i)} \in \mathcal{T}_{m-1}$  then
    if  $p_m^i > p_{m-1}^{Fm(i)}$  then
       $E_m^i = \gamma_p^{\frac{1}{2}} h_{\max} \eta_{m-1}^{Fm(i)}$ 
    else if  $p_m^i = p_{m-1}^{Fm(i)}$  then
       $E_m^i = \gamma_n E_{m-1}^{Fm(i)}$ 
    else if  $p_m^i < p_{m-1}^{Fm(i)}$  then
       $E_m^i = \infty$ 
    else if  $K_i$  is the parent of  $K_{Fm(i)} \in \mathcal{T}_{m-1}$  then
       $E_m^i = \max_{c=0, \dots, n_c-1} \left\{ \gamma_h^{\frac{1}{2}} 2^{p_{m-1}^{Fm(i)+c}} h_K^{(p_m^i - p_{m-1}^{Fm(i)+c})} \eta_{m-1}^{Fm(i)+c} \right\}$ 

```

The value of the constant parameters γ_h , γ_p , γ_n are defined in Sect. 6.

5. Troubled-cell detector

5.1. Refinement-dependent troubled-cell detector

A novel troubled-cell detector is proposed next. It uses the information given by the evolution of the solution through the mesh adaptation process. The idea is that if the solution has a discontinuity front in a certain region, the gradient of the discrete approximation in the elements nearby will be inversely proportional to their size. We are assuming that the width of the sharp layer is of an order much smaller than the characteristic length so we will consider that its width is 0, i.e., a discontinuity. We will present the method for the 2D case but its extension to the 3D case is given in Remark 5. So we will define the mean elemental gradient in an element $K_i \in \mathcal{T}_m$ at adaptivity step m by

$$\mathcal{G}_m^i = \frac{\|\nabla u_m\|_{K_i}}{|K_i|^{\frac{1}{2}}}. \quad (13)$$

To analyze the evolution of the gradient of a numerical solution in 2D we will assume that we have a solution with a discontinuity and that it is approximated by a numerical method. In particular we will focus on what happens on an element that is h -refined (see Fig. 2). Since we expect our method to be capable of capturing the discontinuity smoothly and in a few elements, we will assume that the solution is nodally exact. Moreover, since a discontinuity implies a huge change of the value of the solution, we will conjecture that the difference between the value of the solution in different nodal points in the same side of the discontinuity is negligible. In fact, even though the discontinuities might not be straight, we will assume that the mesh is fine enough so their topology in the element is close to a straight line. This leads to a constant value solution in each side of the discontinuity and, without loss of generality (because we are only interested in the value

Figure 2: Element (and its refinement) with a discontinuity

of the gradient in (13)), it will be considered 0 on one of the sides. We will call H the value of the jump on the discontinuity. With these premises we obtain two possible settings for an element that is crossed by a discontinuity:

- 1 vertex in one side of the discontinuity and 3 vertices in the other (Fig. 2(a)).
- 2 vertices in each side of the discontinuity (Fig. 2(b)).

Once the element is subdivided in its 4 children, the ones that inherit the discontinuity can again have any of the two configurations, as it is shown in Fig. 2. Taking this situation and letting h be the size of the element, it is easy to compute the value of the mean elemental gradient for the element K at the adaptivity step m in any of both cases. For the parent elements (case 1 and 2) an easy calculation leads to the values (notice that, for the sake of comprehensibility, there is a different notation in the superscript of $\mathcal{G}$ in this case):

$$\mathcal{G}_m^{K_1} = \sqrt{\frac{2}{3}} \frac{H}{h} \quad \text{and} \quad \mathcal{G}_m^{K_2} = \frac{H}{h}.$$

Now we want to know the relation between these values and the mean elemental gradients of the children elements K' that contain the layer. Knowing that the children of the element have characteristic length $\frac{h}{2}$ and assuming the same premises on the solution as before, the only two possibilities for the elemental mean gradient to be are

$$\mathcal{G}_{m+1}^{K'_1} = \sqrt{\frac{8}{3}} \frac{H}{h} \quad \text{and} \quad \mathcal{G}_{m+1}^{K'_2} = 2 \frac{H}{h}.$$

Thus the value of the elemental mean gradient in the children elements that contain a discontinuity increases with respect to the previous step. On the other hand, the gradients in the children that do not have a discontinuity are negligible (they are 0 in this limit case) in comparison to the original gradient. In particular, the quotient between both values is expected to be in the following intervals:

$$\begin{cases} \frac{\mathcal{G}_{m+1}^i}{\mathcal{G}_m^{F_{m+1}(i)}} \in \left[\sqrt{\frac{8}{3}}, \sqrt{6} \right] & \text{if } K_i \in \mathcal{T}_{m+1} \text{ contains a discontinuity} \\ \frac{\mathcal{G}_{m+1}^i}{\mathcal{G}_m^{F_{m+1}(i)}} \approx 0 & \text{if } K_i \in \mathcal{T}_{m+1} \text{ does not contain a discontinuity} \end{cases}$$

Remark 5. For the 3D case, taking similar assumptions and approximating the discontinuity by a plane, we end up having 5 different configurations. The mean elemental gradient of those configurations varies along the interval $\left[\frac{H}{\sqrt{3}h}, \frac{H}{h}\right]$. Following previous steps we expect the coefficient $\frac{\mathcal{G}_{m+1}^i}{\mathcal{G}_m^{F_{m+1}(i)}}$ with respect the children that inherit the discontinuity to be in the interval $\left[\frac{2}{\sqrt{3}}, 2\sqrt{3}\right]$.

