

HP-ADAPTIVE *CELATUS* ENRICHED DISCONTINUOUS GALERKIN METHOD FOR SECOND-ORDER ELLIPTIC SOURCE PROBLEMS*

STEFANO GIANI†

Abstract. This paper presents a new way to enrich finite element methods with nonpolynomial functions without adding any function to the finite element space. For this reason, the method is called *celatus*, which is a Latin word meaning “hidden from view.” Since no nonpolynomial function is added to the finite element space, many issues with standard enriched methods are avoided, among which there is the worsening of the condition of the linear system. In the present work, we focus on second-order elliptic source problems with reentering corners and show that the new method is more computationally efficient than standard finite element methods when used with *hp*-adaptivity.

Key words. enriched method, discontinuous Galerkin, *hp*-adaptivity, elliptic problems

AMS subject classifications. 65N30, 65N50, 49M15

DOI. 10.1137/17M1149912

1. Introduction. Finite element methods (FEMs) have been very successful in the past decades in solving many challenging problems in engineering and physics. In the context of second-order elliptic problems, standard FEMs are known to be very flexible and accurate on problems with smooth solutions [22], i.e., solutions $u \in H^n$, $n \geq 2$. However, standard FEMs struggle with nonsmooth solutions [29]. This is mainly due to the fact that standard polynomial sets of basis functions do not approximate well nonsmooth functions. When this happens, the convergence rate of standard FEMs deteriorate and people look for ways to fix it. The two most successful ways are mesh adaptivity and enrichment. Mesh adaptivity is regaining a good convergence by introducing smaller elements where the solutions are nonsmooth [35]. In recent years, the field of automatic mesh adaptivity driven by error estimators has been particularly active [7, 11, 15, 19, 20]. Such technology is particularly appealing because it automatically adapts the mesh without any a priori knowledge of the features of the solution. To obtain an efficient implementation of the automatic mesh adaptive method, in many cases an error estimator has to be derived for the problem under consideration. The enrichment technique consists of transferring a priori knowledge about the solution into the FEM by adding ad hoc functions to the finite element basis set that capture the nonsmooth behavior of the solution that standard FEMs are not able to resolve efficiently [14, 13, 25, 27, 36]. The method works only if the functions used to enrich the finite element space are the exact ones to describe the behavior of the solution. In general, these added functions are not polynomials, and this can create new issues with the FEM, such as stability of the method itself and an increase of the condition number of the linear system, making the problem harder to solve.

We propose a new method to enrich FEMs that does not involve adding functions to the finite element space. In this way, all the issues listed above are avoided. Nevertheless, a big improvement in the convergence of second-order elliptic source problems

*Submitted to the journal’s Computational Methods in Science and Engineering section September 29, 2017; accepted for publication (in revised form) August 20, 2018; published electronically October 16, 2018.

<http://www.siam.org/journals/sisc/40-5/M114991.html>

†Durham University, Durham DH1 3LE, UK (stefano.giani@durham.ac.uk).

is obtained. Also, since no changes have been done by the proposed enrichment to the finite element space, the method can be applied to existing FEM packages without the need to modify the FEM code, which is particularly useful in cases where the source code of the FEM is not available for modifications.

The proposed enriching method consists of using a reliable and efficient error estimator to filter out the nonsmooth part of the solution from the FEM and approximate it using a descent method. Since the method is based on an error estimator, the same error estimator can be used to adapt the mesh obtaining an enriched adaptive FEM with very little extra effort.

The proposed method can be applied to a variety of FEMs, but in the analysis below we focus on the discontinuous Galerkin (DG) symmetric interior penalty method, which is more difficult to enrich compared to other FEMs. The difficulty lies in the fact that the form of the penalty term and the choice of the penalty parameter depend on the characteristics of the finite element space. The penalty parameter is what makes the method stable and for polynomial finite element spaces has been shown [5, 30, 31] a good choice for the penalty term. But such proof is only true for polynomial functions. Enriching the finite element space with nonpolynomial functions could lead to an unstable method if the penalty term is not adjusted accordingly, which may be not straightforward. The proposed method solves this issue since no nonpolynomial functions are added to the finite element space anyway. The *celatus* method can be applied to any FEM for which a reliable and efficient error estimator is available. For conforming meshes, suitable error estimators for h -adaptive continuous Galerkin FEMs with uniform order of polynomials are presented in [34, 35]. Considering hp -adaptive continuous Galerkin methods on conforming meshes, suitable error estimators are presented in [26]. Moving to nonconforming methods, we have the hp -adaptive continuous Galerkin methods on meshes with arbitrary numbers of hanging nodes presented in [33].

To keep the presentation as simple as possible, only problems with nonsmooth solutions arising from nonconvex geometries are considered in the analysis. But the method can be used also for problems with nonsmooth solutions arising from other situations, and the analysis can be applied also in those cases. The fact that the singular behaviors of the solutions are due to reentering corners in the domain is never used in the analysis; what is necessary to assume is that the solution of the problem has a nonsmooth part as in (2) in section 2. Such form of the nonsmooth part is not restricted to problems on nonconvex domains; on the contrary, it is common for a wider class of problems. In [28, 29], it is shown that (2) is also applicable to elliptic problems with singularities arising from changes in the boundary conditions, to problems with singularities arising from discontinuities in the diffusion coefficients in the interior of the domain, and also to problems with singularities arising from discontinuities in the diffusion coefficients along the boundary at possible reentering corners or due to changes in the boundary conditions. Furthermore, such analysis is presented for both two dimensions and three dimensions (3D). In [21], equation (2) is shown to be also applicable to the biharmonic and the elasticity problem. In order to apply the *celatus* method, the singular behavior functions in (2) have to be known for the problem. If not already available, these can be computed solving Sturm–Liouville eigenvalue problems as explained in [28].

The layout of the paper is as follows. Section 2 introduces the model problem and the analysis of the gradient descent method. In section 3 the *celatus* numerical method is presented and its implementation is discussed. Numerical examples to

illustrate the performances of the *celatus* enrichment method are presented in section 4, just before the conclusions in section 5.

2. Model problem and analysis. We consider the following model problem:

$$(1) \quad \begin{aligned} -\Delta u &= f && \text{in } \Omega, \\ u &= g && \text{on } \partial\Omega, \end{aligned}$$

where $\Omega \in \mathbb{R}^2$ is a nonconvex domain with m reentering corners and with $f \in L^2(\Omega)$ and $g \in H^{1/2}(\partial\Omega)$. It is well known that the solution of such problem is in general not smooth [29, 21], i.e., $u \notin H^2(\Omega)$. In fact, the solution u is the sum of a smooth part $s \in H^2(\Omega)$ and a nonsmooth part r , i.e., $u = s + r$, where the nonsmooth part is defined as

$$(2) \quad r := \sum_{i=1}^m \sum_{j=1}^{n_i} c_{i,j} \psi_{i,j}(\ell_i, \theta_i),$$

where (ℓ_i, θ_i) are polar coordinates systems centered at each reentering corner i , n_i are the number of singular behaviors admitted at each reentering corner i , $c_{i,j}$ are real coefficients, and $\psi_{i,j}$ are singular functions with $\psi_{i,j} \in H^{1+\lambda_{i,j}-\epsilon}(\Omega)$, where $\lambda_{i,j} \in (0, 1]$ and for any $\epsilon > 0$. The numbers of singular behaviors n_i at each reentering corner i is equal to the number of eigenvalues $\lambda_{i,j} < 1$ of the Sturm–Liouville eigenvalue problem at each reentering corner; also, the values n_i are always finite. Moreover, for problem (1), as for many other PDE problems, the singular functions $\psi_{i,j}$ and the corresponding values $\lambda_{i,j}$ can be in general analytically computed.

The method proposed in this work consists of using an FEM to approximate the smooth part s of the solution and to use a descendent method to approximate the values of the coefficients $c_{i,j}$ of the nonsmooth part r of the solution. We assume that the functions $\psi_{i,j}$ in (2) are not in the discrete space, and they cannot be approximated easily. In such case, it is more efficient to apply FEMs to only the smooth part of the solution and apply to the nonsmooth part a different technique that takes advantage of the knowledge about the nonsmooth behavior at the reentering corners. In view of the fact that a priori information about the solution is considered, the method can be seen as a FEM with enrichment.

To simplify the notation we arrange the coefficients $c_{i,j}$ in (2) in a vector \mathbf{c} such as

$$\mathbf{c} = [c_{1,1}, \dots, c_{1,n_1}, c_{2,1}, \dots, c_{2,n_2}, \dots, c_{m,n_m}].$$

We can then define $R(\mathbf{c})$ to be a function that for any choice of \mathbf{c} returns the nonsmooth part r defined in (2). It is straightforward to see from the linearity of the problem that for any choice of f and g , we have that problem (1) has a unique solution u , and such solution is the sum of a smooth part s and a nonsmooth part r . In other words, to any choice of f and g corresponds a unique choice for \mathbf{c} such that $r = R(\mathbf{c})$ and that $u - R(\mathbf{c})$ is smooth. We denote such vector \mathbf{c} with $\mathbf{c}_{f,g}$. For any other choice of \mathbf{c} , the split is not correct in the sense that $u - R(\mathbf{c})$ is not smooth.

In order to present the method, we need to introduce an auxiliary problem related to (1): For any choice of \mathbf{c} , $w_{\mathbf{c}}$ is the solution of

$$(3) \quad \begin{aligned} -\Delta w_{\mathbf{c}} &= f + \Delta R(\mathbf{c}) && \text{in } \Omega, \\ w_{\mathbf{c}} &= g - R(\mathbf{c}) && \text{on } \partial\Omega. \end{aligned}$$

Clearly $w_{\mathbf{c}} = u - R(\mathbf{c})$, where u is the solution of (1). Finally, we denote with $w_{h,\mathbf{c}}$ the finite element approximation of $w_{\mathbf{c}}$; this is necessary because in general the solution of (3) cannot be found analytically.

The method presented in this work computes an approximation $u_{h,\mathbf{c}}$ of the solution of (1) summing a nonsmooth part $R(\mathbf{c})$ for a \mathbf{c} that is an approximation of $\mathbf{c}_{f,g}$ and a finite element approximation $w_{h,\mathbf{c}}$ of the remaining part of the solution $u - R(\mathbf{c})$. In other words, $u_{h,\mathbf{c}} := w_{h,\mathbf{c}} + R(\mathbf{c})$ with the slight abuse of notation of having a subscript h in $u_{h,\mathbf{c}}$ even if the part $R(\mathbf{c})$ is not computed through discretization.

To keep the presentation of the method general and valid for different choices for the used FEM, we assume that problem (3) can be written in variational formulation, and we denote the bilinear form as $a(\cdot, \cdot)$. Also, we denote the corresponding energy norm with $|||\cdot|||$, i.e., $|||w|||^2 := a(w, w)$. We also assume that the bilinear form $a(\cdot, \cdot)$ is coercive and continuous with c_a and C_a the coercivity and the continuity constants, respectively.

The next two results show that the quantity $|||u - u_{h,\mathbf{c}}|||$ has a unique minimum. In the first results only one singular function is considered, i.e., $\dim(\mathbf{c}) = 1$. The general case $\dim(\mathbf{c}) > 1$ is considered after.

LEMMA 1. *Assume that the vector \mathbf{c} contains only one value c and that $c_{f,g}$ is the correct value for the split and moreover that ψ , the singular function corresponding to c , is not in the discrete space. Then we have that the quantity $|||u - u_{h,\mathbf{c}}|||$ as a function of c is strictly convex; i.e., it has a unique minimum.*

Proof. Under the assumption that $\dim(\mathbf{c}) = 1$, we have that $R(c) = c\psi$. With P the projection in the energy norm, we have that $w_{h,\mathbf{c}} = P(s + (c_{f,g} - c)\psi)$, and so

$$\begin{aligned} |||u - u_{h,\mathbf{c}}|||^2 &= |||s + c_{f,g}\psi - (w_{h,\mathbf{c}} + c\psi)|||^2 \\ &= |||s - P(s) + (c_{f,g} - c)(\psi - P(\psi))|||^2 \\ &= (c_{f,g} - c)^2 |||\psi - P(\psi)|||^2 \\ &\quad + 2(c_{f,g} - c)a(\psi - P(\psi), s - P(s)) + |||s - P(s)|||^2 =: T(c) . \end{aligned}$$

The first two derivatives of T are

$$T'(c) = -2(c_{f,g} - c) |||\psi - P(\psi)|||^2 - 2a(\psi - P(\psi), s - P(s)) , \quad T''(c) = 2 |||\psi - P(\psi)|||^2 .$$

Clearly T'' is strictly positive under the assumption that ψ is not in the discrete space, i.e., $P(\psi) \neq \psi$. Therefore, applying the Taylor theorem with the Lagrange remainder we have for any $k \in \mathbb{R}$ and for some value \tilde{c} that

$$T(c + k) = T(c) + kT'(c) + \frac{k^2}{2}T''(\tilde{c}) > T(c) + kT'(c) ,$$

which implies that T is a strictly convex function [8] with a unique minimum. \square

The next theorem shows how to find the position of the minimum of $|||u - u_{h,\mathbf{c}}|||$.

