



## A partition of unity approach to adaptivity and limiting in continuous finite element methods

Item Type	Article
Authors	Kuzmin, Dmitri;Quezada de Luna, Manuel;Kees, Christopher E.
Citation	Kuzmin, D., Quezada de Luna, M., & Kees, C. E. (2019). A partition of unity approach to adaptivity and limiting in continuous finite element methods. <i>Computers &amp; Mathematics with Applications</i> , 78(3), 944–957. doi:10.1016/j.camwa.2019.03.021
Eprint version	Post-print
DOI	<a href="https://doi.org/10.1016/j.camwa.2019.03.021">10.1016/j.camwa.2019.03.021</a>
Publisher	Elsevier BV
Journal	Computers & Mathematics with Applications
Rights	NOTICE: this is the author's version of a work that was accepted for publication in <i>Computers and Mathematics with Applications</i> . Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in <i>Computers and Mathematics with Applications</i> , [[Volume], [Issue], [2019-08-01]] DOI: 10.1016/j.camwa.2019.03.021 . © 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <a href="http://creativecommons.org/licenses/by-nc-nd/4.0/">http://creativecommons.org/licenses/by-nc-nd/4.0/</a>
Download date	2024-04-16 05:19:44
Item License	<a href="http://creativecommons.org/licenses/by-nc-nd/4.0/">http://creativecommons.org/licenses/by-nc-nd/4.0/</a>

Link to Item

<http://hdl.handle.net/10754/656485>

# A partition of unity approach to adaptivity and limiting in continuous finite element methods

Dmitri Kuzmin\*    Manuel Quezada de Luna †    Christopher E. Kees‡

## Abstract

The partition of unity finite element method (PUFEM) proposed in this paper makes it possible to blend space and time approximations of different orders in a continuous manner. The lack of abrupt changes in the local mesh size  $h$  and polynomial degree  $p$  simplifies implementation and eliminates the need for using sophisticated hierarchical data structures. In contrast to traditional  $hp$ -adaptivity for finite elements, the proposed approach preserves discrete conservation properties and the continuity of traces at common boundaries of adjacent mesh cells. In the context of space discretizations, a continuous blending function is used to combine finite element bases corresponding to high-order polynomials and piecewise-linear approximations based on the same set of nodes. In a similar vein, spatially partitioned time discretizations can be designed using weights that depend continuously on the space variable. The design of blending functions may be based on *a priori* knowledge (e.g., in applications to problems with singularities or boundary layers), local error estimates, smoothness indicators, and/or discrete maximum principles. In adaptive methods, changes of the finite element approximation exhibit continuous dependence on the data. The presented numerical examples illustrate the typical behavior of local  $H^1$  and  $L^2$  errors.

**Keywords:** conservation laws, finite element methods,  $hp$ -adaptivity, discrete maximum principles, limiting techniques, partitioned time-stepping schemes

## 1 Introduction

Many advanced methods for numerical solution of partial differential equations vary the local mesh size  $h$  or the order  $p$  of polynomial approximations to capture small-scale effects and produce highly accurate predictions to the quantities of interest in an efficient manner. Prominent representatives of such high-resolution schemes include  $hp$ -adaptive finite element methods [13, 30, 33], flux-corrected transport (FCT) algorithms [28, 30, 29], and variable-order time integration schemes [15, 35].

In the process of  $hp$ -adaptation, local error indicators, smoothness sensors, and/or projections of reference solutions are employed to determine an optimal combination of  $h$  and  $p$  for each macrocell

---

\*Institute of Applied Mathematics (LS III), TU Dortmund, kuzmin@math.uni-dortmund.de

†KAUST CEMSE, manuel.quezada@kaust.edu.sa. The work of M. Quezada de Luna was mostly done at the U.S. Army, ERDC-CHL.

‡U.S. Army, ERDC-CHL, christopher.e.kees@usace.army.mil

[9, 10]. Then new finite element spaces are generated using refinement or coarsening procedures which typically require implementation of certain measures to preserve continuity of finite element shape functions and/or fluxes across the internal boundaries of mesh cells [4, 34]. The outcome of this adaptation procedure depends on many binary decisions and small changes of the data may produce entirely different finite element spaces. Moreover, the large number of ways in which anisotropic  $hp$ -refinement/coarsening can be performed makes it very difficult to identify the best choices.

High-resolution finite element schemes based on the FCT methodology and similar ideas [3, 18, 28, 29, 30] enforce monotonicity preservation using *limiters* to construct convex combinations of fluxes or element contributions associated with pairs of alternative approximations. This approach makes it possible to prove discrete maximum principles and existence of a solution to the nonlinear discrete problem [3, 5, 6]. In contrast to  $hp$ -FEM, algebraic limiting techniques of this kind do not change the finite element space. The formation of spurious undershoots or overshoots at nodes belonging to troubled cells is prevented by adding some artificial diffusion to a high-order approximation. The discrete nature of such corrections makes it difficult to design limiter functions that guarantee both monotonicity and optimal accuracy for problems with smooth exact solutions. In addition, the convergence behavior of iterative solvers for resulting nonlinear systems may be unsatisfactory [3, 28].

In this paper, we explore the possibility of combining pairs of finite element spaces or time-stepping schemes using globally defined *blending functions* instead of edge-based or element-based algebraic limiters defined in terms of nodal correction factors. The proposed approach formally represents a *partition of unity finite element method* (cf. [32]). It blends a space of large high-order elements  $V_h^H = S_{ph,p}$ ,  $p \geq 2$  with the space  $V_h^L = S_{h,1}$  corresponding to the piecewise-linear approximation w.r.t. the same nodes. This kind of  $hp$ -adaptation preserves not only global continuity of conforming finite element spaces but also the number of local degrees of freedom per element.

The local order of the time discretization can also be adjusted using partition of unity methods. As shown in [15, 21, 23, 35], the use of space-adaptive time discretizations makes it possible to achieve stability, monotonicity, optimal accuracy, and/or discrete conservation without sacrificing other favorable properties. For example, spatially partitioned embedded Runge-Kutta methods (SPERK) [21, 23] combine complementary advantages of different time discretizations by using convex combinations of semi-discrete equations or numerical fluxes. In flux-based formulations, the degree of implicitness and order of accuracy can be locally adjusted using flux limiters [15, 23, 35]. In the present paper, we consider continuous finite element approximations in which the fluxes are defined not only on the boundaries of mesh cells but also inside. Therefore, the design of spatially partitioned time-stepping schemes calls for the use of modified flux approximations in the variational formulation. The proposed use of continuous blending functions in this context leads to handy generalizations of partitioned time integrators for finite difference and finite volume discretizations of conservation laws.

The presented numerical examples indicate that the use of high-order spatial or temporal approximations in smooth regions delivers the optimal order of accuracy in corresponding subdomains. Additionally, we show that algebraic corrections of the Galerkin approximation make it possible to enforce discrete maximum principles. In the PUFEM context, we use blending functions that restrict such corrections to  $S_{h,1}$  subdomains, while preserving the high-order  $S_{2h,2}$  discretization elsewhere.

## 2 Partitioned space discretizations

Let  $\Omega \subset \mathbb{R}^d$ ,  $d \in \{1, 2, 3\}$  be a bounded domain and  $\mathcal{T}_{ph}$  a conforming finite element mesh consisting of polygonal cells (intervals in 1D, triangles or quadrilaterals in 2D, tetrahedra or hexahedra in 3D). The spacing  $ph$  is determined by the polynomial degree  $p$  of the finite element approximation to be employed and the mesh size  $h$  based on the distance between the neighboring nodal points. The space of  $p$ -th degree polynomials defined on a simplex  $K \in \mathcal{T}_{ph}$  is denoted by  $\mathbb{P}_p(K)$ . The space spanned by products of  $d$  one-dimensional  $p$ -th degree polynomials on a quad/hex element is denoted by  $\mathbb{Q}_p(K)$ . For example,  $\mathbb{P}_1(K)$  and  $\mathbb{Q}_1(K)$  are the spaces of linear and multilinear polynomials, respectively.

A finite element *shape function*  $u_h^{e,H} \in \mathbb{P}_p(K)$  or  $u_h^{e,H} \in \mathbb{Q}_p(K)$  can be expressed in terms of  $N_{\text{dof}}^e$  Lagrange or Bernstein [1, 2, 24] *basis functions*  $\varphi_1^{e,H}, \dots, \varphi_{N_{\text{dof}}^e}^{e,H}$  associated with the nodal points  $\mathbf{x}_1^e, \dots, \mathbf{x}_{N_{\text{dof}}^e}^e$ . In the case  $p > 1$ , not all of them coincide with the vertices of  $\mathcal{T}_{ph}$ . For example, the six nodes of a quadratic triangular element are located at the three vertices and three edge midpoints of  $K$ . A piecewise-linear or multilinear approximation  $u_h^{e,L}$  based on the same set of nodes can be constructed using the  $\mathbb{P}_1$  or  $\mathbb{Q}_1$  Lagrange basis functions  $\varphi_1^{e,L}, \dots, \varphi_{N_{\text{dof}}^e}^{e,L}$ . The superscripts  $H$  and  $L$  refer to the basis functions of the high-order and low-order finite element space, respectively. If  $\varphi_i^{e,H}$  are Lagrange basis functions, then the nodal values of the shape functions  $u_h^{e,H}$  and  $u_h^{e,L}$  coincide. In the context of Bernstein polynomial approximations, the local basis functions  $\varphi_i^{L,e}$  are associated with the nodes of the *Bézier net* [16]. A globally continuous high-order finite element approximation  $u_h$  to the solution  $u$  of an initial/boundary value problem can be sought in the space  $S_{ph,p} \subset H_0^1(\Omega) \cap C(\bar{\Omega})$  of continuous piecewise-polynomial functions such that  $u_h|_K = u_h^e$ . The finite element spaces