This is the information that we will use in order to tackle our methods. To this end, we will define a troubled-cell indicator, say $\mathcal{S}$, which for each element K can take values $\mathcal{S}_m^i \in \{0, 1, 2\}$. Roughly speaking, $\mathcal{S}_m^i = 0$ for elements that are in a region where the solution is smooth, $\mathcal{S}_m^i = 1$ for elements that contains a discontinuity, and $\mathcal{S}_m^i = 2$ for elements that are around the previous elements. For each adaptive iteration m , once the numerical solution u_m is computed, we will find $\mathcal{G}_m^i$ the value of the mean elemental gradient in each element $K_i \in \mathcal{T}_m$ with $\mathcal{S}_m^i > 0$ and compare it with the value of the corresponding element/s in the iteration $m - 1$. We have the following scenarios:

- $K_i \in \mathcal{T}_m$ is the parent of $K_{Fm(i)+c} \in \mathcal{T}_{m-1}$, $c = 0, \dots, n_c - 1$

If any of the elements $K_{Fm(i)+c} \in \mathcal{T}_{m-1}$ had $\mathcal{S}_{m-1}^{Fm(i)+c} = 1$, and therefore contained a discontinuity, its parent K_i will also contain such a layer and so $\mathcal{S}_m^i = 1$. Otherwise, $\mathcal{S}_m^i = 0$.

- $K_i \in \mathcal{T}_m$ corresponds to $K_{Fm(i)} \in \mathcal{T}_{m-1}$

The intuitive idea would be to consider that if an element had a discontinuity inside, it should maintain the troubled-cell indicator. But, in practice, we found out that, if we did not check out the evolution of the gradient of the function in consecutive adaptive iterations, this could lead to over troubled-cell marking some regions in some smooth solutions. Instead, if $\mathcal{S}_m^i = 1$, the mean elemental gradient $\mathcal{G}_m^i$ is always compared to $\mathcal{G}_{m-k}^j$, being $j = F_{m-k+1} \circ F_{m-k+2} \circ \dots \circ F_m(i)$ the first element before the refinement. So, if $\mathcal{S}_{m-1}^{Fm(i)} = 1$ and $\mathcal{G}_m^i > \delta_n \mathcal{G}_{m-k}^j$, $\mathcal{S}_m^i = 1$, otherwise $\mathcal{S}_m^i = 0$.

- $K_i \in \mathcal{T}_m$ is a child of $K_{Fm(i)} \in \mathcal{T}_{m-1}$

Using the reasoning stated before, we will use a threshold value $\delta_n \in (0, \sqrt{\frac{8}{3}})$ to decide which of the children of an element have a discontinuity inside and should inherit the troubled-cell indicator. The troubled-cell indicator value will be given by²

$$\mathcal{S}_m^i = \begin{cases} 1 & \text{if } \mathcal{G}_m^i > \delta_n \mathcal{G}_{m-1}^{Fm(i)} \text{ and } \mathcal{S}_{m-1}^{Fm(i)} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

It is obvious that the smooth regions are refined much less than the sharp layers because the error in the latter is greater for the same size of element. This means that, in a

²See Remark 6 for a justification of why not only elements K_i with $\mathcal{S}_{m-1}^{Fm(i)} = 1$ are checked.

solution combining both kind of regions, the troubled-cell detector is very dynamic near the non-smooth layers, evolving in each adaptive step, but needs much more adaptive steps to be completely removed from the smooth regions. To accelerate this process, we will set a parameter $r_S = 0.001$ such that any element with $\mathcal{G}_m^i < r_S \max_j \mathcal{G}_m^j$ will be automatically set as non-troubled ($\mathcal{S}_m^i = 0$). That is to say that any element with a mean elemental gradient which is less than 0.1% of the mean elemental gradient of the element with the largest value will not be marked as a troubled-cell in that solution.

In addition to these values, once we have selected all the troubled-cells using the previous algorithm, we assign the flag $\mathcal{S}_m^i = 2$ to every element K that shares a face with a troubled-detected element. The scheme is summarized in Alg. 2.

Algorithm 2: Update the values of $\mathcal{S}_m$

```

 $\mathcal{S}_m = 0$ 
for  $i = 1 \dots N_m^{\text{el}}$  do
  if  $K_i$  is a child of  $K_{Fm(i)} \in \mathcal{T}_{m-1}$  then
    if  $\mathcal{G}_m^i > \delta_n \mathcal{G}_{m-1}^{Fm(i)}$  and  $\mathcal{S}_{m-1}^{Fm(i)} > 0$  then
       $\mathcal{S}_m^i = 1$ 
    else if  $K_i$  corresponds to  $K_{Fm(i)} \in \mathcal{T}_{m-1}$  then
      if  $\mathcal{S}_{m-1}^{Fm(i)} = 1$  and  $\mathcal{G}_m^i > \delta_n \mathcal{G}_{m-k}^{Fm(i)}$  then
         $\mathcal{S}_m^i = 1$ 
    else if  $K_i$  is the parent of  $K_{Fm(i)} \in \mathcal{T}_{m-1}$  then
      for  $c = 1, \dots, n_c - 1$  do
        if  $\mathcal{S}_{m-1}^{Fm(i)+c} = 1$  then
           $\mathcal{S}_m^i = 1$ 
    if  $\mathcal{G}_m^i < r_S \max_j \mathcal{G}_m^j$  then
       $\mathcal{S}_m^i = 0$ 
for  $i = 1 \dots N_m^{\text{el}}$  do
  if  $K_i$  share a face with  $K_j$  s.t.  $\mathcal{S}_m^j = 1$  and  $\mathcal{S}_m^i = 0$  then
     $\mathcal{S}_m^i = 2$ 

```

The value of the parameters δ_n is defined in Sect. 6

Remark 6. This extra troubled-cell indicator ($\mathcal{S}_m^i = 2$) has two main advantages: on one hand it allows to add artificial viscosity not only on the elements just over the discontinuity but also on the neighboring regions, which is useful, since the spurious oscillations often appear there. On the other hand, it allows us to introduce a second order control. If an element is marked with $\mathcal{S}_m^i = 2$, the evolution of its mean gradient will also be checked for their children and might be considered in the discontinuity layer in a next adaptivity step, leading to a more effective method in practice.