THEOREM 2. *The position of the minimum of $|||u - u_{h,\mathbf{c}}|||$ is defined as*

$$c^* = c_{f,g} + \frac{a(\psi - P(\psi), s - P(s))}{|||\psi - P(\psi)|||^2} .$$

Proof. From the previous lemma, the minimum is the stationary point c^* such that $T'(c^*) = 0$, which means

$$c^* = \frac{2c_{f,g} |||\psi - P(\psi)|||^2 + 2a(\psi - P(\psi), s - P(s))}{2 |||\psi - P(\psi)|||^2} = c_{f,g} + \frac{a(\psi - P(\psi), s - P(s))}{|||\psi - P(\psi)|||^2} . \quad \square$$

Remark 3. Assuming that $a(\cdot, \cdot)$ is continuous and that $||| \cdot |||$ is the energy norm, i.e., $|||w|||^2 := a(w, w)$:

$$c^* \leq c_{f,g} + C_a \frac{|||\psi - P(\psi)||| |||s - P(s)|||}{|||\psi - P(\psi)|||^2} = c_{f,g} + C_a \frac{|||s - P(s)|||}{|||\psi - P(\psi)|||} .$$

Considering standard a priori converge results for FEMs [22, 32, 31] and assuming that they are sharp, the nominator converges faster to 0 than the denominator when the finite element space is refined since s is smooth; this implies that c^* converges to $c_{f,g}$. The definition of $||| \cdot |||$ depends on the FEM used to solve numerically problem (3). In section 3.1 we introduce a possible choice for the FEM to use, and in (15) we define the corresponding energy norm.

COROLLARY 4. *The second derivative of $|||u - u_{h,c}|||$ converges to 0 when the finite element space is refined uniformly.*

Proof. From

$$T''(c) = 2|||\psi - P(\psi)|||^2 ,$$

we have that T'' converges to 0 as the finite element space is refined uniformly since $P(\psi)$ converges to ψ . □

The next theorem shows that the solution of the *celatus* method converges as the error for the smooth part s only.

THEOREM 5 (convergence for $\dim(\mathbf{c}) = 1$). *Denoting by c^* the position of the minimum of $|||u - u_{h,c}|||$, we have that the solution of the *celatus* method converges to the continuous solution u as*

$$|||u - u_{h,c^*}||| \leq C_{\text{conv}} |||s - P(s)||| .$$

Proof. Defining $c^* = c_{f,g} + \Delta c$, we have applying Taylor’s theorem to $|||u - u_{h,c^*}|||^2 = T(c^*)$

$$(4) \quad T(c^*) = T(c_{f,g} + \Delta c) = T(c_{f,g}) + \Delta c T'(c_{f,g}) + \frac{1}{2} \Delta c^2 T''(c_{f,g})$$

since all derivatives of T of order greater than two are zero. The derivatives of T have been already analyzed in Lemma 1; therefore, we have

$$(5) \quad \begin{aligned} T(c_{f,g}) &= |||s - P(s)|||^2 , \\ T'(c_{f,g}) &= -2a(\psi - P(\psi), s - P(s)) , \\ T''(c_{f,g}) &= 2|||\psi - P(\psi)|||^2 . \end{aligned}$$

From Theorem 2 and using the continuity of $a(\cdot, \cdot)$ we have that

$$|\Delta c| \leq C_a \frac{|||s - P(s)|||}{|||\psi - P(\psi)|||} ,$$

which, substituted in (4) together with (5), leads to

$$T(c^*) \leq |||s - P(s)|||^2 + 2C_a^2 |||s - P(s)|||^2 + C_a^2 |||s - P(s)|||^2 ,$$

which concludes the proof. □

Theorem 5 shows that the nonsmooth behaviour of the solution does not affect the convergence rate of the *celatus* method, which therefore converges faster than standard FEMs.

Remark 6. If a conforming FEM is used, from Theorem 5 we have

$$|||u - u_{h,c^*}||| \leq C_{\text{conv}} \inf_{v_h \in V_h^{\text{FE}}} |||s - v_h||| ,$$

where we denoted with V_h^{FE} the conforming finite element space.

The next three results are the generalizations of the previous three for $\dim(\mathbf{c}) > 1$. From now on we use square brackets to indicate the index of an element in a vector, and we define the vectors \mathbf{e}_i as the vectors with all entries equal to 0 except for $\mathbf{e}_i[i]$, i.e., 1.

LEMMA 7. *Assuming that none of the functions $\psi_{i,j}$ in (2) are in the discrete space, then the quantity $|||u - u_{h,\mathbf{c}}|||$ as a function of \mathbf{c} is strictly convex; i.e., it has a unique minimum.*

Proof. With P the projection in the energy norm, we have that $w_{h,\mathbf{c}} = P(s + R(\mathbf{c}_{f,g} - \mathbf{c}))$, and by the linearity of $R(\cdot)$ in all components of \mathbf{c} we have

$$\begin{aligned} |||u - u_{h,\mathbf{c}}|||^2 &= |||s + R(\mathbf{c}_{f,g}) - (w_{h,\mathbf{c}} + R(\mathbf{c}))|||^2 \\ &= |||s - P(s) + (I - P)R(\mathbf{c}_{f,g} - \mathbf{c})|||^2 \\ &= |||s - P(s)|||^2 + \sum_{z=1}^{\dim(\mathbf{c})} (\mathbf{c}_{f,g}[z] - \mathbf{c}[z])^2 |||(I - P)R(\mathbf{e}_z)|||^2 \\ &\quad + 2 \sum_{z=1}^{\dim(\mathbf{c})} (\mathbf{c}_{f,g}[z] - \mathbf{c}[z]) a(s - P(s), (I - P)R(\mathbf{e}_z)) \\ &\quad + 2 \sum_{z=1}^{\dim(\mathbf{c})} \sum_{t=z+1}^{\dim(\mathbf{c})} (\mathbf{c}_{f,g}[z] - \mathbf{c}[z])(\mathbf{c}_{f,g}[t] - \mathbf{c}[t]) \\ &\quad a((I - P)R(\mathbf{e}_z), (I - P)R(\mathbf{e}_t)) =: T(\mathbf{c}) . \end{aligned}$$

The derivative of T in the component z is

$$\begin{aligned} \frac{\partial T}{\partial \mathbf{c}[z]}(\mathbf{c}) &= -2(\mathbf{c}_{f,g}[z] - \mathbf{c}[z]) |||(I - P)R(\mathbf{e}_z)|||^2 - 2a(s - P(s), (I - P)R(\mathbf{e}_z)) \\ &\quad - 2 \sum_{t=1, t \neq z}^{\dim(\mathbf{c})} (\mathbf{c}_{f,g}[t] - \mathbf{c}[t]) a((I - P)R(\mathbf{e}_z), (I - P)R(\mathbf{e}_t)) . \end{aligned}$$

The second derivative of T in the components z and t is

$$\frac{\partial^2 T}{\partial \mathbf{c}[z] \partial \mathbf{c}[t]}(\mathbf{c}) = 2a((I - P)R(\mathbf{e}_z), (I - P)R(\mathbf{e}_t)) ,$$

which from the assumption that none of the functions $\psi_{i,j}$ are in the discrete space implies using the coercivity of $a(\cdot, \cdot)$ that the Hessain of T is positive definite, which implies that T is a strictly convex function [8]. □

THEOREM 8. *The position of the minimum of $|||u - u_{h,\mathbf{c}}|||$ is defined as*

$$\begin{aligned} \mathbf{c}^*[z] &= \mathbf{c}_{f,g}[z] + \frac{a(s - P(s), (I - P)R(\mathbf{e}_z))}{|||(I - P)R(\mathbf{e}_z)|||^2} \\ &\quad + \frac{\sum_{t=1, t \neq z}^{\dim(\mathbf{c})} (\mathbf{c}_{f,g}[t] - \mathbf{c}[t]) a((I - P)R(\mathbf{e}_z), (I - P)R(\mathbf{e}_t))}{|||(I - P)R(\mathbf{e}_z)|||^2} . \end{aligned}$$

COROLLARY 9. *The second derivative of $|||u - u_{h,\mathbf{c}}|||$ converges to 0 refining uniformly the finite element space.*

Proof. From the definition of the Hessian of T that all terms $\frac{\partial T}{\partial \mathbf{c}[z]}(\mathbf{c})$, for any t and z , tend to 0 as the finite element space is refined uniformly. \square

The next theorem shows the convergence of the *celatus* method in the case $\dim(\mathbf{c}) > 1$. This case is not a simple extension of the $\dim(\mathbf{c}) = 1$ case since in the latter the upper bound for $|c^* - c_{f,g}|$ only depends on the approximation level of the finite element space; i.e., from Theorem 2 we have

$$|c^* - c_{f,g}| \leq C_a \frac{|||s - P(s)|||}{|||\psi - P(\psi)|||}.$$

Instead, in the case $\dim(\mathbf{c}) > 1$ we have from Theorem 8 using the continuity of $a(\cdot, \cdot)$ that

$$(6) \quad \begin{aligned} |\mathbf{c}^* - \mathbf{c}_{f,g}|_\infty &\leq C_a \max_{z=1, \dots, \dim(\mathbf{c})} \frac{|||s - P(s)|||}{|||(I - P)R(\mathbf{e}_z)|||} \\ &+ C_a \max_{z=1, \dots, \dim(\mathbf{c})} \frac{\sum_{t=1, t \neq z}^{\dim(\mathbf{c})} |\mathbf{c}_{f,g}[t] - \mathbf{c}[t]| |||(I - P)R(\mathbf{e}_t)|||}{|||(I - P)R(\mathbf{e}_z)|||}. \end{aligned}$$

Clearly from (6) we have that the quantity $|\mathbf{c}^* - \mathbf{c}_{f,g}|_\infty$, which is the distance between the minimum \mathbf{c}^* of the error in the energy norm and the value $\mathbf{c}_{f,g}$, depends on \mathbf{c} , which is the point from where the position of the minimum \mathbf{c}^* is calculated. In other words, if \mathbf{c} is far enough from $\mathbf{c}_{f,g}$, the distance $|\mathbf{c}^* - \mathbf{c}_{f,g}|_\infty$ can be very large even if a very fine mesh is used. This dependency has to be taken into consideration.

THEOREM 10 (convergence for $\dim(\mathbf{c}) > 1$). *Denoting by \mathbf{c}^* the position of the minimum of $|||u - u_{h,\mathbf{c}}|||$, we have that the solution of the celatus method converges to the continuous solution u as*

$$(7) \quad |||u - u_{h,\mathbf{c}^*}||| \leq C_{\text{conv}} \left(|||s - P(s)||| + Q\rho \right)$$

with $Q = |\mathbf{c}_{f,g} - \mathbf{c}|_\infty$ and

$$\rho := \dim(\mathbf{c}) \max_{t=1, \dots, \dim(\mathbf{c})} |||(I - P)R(\mathbf{e}_t)|||.$$

Proof. Defining $\mathbf{c}^* = \mathbf{c}_{f,g} + \Delta\mathbf{c}$, we have applying Taylor's theorem to $|||u - u_{h,\mathbf{c}^*}|||^2 = T(\mathbf{c}^*)$

$$(8) \quad T(\mathbf{c}^*) = T(\mathbf{c}_{f,g} + \Delta\mathbf{c}) = T(\mathbf{c}_{f,g}) + \Delta\mathbf{c} \cdot \nabla T(\mathbf{c}_{f,g}) + \frac{1}{2} \Delta\mathbf{c}^T \nabla^2 T(\mathbf{c}_{f,g}) \Delta\mathbf{c}$$

since all derivatives of T of order greater than two are zero. The derivatives of T have been already analyzed in Lemma 7; therefore, we have

$$(9) \quad \begin{aligned} T(\mathbf{c}_{f,g}) &= |||s - P(s)|||^2, \\ \frac{\partial T}{\partial \mathbf{c}[z]}(\mathbf{c}_{f,g}) &\leq 2C_a |||s - P(s)||| |||(I - P)R(\mathbf{e}_z)|||, \\ \frac{\partial^2 T}{\partial \mathbf{c}[z] \partial \mathbf{c}[t]}(\mathbf{c}_{f,g}) &\leq 2C_a |||(I - P)R(\mathbf{e}_t)||| |||(I - P)R(\mathbf{e}_z)|||. \end{aligned}$$

From Theorem 8 and using the continuity of $a(\cdot, \cdot)$ we have that

$$\Delta \mathbf{c}[z] \leq C_a \frac{\|s - P(s)\|}{\|(I - P)R(\mathbf{e}_z)\|} + C_a Q \frac{\rho}{\|(I - P)R(\mathbf{e}_z)\|}.$$

Substituting the result in (8) together with (9) leads to

$$\begin{aligned} T(\mathbf{c}^*) &\leq \|s - P(s)\|^2 + 2C_a^2 \|s - P(s)\|^2 + 2C_a^2 Q \rho \|s - P(s)\| \\ &\quad + C_a^3 \|s - P(s)\|^2 + C_a^3 Q^2 \rho^2 + 2C_a^3 Q \rho \|s - P(s)\|, \end{aligned}$$

which concludes the proof. \square

COROLLARY 11. *Assuming that $\mathbf{c} = \mathbf{c}_{f,g}$ we have from Theorem 10 that*

$$\|u - u_{h,\mathbf{c}^*}\| \leq C_{\text{conv}} \|s - P(s)\|.$$

Corollary 11 shows that in the best-case scenario also for $\dim(\mathbf{c}) > 1$ the method converges as the case $\dim(\mathbf{c}) = 1$. On the other hand, Theorem 10 shows that in the worst-case scenario the convergence may be dominated by ρ , which converges as the most nonsmooth part of the solution. This suggests that in order to have a fast convergence, it is crucial to try to reduce Q moving \mathbf{c} closer to $\mathbf{c}_{f,g}$. This is what inspired us to develop the gradient descent method in section 3.2.