$$V_h^H = \text{span}\{\varphi_1^H, \dots, \varphi_{N_{\text{dof}}^H}^H\} = S_{ph,p}$$

and

$$V_h^L = \text{span}\{\varphi_1^L, \dots, \varphi_{N_{\text{dof}}^L}^L\} = S_{h,1}$$

are spanned by the same number  $N_{\text{dof}}$  of global basis functions and these basis functions are associated with the same nodes. However, the approximation properties of the spaces  $V_h^H$  and  $V_h^L$  are quite different. Adopting the design philosophy behind  $hp$ -FEM and FCT-like algebraic fixes, we would like to use the high-order approximation  $u_h^{e,H}$  in ‘smooth’ cells and the low-order approximation  $u_h^{e,L}$  in ‘troubled’ cells. The basis functions spanning an  $hp$ -adaptive finite element space

$$V_h(\alpha_h) := \text{span}\{\varphi_1, \dots, \varphi_{N_{\text{dof}}}\} \subset S_{h,p+1} \quad (1)$$

can be defined by

$$\varphi_i(\mathbf{x}) = \alpha_h(\mathbf{x})\varphi_i^H(\mathbf{x}) + (1 - \alpha_h(\mathbf{x}))\varphi_i^L(\mathbf{x}), \quad \mathbf{x} \in \bar{\Omega}, \quad i = 1, \dots, N_{\text{dof}}, \quad (2)$$

where  $\alpha_h$  is a blending function which yields a convex combination of  $\varphi_i^H$  and  $\varphi_i^L$ . The use of a piecewise-constant basis selector ( $\alpha_h^e \equiv 1$  or  $\alpha_h^e \equiv 0$  in  $K^e \in \mathcal{T}_{ph}$ ) may produce a discontinuous global

approximation. Instead, we propose the use of continuous piecewise-linear blending functions

$$\alpha_h = \sum_{i=1}^{N_{\text{dof}}} \alpha_i \varphi_i^L \in V_h^L. \quad (3)$$

The partition of unity (PU) parameters  $\alpha_i \in [0, 1]$  may be defined using bound-preserving limiters or smoothness indicators that depend on the solution in a continuous manner, see Section 4.

If  $\alpha_h$  is sufficiently smooth, the partitioned scheme is guaranteed to be at least as accurate as the  $V_h^L$  approximation. To show this, we consider a generic boundary value problem of the form

$$a(u, v) = b(v) \quad \forall v \in V, \quad (4)$$

where  $a : V \times V \rightarrow \mathbb{R}$  is a coercive and continuous bilinear form and  $b : V \rightarrow \mathbb{R}$  is a continuous linear form on  $V := H_0^1(\Omega)$ . Let  $u_h \in V_h(\alpha_h)$  be the solution of the discrete problem

$$a(u_h, v_h) = b(v_h) \quad \forall v_h \in V_h(\alpha_h). \quad (5)$$

Using the Galerkin orthogonality, coercivity, and continuity, the best approximation property

$$\exists C > 0 \quad \forall w_h \in V_h(\alpha_h) \quad \|u - u_h\|_{1,\Omega} \leq C \|u - w_h\|_{1,\Omega} \quad (6)$$

can be shown as in Céa's lemma. Let  $w_h = \alpha_h I_{ph,p}u + (1 - \alpha_h)I_{h,1}u \in V_h(\alpha_h)$  be defined in terms of the interpolants  $I_{h,1}u \in V_h^L$  and  $I_{ph,p}u \in V_h^H$  such that  $I_{h,1}u(\mathbf{x}_i) = u(\mathbf{x}_i) = I_{ph,p}u(\mathbf{x}_i)$  at each node  $\mathbf{x}_i$  of the submesh  $\mathcal{T}_h = \mathcal{T}_{1h}$ , i.e., of the mesh on which the piecewise-linear approximation  $V_h^L$  is defined. Suppose that  $|\nabla \alpha_h(\mathbf{x})| \leq C_\alpha \in \mathbb{R}_+$  for all meshes and all  $\mathbf{x} \in \Omega$ . For any  $w \in V$  we have

$$\begin{aligned} \|\alpha_h w\|_{1,\Omega}^2 &= (\alpha_h w, \alpha_h w)_{0,\Omega} + (w \nabla \alpha_h + \alpha_h \nabla w, w \nabla \alpha_h + \alpha_h \nabla w)_{0,\Omega} \\ &= \|\alpha_h w\|_{0,\Omega}^2 + \|\alpha_h \nabla w\|_{0,\Omega}^2 + (w^2, |\nabla \alpha_h|^2)_{0,\Omega} + 2(\alpha_h \nabla \alpha_h, w \nabla w)_{0,\Omega}. \end{aligned}$$

The last term can be estimated as follows:

$$\begin{aligned} 2(\alpha_h \nabla \alpha_h, w \nabla w)_{0,\Omega} &= 2 \int_{\Omega} \alpha_h w \nabla \alpha_h \cdot \nabla w \, d\mathbf{x} \leq 2 \int_{\Omega} \alpha_h |w| |\nabla \alpha_h| |\nabla w| \, d\mathbf{x} \\ &\leq 2C_\alpha \int_{\Omega} \alpha_h \left( \frac{1}{2}|w|^2 + \frac{1}{2}|\nabla w|^2 \right) \, d\mathbf{x} \leq C_\alpha \|w\|_{1,\Omega}^2, \end{aligned}$$

where we have used Young's inequality and the fact that  $0 \leq \alpha_h \leq 1$ . It follows that

$$\|\alpha_h w\|_{1,\Omega}^2 \leq \|w\|_{0,\Omega}^2 + \|\nabla w\|_{0,\Omega}^2 + C_\alpha^2 \|w\|_{0,\Omega}^2 + C_\alpha \|w\|_{1,\Omega}^2 \leq (1 + C_\alpha)^2 \|w\|_{1,\Omega}^2.$$

Using the triangle inequality, we obtain the worst-case *a priori* error estimate

$$\begin{aligned} \|u - w_h\|_{1,\Omega} &\leq \|\alpha_h(u - I_{ph,p}u)\|_{1,\Omega} + \|(1 - \alpha_h)(u - I_{h,1}u)\|_{1,\Omega} \\ &\leq (1 + C_\alpha)(\|u - I_{ph,p}u\|_{1,\Omega} + \|u - I_{h,1}u\|_{1,\Omega}) \leq (1 + C_\alpha)(C_1 + C_p h^{p-1})h|u|_{2,\Omega} \quad (7) \end{aligned}$$

for  $u \in H^2(\Omega)$  under usual assumptions regarding the mesh refinement strategy. The optimal order  $p$  of the high-order finite element approximation is preserved for blending functions  $\alpha_h$  satisfying  $\|(1 - \alpha_h)(u - I_{h,1}u)\|_{1,\Omega} \leq C\|\alpha_h(u - I_{ph,p}u)\|_{1,\Omega}$  for some  $C > 0$ . An *a priori* estimate of the global  $L^2$  error can be obtained using the Aubin-Nitsche duality argument (see, e.g., [25]). In practice, the optimal  $H^1$  and  $L^2$  convergence behavior can be achieved at least in core of subdomains where  $\alpha_h$  equals 1, i.e., on patches of  $p$ -th order elements located sufficiently far from the subdomains where  $\alpha_h < 1$ . A numerical study of local errors for the Poisson equation is presented in Section 5.1.

### 3 Partitioned time discretizations

Continuous blending functions can also be used to combine time discretizations of different orders. The Galerkin finite element discretization of the time-dependent scalar conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0 \quad \text{in } \Omega \subset \mathbb{R}^d, \quad d \in \{1, 2, 3\} \quad (8)$$

leads to a semi-discrete variational problem of the form

$$\frac{d}{dt} \int_{\Omega} \varphi_i u_h d\mathbf{x} = \int_{\Omega} \nabla \varphi_i \cdot \mathbf{f}(u_h) d\mathbf{x} - \int_{\partial\Omega} \varphi_i \mathbf{f}(u_h) \cdot \mathbf{n} ds, \quad i = 1, \dots, N_{\text{dof}}, \quad (9)$$

where  $\mathbf{n}$  is the unit outward normal. Summing over  $i$  and using the partition of unity property

$$\sum_{i=1}^{N_{\text{dof}}} \varphi_i \equiv 1 \quad (10)$$

of Lagrange and Bernstein basis functions, we find that  $u_h$  satisfies the integral conservation law

$$\frac{d}{dt} \int_{\Omega} u_h d\mathbf{x} = - \int_{\partial\Omega} \mathbf{f}(u_h) \cdot \mathbf{n} ds. \quad (11)$$

Let  $u_h^n$  denote the approximation to  $u_h$  at the discrete time level  $t^n = n\Delta t$ , where  $\Delta t$  is the time step. Introducing the blending function  $\theta_h^n : \Omega \rightarrow [0, 1]$ , we define the convex average

$$u_h^{n+\theta} := \theta_h^n u_h^{n+1} + (1 - \theta_h^n) u_h^n \quad (12)$$

and discretize (9) in time using the following generalization of the two-level  $\theta$ -scheme

$$\int_{\Omega} \varphi_i u_h^{n+1} d\mathbf{x} = \int_{\Omega} \varphi_i u_h^n d\mathbf{x} + \Delta t \left[ \int_{\Omega} \nabla \varphi_i \cdot \mathbf{f}(u_h^{n+\theta}) d\mathbf{x} - \int_{\partial\Omega} \varphi_i \mathbf{f}(u_h^{n+\theta}) \cdot \mathbf{n} ds \right]. \quad (13)$$

The use of  $\theta_h \equiv \theta \in \{0, \frac{1}{2}, 1\}$  yields the forward Euler, Crank-Nicolson, and backward Euler methods, respectively. The discrete conservation property is preserved for arbitrary blending functions since

$$\int_{\Omega} u_h^{n+1} d\mathbf{x} = \int_{\Omega} u_h^n d\mathbf{x} - \Delta t \int_{\partial\Omega} \mathbf{f}(u_h^{n+\theta}) \cdot \mathbf{n} ds. \quad (14)$$

The proposed approach leads to a continuous finite element version of flux-based partitioned schemes for finite difference and finite volume discretizations of conservation laws [15, 23, 35]. For a comprehensive review and analysis of such schemes, we refer the reader to Ketcheson et al. [21, 23]. We envisage that similar ideas can be used to design spatially partitioned Runge-Kutta schemes for finite element discretizations. The need to invert consistent mass matrices in explicit schemes can be avoided, e.g., using deferred correction methods [1] or truncated Neumann series approximations [17, 18].