Remark 7. For implementation issues this algorithm will be performed in 2 steps. At each iteration m , before computing u_m , we will evaluate the predicted gradient G_m defined by:

- *h-coarsening*: $G_m^i = \min_{c=0, \dots, n_c-1} \mathcal{G}_{m-1}^{Fm(i)+c}$
- *No modification*: $G_m^i = G_{m-1}^{Fm(i)}$
- *h-refinement*: $G_m^i = \mathcal{G}_{m-1}^{Fm(i)}$

When the numerical solution is computed, the actual values of $\mathcal{G}_m$ are computed and compared to the predicted ones, if necessary. For example, if an element K_i is an *h*-refinement of $K_{Fm(i)} \in \mathcal{T}_{m-1}$ with $\mathcal{S}_{m-1}^{Fm(i)}$, we would check if $\mathcal{G}_m^i > \delta_n G_m^i$ to decide whether or not to assign $\mathcal{S}_m^i = 1$.

The value δ_n is to be chosen. It has to be big enough so that, when solving a smooth problem, there are no troubled-cell elements detected after a reasonable number of adaptive iterations (to avoid setting elements to *h*-refine instead of *p*-refine in smooth regions). This, in fact, means that a reasonable interval for δ_n is $(1, \sqrt{\frac{8}{3}})$ because it would make no sense to consider that in the regions where the gradient decays, we keep on considering that there is a discontinuity. On the other hand δ_n has to be small enough so it does detect the discontinuities and sharp layers. The choice of the value will be introduced in the experimental Sect. 6. In our numerical experiments we will start by setting $\mathcal{S}_0 = 1$ and then proceed to *h*-refine all the elements at least 2 iterations.

5.2. Activation of the shock capturing

The troubled-cell indicator will identify a region of linear elements from which different stabilization techniques can benefit. In particular, in this work, the previously introduced dbGJV viscosity will be used. The value of the shock detector S_F defined in (6) will be computed in all the faces that are of linear order, $p_F = 1$, and have, at least, one neighbor element K with $\mathcal{S}_m^i > 0$. This implies that the artificial viscosity can be activated in not troubled-cell elements. It is motivated by the fact that the troubled-cells are usually the ones over the steep layer but the spurious oscillations tend to be also on the surroundings of this layer. It has been proven numerically that adding artificial viscosity in those surrounding elements improves considerably the behavior of the method.

5.3. Combination of the *hp*-adaptive process and the troubled-cell detector

Up to this point a set of parameters and processes have been introduced. So, before proceeding to the numerical results, it might be useful to summarize the whole method in order to make it clearer.

At the first step, the troubled-cell detector will be set as $\mathcal{S}_m^i = 1 \ \forall K_i$ and the first adaptivity process will consist in *h*-refining all the elements so we can establish values for the predicted error E_m , (introduced in Alg. 1), the predicted gradient G_m (Remark 7 in Sect. 5.1) and the troubled-cell detector $\mathcal{S}_m$. In fact all the elements of the mesh will be simply *h*-refined during the first adaptive steps. In our numerical experiments this will be done the first 2 steps as it is shown in the first part of Alg. 3. This can affect the convergence but it helps detecting the non-smooth layers from the beginning. The parameters of the algorithm are set in such a way they favor the *h*-coarsening if some

regions have been unnecessarily h -refined (see the introduction of Sect. 6). Moreover it is assumed that the initial mesh is very coarse, so it makes sense to h -refine all the elements at the beginning.

In Sect. 4.1 we already introduced the smooth predictor proposed in [32] together with our modifications. Now it only lasts to show how the troubled-cell detector affects the choice of the kind of refinement. Basically, we will follow the smooth predictor scheme for the adaptivity unless the element K_i is set as a troubled-cell ($\mathcal{S}_m^i > 0$), in which case we will avoid the p -refinement. Moreover we have added the coarsening possibility. There are three values associated to this process: T_{ref} , the amount of elements to be refined, $T_{\text{coars,p}}$ and $T_{\text{coars,h}}$, the amount of elements to be p and h refined respectively. They are defined as follows:

$$\begin{aligned} T_{\text{ref}} &= N_m^{\text{nd}} - \lfloor \sigma_{\text{ref}} N_m^{\text{nd}} \rfloor, \\ T_{\text{coars,p}} &= \lceil \sigma_{\text{coarsen}}^p N_m^{\text{nd}} \rceil, \\ T_{\text{coars,h}} &= \lceil \sigma_{\text{coarsen}}^h N_m^{\text{nd}} \rceil. \end{aligned}$$

The parameters σ_{ref} , $\sigma_{\text{coarsen}}^p$ and $\sigma_{\text{coarsen}}^h$ are parameters of the method that must be chosen. Notice that an element will only be h -coarsened if all its children are set to be h -coarsened (which rarely happen if the solution has been properly refined). Instead the p -coarsening occurs always that an element is set to be p -coarsened which can randomly happen if a solution has been correctly refined everywhere. So we have defined two different values for the different coarsening options and we will set the parameter $\sigma_{\text{coarsen}}^p \ll \sigma_{\text{coarsen}}^h$ to minimize such problem. Also, T_{ref} is an approximated value because, in order to homogeneously refine regions with the same order of error, we have included a correction factor such that all the elements that have the 90% of the error of the last element refined are also marked for refinement; the objective of this is to minimize abnormal adaptation steps and keep the symmetry of the problem, if there is so. Also all the elements with 10% of the that threshold error are marked for h -coarsening. The process is specified in Alg. 3.