Unfortunately, the quantity $\|u - u_{h,\mathbf{c}}\|$ is not computable for most problems, so it cannot be used directly to find \mathbf{c}^* . We propose to use a reliable and efficient a posteriori error estimator η to approximate \mathbf{c}^* . The main characteristic of a reliable and efficient a posteriori error estimator is that its value is linearly depending on the value of the true error; i.e., there are two positive constants β_1 and β_2 not depending on the size or order of elements in the mesh such that

$$(10) \quad \beta_1 \|u - u_{h,\mathbf{c}}\| \leq \eta \leq \beta_2 \|u - u_{h,\mathbf{c}}\|.$$

In other words, the true error forces the error estimator to behave in a similar way, making possible in practice to find with good approximation the position of the minimum of $\|u - u_{h,\mathbf{c}}\|$.

Reliable and efficient a posteriori error estimators are available for different types of FEMs, and they are normally used to drive mesh adaptivity. The first error estimators were derived for conforming FEMs to be used with h -adaptivity [35]. Later, error estimators for conforming FEMs suitable for hp -adaptivity appeared [11]. Even more recently, examples of reliable and efficient a posteriori error estimators for nonconforming FEMs like for DG methods started to appear. The first example of such error estimator for the DG method can be traced back to [24]. In the past few years more refined analyses have appeared addressing the issue of the robustness of the efficiency for a posteriori error estimators for high-order methods. Two examples of p -robust a posteriori error estimates based on flux reconstructions can be found in [9, 12].

The derivation of reliable and efficient a posteriori error estimators is a topic on its own that deserves to be discussed in separate publications. For the purposes of this work, we assume a reliable and efficient a posteriori error estimator for the FEM used within the *celatus* method is already available.

3. Numerical method and implementation. In this section we introduce the numerical method and discuss its implementation.

3.1. FEM. We assume that problem (3) is numerically approximated using an FEM. Many different FEMs can be used for this task. Among all of them we decided to use the symmetric interior penalty discontinuous Galerkin (SIPDG) method [5].

Any mesh \mathcal{T} used in this work is a subdivision of Ω with K denoting a generic triangular or quadrilateral element. We assume that the subdivision \mathcal{T} is shape regular and constructed via affine mappings $F_K : \hat{K} \rightarrow K$, where \hat{K} is either the reference triangle or the reference quadrilateral. We allow for a maximum of one hanging node per edge, and we denote $\mathcal{E}(\mathcal{T})$ and $\mathcal{E}^{int}(\mathcal{T}) \subset \mathcal{E}(\mathcal{T})$ the set of all edges of the mesh \mathcal{T} and the subset of all interior edges, respectively, and by $\mathcal{E}^{BC}(\mathcal{T}) \subset \mathcal{E}(\mathcal{T})$ the subset of all boundary edges. We define h_K and h_E to be the diameter of the element K and the length of the edge E , respectively.

Now we introduce the polynomial degrees for the approximation in our DG method. Hence, for each element K of the mesh \mathcal{T} we associate a polynomial degree $p_K \geq 1$, and we introduce the vector $\mathbf{p} = \{p_K : K \in \mathcal{T}\}$. We assume that \mathbf{p} is of bounded local variation between pairs of neighboring elements. For any $E \in \mathcal{E}(\mathcal{T})$, we introduce the edge polynomial degree p_E by

$$(11) \quad p_E = \begin{cases} \max(p_K, p_{K'}) & \text{if } E = \partial K \cap \partial K' \in \mathcal{E}^{int}(\mathcal{T}), \\ p_K & \text{if } E = \partial K \cap \partial \Omega \in \mathcal{E}^{BC}(\mathcal{T}). \end{cases}$$

Hence, for a given partition \mathcal{T} of Ω and a vector \mathbf{p} on \mathcal{T} , we define the DG finite element space by

$$(12) \quad V_{\mathbf{p}}(\mathcal{T}) = \{v \in L^2(\Omega) : v|_K \in \mathcal{P}_{p_K}(K), \quad K \in \mathcal{T}\},$$

where for triangular elements $\mathcal{P}_{p_K}(K)$ is the space of polynomials of degree less or equal p_K and for quadrilateral elements is the space of polynomials of degree less or equal p_K with respect to each variable. We also denote h_{\max} and p_{\min} as the diameter of the biggest element in \mathcal{T} and the minimum value in \mathbf{p} , respectively.

Thus, the DG approximation of problem (3) reads as follows: For any vector \mathbf{c} , find $w_{h,\mathbf{c}} \in V_{\mathbf{p}}(\mathcal{T})$ such that

$$(13) \quad a^{\text{DG}}(w_{h,\mathbf{c}}, v_h) = l_{\mathbf{c}}^{\text{DG}}(v_h) \quad \forall v_h \in V_{\mathbf{p}}(\mathcal{T}),$$

where the bilinear forms

$$(14) \quad \begin{aligned} a^{\text{DG}}(u, v) &:= \sum_{K \in \mathcal{T}} \int_K \nabla u \cdot \nabla v \, d\mathbf{x} \\ &\quad - \sum_{E \in \mathcal{E}^{int}(\mathcal{T}) \cup \mathcal{E}^{BC}(\mathcal{T})} \int_E \{ \{ \nabla u \} \} \cdot [v] + \{ \{ \nabla v \} \} \cdot [u] \, ds \\ &\quad + \sum_{E \in \mathcal{E}^{int}(\mathcal{T}) \cup \mathcal{E}^{BC}(\mathcal{T})} \frac{\gamma p_E^2}{h_E} \int_E [u] \cdot [v] \, ds \\ l_{\mathbf{c}}^{\text{DG}}(v) &:= \sum_{K \in \mathcal{T}} \int_K (f + \Delta R(\mathbf{c}))v \, d\mathbf{x} \\ &\quad - \sum_{E \in \mathcal{E}^{BC}(\mathcal{T})} \int_E (g - R(\mathbf{c}))\nabla v \cdot \mathbf{n}_E \, ds \\ &\quad + \sum_{E \in \mathcal{E}^{BC}(\mathcal{T})} \frac{\gamma p_E^2}{h_E} \int_E (g - R(\mathbf{c}))v \, ds, \end{aligned}$$

where γ is the penalty constant; $\llbracket \cdot \rrbracket$ and $\{\!\!\{ \cdot \}\!\!\}$ are the jump and average operators defined in [5], respectively; and \mathbf{n}_E is the unit vector normal to E and pointing out of Ω .

The energy norm for problem (13) is the DG norm defined as

$$(15) \quad |||u|||_{\text{DG}} = \left(\sum_{K \in \mathcal{T}} \|\nabla u\|_{0,K}^2 + \sum_{E \in \mathcal{E}^{\text{int}}(\mathcal{T})} \frac{\gamma p_E^2}{h_E} \|\llbracket u \rrbracket\|_{0,E}^2 + \sum_{E \in \mathcal{E}^{\text{BC}}(\mathcal{T})} \frac{\gamma p_E^2}{h_E} \|u\|_{0,E}^2 \right)^{1/2},$$

where $\|\cdot\|_{0,K}$ and $\|\cdot\|_{0,E}$ are, respectively, the L^2 -norm on an element K and on an edge E .

Considering the SIPDG method, the result in Theorem 5 can be extended using the convergence result in [31].

COROLLARY 12 (convergence of the SIPDG method for $\dim(\mathbf{c}) = 1$). *Denoting by c^* the position of the minimum of $|||u - u_{h,c^*}|||_{\text{DG}}$, we have that the solution of the celatus method converges to the continuous solution u as*

$$|||u - u_{h,c^*}|||_{\text{DG}} \leq C_{\text{conv}} \frac{h_{\max}^{\mu-1}}{p_{\min}^{r-3/2}} \|s\|_{r,\Omega},$$

where the smooth part s of the solution is assumed to be in $s \in H^r(\Omega)$ with $r \geq 2$, where $\|\cdot\|_{r,\Omega}$ is the Sobolev norm on $H^r(\Omega)$ and with $\mu = \min(p_{\min} + 1, r)$.

This result clearly explains the better performance of the *celatus* method compared to the SIPDG method applied directly to (1) as reported in section 4.3. In the standard case the convergence can be found using the results in [31] with the interpolation operator in [6],

$$(16) \quad |||u - u_h|||_{\text{DG}} \leq C_{\text{conv}} \frac{h_{\max}^{\mu-1}}{p_{\min}^{k-3/2}} \|u\|_{k,\Omega},$$

with $u \in H^k(\Omega)$ and with $\mu = \min(p_{\min} + 1, k)$. In case of nonsmooth solutions u , $k < r$ because $H^k(\Omega)$ is the Sobolev space containing the entire solution u , where $H^r(\Omega)$ is the Sobolev space containing only the smooth part of the solution. Moving to the general case, i.e., $\dim(\mathbf{c}) > 1$, we have from Theorem 10 that the worst possible convergence of the *celatus* method with SIPDG is asymptotically the same as the convergence of SIPDG alone (16).

COROLLARY 13 (convergence of the SIPDG method for $\dim(\mathbf{c}) > 1$). *Denoting by \mathbf{c}^* the position of the minimum of $|||u - u_{h,\mathbf{c}^*}|||_{\text{DG}}$, we have that the solution of the celatus method converges to the continuous solution u as*

$$|||u - u_{h,\mathbf{c}^*}|||_{\text{DG}} \leq C_{\text{conv}} \left(\frac{h_{\max}^{\mu_1-1}}{p_{\min}^{r-3/2}} \|s\|_{r,\Omega} + R \dim(\mathbf{c}) \frac{h_{\max}^{\mu_2-1}}{p_{\min}^{k-3/2}} \right),$$

where the smooth part s of the solution is assumed to be in $s \in H^r(\Omega)$ and

$$k := \min\{m : 1 \leq t \leq \dim(\mathbf{c}), R(\mathbf{e}_t) \in H^m(\Omega)\},$$

is the regularity of the least regular part of the solution and with $\mu_1 = \min(p_{\min} + 1, r)$ and $\mu_2 = \min(p_{\min} + 1, k)$.

The error estimator $\eta_{\mathbf{c}}$ for our SIPDG method (13) is defined as

$$(17) \quad \eta_{\mathbf{c}} = \sqrt{\sum_{K \in \mathcal{T}} \left(\eta_{\mathbf{c},R,K}^2 + \eta_{\mathbf{c},J,K}^2 + \eta_{\mathbf{c},F,K}^2 \right)},$$

where the three terms are defined as

$$\begin{aligned} \eta_{\mathbf{c},R,K}^2 &= \frac{h_K^2}{p_K^2} \left\| f + \Delta R(\mathbf{c}) + \Delta w_{h,\mathbf{c}} \right\|_{0,K}^2, \\ \eta_{\mathbf{c},J,K}^2 &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| \llbracket w_{h,\mathbf{c}} \rrbracket \right\|_{0,E}^2 + \sum_{E \in \mathcal{E}^{\text{BC}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| w_{h,\mathbf{c}} - (g - R(\mathbf{c})) \right\|_{0,E}^2, \\ \eta_{\mathbf{c},F,K}^2 &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{h_E}{p_E} \left\| \llbracket \nabla w_{h,\mathbf{c}} \rrbracket \right\|_{0,E}^2. \end{aligned}$$

This error estimator is an example of an energy norm residual-based error estimator [16, 20, 35, 38], which is proven to be reliable in the energy norm. In the case of the SIPDG method used in this paper, the reliability in the energy norm implies

$$\|w_{\mathbf{c}} - w_{h,\mathbf{c}}\|_{\text{DG}} \leq C(\eta_{\mathbf{c}} + \Theta_{\mathbf{c}}),$$

where the constant C does not depend on the size of the elements in the mesh or the order of polynomials used in the finite element space and where $\Theta_{\mathbf{c}}$ is the data approximation term depending on the approximation of $f + \Delta R(\mathbf{c})$ and $g - R(\mathbf{c})$ in the finite element space and which is an asymptotically higher-order term compared to $\eta_{\mathbf{c}}$. Because of $\Theta_{\mathbf{c}}$ is a higher-order term, it is never computed in practice, and from the literature for both smooth and nonsmooth problems, it clearly can be omitted.

3.2. Gradient descent method. The method presented in this section is what distinguishes the present work from others. Here, we use a gradient descent method to approximate \mathbf{c}^* , which is the position of the minimum for the energy norm as shown in Theorem 8. Denoting by \mathbf{c}_h^* the approximation of \mathbf{c}^* computed finding the minimum of the error estimator $\eta_{\mathbf{c}}$, we have from (10) that \mathbf{c}_h^* has to be closed to \mathbf{c}^* for a fine enough finite element space. We introduce the function $F_{\mathbf{c},z}(\cdot)$

$$(18) \quad F_{\mathbf{c},z}(\alpha) := \eta_{\mathbf{c} + \alpha \mathbf{e}_z}^2 - \eta_{\mathbf{c}}^2,$$

which describes the variation of the error estimator along the z component of \mathbf{c} . In order to find the minimum of the residual along the z component, we can simply apply the gradient descent method to $F_{\mathbf{c},z}(\cdot)$. This consists of solving for α the equation

$$(19) \quad F'_{\mathbf{c},z}(\alpha) = 0.$$

Then to complete the computation of the position of the minimum, the same process can be repeated for all remaining components of \mathbf{c} .