A typical application which calls for the use of a non-constant blending function  $\theta_h$  in the generalized  $\theta$  scheme (12),(13) is the *small cut cell problem*. Explicit schemes based on the embedded boundary cut cell methodology are subject to time step restrictions depending on the size of the smallest subcells. As shown by May and Berger [31], an implicit treatment of cut cells makes it possible to circumvent these severe time restrictions while using computationally efficient explicit time integration in regular cells. The design of such implicit-explicit (IMEX) methods is generally rather difficult due to the need to preserve consistency and conservation properties in applications to evolutionary problems. The proposed use of continuous blending functions provides a very simple and flexible tool for combining different time discretizations in a consistent and conservative manner. The numerical study in Section 5.5 indicates that second-order overall accuracy can be achieved for  $\Delta t = O(h^2)$  using a spatially partitioned  $\theta_h$  scheme. Moreover, the use of  $\theta_h = \frac{1}{2}$  in subdomains where rapid changes of the exact solution occur is likely to have the same positive impact on the local accuracy of temporal discretization as the use of  $\alpha_h = 1$  in the context of partitioned space discretizations.

## 4 Design of blending functions

Optimal design of blending functions for space and time discretizations depends on the local smoothness of the solution and on the adaptation criteria. Many useful error estimation techniques and refinement strategies can be found in the literature on *hp*-FEM. The proposed partition of unity approach can be used to prevent violations of discrete maximum principles which are possible even for the Poisson equation if high-order finite elements are employed [12, 20]. If the piecewise-linear discretization is guaranteed to be monotone (which is the case, e.g., for the Galerkin discretization of the Laplace operator under certain assumptions regarding the shapes of mesh elements), then formation of undershoots and overshoots can be ruled out by setting  $\alpha_i = 0$  at nodes of elements containing local extrema. In general, discrete maximum principles can be enforced by setting  $\alpha_i = 0$  **and** using artificial diffusion operators / limiter functions developed in [3, 5, 6, 28, 18]. The loss of accuracy at smooth local extrema can be avoided using continuous interpolants of nodal smoothness indicators [9, 26, 29]. Numerical examples illustrating these ideas are presented in Sections 5.2-5.4.

## 5 Numerical experiments

In this section, we perform numerical studies illustrating the use of continuous blending functions as a tool for the design of adaptive finite element spaces and explicit-implicit time discretizations. In appendix A we briefly describe some details of the numerical implementation.

## 5.1 Poisson equation in 2D

In the first numerical example, we study the local convergence behavior of a variable-order finite element approximation  $u_h$  to the solution of the 2D elliptic boundary value problem

$$\begin{aligned} -\Delta u &= f, & \text{in } \Omega, \\ u &= 0, & \text{on } \partial\Omega. \end{aligned}$$

The domain of interest is  $\Omega = (0, 1)^2$ . The right-hand side

$$f(x, y) = 8\pi^2 \sin(2\pi x) \sin(2\pi y)$$

of the Poisson equation is obtained by differentiating the exact solution

$$u(x, y) = \sin(2\pi x) \sin(2\pi y).$$

Let  $u_h^H = u_{2h,2}$  denote the  $\mathbb{Q}_2$  approximation on a uniform Cartesian mesh  $\mathcal{T}_{2h}$ . Then  $u_h^L = u_{h,1}$  is the  $\mathbb{Q}_1$  approximation on the submesh  $\mathcal{T}_h$ . The corresponding finite element spaces  $V_h^H = S_{2h,2}$  and  $V_h^L = S_{h,1}$  are combined using a blending function  $\alpha_h \in V_h^L$ , as defined by (3) and shown in Fig. 1(a). In this test, we use  $\alpha_i = \mathbb{H}(x_i - 0.5)$ , where  $\mathbb{H} : [0, 1] \rightarrow [0, 1]$  is the Heaviside function defined by

$$\mathbb{H}(x) := \begin{cases} 1 & \text{if } x > 0.5, \\ \frac{1}{2} & \text{if } x = 0.5, \\ 0 & \text{if } x < 0.5. \end{cases} \quad (15)$$

The continuous interpolant  $\alpha_h$  is well-defined for  $h = \frac{1}{2^k}$ ,  $k \in \mathbb{N}$  since  $x = 0.5$  remains a grid line in the process of mesh refinement. The total number of degrees of freedom is  $N_{\text{dof}} = (N_x - 2) \times (N_y - 2)$ , where  $N_x = \frac{1}{h} + 1 = N_y$  is the number of grid lines parallel to the  $x$ - and  $y$ -axes. The condition that  $|\nabla \alpha_h| \leq C_\alpha$  does not hold in this example but it is not uncommon that restrictive assumptions behind the derivation of *a priori* error estimates are violated in practice. It turns out that the above definition of the blending function  $\alpha_h$  does deliver optimal rates of convergence inside  $\mathbb{Q}_1$  and  $\mathbb{Q}_2$  subdomains. In Figs 1(b,c), we show the distribution of the pointwise error

$$e(x, y) = u(x, y) - u_h(x, y)$$

for the partitioned finite element approximation  $u_h \in V_h(\alpha_h)$  on meshes corresponding to  $321^2$  and  $641^2$  DOFs. As expected, smaller errors are observed in the subdomain discretized using  $\mathbb{Q}_2$  elements.

For a quantitative assessment of local errors, we compute the  $L^2$  and  $H^1$  norms  $\|e\|_{0,D}$  and  $\|e\|_{1,D}$  of the absolute error for  $D = \{\Omega, \Omega_1, \Omega_2\}$ , where  $\Omega_1 = \{(x, y) \in \Omega \mid x < 0.5 - h\}$  and  $\Omega_2 = \{(x, y) \in \Omega \mid x > 0.5 + h\}$ . By definition of  $\alpha_i$ , the  $\mathbb{Q}_1$  approximation on small cells of the submesh  $\mathcal{T}_h$  is used in  $\Omega_1$ , whereas the  $\mathbb{Q}_2$  approximation on large cells of the mesh  $\mathcal{T}_{2h}$  is employed in  $\Omega_2$ . A continuous transition between the corresponding finite element spaces occurs in the buffer zone  $\Omega \setminus (\Omega_1 \cup \Omega_2)$ . The results of a grid convergence study are shown in Table 1. Note that the optimal  $\mathbb{Q}_2$  convergence rates are obtained in the subdomain  $\Omega_2$ . The errors in  $\Omega$  and  $\Omega_1$  exhibit the  $\mathbb{Q}_1$  convergence behavior.

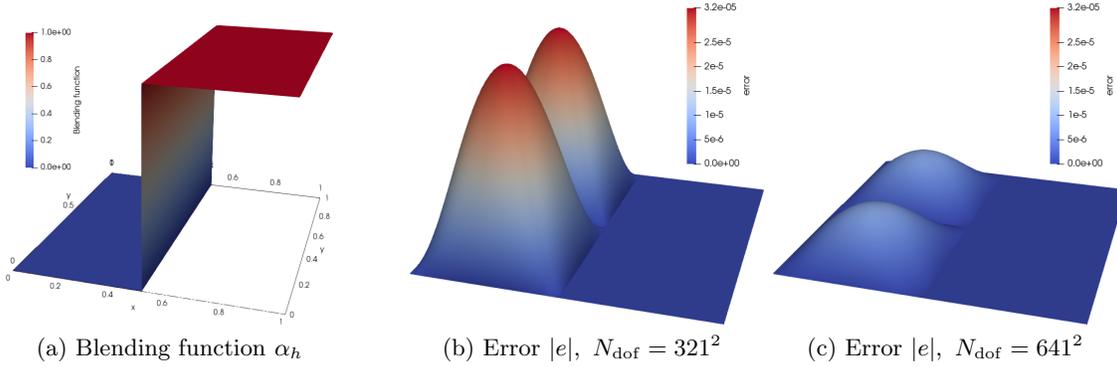


Figure 1: Blending function and error for the 2D Poisson test.