Summarizing, given $\mathcal{T}_{m-1}$ (mesh), $\mathcal{G}_{m-1}$ (mean elemental gradients), η_{m-1} (*a posteriori* error estimators), $\mathcal{S}_{m-1}$ (troubled-cell flags), $\mathbf{p}_{m-1}$ (elemental polynomial order), the adaptivity flags, and u_{m-1} (discrete solution), we follow the following steps to compute $\mathcal{T}_m$, $\mathcal{G}_m$, η_m , $\mathcal{S}_m$, $\mathbf{p}_m$, and u_m :

1. Compute the mesh $\mathcal{T}_m$ and the order array p_m using $\mathcal{T}_{m-1}$ and p_{m-1} and the adaptivity flags of the previous iteration.
2. Inherit the troubled-cell detector values.
3. Set predicted error E_m (following the Alg. 1) and the predicted gradient G_m (introduced in Remark 7 in Sect. 5.1).
4. Compute the matrix and vector and obtain u_m . When using shock capturing, non-linear iterations with relaxation parameters are used. In particular we have implemented the relaxation scheme proposed in [26], using the same parameter values therein.
5. Compute *a posteriori* error estimator values η_m (see Equation (9)) and $\mathcal{G}_m$ (see Equation (13)).

Algorithm 3: Compute the adaptivity flags for each element in $\mathcal{T}_m$ using the *a posteriori* error estimator of the solution u_m .

```

if  $m \leq 2$  then
  | Set all elements in  $\mathcal{T}_m$  to be  $h$ -refined;
else
   $I = \text{sort}(\eta_m)$           /*  $I$  is such that  $\eta_m^{K_{I_i}} \leq \eta_m^{K_{I_{i+1}}} \ \forall i = 1, \dots, N_m^{\text{nd}}$  */

   $\text{First\_refined\_elem} = N_m^{\text{nd}} - T_{\text{ref}}$ 
   $T_\eta = 0.9 \cdot \eta_m^{\text{first\_refined\_elem}}$           /* Upper threshold error */

   $B_\eta = 0.1 \cdot \eta_m^{\text{first\_refined\_elem}}$           /* Lower threshold error */

  for  $i \leq T_{\text{coars,p}}$  do
    | Set  $K_{I_i}$  to  $p$ -coarse
   $i = 0$ 
  while  $i \leq T_{\text{coars,h}}$  and  $\eta_m^{I_i} < B_\eta$  do
    | Set  $K_{I_i}$  to  $h$ -coarsen
    |  $i = i + 1$ 

  for  $T_{\text{coars,h}} < i \leq N_m^{\text{nd}} - T_{\text{ref}}$  do
    | if  $\eta_m^{I_i} > T_\eta$  then
      |    $\text{first\_refined\_element} = i$ 
      |   Exit loop
    | if  $\mathcal{S}_m^{I_i} > 0$  and  $p_m^i > 1$  then
      |   | Set  $K_{I_i}$  to  $p$ -coarse to linear order
    | for  $\text{first\_refined\_element} < i \leq N_m^{\text{nd}}$  do
      | if  $\eta_m^{I_i} < E_m^i$  and  $\mathcal{S} = 0$  then
        |   | Set  $K_{I_i}$  to  $p$ -refine
      | else
        |   | Set  $K_{I_i}$  to  $h$ -refine

```

6. Set the troubled-cell indicator values $\mathcal{S}_m$ (following the scheme in Alg. 2 in Sect. 5.1). To compute them, we use the information stored in G_m .
7. Compute the adaptivity flags with the Alg. 3.

6. Numerical Results

In the following numerical experiments we want to show the performance of the troubled-cell detector and to find the optimum values of some of the parameters of the method in such a way that we ensure that it detects correctly the layers and, at the same time, it is deactivated when the solution is smooth.

In all the examples, the mesh consists of square elements so we have chosen the length of any of the edges of an element K_i as the characteristic length h_{K_i} . Following [4], we set the parameters of the equations (2)-(3)-(4) to be $c^{\text{ip}} = 10$ and $c^{\text{bms}} = 0.5$. Concerning ξ , since we are interested in recovering the DMP features introduced in [4], we would like to have the incomplete method around the non-smooth regions and the symmetric method elsewhere. To achieve such a property we will use the same value we use to scale ε_{K_i} , $S_{K_i} := \max_{F \in \mathcal{E}_m(K_i)} S_F^{\Gamma_{K_i}(F)}$. This parameter takes values in the interval $[0, 1]$; $S_{K_i} \approx 1$ near the local extrema of the function and $S_{K_i} = 0$ in the smooth regions. Thus, we will define $\xi = 1 - S_{K_i}$ to attain the desired property.

Since we know that the troubled-cell detector can be activated in the first steps even when the solution is smooth and, moreover, since we know (or at least we expect) that linear order will be imposed where it is not, we can choose the parameters to be such that we favor the p -refinement and the h -coarsening. Following this principle, we will justify the choice of some of the parameters that has been done beforehand.