In order to express the first derivative of $F_{\mathbf{c},z}(\cdot)$ we need the solutions from an auxiliary family of problems: For any i and j from (2), let z the index in \mathbf{c} for the position of the entry $c_{i,j}$. Then u_z is the solution of

$$(20) \quad \begin{aligned} -\Delta u_z &= \Delta R(\mathbf{e}_z) && \text{in } \Omega, \\ u_z &= -R(\mathbf{e}_z) && \text{on } \partial\Omega. \end{aligned}$$

Clearly $u_z \equiv -\psi_z$. Applying the SIPDG methods to problems (20) for all values of z between 1 and $\dim(\mathbf{c})$, we obtain a family of discrete problems: For any z , $u_{h,z}$ is the solution of

$$(21) \quad a^{\text{DG}}(u_{h,z}, v_h) = l_z^{\text{DG}}(v_h) \quad \forall v_h \in V_{\mathbf{p}}(\mathcal{T}) ,$$

where

$$(22) \quad \begin{aligned} l_z^{\text{DG}}(v) := & \sum_{K \in \mathcal{T}} \int_K \Delta R(\mathbf{e}_z) v \, d\mathbf{x} \\ & - \sum_{E \in \mathcal{E}^{\text{BC}}(\mathcal{T})} \int_E -R(\mathbf{e}_z) \nabla v \cdot \mathbf{n}_E \, ds + \sum_{E \in \mathcal{E}^{\text{BC}}(\mathcal{T})} \frac{\gamma p_E^2}{h_E} \int_E -R(\mathbf{e}_z) v \, ds . \end{aligned}$$

Now we have all the ingredients to express (19): From (18) and (17) we have

$$F_{\mathbf{c},z}(\alpha) = \sum_{K \in \mathcal{T}} (\eta_{\mathbf{c}+\alpha\mathbf{e}_z,R,K}^2 - \eta_{\mathbf{c},R,K}^2 + \eta_{\mathbf{c}+\alpha\mathbf{e}_z,J,K}^2 - \eta_{\mathbf{c},J,K}^2 + \eta_{\mathbf{c}+\alpha\mathbf{e}_z,F,K}^2 - \eta_{\mathbf{c},F,K}^2) ,$$

where

$$\begin{aligned} \eta_{\mathbf{c}+\alpha\mathbf{e}_z,R,K}^2 &= \frac{h_K^2}{p_K^2} \left\| f + \Delta R(\mathbf{c} + \alpha\mathbf{e}_z) + \Delta w_{h,\mathbf{c}+\alpha\mathbf{e}_z} \right\|_{0,K}^2 \\ &= \frac{h_K^2}{p_K^2} \left\| (f + \Delta R(\mathbf{c}) + \Delta w_{h,\mathbf{c}}) + \alpha(\Delta R(\mathbf{e}_z) + \Delta u_{h,z}) \right\|_{0,K}^2 , \\ \eta_{\mathbf{c}+\alpha\mathbf{e}_z,J,K}^2 &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| \llbracket w_{h,\mathbf{c}+\alpha\mathbf{e}_z} \rrbracket \right\|_{0,E}^2 \\ &\quad + \sum_{E \in \mathcal{E}^{\text{BC}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| w_{h,\mathbf{c}+\alpha\mathbf{e}_z} - (g - R(\mathbf{c} + \alpha\mathbf{e}_z)) \right\|_{0,E}^2 \\ &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| \llbracket w_{h,\mathbf{c}} + \alpha u_{h,z} \rrbracket \right\|_{0,E}^2 \\ &\quad + \sum_{E \in \mathcal{E}^{\text{BC}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| (w_{h,\mathbf{c}} - (g - R(\mathbf{c})) + \alpha(u_{h,z} + R(\mathbf{e}_z))) \right\|_{0,E}^2 , \\ \eta_{\mathbf{c}+\alpha\mathbf{e}_z,F,K}^2 &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{h_E}{p_E} \left\| \llbracket \nabla w_{h,\mathbf{c}+\alpha\mathbf{e}_z} \rrbracket \right\|_{0,E}^2 \\ &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{h_E}{p_E} \left\| \llbracket \nabla w_{h,\mathbf{c}} + \alpha \nabla u_{h,z} \rrbracket \right\|_{0,E}^2 , \end{aligned}$$

and consequently

$$\begin{aligned} \eta_{\mathbf{c}+\alpha\mathbf{e}_z,R,K}^2 - \eta_{\mathbf{c},R,K}^2 &= \frac{h_K^2}{p_K^2} \alpha^2 \left\| \Delta R(\mathbf{e}_z) + \Delta u_{h,z} \right\|_{0,K}^2 \\ &\quad + 2\alpha \frac{h_K^2}{p_K^2} \int_K (f + \Delta R(\mathbf{c}) + \Delta w_{h,\mathbf{c}})(\Delta R(\mathbf{e}_z) + \Delta u_{h,z}) \, d\mathbf{x} \\ &= \alpha^2 G_{R,K,z} + 2\alpha D_{\mathbf{c},R,K,z} , \end{aligned}$$

$$\begin{aligned}
 \eta_{\mathbf{c}+\alpha\mathbf{e}_z,J,K}^2 - \eta_{\mathbf{c},J,K}^2 &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left(\alpha^2 \left\| \llbracket u_{h,z} \rrbracket \right\|_{0,E}^2 + 2\alpha \int_E \llbracket w_{h,\mathbf{c}} \rrbracket \llbracket u_{h,z} \rrbracket ds \right) \\
 &\quad + \sum_{E \in \mathcal{E}^{\text{BC}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left(\alpha^2 \left\| u_{h,z} + R(\mathbf{e}_z) \right\|_{0,E}^2 \right. \\
 &\quad \left. + 2\alpha \int_E (w_{h,\mathbf{c}} - (g - R(\mathbf{c}))) (u_{h,z} + R(\mathbf{e}_z)) ds \right) \\
 &= \alpha^2 G_{J,K,z} + 2\alpha D_{\mathbf{c},J,K,z} , \\
 \eta_{\mathbf{c}+\alpha\mathbf{e}_z,F,K}^2 - \eta_{\mathbf{c},F,K}^2 &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{h_E}{p_E} \left(\alpha^2 \left\| \llbracket \nabla u_{h,z} \rrbracket \right\|_{0,E}^2 \right. \\
 &\quad \left. + 2\alpha \int_E \llbracket \nabla w_{h,\mathbf{c}} \rrbracket \llbracket \nabla u_{h,z} \rrbracket ds \right) \\
 &= \alpha^2 G_{F,K,z} + 2\alpha D_{\mathbf{c},F,K,z} ,
 \end{aligned}$$

where

$$\begin{aligned}
 G_{R,K,z} &= \frac{h_K^2}{p_K^2} \left\| \Delta R(\mathbf{e}_z) + \Delta u_{h,z} \right\|_{0,K}^2 , \\
 D_{\mathbf{c},R,K,z} &= \frac{h_K^2}{p_K^2} \int_K (f + \Delta R(\mathbf{c}) + \Delta w_{h,\mathbf{c}}) (\Delta R(\mathbf{e}_z) + \Delta u_{h,z}) dx , \\
 G_{J,K,z} &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{\gamma^2 p_E^3}{h_E} \left\| \llbracket u_{h,z} \rrbracket \right\|_{0,E}^2 + \sum_{E \in \mathcal{E}^{\text{BC}}(K)} \frac{\gamma^2 p_E^3}{h_E} 2 \left\| u_{h,z} + R(\mathbf{e}_z) \right\|_{0,E}^2 , \\
 D_{\mathbf{c},J,K,z} &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{\gamma^2 p_E^3}{h_E} \int_E \llbracket w_{h,\mathbf{c}} \rrbracket \llbracket u_{h,z} \rrbracket ds \\
 &\quad + \sum_{E \in \mathcal{E}^{\text{BC}}(K)} \frac{\gamma^2 p_E^3}{h_E} 2 \int_E (w_{h,\mathbf{c}} - (g - R(\mathbf{c}))) (u_{h,z} + R(\mathbf{e}_z)) ds , \\
 G_{F,K,z} &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{h_E}{p_E} \left\| \llbracket \nabla u_{h,z} \rrbracket \right\|_{0,E}^2 , \\
 D_{\mathbf{c},F,K,z} &= \frac{1}{2} \sum_{E \in \mathcal{E}^{\text{int}}(K)} \frac{h_E}{p_E} \int_E \llbracket \nabla w_{h,\mathbf{c}} \rrbracket \llbracket \nabla u_{h,z} \rrbracket ds .
 \end{aligned}$$

Putting all together we obtain

$$\begin{aligned}
 F_{\mathbf{c},z}(\alpha) &= \alpha^2 \sum_{K \in \mathcal{T}} (G_{R,K,z} + G_{J,K,z} + G_{F,K,z}) \\
 (23) \quad &\quad + 2\alpha \sum_{K \in \mathcal{T}} (D_{\mathbf{c},R,K,z} + D_{\mathbf{c},J,K,z} + D_{\mathbf{c},F,K,z}) .
 \end{aligned}$$

Finally, we can express (19) taking the derivative of (23):

$$\begin{aligned}
 0 = F'_{\mathbf{c},z}(\alpha) &= 2\alpha \sum_{K \in \mathcal{T}} (G_{R,K,z} + G_{J,K,z} + G_{F,K,z}) \\
 (24) \quad &\quad + 2 \sum_{K \in \mathcal{T}} (D_{\mathbf{c},R,K,z} + D_{\mathbf{c},J,K,z} + D_{\mathbf{c},F,K,z}) .
 \end{aligned}$$

It is straightforward to see that the solution α of (24) is

$$(25) \quad \alpha = \frac{-\sum_{K \in \mathcal{T}} (D_{\mathbf{c},R,K,z} + D_{\mathbf{c},J,K,z} + D_{\mathbf{c},F,K,z})}{\sum_{K \in \mathcal{T}} (G_{R,K,z} + G_{J,K,z} + G_{F,K,z})}.$$

Algorithm 1 describes in algorithmic form the descent method to find the minimum of $\eta_{\mathbf{c}}$ based on (25). Due to the nature of the problem, the position of the minimum along each component can be found separately, and this is exploited in Algorithm 1. Compared to a standard FEM that consists of only solving a linear problem, Algorithm 1 is more expensive to run since multiple linear problems have to be solved, but the accuracy of Algorithm 1 is much higher, compensating for its cost as reported in sections 4.3 and 4.4. However, the total computational cost of Algorithm 1 heavily depends on the linear solver used. If a direct solver is used to solve the linear systems, then the computational cost can be reduced because all linear problems in Algorithm 1 have the same stiffness matrix. So, the factorization can be done only once, and then the solution can be computed multiple times for different right-hand sides. If an iterative solver is used, the choice of the solver and the preconditioner may have a big impact on the complexity of Algorithm 1. It has been reported in the literature [37] that the complexity of certain multigrid methods is $\mathcal{O}(N)$ for 3D problems with N the dimension of the problem. Comparing this with direct solvers, we have that the complexity of Gauss elimination for dense matrices [10] is $\mathcal{O}(N^3)$, and for sparse matrices the complexity is estimated in [1] as $\mathcal{O}(N^{4/3})$ for 3D problems. Other direct methods for sparse matrices have complexities between these two extrema. In view of this, it could be still cheaper for big enough N such that $\text{maxits} \times \dim(\mathbf{c}) < N$ to run Algorithm 1 calling an iterative solver multiple times rather than running Algorithm 1 with a direct solver and factorizing the

Algorithm 1. Gradient descent algorithm: *For parameters: the mesh \mathcal{T} , the finite element space $V_{\mathbf{p}}(\mathcal{T})$, the initial guess \mathbf{c} , the maximum number of iterations maxits , and the acceptable tolerance on the position of the minimum tol . The algorithm returns the vector \mathbf{c}_h^* of the minimum the solution $u_{h,\mathbf{c}_h^*} = w_{h,\mathbf{c}_h^*} + R(\mathbf{c}_h^*)$ and the residual $\eta_{\mathbf{c}_h^*}$.*

```

for  $z = 1, \dots, \dim(\mathbf{c})$  do
  Compute  $u_{h,z}$  solving problem (20)
end for
 $n := 0$ 
 $\mathbf{c}_0 := \mathbf{c}$ 
repeat
   $\mathbf{c}_{\text{old}} := \mathbf{c}_n$ 
  Compute  $w_{h,\mathbf{c}_n}$  solving problem (13)
  for  $z = 1, \dots, \dim(\mathbf{c}_n)$  do
    Compute  $\alpha$  using (25)
     $\mathbf{c}_n[z] := \mathbf{c}_{\text{old}}[z] + \alpha$ 
  end for
   $n := n + 1$ 
until  $|\mathbf{c}_n - \mathbf{c}_{\text{old}}| < \text{tol}$  OR  $n \geq \text{maxits}$ 
Return  $\mathbf{c}_h^* := \mathbf{c}_n$ 
Return  $u_{h,\mathbf{c}_h^*}$  which is the sum of  $w_{h,\mathbf{c}_n}$  and  $R(\mathbf{c}_n)$ , i.e.  $u_{h,\mathbf{c}_h^*} := w_{h,\mathbf{c}_n} + R(\mathbf{c}_n)$ 
Compute and return  $\eta_{\mathbf{c}_h^*}$ 

```

stiffness matrix only once. In case less efficient iterative solvers are used, it could be more convenient to use a direct solver. In sections 4.3 and 4.4, we further analyze the computational cost of Algorithm 1.