$N_x \times N_y$	Domain: $\Omega$				Domain: $\Omega_1$				Domain: $\Omega_2$			
	$\ e\ _{0,\Omega}$	—	$\ e\ _{1,\Omega}$	—	$\ e\ _{0,\Omega_1}$	—	$\ e\ _{1,\Omega_1}$	—	$\ e\ _{0,\Omega_2}$	—	$\ e\ _{1,\Omega_2}$	—
$21^2$	3.49E-3	—	2.87E-1	—	3.42E-3	—	2.83E-1	—	6.55E-4	—	4.35E-2	—
$41^2$	8.63E-4	2.01	1.42E-1	1.01	8.59E-4	1.99	1.42E-1	0.99	8.56E-5	2.93	1.12E-2	1.95
$81^2$	2.15E-4	2.00	7.12E-2	1.00	2.14E-4	1.99	7.12E-2	0.99	1.09E-5	2.97	2.84E-3	1.97
$161^2$	5.38E-5	2.00	3.56E-2	1.00	5.38E-5	1.99	3.56E-2	0.99	1.38E-6	2.98	7.17E-4	1.98
$321^2$	1.34E-5	2.00	1.78E-2	1.00	1.34E-5	1.99	1.78E-2	0.99	1.73E-7	2.99	1.79E-4	1.99
$641^2$	3.36E-6	2.00	8.90E-3	1.00	3.36E-6	1.99	8.90E-3	0.99	2.17E-8	2.99	4.51E-5	1.99

Table 1: Grid convergence history for the 2D Poisson test.

## 5.2 Steady advection-diffusion in 2D

In the second numerical example, we solve the 2D stationary advection-diffusion equation

$$\mathbf{v} \cdot \nabla u - \epsilon \Delta u = 0 \quad \text{in } \Omega = (0, 1)^2 \quad (16)$$

Following Brezzi et al. [11], we use the constant velocity field  $\mathbf{v} = (1, 3)$  and the diffusion coefficient  $\epsilon = 0.01$ . The Dirichlet boundary conditions for this test problem are given by [11]

$$\begin{aligned} u(0, y) &= 1, \quad \forall y \in [0, 1], \\ u(x, 1) &= 0, \quad \forall x \in [0, 1], \\ u(1, y) &= 0, \quad \forall y \in [0, 1], \\ u(x, 0) &= 1, \quad \forall x \in \left[0, \frac{1}{3}\right), \\ u(x, 0) &= 0, \quad \forall x \in \left(\frac{1}{3}, 1\right]. \end{aligned}$$

The exact solution exhibits a boundary layer and an internal layer. Once again, we combine the finite element spaces  $V_h^H = S_{2h,2}$  and  $V_h^L = S_{h,1}$  using a blending function  $\alpha_h \in V_h^L$  such that  $u_h|_K \in \mathbb{Q}_2(K)$

if  $\alpha_h|_K \equiv 1$  and  $u_h|_K \in \mathbb{Q}_1(K)$  if  $\alpha_h|_K \equiv 0$ . The  $\mathbb{Q}_2$  approximation is well suited for resolving the smooth part of the parabolic internal layer but may produce undershoots or overshoots near the discontinuity point  $(\frac{1}{3}, 0)$  or along the boundary  $y = 1$ . If the  $\mathbb{Q}_1$  approximation is employed in these regions, violations of discrete maximum principles can be prevented using algebraic correction of finite element matrices [7, 28]. To show the potential benefit of using adaptive finite element spaces in this context, we define the nodal values  $\alpha_i$  of the piecewise-linear blending function (3) as follows:

$$\alpha_i(x, y) = \begin{cases} 1, & \text{if } 2h \leq y \leq 0.8, \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

This definition produces a thin layer of pure  $\mathbb{Q}_1$  elements along the inflow boundary  $y = 0$  and a larger  $\mathbb{Q}_1$  subdomain along the outflow boundary  $y = 1$ . Additionally, the  $\mathbb{Q}_1$  Galerkin discretization of the advective term  $\mathbf{v} \cdot \nabla u$  is stabilized by adding a discrete diffusion operator defined as in [7, 28]. The modified form of the equation corresponding to the test function  $\varphi_i \in H_0^1(\Omega)$  reads

$$\sum_j (k_{ij} - d_{ij} + l_{ij}) u_j = 0,$$

where  $u_j = u_h(\mathbf{x}_j)$  is the nodal value at the point  $\mathbf{x}_j \in \bar{\Omega}$ ,

$$k_{ij} = \int_{\Omega} \varphi_i(\mathbf{v} \cdot \nabla \varphi_j) d\mathbf{x}, \quad l_{ij} = \epsilon \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j d\mathbf{x}$$

are the coefficients of the Galerkin discretization and

$$d_{ij} = \begin{cases} \max((1 - \alpha_i)k_{ij}, 0, (1 - \alpha_j)k_{ji}) & \text{if } j \neq i, \\ -\sum_{k \neq i} d_{ik} & \text{if } j = i \end{cases}$$

are the coefficients of the artificial diffusion operator (cf. [3, 7]). For each  $\mathbf{x}_i \in \partial\Omega$ , the value of  $u_i = u_h(\mathbf{x}_i)$  is determined by the strongly imposed Dirichlet boundary conditions.

Computations are performed on uniform Cartesian meshes. In Figure 2, we present numerical solutions to (16) obtained using three finite element spaces of equal dimensions corresponding to  $N_{\text{dof}} = N_x \times N_y = 101^2$  degrees of freedom. The pure  $\mathbb{Q}_2$  approximation ( $u_h^H$ , middle panel) exhibits spurious undershoots and overshoots. A proof of the discrete maximum principle (DMP) for the pure  $\mathbb{Q}_1$  approximation ( $u_h^L$ , left panel) can be found in [7]. The solution  $u_h$  shown in the right panel was calculated using the space  $V_h(\alpha_h)$  with  $\alpha_h$  defined by (17). It looks similar to  $u_h^L$  and is also bounded by the Dirichlet boundary values. This indicates that the use of  $\mathbb{Q}_1$  elements and artificial diffusion operators may be restricted to small subdomains containing small-scale features that cannot be resolved properly on a given mesh. In a fully adaptive version of the proposed methodology, the nodal values of the blending function  $\alpha_h$  may be adjusted to maximize the size of  $\mathbb{Q}_2$  subdomains subject to DMP-like constraints. Examples of DMP-satisfying partitioned finite element schemes that select appropriate values of the tuning parameters  $\alpha_i \in [0, 1]$  automatically will be presented below.

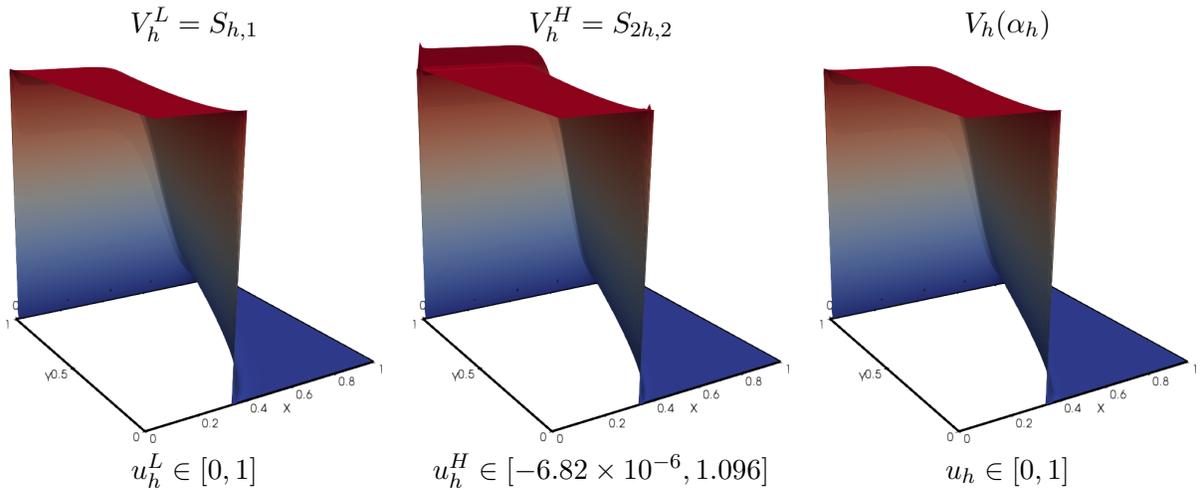


Figure 2: Solutions of the stationary advection-diffusion equation (16) obtained using finite element spaces  $V_h^H = V_h(0)$ ,  $V_h^L = V_h(1)$ , and  $V_h(\alpha_h)$ , where  $\alpha_h$  is the blending function defined by (17).

### 5.3 Anisotropic diffusion equation in 2D

In the next numerical experiment, we consider the 2D anisotropic diffusion problem that was used in [27, §8.1] and in many other publications to test DMP-satisfying numerical methods for elliptic problems of this kind. The Dirichlet boundary value problem to be solved is defined by

$$-\nabla \cdot (\mathcal{D}\nabla u) = 0, \quad \text{in } \Omega, \quad (18a)$$

$$u = -1, \quad \text{on } \partial\Omega_O, \quad (18b)$$

$$u = 1, \quad \text{on } \partial\Omega_I, \quad (18c)$$

where  $\Omega = (0, 1)^2 \setminus [4/9, 5/9]^2$  and  $\partial\Omega_O$  and  $\partial\Omega_I$  are the outer and inner boundaries, respectively. The diffusion tensor  $\mathcal{D}$  admits the spectral decomposition

$$\mathcal{D} = \mathcal{R}(-\theta) \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \mathcal{R}(\theta), \quad \mathcal{R} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}.$$

In our numerical study, we use the parameter settings  $\theta = \frac{\pi}{6}$ ,  $k_1 = 1000$  and  $k_2 = 1$ . We first consider the Galerkin discretization corresponding to a generic partitioned finite element space  $V_h(\alpha_h)$ . The Galerkin equation associated with an internal node  $\mathbf{x}_i \in \Omega$  is given by

$$\sum_j l_{ij} u_j = 0, \quad (19)$$

where  $u_j = u_h(\mathbf{x}_j)$  is the nodal value at the point  $\mathbf{x}_j \in \bar{\Omega}$  and  $l_{ij} = \int_{\Omega} \nabla \varphi_i \cdot (\mathcal{D}\nabla \varphi_j) dx$ . The Dirichlet boundary values  $u_i = u_h(\mathbf{x}_i)$  for  $\mathbf{x}_i \in \partial\Omega$  are imposed strongly.