We start setting the values of γ_h , γ_p and γ_n in Alg. 1. Even though the value used for γ_h in [33] is 4, we allow it to take value $\gamma_h = 10$, so our predicted error is bigger and the smooth assumption is fostered. Also γ_p is set to be $\gamma_p = 10$. In fact, the definition of γ_p in [33] was different since they did not use the characteristic length h_{K_i} to compute the predicted error. On the other hand, we have set the predicted error to be $E_m^i = \gamma_n E_{m-1}^{Fm(i)}$ if K_i is identical to $K_{Fm(i)} \in \mathcal{T}_{m-1}$. Since the element has not changed, we could accept that the predicted error should remain the same, and thus, let $\gamma_n = 1$. Instead, we will set it to be slightly larger, $\gamma_n = 1.1$, so that, if the element is not refined after many steps, its predicted value can rise and promote the p -refinement.

Finally we will set the percentage of elements to be refined to be $\sigma_{\text{ref}} = 0.75$ and the coarsening percentage to be $\sigma_{\text{coarsen}}^h = 0.20$; it means that the elements corresponding to the top 25% of the error are (h - or p -) refined and the bottom 20% of the elements are marked for h -coarsening (which not necessarily means they are coarsened, since all children have to be marked to be coarsened). With this last option, we try to compensate the fact that some smooth regions might have been h -refined if they were erroneously considered to be on a troubled region; we want to promote the coarsening of those regions to recover the order of convergence of the hp -adaptivity methods. Finally we set $\sigma_{\text{coarsen}}^p = 0$ to virtually avoid p -coarsening (in fact p -coarsening keeps on taking place near the sharp layers if an element is detected as a troubled-cell).

In the following numerical test we will first set the value of δ_n ensuring that it works for a smooth function. Then, we will show how the method is capable to correctly determine

c^{ip}	c^{bms}	ξ	γ_h	γ_p	γ_n	σ_{ref}	$\sigma_{\text{coarsen}}^h$	$\sigma_{\text{coarsen}}^p$
10	0.5	$1 - S_{K_i}$	10	10	1.1	0.75	0.20	0

Table 1: Values of the fixed parameters of the method.

where the sharp layers are and apply the h - or p -refinement accordingly. The performance of the shock capturing in terms of diminishing the oscillations around the non-smooth layers will also be shown.

6.1. Smooth Solution

The only constant we did not set beforehand is δ_n . The value of this parameter is crucial to define the behavior of the troubled-cell detector since it defines the threshold between what we consider a smooth region or not. As it was explained in Sect. 5.1, δ_n should take values in the interval $(1, \sqrt{\frac{8}{3}})$. The closer it is to 1 the easiest it is for an element to be considered a troubled cell. Certainly we cannot choose it to be too large, since we want to ensure that the troubled-cell detector is activated around the sharp layers; we would like δ_n to be as small as possible.

In order to set this parameter, we will choose a smooth function with variable slope and check how many steps it takes for the method to remove any troubled-cell flag for different values of δ_n . Concretely, we will choose the solution to be a sinus (not aligned to the mesh) so we will solve problem (2) in $\Omega = [0, 1]^2$ with $\mu = 1$, $\beta = 0$, $f = \pi^2(1 + \lambda^2) \sin(\pi(x + \lambda(y - 1)))$ and the boundary conditions $g(x, y) = \sin(\pi(x + \lambda(y - 1)))$, with $\lambda = \frac{1}{\tan \pi/3}$. Even when it looks like a simple solution, it can be difficult to deal with the regions near the maximum and the minimum since the discrete solution can be almost flat in the elements on the extrema while the refined children can have greater slope. Thus it is a good solution to check the proposed formulation. We will run the method with $\delta_n = 1.1, 1.2, 1.3, 1.4, 1.5, \sqrt{\frac{8}{3}}$ and check the evolution of the activation of the detector. Notice that, in this case, the artificial viscosity is not activated since $\|\beta\|_{\infty, K_i} = 0$, thus $\varepsilon_{K_i} = 0 \forall i$.

We take an initial mesh of 4×4 elements and $\mathcal{S}_0 = 1$ and check the evolution of the rate of troubled-cell detected elements. We are interested in knowing how many adaptive steps are necessary to completely remove the troubled-cell flag and how it affects the convergence of the problem. The rate of troubled-cell elements, $\frac{\#\{K \text{ s.t. } \mathcal{S}_m^i > 0\}}{N_m^{\text{el}}}$, is plotted in Fig. 3 as well as the convergence of the *a posteriori* error estimator η and the actual L^2 error. We can observe that for $\delta_n \geq 1.2$ the troubled-cell detector is progressively removed and the optimal convergence recovered after some steps. The plots also show how $\delta_n = 1.1$ seems to be incapable to remove the troubled cells at all and the convergence of the methods is jeopardized. In consequence, we have chosen $\delta_n = 1.2$ and $\delta_n = 1.4$ to be the parameters used in the rest of the numerical test. The value $\delta_n = 1.2$ is the smallest one capable of removing all the troubled-cell detector flags while the method with $\delta_n = 1.4$ seems to be more effective. The plot also verifies how the orders of convergence of the *a posteriori* error and the actual L^2 -error agree.

(a) Rate of troubled-cell elements

(b) Convergence of the a posteriori error estimator η and the weighted L^2 -error $\mu^{-\frac{1}{2}} \|u - u_m\|_{L(\Omega)}$.

Figure 3: Smooth solution. Troubled-cell activation for different values of δ_n .