4. Numerical experiments. In this section we present a variety of numerical examples to illustrate the performances of the *celatus* method. We are going to consider two domains:

- (i) An L-shape domain constructed removing a quarter from a square domain, i.e., $\Omega = [-1, 1]^2 / [0, 1]^2$.
- (ii) A T-shape domain constructed removing two square portions from a square domain, i.e., $\Omega = ([0, 3] \times [0, 2]) / ([0, 1]^2 \cup [2, 3] \times [0, 1])$.

We also are going to use two sequences of meshes:

- (i) A uniformly h -refined sequence of meshes for the L-shape domain. Such sequence is constructed starting from a sequence of structured square element meshes on the square domain $[-1, 1]^2$ with 17×17 , 33×33 , 65×65 , and 129×129 nodes. From this sequence, the sequence for the L-shape domain is constructed removing a quarter of the mesh from each of the structured meshes. This is a uniformly h -refined sequence of meshes because each mesh in the sequence can be seen as the complete h -refinement of the previous mesh in the sequence.
- (ii) A uniformly p -refined sequence of meshes for the L-shape domain. Such sequence is constructed taking the first mesh in the h -refined sequence of meshes for the L-shape domain and letting the polynomial order increase from 1 to 4 on all elements. This is a uniformly p -refined sequence of meshes because each mesh in the sequence can be seen as the complete p -refinement of the previous mesh in the sequence.

In the rest of the section we use the following five problems:

- (i) On the L-shape domain let define f and g such that problem (1) has solution $u = \sin(\pi(x+1)) \sin(\pi(y+1))$.
- (ii) On the L-shape domain let define f and g such that problem (1) has solution $u = (x^2 + y^2)^{1/3} + \sin(\pi(x+1)) \sin(\pi(y+1))$.
- (iii) On the L-shape domain let define $f = 1$ and $g = 0$.
- (iv) On the L-shape domain let define f and g such that problem (1) has solution $u = 10(x^2 + y^2)^{1/3} + \sin(\pi(x+1)) \sin(\pi(y+1))$.
- (v) On the T-shape domain let define $f = 1$ and $g = 0$.

All the codes used to produce the results in this section are written in FORTRAN, and all linear systems are solved using Multifrontal Massively Parallel Solver (MUMPS); see, e.g. [2, 3, 4]. All computations are performed on an Intel Core i7 PC with 16 GB of RAM at 3.60 GHz. The codes only take the default optimization of the machine; i.e., they are not parallel codes.

4.1. Reliability of the error estimator in finding the minimum. In this subsection, we numerically explore the accuracy of relation (10) or, in other words, how accurate is $\eta_{\mathbf{c}}$ when used to find $\mathbf{c}_{f,g}$. As discussed in section 2, \mathbf{c}^* is unique, and when \mathbf{c}^* is close to $\mathbf{c}_{f,g}$, $w_{\mathbf{c}} = s + R(\mathbf{c}_{f,g} - \mathbf{c}^*)$ is close to be a smooth function. This is also true for its approximation $w_{h,\mathbf{c}}$. Since the error estimator $\eta_{\mathbf{c}}$ is sensitive to any singular behavior, it can be used to estimate the smoothness of $w_{h,\mathbf{c}}$ and therefore the region around the optimal value $\mathbf{c}_{f,g}$ where the solution is almost smooth. The accuracy of this method is based on the accuracy of the error estimator $\eta_{\mathbf{c}}$ to detect the presence in the computed solution $w_{h,\mathbf{c}}$ of any singular behavior. Such capability has been already established in the literature, and it is at the basis of many

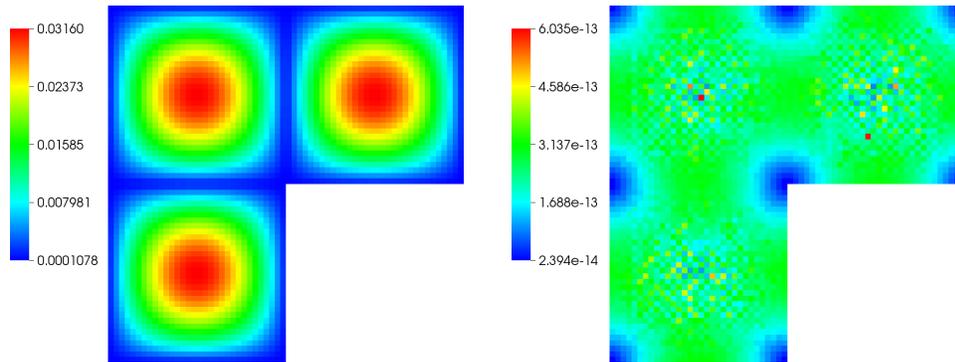


FIG. 1. Distribution of the error estimator values on the mesh for the smooth solution and for $p = 1$ (left) and $p = 6$ (right).

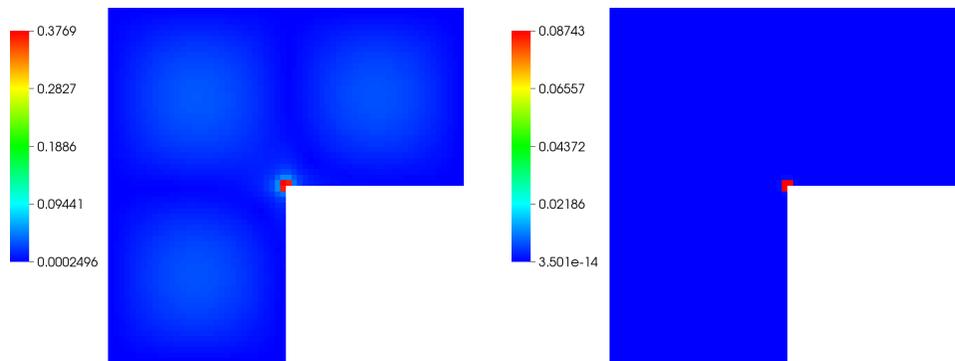


FIG. 2. Distribution of the error estimator values on the mesh for the nonsmooth solution and for $p = 1$ (left) and $p = 6$ (right).

mesh adaptivity techniques [17, 18, 38]. In order to show this in the context of the present paper, let η_1 and η_2 be the error estimators of the SIPDG solutions $u_{h,1}$ and $u_{h,2}$, where $u_{h,1}$ is an approximation of the solution of problem (i) and $u_{h,2}$ is an approximation of the solution of problem (ii). Clearly the solution $u_{h,1}$ is smooth, and the solution $u_{h,2}$ is not smooth. In Figure 1 are the plots of the distribution of values of η_1 on the second mesh in the uniformly h -refined sequence and for p on all elements equal to either 1 or 6. In Figure 2 are the plots of the distribution of values of η_2 on the same mesh for the same values of p . In the case of η_1 the values of the error estimator are all closer to 0 compared to η_2 , and they decrease very rapidly when p increase. Comparing η_1 with η_2 , it is clear that in the latter the error estimator has correctly identified the presence of singular behavior at the reentering corner marking the region with very high values, and this is the same for all values of p since singular behavior cannot be resolved efficiently with high-order polynomials. To further show that the error estimator η_c is suitable for searching for the position of the minimum \mathbf{c}^* , we now move our attention to the *celatus* method applied to problem (ii) and with $\psi(r, \theta) = (x^2 + y^2)^{1/3}$. Such choice of ψ is able to capture the singular behavior at the reentering corner, and in view of the decomposition $u = s + c_{f,g}\psi$, clearly the optimal value for $c_{f,g}$ is 1. We then consider values of c in the range $[-5, 5]$, and for each value of c we numerically solve problem (3) and then compute η_c for each c . In

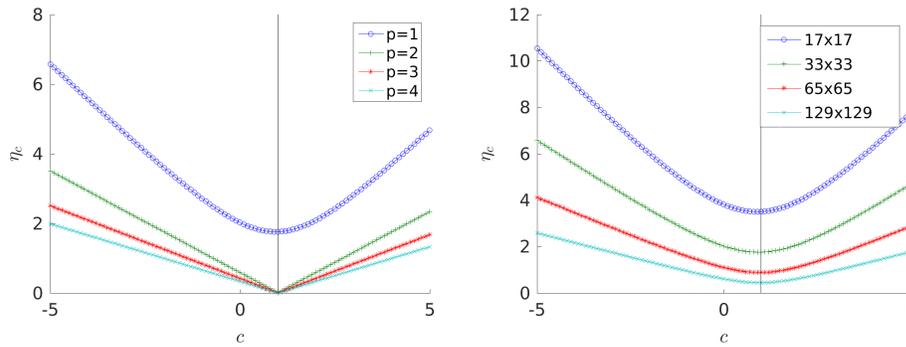


FIG. 3. Value of η_c as a function of c for different values of p on the left and for different meshes with $p = 1$ on the right.

TABLE 1

Error in the value of c_n in Algorithm 1 for a series of structured meshes with $p = 1$ and corresponding values for η_{c_n} for each iteration of Algorithm 1.

Mesh	Iteration	$ c_n - c_{f,g} $	η_{c_n}
17×17	0	1	1.6367
	1	0.1619	0.8289
33×33	0	1	1.0465
	1	0.1029	0.4240
65×65	0	1	0.6658
	1	0.0653	0.2149
129×129	0	1	0.4221
	1	0.0414	0.1084

Figure 3 η_c is plotted as a function of c . On the left we used the uniformly p -refined sequence of meshes, on the right the uniformly h -refined sequence of meshes with $p = 1$. The value of $c = 1$ is marked by vertical black lines, and clearly the minimum of η_c is reached in the correct neighborhood for all simulations suggesting that in this case $c_h^* \approx 1$, which is correct since by construction $c_{f,g} = 1$. Moreover, we can notice that the value of the error estimator is decreasing with p and also by refining the mesh. This is explainable due to the fact that a richer finite element space can better approximate the solution. Also, it is clear that increasing p or reducing the sizes of the elements, the second derivative of η_c as a function of c is decreasing; this is exactly the behavior of $\| \|u - u_{h,c}\| \|$ described in Corollary 4, suggesting that the error estimator has a similar behavior.

4.2. Accuracy in localizing the minimum. Next, we show the performances of the *celatus* method, Algorithm 1, in localizing the minimum. In order to do that the considered test case is problem (ii) and with $\psi(r, \theta) = (x^2 + y^2)^{1/3}$. As already explained in the previous subsection, the optimal value for $c_{f,g}$ is 1. The parameters for Algorithm 1 are initial value of $c = 0$, $\text{maxits} = 100$, and $\text{tol} = 1E - 6$.

In Table 1 we report the behavior of the quantity $|c_n - c_{f,g}|$ during the iterations of Algorithm 1 for the uniformly h -refined sequence of meshes with $p = 1$. For all meshes the algorithm takes two iterations before converging, and it can be seen that the error decreases using finer meshes. This is in line with Theorem 2, which shows convergence of c^* to $c_{f,g}$ when the finite element space gets richer. A similar behavior can be seen for the error estimator η_{c_n} ; this can be easily explained by the fact that the approximation error reduces when the finite element space gets finer.

In Table 2 we report the behavior of the quantity $|c_n - c_{f,g}|$ during the iterations of Algorithm 1 for the uniformly p -refined sequence of meshes. For all meshes the algorithm takes two iterations to converge, which is the same as before, but, compared with Table 1, it is clear that in this case the error and the residual decrease much faster. This suggests much better approximation of the location of the minimum, and smaller residuals imply better approximation of the solution. This is confirmed by Figure 4 (left), where the error $|c_h^* - c_{f,g}|$ is plotted for the two sequences of meshes. On closer inspection, it can be seen that the error on the uniformly p -refinement sequence converges exponentially. This is what we consider the main advantage of our method compared to any standard polynomial FEM. With our method it is possible to achieve exponential convergence for nonsmooth solutions with uniform p -refinement alone. Standard FEMs can only achieve polynomial convergence with p -refinement alone for nonsmooth solutions since increasing the order of p , without reducing the size of the elements where the solution is nonsmooth, is not enough. To get exponential convergence for nonsmooth solutions with standard FEMs, it is necessary to use hp -adaptivity. With our method this is not necessary since the nonsmooth part of the solution is removed from the finite element problem and handled by the gradient descent method. This improves the convergence of the finite element part and in return improves the convergence of the overall method. This aspect is further explored in the next sections.

Next, we consider problem (iii), for which the analytical solution is unknown. Solving the Sturm–Liouville eigenvalue problem around the reentering corner, we have that there is a single eigenvalue less than 1, and the corresponding singular behavior

TABLE 2

Error in the value of c_n in Algorithm 1 for a series of structured meshes with 192 elements and different values of p and corresponding values for η_{c_n} for each iteration of Algorithm 1.