In Figure 3, we show the Galerkin solutions of (19) obtained using the spaces  $V_h^L = V_h(0)$  and  $V_h^H = V_h(1)$  for  $h = \frac{1}{40}$ . In both cases, we observe violations of the lower bound  $\min_{\Omega} u = -1$ . To enforce discrete maximum principles as in §5.2, we define the artificial diffusion coefficients

$$d_{ij} = \begin{cases} \max((1 - \alpha_i)l_{ij}, 0, (1 - \alpha_j)l_{ji}) & \text{if } j \neq i, \\ -\sum_{k \neq i} d_{ik} & \text{if } j = i \end{cases} \quad (20)$$

and replace (19) with

$$\sum_j (l_{ij} - d_{ij}) u_j = 0. \quad (21)$$

In the left panel of Figure 4, we show the solution of (21) obtained with  $\alpha_h = 0$ . The nodal values of  $u_h$  are now in the range  $[-1, 1]$ . However, large amounts of numerical dissipation are introduced. To localize this dissipation, we use a blending function  $\alpha_h$  whose nodal values are given by

$$\alpha_i(\mathbf{x}) = \begin{cases} 0, & \text{if } \text{dist}(\mathbf{x}, \partial\Omega_O) \leq 4h \text{ or } \mathbf{x} \in \partial\Omega_I, \\ 1, & \text{otherwise,} \end{cases} \quad (22)$$

see the right panel of Figure 4. The corresponding numerical solution is shown in the middle panel of Figure 4. Note that this solution stays in the admissible range and is less diffusive than the one obtained with  $\alpha_h = 0$ . Clearly, the *ad hoc* definition (22) of the blending function  $\alpha_h$  is based on *a priori* knowledge of the expected solution behavior. In §5.4, we explore some possibilities for automatic selection of  $\alpha_h$  on the basis of DMPs and extremum-preserving smoothness criteria.

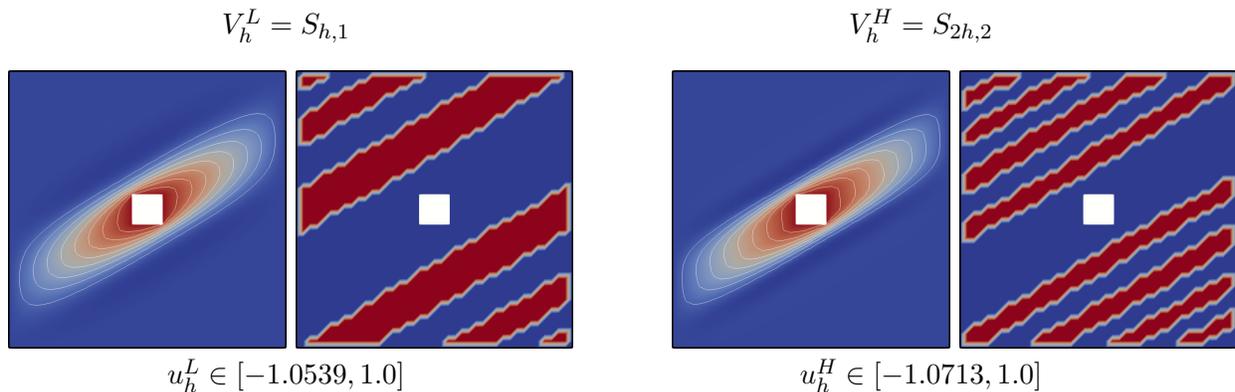


Figure 3: Galerkin solutions of the anisotropic diffusion equation (18) obtained using (a)  $V_h^L = V_h(0)$  and (b)  $V_h^H = V_h(1)$ . The colormaps in the left panels span the entire range of solution values. The colormaps in the right panels expose violations of the lower bound  $\min_{\Omega} u = -1$ .

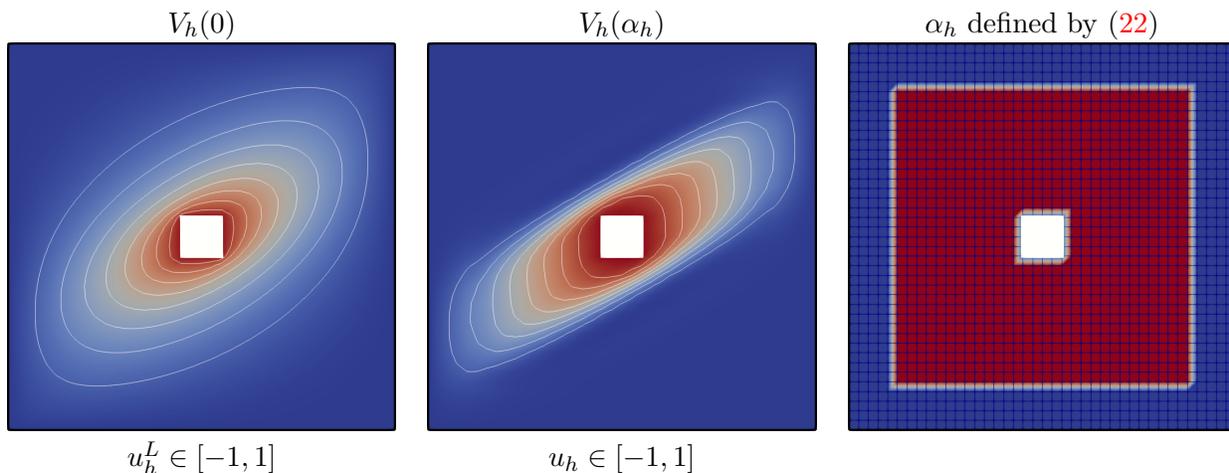


Figure 4: Solutions of the anisotropic diffusion equation (18) obtained using (21). The blending function  $\alpha_h$  of the partitioned finite element space  $V_h(\alpha_h)$  is shown in the right panel.

#### 5.4 Automated selection of blending functions

In the above examples, we used low-order artificial diffusion to enforce discrete maximum principles in subdomains where  $\alpha_h < 1$ . The results presented in Figure 4 indicate that the definition of  $\alpha_h$  may have a strong impact on the accuracy of such approximations. In the absence of antidiffusive correction terms, the levels of numerical diffusion can be reduced by using  $\alpha_i = 1$  in ‘safe’ subdomains, where the Galerkin solution varies smoothly and no violations of DMPs are detected.

In this section, we give an example of an algorithm that determines  $\alpha_i$  automatically using a DMP-based sensor  $\beta_i$  and a smoothness indicator  $\gamma_i$ . The former is defined by (cf. [3, 8, 19])

$$\beta_i = 1 - \left[ \frac{\left| \sum_{j \in \mathcal{N}_i} u_j - u_i \right| + \epsilon}{\sum_{j \in \mathcal{N}_i} |u_j - u_i| + \epsilon} \right]^2, \quad (23)$$

where  $\mathcal{N}_i$  is an integer set containing node  $i$  and its common-element neighbors. The small number  $\epsilon > 0$  is used to prevent division by zero. The so-defined DMP sensor  $\beta_i$  equals zero if  $u_i$  is a local maximum or minimum. On the other hand, it is easy to verify that  $\beta_i = 1$  if  $\mathbf{x}_i \in \Omega$  is an internal node of a uniform mesh and  $u_h$  is linear on the patch of elements  $\mathcal{E}_i$  containing this node.

The smoothness indicator  $\gamma_i$  is designed to measure the local regularity of a piecewise-biquadratic approximation  $u_{2h,2}$  at a vertex  $\mathbf{x}_i$  of a (possibly unstructured) coarse mesh  $\mathcal{T}_{2h}$ . A node-based counterpart of the element-based smoothness sensor proposed in [29] is given by

$$\gamma_i = \frac{\max [0, \min(\eta_i^2, C \min_{e \in \mathcal{E}_i} \eta_i \eta_i^e)]}{\eta_i^2 + \epsilon}, \quad (24)$$

where  $\eta_i^e = \eta_{2h,2}^e(\mathbf{x}_i)$  denotes the one-sided limit of the Hessian determinant

$$\eta_{2h,2}^e = \frac{\partial^2 u_{2h,2}^e}{\partial x^2} \frac{\partial^2 u_{2h,2}^e}{\partial y^2} - \frac{\partial^2 u_{2h,2}^e}{\partial x \partial y} \frac{\partial^2 u_{2h,2}^e}{\partial y \partial x}$$