(a) wIPDG. No troubled-cell activation (b) wIPDG + dbGJV, $\delta_n = 1.2$ (c) wIPDG + dbGJV, $\delta_n = 1.4$

Figure 4: Solutions at the adaptive step $m = 12$.

6.2. Propagation of discontinuities

Let us analyze how good the method is capturing the proper boundary and internal layers. A typical steady-state test is to study the propagation of a boundary discontinuity for a convection-dominated equation. To do so, problem (2) is solved in $\Omega = [0, 1]^2$ with $\mu = 10^{-8}$, $\beta = (\cos(-\pi/3), \sin(-\pi/3))$, $f = 0$, and the boundary conditions $g = 1$ on $y = 1$ and on $\{x = 0\} \cap \{y \geq 0.7\}$ and $g = 0$ on the rest of the boundary. The solution of this problem has two boundary layers (on $x = 1$ and $y = 0$) and an interior layer of width proportional to the value of μ .

We start with a 4×4 element mesh and adaptivity from there. We want to control the spurious oscillations that arise around the sharp layers (see Fig. 4(a)). To do so, we must correctly identify the layers and add the adequate artificial viscosity. In Fig. 5, one can appreciate the evolution of the troubled-cell flags through the mesh adaptivity process and see that the layers are correctly captured by the method for both values of δ_n . This leads to the solutions plotted in Fig. 4(b) and Fig. 4(c). To be more precise, we can follow the evolution of the oscillations through the adaptivity process and see how they are controlled by each method. We define the oscillation in a point $x \in \omega$ as:

$$\text{OSC}_m(x) := \max\{0, u_m(x) - 1, -u_m(x)\}.$$

Then, we define two parameters that gives an idea of how much the DMP has been violated, namely the *mean* OSC defined as the mean of the value of the oscillation in all the degrees of freedom and the *maximum* OSC defined as the maximum value of OSC_m in Ω . We can see in Fig. 6(a) how the artificial viscosity technique is capable to control the spurious oscillations, specially when the mesh is very refined, giving a mean OSC 5 orders of magnitude lower than the one obtained with the wIPDG method without the extra stabilization. The Maximum OSC, on the other hand, is two orders of magnitude lower, as it is shown in Fig. 6(b).

We can observe a slightly oscillatory behavior of the mean OSC with the number of DOFs in in Fig. 6(a). It happens because, as we can observe in Fig. 5(e) and Fig. 5(f) some small regions of the layer may become undetected. The reason is the fact

(a) $m = 3, \delta_n = 1.2$

(b) $m = 6, \delta_n = 1.2$

(c) $m = 12, \delta_n = 1.2$

(d) $m = 3, \delta_n = 1.4$

(e) $m = 6, \delta_n = 1.4$

(f) $m = 12, \delta_n = 1.4$

Figure 5: Troubled-cell for different steps and values of δ_n .

Figure 6: Mean and Maximum oscillation vs. the amount of DOFs.

that there is not a real discontinuity in practice, due to the smearing produced by the artificial viscosity. When the region is over-resolved, at the FE scale the solution starts to be smooth, and artificial viscosity is switched off. This is specially crucial in the region where the exponential boundary layer and the interior layer meet since the size of the elements is even smaller than in the rest of the interior layer. This behavior could be corrected by further complicating the adaptive strategy.

6.3. Propagation with smooth regions

The hp -adaptivity was not completely exploited in the previous example, since the solution was flat in the smooth region, and thus, perfectly captured by low order approximations. Now, let us evaluate the performance of our approach in terms of distribution of h and p adaptivity. We basically took the same solution as before but, instead of imposing $g = 1$ on $y = 1$ and on $\{x = 0\} \cap \{y \geq 0.7\}$, we impose $g = 1 + 0.1 \cos \left(6\pi \left(x - \frac{1-y}{\tan(\frac{\pi}{3})} \right) \right)$. The resulting solution is sinusoidal in the previously flat region (see Fig. 7(a)).

The order map of the solution in Fig. 7(a) for different values of δ_n is plotted in Fig. 7(b) and Fig. 7(c). It can be observed that the method is indeed capable to p -refine in the smooth regions in both cases. Further, a smaller value of δ_n does not imply necessarily that the mean order in the smooth region is lower.

6.4. Parabolic boundary layers

Let us now check how the method performs when using parabolic boundary layers. We solve problem (2) with $\Omega = [0, 1]^2$, $\mu = 10^{-8}$, $\beta = (1, 0)$, $f = 1$, and homogeneous boundary conditions $g = 0$ on $\partial\Omega$. This leads to an exponential layer on $x = 1$ and parabolic layers on $y = 0$ and $y = 1$. In Fig. 8, one can see how in the first adaptivity steps the method is capable to correctly control the oscillations. We have plotted the solution in Fig. 8(b) and Fig. 8(c) with the troubled-cell detector activation map so one can appreciate how its activation is related to the control of spurious oscillations.

In this case we define $\text{OSC}_m(x) = \max\{0, u_m(x) - x\}$ and in Fig. 9 we plot an approximation of its maximum value computed on the quadrature points. We can see

Figure 7: Stabilized solution at $m = 12$. Order map range $p \in \{1, 4\}$.

Figure 8: Solutions at the adaptive step $m = 5$.