Mesh	Iteration	$ c_n - c_{f,g} $	η_{c_n}
$p = 1$	0	1	1.6367
	1	0.1619	0.8289
$p = 2$	0	1	0.9265
	1	0.0003	0.0153
$p = 3$	0	1	0.6623
	1	4.8795e-05	0.0003
$p = 4$	0	1	0.5246
	1	5.7818e-08	3.8663e-06

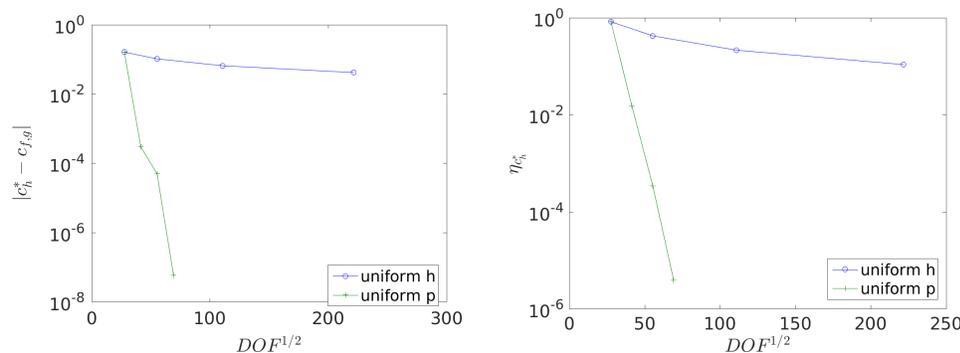


FIG. 4. (left) Error of the position of the minimum in Algorithm 1 for the two sequences of meshes. (right) Value of η_{c_n} per iteration of Algorithm 1 for two sequences of meshes.

TABLE 3

Value of c_n in Algorithm 1 for the uniformly h -refined sequence of meshes with $p = 1$ and corresponding values for η_{c_n} for each iteration of Algorithm 1.

Mesh	Iteration	c_n	η_{c_n}
17×17	0	0	0.3767
	1	0.5044	0.3553
33×33	0	0	0.1962
	1	0.4789	0.1807
65×65	0	0	0.1024
	1	0.4551	0.9129e-01
129×129	0	0	0.5390e-01
	1	0.4372	0.4590e-01

TABLE 4

Value of c_n in Algorithm 1 for the uniformly p -refined sequence of meshes and corresponding values for η_{c_n} for each iteration of Algorithm 1.

Mesh	Iteration	c_n	η_{c_n}
$p = 1$	0	0	0.3767
	1	0.5044	0.3553
$p = 2$	0	0	0.6300e-01
	1	0.4023	0.1061e-01
$p = 3$	0	0	0.4850e-01
	1	0.4020	0.2107e-02
$p = 4$	0	0	0.4114e-01
	1	0.4020	0.8097e-03

is $\psi(r, \theta) = r^{2/3} \sin(2/3 \theta)$. Even if the singular behavior is known, the solution u remains unknown since the exact value of $c_{f,g}$ and the smooth part s are not known. We choose the parameters for Algorithm 1 as before, and in Table 3 we report the behavior of the quantity c_n during the iterations of Algorithm 1 for the uniformly h -refined sequence of meshes with $p = 1$. For all meshes the algorithm takes two iterations before converging. Since the exact value of $c_{f,g}$ is not known, it is difficult to estimate the convergence, but by the naked eye the values c_n^* are settling down. Also, the value of the error estimator is decreasing refining the mesh, suggesting a decrease of the approximation error for the smooth part of the solution. As in the previous example, this leads to a decay of the total error $\| \|u - u_{h,c}\| \|_{DG}$.

In Table 4 we report the behavior of c_n and η_{c_n} on the uniformly p -refined sequence of meshes.

Comparing Tables 3 and 4 to the two previous tables, the same conclusions can be reached for the convergence speeds on the two sequences of meshes. This is confirmed by Figure 5, where the behavior of the error estimator is reported.

4.3. Comparison with standard FEMs. In this subsection, we compare the performances of the *celatus* method against a standard FEM. In particular we are going to compare against the SIPDG method in section 3.1 applied directly to problem (1). We consider problem (iv) with $\psi(r, \theta) = (x^2 + y^2)^{1/3}$, which implies that the optimal value for $c_{f,g}$ is 10. The parameters for Algorithm 1 are initial value of $c = 0$, $\text{maxits} = 100$, and $\text{tol} = 1E - 8$. To do the comparison we start from a mesh of 12 square elements with $p = 1$, and then we apply either uniform- h or uniform- p refinement. In Figure 6 we report the decay of the errors in the L^2 and the DG norms for our method and for the standard SIPDG method. As can be seen, our method is always better than the standard method and really outperforms the standard one with uniform- p refinement. Considering only uniform- h refinement, we have that the

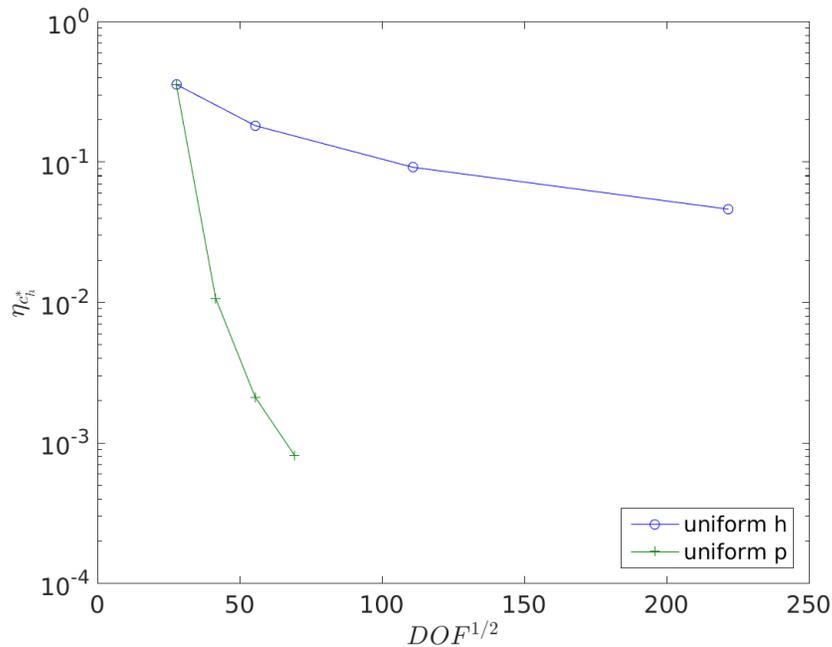


FIG. 5. Minimum value of η_{c_n} per iteration of Algorithm 1 for the two sequences of meshes.

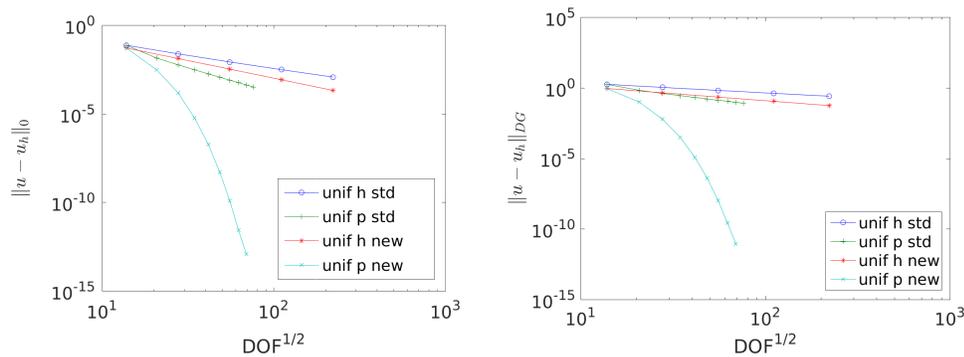


FIG. 6. (left) Convergence for the methods in the L^2 norm: std stands for standard SIPDG, and new stands for the celatus method. (right) Convergence for the methods in the DG norm.

two methods converge only polynomially, which is not surprising since finite elements can only converge as such if the order of the elements is kept constant. However, increasing the order of polynomials, FEMs can potentially converge exponentially fast, but only if the solution is smooth. The solution of the considered problem is not smooth at all, and so the standard method converges only polynomially also with uniform- p refinement. However, with our proposed method, the nonsmooth part is almost completely removed from the solution of the finite element problem. Therefore, from Corollary 12 the convergence of the overall method follows the convergence of the FEM part, which is exponential since only the smooth part is approximated by the FEM. In other words, our method converges exponentially fast even on structured meshes for nonsmooth problems where standard methods do not.

4.4. Comparison with standard adaptive FEMs. In this subsection, we compare the performances of the *celatus* method against a standard FEM with adaptivity, and we are going to consider problem (iii). For this problem the exact solution is unknown. For the comparison, we set the parameters of Algorithm 1 as $c = 0$, $\text{maxits} = 100$, and $\text{tol} = 1E - 6$. The initial mesh is the same used in the previous subsection comprised of 12 square elements with $p = 1$. The comparison is done using uniform- h , adaptive- h and adaptive- hp refinement. In the adaptive schemes, we use a fixed fraction criteria with a threshold of 15% to mark elements for refinement, and for the adaptive- hp refinement, we use the estimation of the local smoothness of the computed solution to choose between h or p refinement [23]. The adaptive algorithms used for the standard method and for the proposed method are reported in Algorithm 2 and Algorithm 3, respectively.

The convergence plots of the error estimator using the different adaptive techniques are reported in Figure 7. For both uniform- h and adaptive- h the proposed method is slightly better than the standard method. However, for the adaptive- hp refinement the improvement is more dramatic. Even if the standard method is already converging exponentially as expected, the proposed method converges even faster. This is again due to the fact that in the proposed method the FEM focuses mainly on the smooth part of the solution, and so the hp -adaptive strategy recognizes the smoothness and adapts to give an even faster convergence. The fact that the solution of the FEM in the proposed method does not include the nonsmooth part is confirmed by Figure 8, where the FEM solutions for the standard and proposed methods are compared. Clearly the proposed method has successfully captured the nonsmooth part removing it from the region around the reentering corner in the FEM. It is interesting

Algorithm 2. Standard adaptive FEM: For parameters: the initial mesh \mathcal{T} , the initial finite element space $V_{\mathbf{p}}(\mathcal{T})$, and the number of meshes n to compute.

```

 $\mathcal{T}_1 := \mathcal{T}$ 
 $V_{\mathbf{p},1}(\mathcal{T}_1) := V_{\mathbf{p}}(\mathcal{T})$ 
for  $i = 1, \dots, n$  do
  Compute  $u_h$  solving problem (1) using the SIPDG method on the mesh
   $\mathcal{T}_i$  and using the FE space  $V_{\mathbf{p},i}(\mathcal{T}_i)$ .
  Compute the error estimator  $\eta$ .
  Use  $\eta$  to adapt the mesh and the local polynomial degrees to
  construct  $\mathcal{T}_{i+1}$  and  $V_{\mathbf{p},i+1}(\mathcal{T}_{i+1})$ .
end for

```

Algorithm 3. Celatus adaptive FEM: For parameters: the initial mesh \mathcal{T} , the initial finite element space $V_{\mathbf{p}}(\mathcal{T})$, and the number of meshes n to compute and the initial guess \mathbf{c} , the maximum number of iterations maxits , and the acceptable tolerance on the position of the minimum tol .

```

 $\mathcal{T}_1 := \mathcal{T}$ 
 $V_{\mathbf{p},1}(\mathcal{T}_1) := V_{\mathbf{p}}(\mathcal{T})$ 
for  $i = 1, \dots, n$  do
   $(\mathbf{c}_h^*, u_h, \mathbf{c}_h^*, \eta_{\mathbf{c}_h^*}) := \text{Algorithm 1}(\mathcal{T}_i, V_{\mathbf{p},i}(\mathcal{T}_i), \text{maxits}, \text{tol})$ .
  Use  $\eta_{\mathbf{c}_h^*}$  to adapt the mesh and the local polynomial degrees to
  construct  $\mathcal{T}_{i+1}$  and  $V_{\mathbf{p},i+1}(\mathcal{T}_{i+1})$ .
end for

```

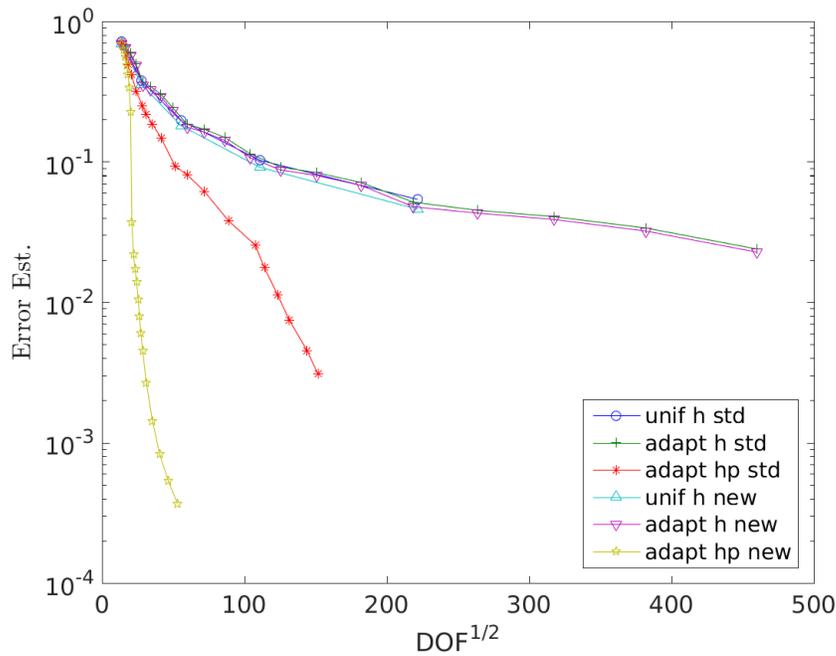


FIG. 7. Convergence plots of the standard and celatus method.