at node  $i$  of element  $K^e \in \mathcal{T}_{2h}$  and  $\eta_i = \frac{1}{|\mathcal{E}_i|} \sum_{e \in \mathcal{E}_i} \eta_i^e$  is the arithmetic mean of all one-sided limits. The so-defined smoothness indicator  $\gamma_i$  equals zero whenever  $\eta_i^e \eta_i^{e'} \leq 0$  for any  $e \neq e'$ . The value  $\gamma_i = 0$  implies that node  $i$  belongs to at least one troubled cell in which  $\alpha_h$  should be set equal to zero. The value  $\gamma_i = 1$  implies that  $u_{2h,2}$  is smooth in all cells containing node  $i$  and it is safe to use  $\alpha_h = 1$  in these cells. This situation occurs if all one-sided Hessian determinants  $\eta_i^e$  have the same sign and their magnitudes do not differ by more than a factor of  $C$ . In the present implementation, we use  $C = 3$  and approximate  $\eta_i^e$  by the cell average  $\bar{\eta}^e = \frac{1}{|K^e|} \int_{K^e} \eta_{2h,2}^e d\mathbf{x}$ . After calculating the value of  $\gamma_i$  for each vertex of the coarse mesh  $\mathcal{T}_{2h}$ , we use bilinear interpolation of these values to define  $\gamma_j = \sum_i \gamma_i \varphi_i^L(\mathbf{x}_j)$  at the remaining vertices of the submesh  $\mathcal{T}_h$ . Given the values of  $\beta_i$  and  $\gamma_i$  at all vertices of  $\mathcal{T}_h$ , we finally define the blending function  $\alpha_h$  using bilinear interpolation of its nodal values

$$\alpha_i = \max\{\beta_i, \gamma_i\}. \quad (25)$$

This definition implies that the unconstrained high-order Galerkin approximation will be used in smooth subdomains where  $\beta_h = 1$  or  $\gamma_h = 1$ . In the remaining subdomains, the use of  $\alpha_h < 1$  will activate the DMP-preserving artificial diffusion operator (20). In the numerical examples of this section, we calculate the indicators  $\beta_i$  and  $\gamma_i$  just once using the Galerkin solution.

To check if definition (25) ensures preservation of high accuracy for problems with smooth solutions, we perform grid convergence studies for the 2D Poisson problem considered in §5.1. In Figures 5 and 6, we present the DMP sensor  $\beta_h$ , the smoothness indicator  $\gamma_h$ , and the blending function  $\alpha_h$  calculated using the nodal values of the Galerkin approximations  $u_h^L \in V_h(0)$  and  $u_h^H \in V_h(1)$ , respectively, for  $h = \frac{1}{40}$ . In Table 2, we present the convergence history for a family of adaptive spaces  $V_h(\alpha_h)$  generated using blending functions defined by (25) on each level of mesh refinement. For comparison purposes, we show the convergence history for  $\alpha_h$  with  $\alpha_i = \beta_i$  in Table 3. In this experiment, the  $L^2$  convergence rates drop to second order due to the fact that  $\beta_i = 0$  even at smooth local extrema.

To demonstrate that formula (25) is a usable troubled cell detector as well, we use it to design  $hp$ -adaptive spaces for the singularly perturbed advection-diffusion equation (16) and the anisotropic diffusion equation (18). As before, we calculate the indicators  $\beta_h$ ,  $\gamma_h$ , and  $\alpha_h$  using Galerkin approximations in  $V_h(0)$  and  $V_h(1)$ . All computations are performed with  $h = \frac{1}{40}$ . In Figures 7 and 8, we present the indicators and the finite element solution  $u_h \in V_h(\alpha_h)$  of the advection-diffusion problem. In Figures 9 and 10, we show the same set of plots for the anisotropic diffusion problem. It can be seen that the numerical solutions of both problems are free of under- and overshoots that were present in the underlying Galerkin approximations. For this reason, no further adaptation of  $\alpha_h$  is performed in this study. In general, it is advisable to check the validity of DMPs *a posteriori*. If necessary, the values of  $\alpha_i$  may be decreased monotonically until a bound-preserving finite element space  $V_h(\alpha_h)$  is constructed. To make sure that  $\alpha_h$  vanishes identically in elements containing a local extremum,

$h$	Ind. based on Galerkin solution $u_h^L \in V_h(0)$						Ind. based on Galerkin solution $u_h^H \in V_h(1)$					
	$\min(u_h)$	$\max(u_h)$	$\ e\ _{0,\Omega}$	EOC	$\ e\ _{1,\Omega}$	EOC	$\min(u_h)$	$\max(u_h)$	$\ e\ _{0,\Omega}$	EOC	$\ e\ _{1,\Omega}$	EOC
5E-2	-0.9443	0.9443	2.73E-2	–	2.2019	–	-0.9443	0.9443	2.73E-2	–	2.2021	–
2.5E-2	-0.9901	0.9902	4.78E-3	2.51	0.4324	2.34	-0.9901	0.9902	4.78E-3	2.51	0.4324	2.34
1.25E-2	-0.9990	0.9990	5.56E-4	3.10	0.0675	2.67	-0.9990	0.9990	5.57E-4	3.10	0.0675	2.67
6.25E-3	-0.9998	0.9998	6.94E-5	3.00	0.0133	2.34	-0.9998	0.9998	6.94E-5	3.00	0.0133	2.34

Table 2: Grid convergence history for the 2D Poisson test with  $\alpha_i = \max\{\beta_i, \gamma_i\}$ .

$h$	Ind. based on Galerkin solution $u_h^L \in V_h(0)$						Ind. based on Galerkin solution $u_h^H \in V_h(1)$					
	$\min(u_h)$	$\max(u_h)$	$\ e\ _{0,\Omega}$	EOC	$\ e\ _{1,\Omega}$	EOC	$\min(u_h)$	$\max(u_h)$	$\ e\ _{0,\Omega}$	EOC	$\ e\ _{1,\Omega}$	EOC
5E-2	-0.9059	0.9059	4.24E-2	–	3.9658	–	-0.9059	0.9059	4.24E-2	–	3.9669	–
2.5E-2	-0.9556	0.9556	1.19E-2	1.83	1.3638	1.53	-0.9556	0.9556	1.19E-2	1.83	1.3638	1.54
1.25E-2	-0.9833	0.9833	3.15E-3	1.91	0.4463	1.61	-0.9833	0.9833	3.15E-3	1.91	0.4463	1.61
6.25E-3	-0.9942	0.9942	8.18E-4	1.94	0.1364	1.71	-0.9942	0.9942	8.18E-4	1.94	0.1364	1.71

Table 3: Grid convergence history for the 2D Poisson test with  $\alpha_i = \beta_i$ .

definition (25) may be replaced by the safer choice  $\alpha_i = \min_{j \in \mathcal{N}_i} \max(\beta_j, \gamma_j)$ . In applications to time-dependent problems with propagating fronts, additional protection layers should be included to guarantee that steep gradients stay in safe  $\mathbb{P}_1/\mathbb{Q}_1$  regions until the end of the current time step.

## 5.5 Unsteady diffusion in 1D

To study the numerical behavior of the partitioned  $\theta_h$  scheme (12),(13) in the context of embedded boundary / cut cell finite element methods, we consider the one-dimensional diffusion equation

$$\frac{\partial u}{\partial t} - d \frac{\partial^2 u}{\partial x^2} = 0 \quad \text{in } \Omega = (0, 1) \quad (26)$$

with the diffusion coefficient  $d = 10^{-2}$ . The exact solution is the Gaussian hill (cf. [14], p. 243)

$$u(x, t) = \frac{5}{7\sigma(t)} \exp \left\{ - \left( \frac{x - 0.5}{l\sigma(t)} \right)^2 \right\}, \quad l = \frac{7\sqrt{2}}{300}, \quad \sigma(t) = \sqrt{1 + \frac{4dt}{l^2}}. \quad (27)$$

The initial data and boundary conditions for the numerical experiment are also defined by this formula.

We discretize (26) in space using linear finite elements. Given a uniform mesh with the grid points

$$x_i = ih, \quad h = \frac{1}{2N}, \quad i = 0, \dots, 2N,$$

we subdivide the two elements containing the midpoint  $x_N = 0.5$  into  $[0.5 - h, 0.5 - \epsilon] \cup [0.5 - \epsilon, 0.5]$  and  $[0.5, 0.5 + \epsilon] \cup [0.5 + \epsilon, 0.5 + h]$ , where  $\epsilon = 10^{-4} \ll h$  is the size of the small cut cells.

The time step  $\Delta t$  is selected so that the ratio  $\lambda = \frac{\Delta t}{h^2} = \frac{1}{4}$  remains constant in the process of mesh refinement. On a uniform mesh with spacing  $h$ , the forward Euler time discretization ( $\theta_h \equiv 0$ ) is

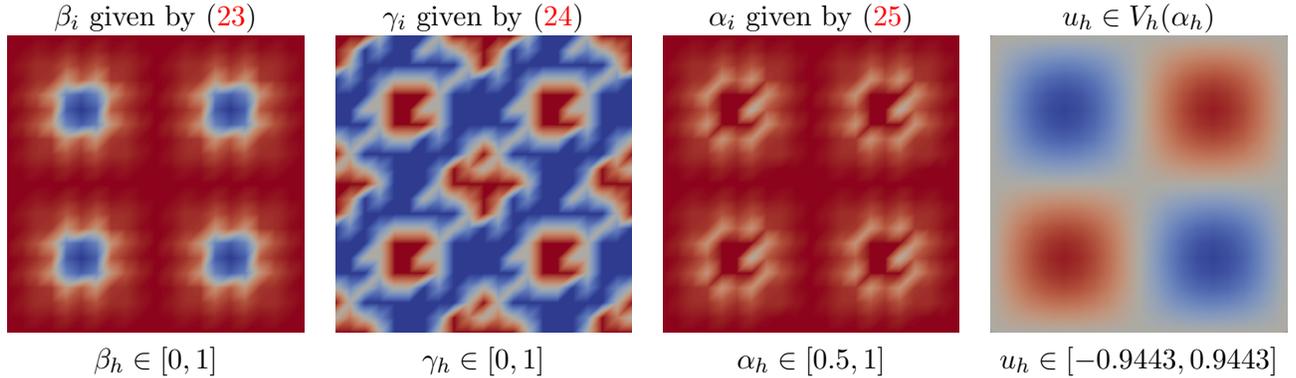


Figure 5: Indicators based on the Galerkin solution  $u_h^L \in V_h(0)$  of the 2D Poisson equation are shown in the left three panels. The solution  $u_h \in V_h(\alpha_h)$  is shown in the right panel.