how, even though the order of the oscillations is one order lower than in the original method, there is some point in which the oscillation tend to increase. This is specially noticeable for $\delta_n = 1.4$. This phenomena is due to the fact that there are two regions in which the troubled-cell detector is being deactivated after a sufficient refinement of the mesh. On one hand there is the region on the parabolic layers close to $x = 0$, as it can be appreciated in Fig. 10(b), where the gradient is smaller than in the rest of the layer. On the other hand, as we saw in Sect. 6.2, the region where the exponential and the parabolic layer intersect is crucial. In this case, for $\delta_n = 1.4$, the troubled cell deactivation in that region leads to the peak one can appreciate in Fig. 8 that corresponds to the oscillations shown in Fig. 10(c).

6.5. Curved characteristics

Finally, we show how the troubled-cell detector works with curved characteristics. To do so, we solve problem (2) with $\Omega = [0, 1]^2$, $\mu = 10^{-8}$, $\beta = (-2\pi y, 2\pi x)$, $f = 0$, and

Figure 9: Maximum oscillation vs the amount of DOFs.

(a) wIPdG. No troubled-cell activation (b) wIPdG + dbGJV, $\delta_n = 1.2$ (c) wIPdG + dbGJV, $\delta_n = 1.4$

Figure 10: Solutions at the adaptive step $m = 10$.

Figure 11: Troubled-cell for different steps and values of δ_n .

$g = 0$ on all the boundary except for $y = 0$, where we take the step function

$$g(x, 0) = \begin{cases} 1 & x \in (0.35, 0.65), \\ 0 & \text{elsewhere.} \end{cases}$$

This problem leads to a curved step function with internal layers and a boundary layer on $x = 0$. In Fig. 11 it is shown how the troubled-cell detectors perform in capturing such kind of solution. As it was observed in previous examples, it seems that the method with $\delta_n = 1.2$ is capable to better capture the layers in the final adaptivity steps. The gaps in the troubled-cell activation layer originate the oscillation observed in Fig. 12(c).

7. Conclusions

In this paper a new troubled-cell detector has been proposed to work with adaptive meshes. In the numerical section, it has been shown how it works with different kind of solutions with parabolic and exponential boundary and interior layers. In general, the method is capable to correctly detect the sharp layers until a certain level of refinement. It has been shown that, the smaller the value of the parameter δ_n (which determines the amount of variation of the gradient we admit to consider a solution smooth) the better the method performs. Even for the solutions with smooth regions together with sharp layers, the method is capable to let the order of the solution rise in the smooth region while keeping low order and nonlinear stabilization in the sharp layers of the solution.

Figure 12: Solutions at the adaptive step $m = 9$.

On the other hand the extension of the bGJV method proposed in [3] has been shown to be effective in dealing with the spurious oscillations in adaptive meshes. Thus the implementation of a troubled-cell detector ensuring such conditions in the non-smooth layers has been crucial to blend adaptivity and monotonicity-preserving shock capturing.

Acknowledgments

This work has been partially funded by the European Research Council under the FP7 Programme Ideas through the Starting Grant No. 258443 - COMFUS: Computational Methods for Fusion Technology. A. Hierro gratefully acknowledges the support received from the Catalan Government through a FI fellowship.

- [1] M. AINSWORTH AND B. SENIOR, *hp-Finite Element Procedures on Non-Uniform Geometric Meshes: Adaptivity and Constrained Approximation*, in Grid Generation and Adaptive Algorithms, M. W. Bern, J. E. Flaherty, and M. Luskin, eds., no. 113 in The IMA Volumes in Mathematics and its Applications, Springer New York, Jan. 1999, pp. 1–27.
- [2] S. BADIA AND J. BAIGES, *Adaptive Finite Element Simulation of Incompressible Flows by Hybrid Continuous-Discontinuous Galerkin Formulations*, SIAM Journal on Scientific Computing, 35 (2013), pp. A491–A516.
- [3] S. BADIA AND A. HIERRO, *On Monotonicity-Preserving Stabilized Finite Element Approximations of Transport Problems*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2673–A2697.
- [4] S. BADIA AND A. HIERRO, *On discrete maximum principles for discontinuous Galerkin methods*, Computer Methods in Applied Mechanics and Engineering, 286 (2015), pp. 107–122.
- [5] W. BANGERTH AND O. KAYSER-HEROLD, *Data Structures and Requirements for Hp Finite Element Software*, ACM Trans. Math. Softw., 36 (2009), pp. 4:1–4:31.

- [6] R. BISWAS, K. D. DEVINE, AND J. E. FLAHERTY, *Parallel, adaptive finite element methods for conservation laws*, Applied Numerical Mathematics, 14 (1994), pp. 255–283.
- [7] F. BREZZI, L. D. MARINI, AND E. SÜLI, *Discontinuous Galerkin methods for first-order hyperbolic problems*, Mathematical Models and Methods in Applied Sciences, 12 (2004), pp. 1893–1903.
- [8] E. BURMAN, *On nonlinear artificial viscosity, discrete maximum principle and hyperbolic conservation laws*, BIT Numerical Mathematics, 47 (2007), pp. 715–733.
- [9] E. BURMAN AND A. ERN, *Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 3833–3855.
- [10] —, *Stabilized Galerkin approximation of convection-diffusion-reaction equations: discrete maximum principle and convergence*, Mathematics of Computation, 74 (2005), pp. 1637–1652.
- [11] E. BURMAN AND P. HANSBO, *Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 1437–1453.
- [12] B. COCKBURN AND C.-W. SHU, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework*, Mathematics of Computation, 52 (1989), pp. 411–435.
- [13] R. CODINA, *Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods*, Computer Methods in Applied Mechanics and Engineering, 190 (2000), pp. 1579–1599.
- [14] L. DEMKOWICZ, *Computing with hp-Adaptive Finite Elements. Vol 1. One and Two Dimensional Elliptic and Maxwell Problems*, CRC Press, Oct. 2006.