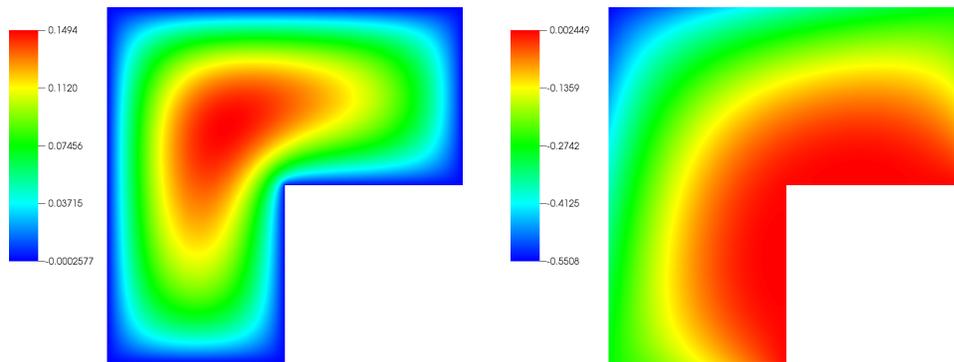


FIG. 8. (left) FE solution computed by the standard method. (right) FE solution computed by the celatus method.

to notice that the mesh-adaptivity is based on the computed solutions w_{h,c_h^*} , which approximates well the smooth part of u ; therefore, the mesh is adapted mainly to reduce the error $\|s - P(s)\|_{DG}$ in the smooth part of u . This is enough in view of Corollary 12, which shows that the overall error of the *celatus* method is dominated by the error from the smooth part. Finally, to see how the adaptive-*hp* strategy performs with the two methods, the 20th adapted meshes from both methods are displayed in Figure 9. For the standard method, the adaptive scheme has recognized the singular behavior around the reentering corner and captured it using heavily *h*-refinement. In the case of the *celatus* method, the adaptive scheme has recognized the smoothness of the solution, and more *p*-refinement has been automatically applied.

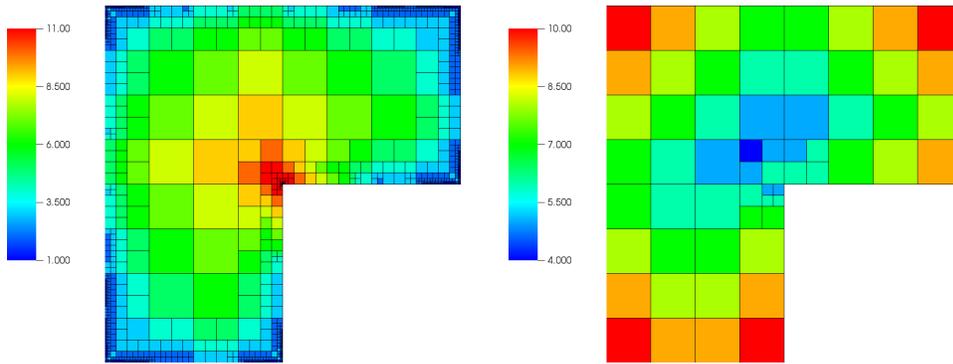


FIG. 9. (left) *hp*-adapted mesh for the standard method. (right) *hp*-adapted mesh for the celatus method. The color scheme is used to denote the order of polynomials for the elements.

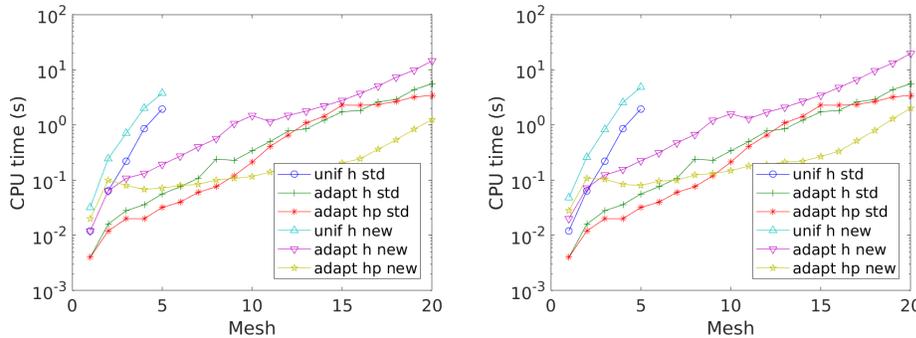


FIG. 10. (left) CPU time in seconds to solve each mesh factorizing the stiffness matrix only once. (right) CPU time in seconds to solve each mesh factorizing the stiffness matrix for each problem.

To further compare the two methods, in Figure 10 we report the CPU time for all considered adaptive strategies. Even if we use a direct solver, we have also included the case where the factorization is done every time a linear system has to be solved. In this way, we can more precisely capture the behavior of the *celatus* method in case of a different solver is used which does not produce a factorization that can be used multiple times. It is interesting to notice that for uniform-*h* refinement and for adaptive-*h* refinement, the *celatus* method is more expensive since more linear systems have to be solved. But for adaptive-*hp* refinement, the *celatus* method is less expensive after few meshes. This is due to the fact that by removing the singular behavior from the finite element solution, the mesh refinement strategy follows a completely different path that introduces fewer degrees of freedom.

Figure 10 does not consider the entire picture; our method is sometimes more expensive to use, but it is also more accurate. So, in order to consider also the accuracy, in Figure 11 we report the CPU time for different methods to reach a given accuracy. The bottom-left corner is where the best methods sit, and it is clear that the *celatus* method with adaptive-*hp* refinement is the best one reaching far better accuracy with smaller CPU time. Another way to compare different methods considering both accuracy and CPU time is by using the CPU efficiency index, which is computed taking the product of $(\text{CPU time})^{-1}$ and $\eta_{c_h}^{-1}$. In this way, high values of CPU time or error estimator translate into lower values of CPU efficiency. Also, in

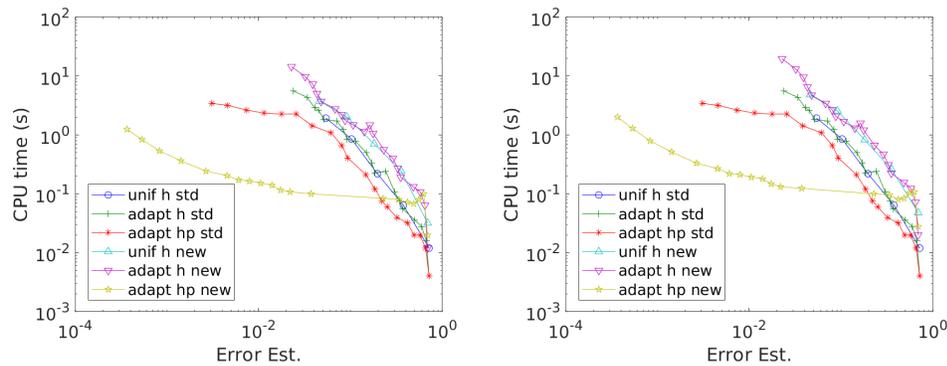


FIG. 11. Graphs reporting CPU time for different methods to reach a certain precision. The value of the error estimator is used to assess the precision. Best results are near the bottom-left corner. Each curve represents a method, and all of the simulations start in the bottom-right corner. (left) CPU time in seconds factorizing the stiffness matrix only once on each mesh. (right) CPU time in seconds factorizing the stiffness matrix for each problem.

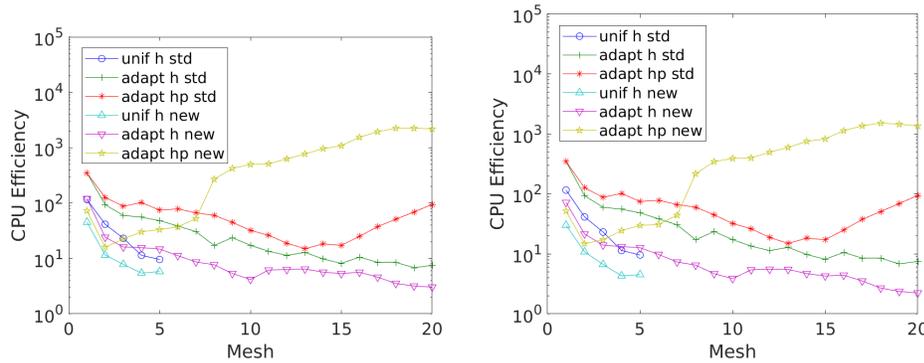


FIG. 12. (left) CPU efficiency to solve each mesh factorizing the stiffness matrix only once. (right) CPU efficiency to solve each mesh factorizing the stiffness matrix for each problem.

this respect (see Figure 12), the benefit of using the *celatus* method with adaptive-*hp* refinement is clear.

Next, we consider the T-shape domain with problem (v). Since the domain has two reentering corners, the irregular part is a linear combination of two nonsmooth functions, each one centered in one of the reentering corners. For this problem the exact solution is unknown. We use for Algorithm 1 with initial value of $\mathbf{c} = (0, 0)$, $\text{maxits} = 100$, and $\text{tol} = 1E - 6$. The initial mesh is a structured mesh of 64 square elements with $p = 1$. The comparison is done using uniform-*h*, adaptive-*h*, and adaptive-*hp* refinement as before. The convergence plots of the error estimator using the different adaptive techniques are reported in Figure 13. As already noticed in the previous subsection, for the adaptive-*hp* refinement, the improvement using our method is more dramatic. Even if the standard method is already converging exponentially as expected, the proposed method converges even faster. This seems to indicate that we are not in the worst scenario of Corollary 13 since the *celatus* method is converging faster than SIPDG alone. The solution of the FEM in the proposed method is much smoother as confirmed by Figure 14, where the FEM solutions for the standard and proposed methods are compared. Finally, to see how the adaptive-*hp* strategy

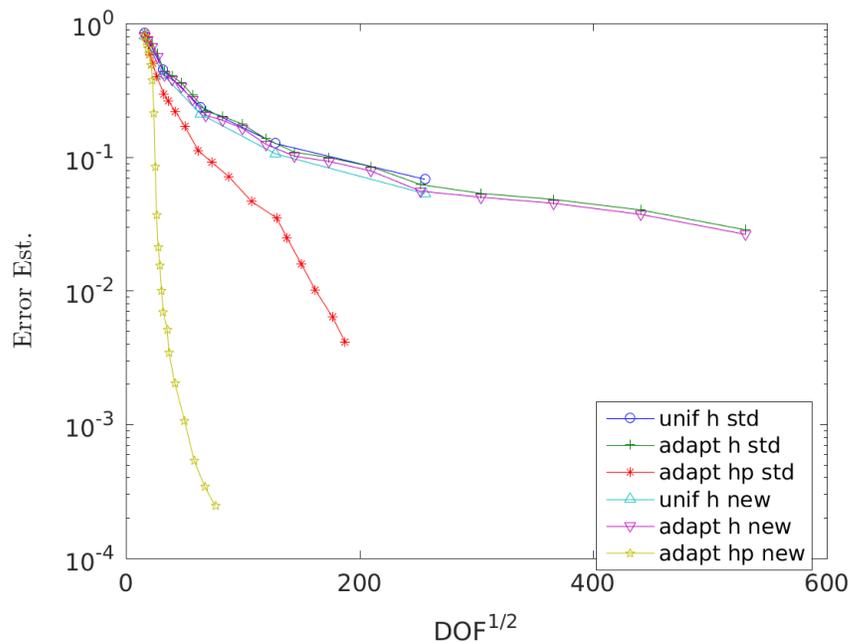


FIG. 13. Convergence plots of the standard and proposed method.

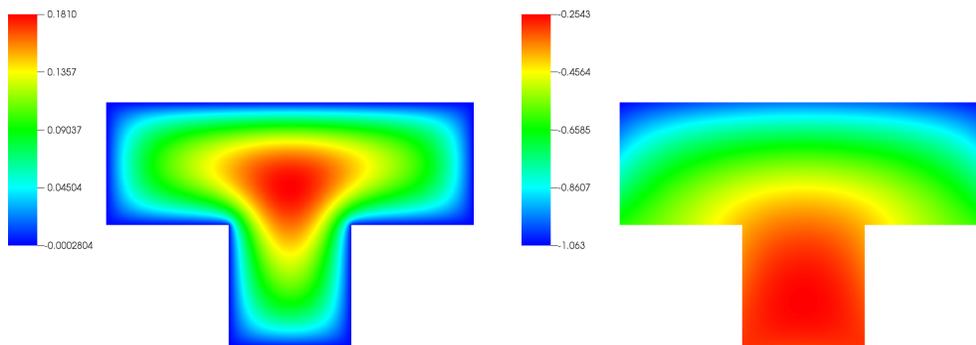


FIG. 14. (left) FE solution computed by the standard method. (right) FE solution computed by the proposed method.

performs in this case, the twentieth adapted meshes from both methods are displayed in Figure 15. For the standard method, the adaptive scheme has recognized the singular behavior around the reentering corners and approximated them using heavily h -refinement. In the case of the proposed method, the adaptive scheme has recognized the smoothness of the solution, and more p -refinement has been automatically applied.