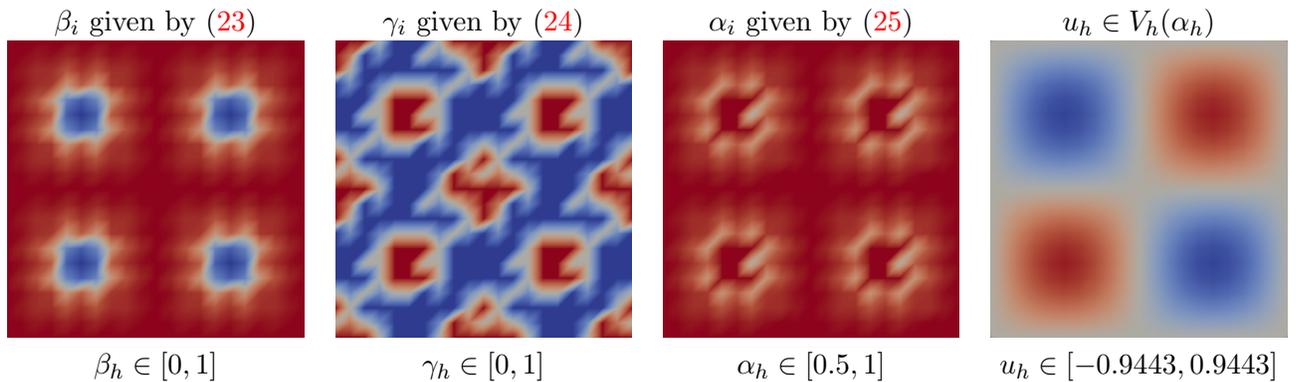


Figure 6: Indicators based on the Galerkin solution  $u_h^H \in V_h(1)$  of the 2D Poisson equation are shown in the left three panels. The solution  $u_h \in V_h(\alpha_h)$  is shown in the right panel.

stable for  $\lambda \leq \frac{1}{2}$  and second-order accurate due to the fact that  $\Delta t = \lambda h^2$ . However, the introduction of small cut cells makes it unstable and the code crashes for  $\Delta t > \frac{1}{2}\epsilon^2$ . To remedy this instability, we employ the partitioned  $\theta$  scheme defined by the continuous piecewise-linear blending function

$$\theta_h(x) := \begin{cases} 1 & \text{if } |x - 0.5| \leq h, \\ 2 - \frac{1}{h}|x - 0.5| & \text{if } h \leq |x - 0.5| \leq 2h, \\ 0 & \text{if } |x - 0.5| \geq 2h. \end{cases} \quad (28)$$

That is, we use the unconditionally stable backward Euler scheme in cut cells, the forward Euler scheme in regular cells that are not adjacent to a cut cell, and a semi-implicit blend in-between.

The numerical solution presented in Fig. 11 was calculated using  $h = \frac{1}{50}$  and the final time  $T = 0.5$ . A zoom of the cut cell subdomain is included to show how strongly the mesh size varies in this test.

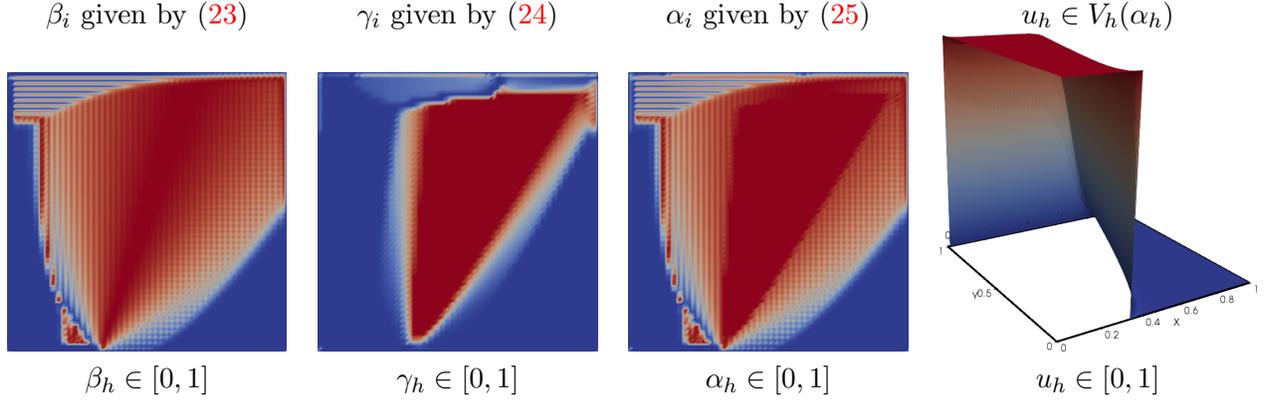


Figure 7: Indicators based on the nodal values of the Galerkin approximation  $u_h^L \in V_h(0)$  and the corresponding solution  $u_h \in V_h(\alpha_h)$  to the advection-diffusion equation (16).

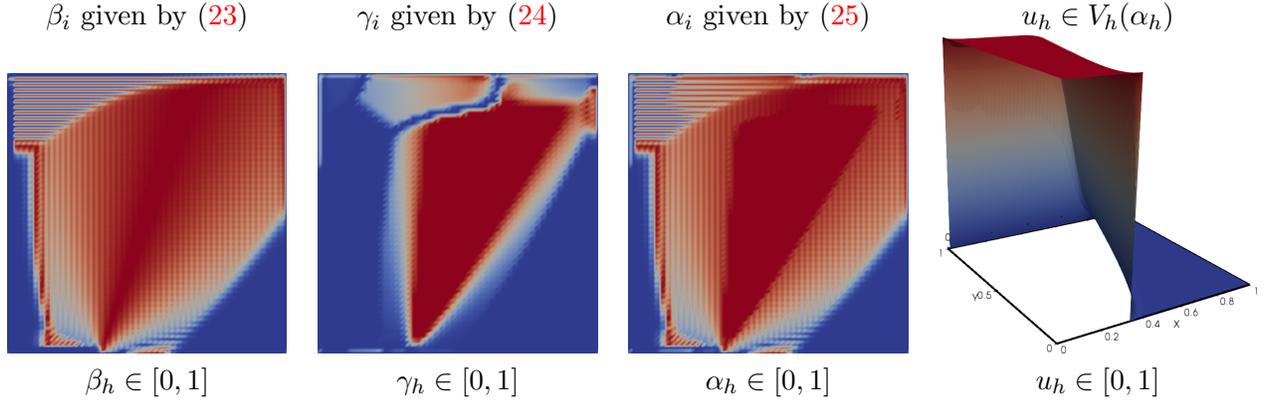


Figure 8: Indicators based on the nodal values of the Galerkin approximation  $u_h^H \in V_h(1)$  and the corresponding solution  $u_h \in V_h(\alpha_h)$  to the to the advection-diffusion equation (16).

The errors w.r.t. the maximum norm  $\|u - u_h\|_{C([0,1])} = \max_{x \in [0,1]} |u(x) - u_h(x)|$  are listed in Table 4. The partitioned  $\theta_h$  scheme is stable and second-order accurate under the same time step restriction as forward Euler. At the same time, it is more efficient than backward Euler since just a few nodes are treated implicitly and the matrix of the linear system to be solved is a block-diagonal matrix. The size of the diagonal blocks is determined by the number of adjacent implicit ( $\theta_i > 0$ ) nodes. Diagonal blocks of size  $1 \times 1$  correspond to explicit ( $\theta_i = 0$ ) nodes.

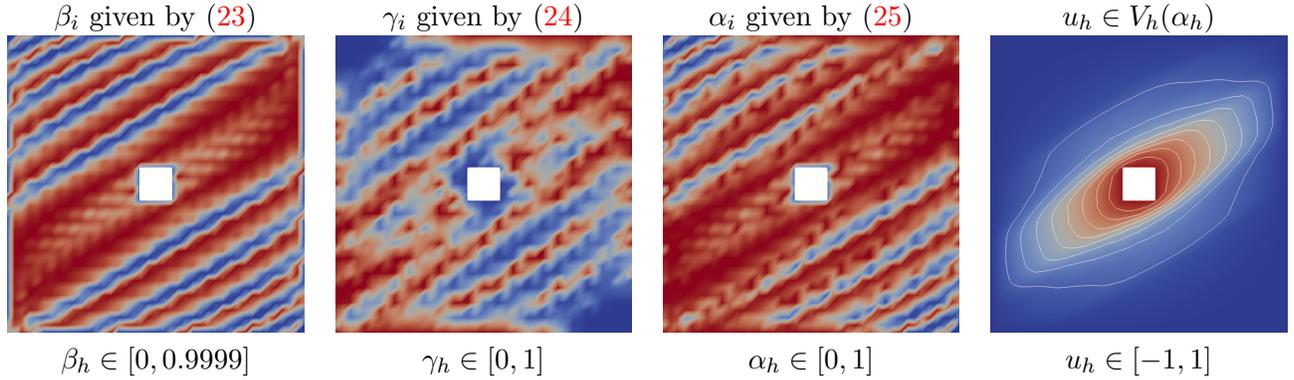


Figure 9: Indicators based on the Galerkin solution  $u_h^L \in V_h(0)$  of the anisotropic diffusion equation (18) are shown in the left three panels. The solution  $u_h \in V_h(\alpha_h)$  is shown in the right panel.