- [15] L. DEMKOWICZ, J. KURTZ, D. PARDO, M. PASZYNSKI, W. RACHOWICZ, AND A. ZDUNEK, *Computing with hp-Adaptive Finite Elements, Vol. 2: Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*, Chapman and Hall/CRC, Boca Raton, 1 edition ed., Nov. 2007.
- [16] L. DEMKOWICZ, W. RACHOWICZ, AND P. DEVLOO, *A Fully Automatic hp-Adaptivity*, Journal of Scientific Computing, 17 (2002), pp. 117–142.
- [17] A. ERN AND J. GUERMOND, *Weighting the Edge Stabilization*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 1655–1677.
- [18] S. GIANI, D. SCHÖTZAU, AND L. ZHU, *An a-posteriori error estimate for -adaptive DG methods for convection-diffusion problems on anisotropically refined meshes*, Computers & Mathematics with Applications, 67 (2014), pp. 869–887.

- [19] D. GILBARG AND N. S. TRUDINGER, *Elliptic Partial Differential Equations of Second Order*, Springer, Mar. 2015.
- [20] J. GUERMOND, M. NAZAROV, B. POPOV, AND Y. YANG, *A Second-Order Maximum Principle Preserving Lagrange Finite Element Technique for Nonlinear Scalar Conservation Equations*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 2163–2182.
- [21] J.-L. GUERMOND, R. PASQUETTI, AND B. POPOV, *Entropy viscosity method for nonlinear conservation laws*, Journal of Computational Physics, 230 (2011), pp. 4248–4267.
- [22] B. GUO AND I. BABUSKA, *The hp version of the finite element method*, Computational Mechanics, 1 (1986), pp. 21–41.
- [23] T. L. HORVÁTH AND M. E. MINCSOVICS, *Discrete maximum principle for interior penalty discontinuous Galerkin methods*, Central European Journal of Mathematics, 11 (2013), pp. 664–679.
- [24] P. HOUSTON, B. SENIOR, AND E. SÜLI, *Sobolev regularity estimation for hp-adaptive finite element methods*, in Numerical Mathematics and Advanced Applications, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer Milan, Jan. 2003, pp. 631–656.
- [25] V. JOHN AND P. KNOBLOCH, *On spurious oscillations at layers diminishing (SOLD) methods for convectiondiffusion equations: Part I A review*, Computer Methods in Applied Mechanics and Engineering, 196 (2007), pp. 2197–2215.
- [26] ———, *On spurious oscillations at layers diminishing (SOLD) methods for convectiondiffusion equations: Part II Analysis for and finite elements*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 1997–2014.
- [27] A. KLÖCKNER, T. WARBURTON, AND J. S. HESTHAVEN, *Viscous Shock Capturing in a Time-Explicit Discontinuous Galerkin Method*, Mathematical Modelling of Natural Phenomena, 6 (2011), pp. 57–83.
- [28] L. KRIVODONOVA, *Limiters for high-order discontinuous Galerkin methods*, Journal of Computational Physics, 226 (2007), pp. 879–896.
- [29] P. KUS, P. SOLIN, AND D. ANDRS, *Arbitrary-level hanging nodes for adaptive hp-FEM approximations in 3d*, Journal of Computational and Applied Mathematics, 270 (2014), pp. 121–133.
- [30] G. LUBE AND G. RAPIN, *Residual-based stabilized higher-order FEM for advection-dominated problems*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 4124–4138.
- [31] C. MAVRIPLIS, *Adaptive mesh strategies for the spectral element method*, Computer Methods in Applied Mechanics and Engineering, 116 (1994), pp. 77–86.

- [32] J. M. MELENK, *hp-Finite Element Methods for Singular Perturbations*, Springer Science & Business Media, Oct. 2002.
- [33] J. M. MELENK AND B. I. WOHLMUTH, *On residual-based a posteriori error estimation in hp-FEM*, *Advances in Computational Mathematics*, 15 (2001), pp. 311–331.
- [34] W. F. MITCHELL, *Adaptive refinement for arbitrary finite-element spaces with hierarchical bases*, *Journal of Computational and Applied Mathematics*, 36 (1991), pp. 65–78.
- [35] W. F. MITCHELL AND M. A. MCCLAIN, *A Survey of hp-Adaptive Strategies for Elliptic Partial Differential Equations*, in *Recent Advances in Computational and Applied Mathematics*, T. E. Simos, ed., Springer Netherlands, Jan. 2011, pp. 227–258.
- [36] P.-O. PERSSON AND J. PERAIRE, *Sub-Cell Shock Capturing for Discontinuous Galerkin Methods*, in 44th AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics.
- [37] J. QIU AND C. SHU, *A Comparison of Troubled-Cell Indicators for Runge–Kutta Discontinuous Galerkin Methods Using Weighted Essentially Nonoscillatory Limiters*, *SIAM Journal on Scientific Computing*, 27 (2005), pp. 995–1013.
- [38] E. SÜLI, P. HOUSTON, AND C. SCHWAB, *hp-finite element methods for hyperbolic problems*, in *The Mathematics of Finite Elements and Applications X*, Elsevier, June 2000, p. 432.
- [39] M. J. VUIK AND J. K. RYAN, *Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes*, *Journal of Computational Physics*, 270 (2014), pp. 138–160.
- [40] H. ZHU AND J. QIU, *An h-adaptive RKDG method with troubled-cell indicator for two-dimensional hyperbolic conservation laws*, *Advances in Computational Mathematics*, 39 (2012), pp. 445–463.