To further compare the two methods on the T-shape domain, in Figure 16 we report the CPU time for all considered adaptive strategies. The same conclusions discussed for the L-shape domain are also valid in this case.

In Figure 17 we report the CPU time for different methods to reach a given accuracy, and in Figure 18 we report the CPU efficiency for all considered adaptive strategies. Also, for the T-shape domain, the *celatus* method with adaptive- hp refinement is much better than the standard method.

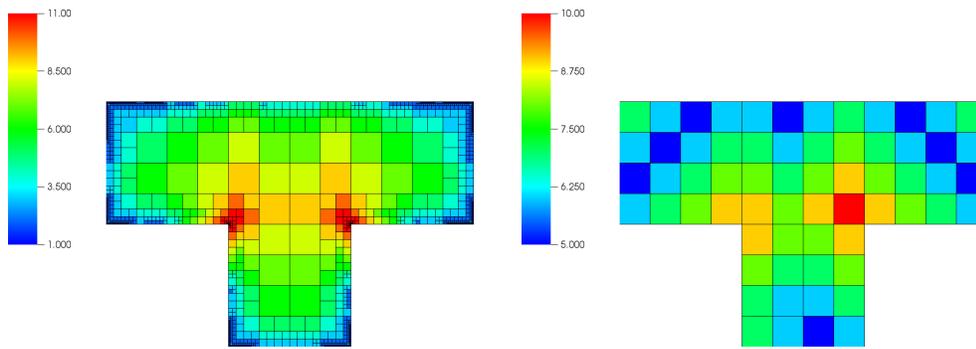


FIG. 15. (left) *hp*-adapted mesh for the standard method. (right) *hp*-adapted mesh for the proposed method. The color scheme is used to denote the order of polynomials used in the elements.

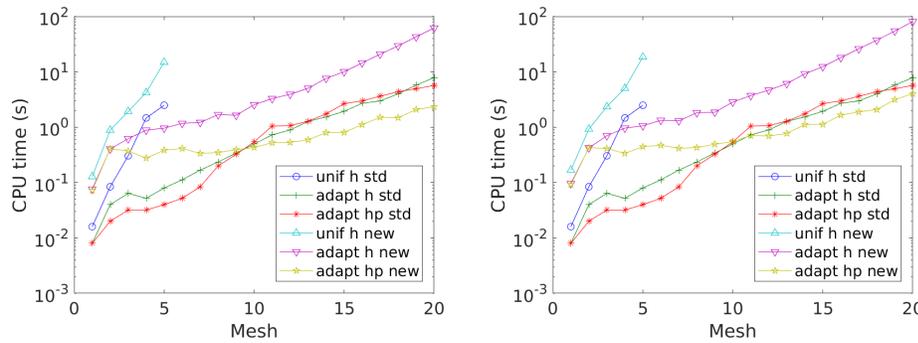


FIG. 16. (left) CPU time in seconds to solve each mesh factorizing the stiffness matrix only once. (right) CPU time in seconds to solve each mesh factorizing the stiffness matrix for each problem.

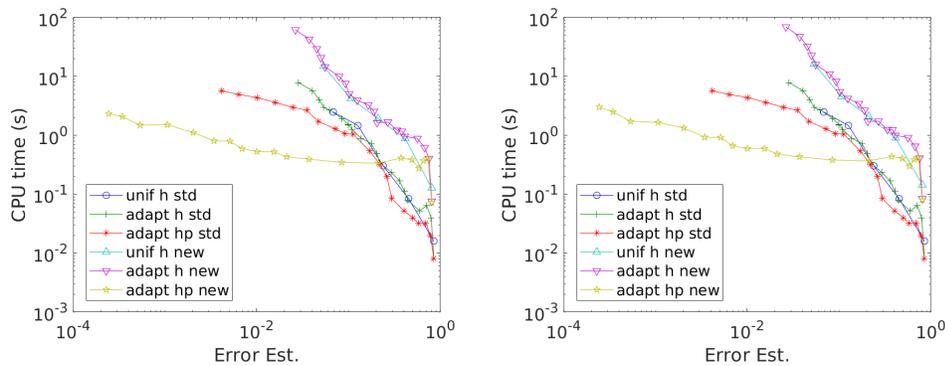


FIG. 17. Graphs reporting CPU time for different methods to reach a certain precision. The value of the error estimator is used to assess the precision. Best results are near the bottom-left corner. Each curve represents a method, and all of the simulations start in the bottom-right corner. (left) CPU time in seconds factorizing the stiffness matrix only once on each mesh. (right) CPU time in seconds factorizing the stiffness matrix for each problem.

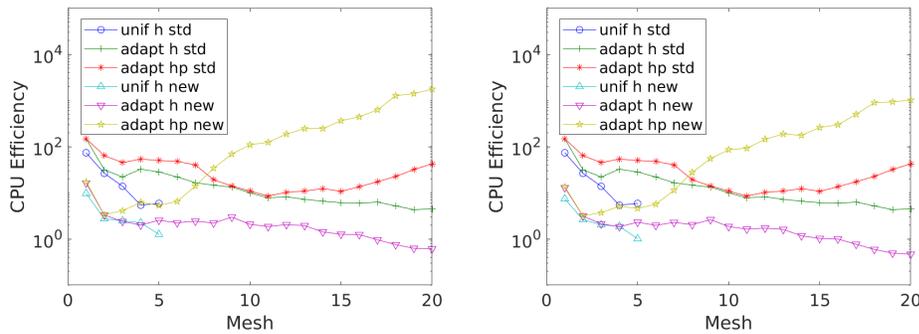


FIG. 18. (left) CPU efficiency to solve each mesh factorizing the stiffness matrix only once. (right) CPU efficiency to solve each mesh factorizing the stiffness matrix for each problem.

5. Conclusions. The paper has presented, for the first time, a new way called *celatus* to enrich a FEM. The novelty of the presented approach is that no nonpolynomial functions are added to the finite element space avoiding any issue with the condition number of the system. The implementation presented in the paper is based on a DG FEM, but the enrichment can be applied to other FEMs as well. The *celatus* enrichment uses an error estimator to capture the nonsmooth part of the solution; in view of this it seems natural to use the *celatus* enrichment with mesh adaptivity since an error estimator is already available. The paper has provided the framework for the new enrichment and tested against a standard FEM on second-order problems with reentering corners. This will allow engineers and scientists to apply the enrichment method to obtain more accurate approximations of solutions even on existing FEM codes.

REFERENCES

- [1] P. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, AND T. MARY, *On the complexity of the block low-rank multifrontal factorization*, SIAM J. Sci. Comput., 39 (2017).
- [2] P. R. AMESTOY, I. S. DUFF, J. KOSTER, AND J.-Y. L'EXCELLENT, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.
- [3] P. R. AMESTOY, I. S. DUFF, AND J.-Y. L'EXCELLENT, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 501–520.
- [4] P. R. AMESTOY, A. GUERMOUCHE, J.-Y. L'EXCELLENT, AND S. PRALET, *Hybrid scheduling for the parallel solution of linear systems*, Parallel Comput., 32 (2006), pp. 136–156.
- [5] D. N. ARNOLD, F. BREZZI, B. COCKBURN, AND L. D. MARINI, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., (2001), pp. 1749–1779.
- [6] I. BABUŠKA AND M. SURI, *The h-p Version of the Finite Element Method with Quasiuniform Meshes.*, Tech. report, 1986.
- [7] C. BERNARDI AND R. VERFÜRTH, *Adaptive finite element methods for elliptic equations with non-smooth coefficients*, Numer. Math., 85 (2000), pp. 579–608.
- [8] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, New York, 2004.
- [9] D. BRAESS, V. PILLWEIN, AND J. SCHBERL, *Equilibrated residual error estimates are p-robust*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 1189–1197.
- [10] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, New York, 1986.
- [11] T. EIBNER AND J. M. MELENK, *An adaptive strategy for hp-FEM based on testing for analyticity*, Comput. Mech., 39 (2006), pp. 575–595.

- [12] A. ERN AND M. VOHRALK, *Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and mixed discretizations*, SIAM J. Numer. Anal., 53 (2015), pp. 1058–1081.
- [13] C. FARHAT, I. HARARI, AND L. P. FRANCA, *The discontinuous enrichment method*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6455–6479.
- [14] C. FARHAT, I. HARARI, AND U. HETMANIUK, *A discontinuous Galerkin method with Lagrange multipliers for the solution of Helmholtz problems in the mid-frequency regime*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 1389–1419.
- [15] E. GEORGIOULIS, E. HALL, AND P. HOUSTON, *Discontinuous Galerkin methods for advection-diffusion-reaction problems on anisotropically refined meshes*, SIAM J. Sci. Comput., 30 (2007), pp. 246–271.
- [16] S. GIANI, *An a posteriori error estimator for hp-adaptive discontinuous Galerkin methods for computing band gaps in photonic crystals*, J. Comput. Appl. Math., 236 (2012), pp. 4810–4826.
- [17] S. GIANI, *An a posteriori error estimator for hp-adaptive discontinuous Galerkin methods for computing band gaps in photonic crystals*, J. Comput. Appl. Math., 236 (2012), pp. 4810–4826.
- [18] S. GIANI AND E. HALL, *An a posteriori error estimator for hp-adaptive discontinuous Galerkin methods for elliptic eigenvalue problems*, Math. Models Methods Appl. Sci., 22 (2011), p. 1299001.
- [19] S. GIANI AND P. HOUSTON, *Goal-oriented adaptive composite discontinuous Galerkin methods for incompressible flows*, J. Comput. Appl. Math., 270 (2014), pp. 32–42.
- [20] S. GIANI, D. SCHÖTZAU, AND L. ZHU, *An a-posteriori error estimate for adaptive DG methods for convection-diffusion problems on anisotropically refined meshes*, Comput. Math. Appl., 67 (2014), pp. 869–887.
- [21] P. GRISVARD, *Singularities in Boundary Value Problems*, Masson, Paris, 1992.
- [22] W. HACKBUSCH, *Elliptic Differential Equations*, Springer-Verlag, 1992.
- [23] P. HOUSTON AND E. SÜLI, *A note on the design of hp-adaptive finite element methods for elliptic partial differential equations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 229–243.
- [24] O. A. KARAKASHIAN AND F. PASCAL, *A posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems*, SIAM J. Numer. Anal., 41 (2003), pp. 2374–2399.
- [25] R. B. KELLOGG AND C. XENOPHONTOS, *An enriched subspace finite element method for convection-diffusion problems*, Int. J. Numer. Anal. Model., 7 (2009).
- [26] J. MELENK AND B. WOHLMUTH, *On residual-based a posteriori error estimation in hp-FEM*, Adv. Comput. Math., 15 (2001), pp. 311–331.
- [27] J. M. MELENK AND I. BABUŠKA, *The partition of unity finite element method: Basic theory and applications*, Comput. Methods Appl. Mech. Engrg., 139 (1996), pp. 289–314.
- [28] M. PETZOLDT, *Regularity and Error Estimators for Elliptic Problems with Discontinuous Coefficients*, Ph.D. thesis, Weierstrass Institut, 2001.
- [29] M. PETZOLDT, *A posteriori error estimators for elliptic equations with discontinuous coefficients*, Adv. Comput. Math., 16 (2002), pp. 47–75.
- [30] D. A. D. PIETRO AND A. ERN, *Mathematical Aspects of Discontinuous Galerkin Methods*, Mathématiques et Applications, Springer-Verlag, Berlin, 2012.
- [31] S. PRUDHOMME, F. PASCAL, J. ODEN, AND A. ROMKES, *Review of a Priori Error Estimation for Discontinuous Galerkin Methods*, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.2750>.
- [32] C. SCHWAB, *p- and hp- Finite Element Methods: Theory and Applications to Solid and Fluid Mechanics*, Oxford University Press, Oxford, 1999.
- [33] P. SOLIN, K. SEGETH, AND I. DOLEZEL, *Higher Order Finite Element Methods*, Chapman and Hall/CRC, Boca Raton, FL, 2003.
- [34] R. VERFRTH, *A Posteriori Error Estimation Techniques for Finite Element Methods*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2013.
- [35] R. VERFÜRTH, *A Review of Posteriori Error Estimation and Adaptive Mesh Refinement Techniques*, Wiley-Teubner, New York, 1996.
- [36] D. WANG, R. TEZAUER, J. TOIVANEN, AND C. FARHAT, *Overview of the discontinuous enrichment method, the ultra-weak variational formulation, and the partition of unity method for acoustic scattering in the medium frequency regime and performance comparisons*, Int. J. Numer. Methods Engrg., 89 (2012), pp. 403–417.
- [37] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley & Sons, New York, 1992.
- [38] L. ZHU, S. GIANI, P. HOUSTON, AND D. SCHÖTZAU, *Energy norm a posteriori error estimation for hp-adaptive discontinuous Galerkin methods for elliptic problems in three dimensions*, Math. Models Methods Appl. Sci., 21 (2011), pp. 267–306.