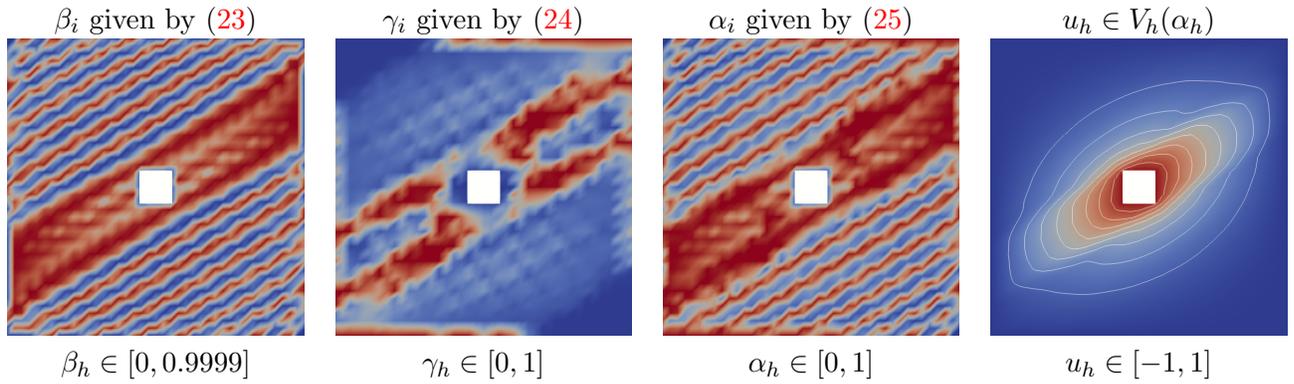


Figure 10: Indicators based on the Galerkin solution  $u_h^H \in V_h(1)$  of the anisotropic diffusion equation (18) are shown in the left three panels. The solution  $u_h \in V_h(\alpha_h)$  is shown in the right panel.

$h$	$\ u - u_h\ _{C([0,1])}$	$EOC$
1/100	3.2765E-04	
1/200	8.6262E-05	1.93
1/400	2.2416E-05	1.94
1/800	5.5032E-06	2.03

Table 4: Diffusion of a 1D Gaussian hill: numerical errors w.r.t. the maximum norm and the experimental order of convergence for the  $\mathbb{P}_1$  Galerkin  $\theta_h$  discretization with  $\Delta t = 0.25h^2$ .

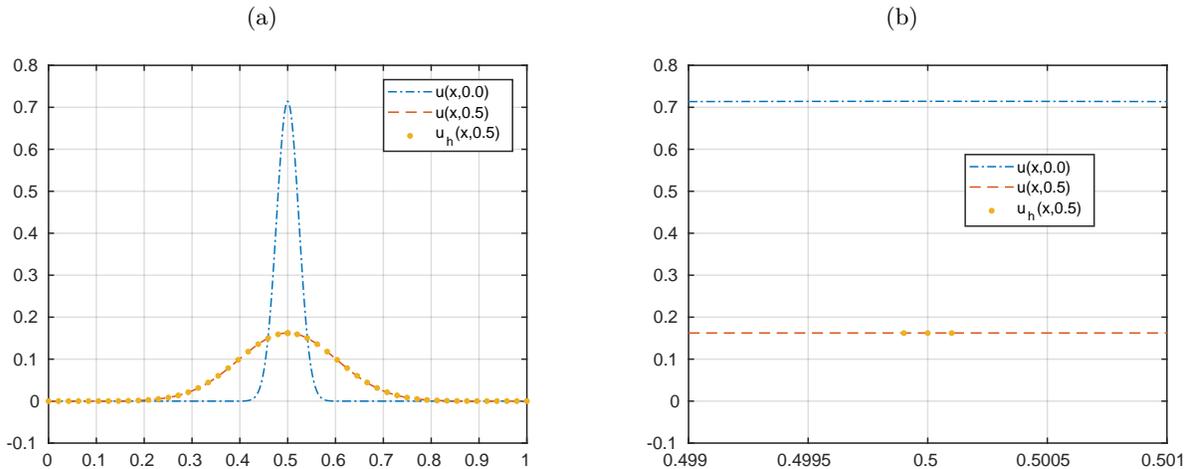


Figure 11: Diffusion of a 1D Gaussian hill: (a) initial data, exact and numerical solutions at  $T = 0.5$ , (b) a zoom of the small cut cell region.  $\mathbb{P}_1$  Galerkin  $\theta_h$  discretization with  $h = \frac{1}{50}$ ,  $\Delta t = \frac{1}{4}h^2$ .

## 6 Conclusions

This work indicates that the partition of unity (PU) approach to blending finite element spaces and time discretizations has significant merit in the context of  $hp$ -adaptation for conforming finite element spaces and explicit-implicit partitioned time discretizations of evolutionary problems. The proposed PU hybridization of high-order basis functions with their piecewise-linear counterparts preserves global continuity of the finite element approximation as well as the number of degrees of freedom and the sparsity pattern of discrete operators. Partitioned time discretizations are well-suited for applications in which the optimal time step size (determined by stability and accuracy considerations) exhibits strong spatial variations. The use of adaptive and possibly time-dependent blending functions based on discrete maximum principles and/or local error indicators would lead to a new generation of high-resolution finite element schemes combining complementary advantages of existing  $hp$ -FEM, limiting techniques, IMEX time integrators, and local time stepping schemes. It is hoped that the present paper provides sufficient motivation for further development and analysis of such methods.

## 7 Acknowledgments

The work of Dmitri Kuzmin was supported by the German Research Association (DFG) under grant KU 1530/23-1. He would like to thank Hennes Hajduk (TU Dortmund University) and Friedhelm Schieweck (Otto von Guericke University Magdeburg) for helpful discussions. The work of Manuel Quezada de Luna was supported in part by an appointment to the Postgraduate Research Participation Program at the U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory (ERDC-CHL) administrated by the Oak Ridge Institute for Science and Education through an

interagency agreement between the U.S. Department of Energy and ERDC. Permission was granted by the Chief of Engineers, US Army Corps of Engineers, to publish this information.

## References

- [1] R. Abgrall. High order schemes for hyperbolic problems using globally continuous approximation and avoiding mass matrices. J. Sci. Comput., 73:461–494, 2017.
- [2] M. Ainsworth, G. Andriamaro, and O. Davydov. Bernstein-Bézier finite elements of arbitrary order and optimal assembly procedures. SIAM J. Sci. Comput., 33(6):3087–3109, 2011.
- [3] S. Badia and J. Bonilla. Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization. Computer Methods Appl. Mech. Engrg., 313:133–158, 2017.
- [4] W. Bangerth and O. Kayser-Herold. Data structures and requirements for *hp* finite element software. ACM Transactions on Mathematical Software, 36:4/1–4/31, 2009.
- [5] G. Barrenechea, V. John, and P. Knobloch. Analysis of algebraic flux correction schemes. SIAM J. Numer. Anal., 54(4):2427–2451, 2016.
- [6] G. Barrenechea, V. John, and P. Knobloch. A linearity preserving algebraic flux correction scheme satisfying the discrete maximum principle on general meshes. Mathematical Models and Methods in Applied Sciences (M3AS), 27(3):525–548, 2017.
- [7] G. R. Barrenechea, V. John, and P. Knobloch. Analysis of algebraic flux correction schemes. SIAM J. Numer. Anal., 54(4):2427–2451, 2016.
- [8] G. R. Barrenechea, E. Burman, and F. Karakatsani. Edge-based nonlinear diffusion for finite element approximations of convection-diffusion equations and its relation to algebraic flux-correction schemes. Numerische Mathematik, 135(2):521–545, 2017.
- [9] M. Bittl and D. Kuzmin. An *hp*-adaptive flux-corrected transport algorithm for continuous finite elements. Computing, 95(1):S27–S48, 2013.
- [10] M. Bittl and D. Kuzmin. The reference solution approach to *hp*-adaptivity in finite element flux-corrected transport algorithms. In I. Lirkov, S. Margenov, and J. Wasniewski, editors, Large Scale Scientific Computing, pages 197–204. Springer, 2014.
- [11] F. Brezzi, L. Marini, and A. Russo. On the choice of a stabilizing subgrid for convection–diffusion problems. Computer Methods in Applied Mechanics and Engineering, 194(2-5):127–148, 2005.
- [12] T. Brunner. Preserving positivity of solutions to the diffusion equation for higher-order finite elements in under resolved regions. Technical Report LLNL-PROC-66674, Lawrence Livermore National Laboratory, 2015.

- [13] L. Demkowicz. Computing with hp-Adaptive Finite Elements. Chapman & Hall/CRC, 2006.
- [14] J. Donea and A. Huerta. Finite Element Methods for Flow Problems. John Wiley & Sons, 2003.
- [15] K. Duraisamy, K. Baeder, and J.-G. Liu. Concepts and application of time-limiters to high resolution schemes. J. of Sci. Comput., 19:139–162, 2003.
- [16] T. N. T. Goodman. Further variation diminishing properties of Bernstein polynomials on triangles. Constructive Approximation, 3:297–305, 1987.
- [17] J.-L. Guermond and R. Pasquetti. A correction technique for the dispersive effects of mass lumping for transport problems. Computer Methods Appl. Mech. and Engrg., 253(1):186–198, 2013.
- [18] J.-L. Guermond and R. Pasquetti. A correction technique for the dispersive effects of mass lumping for transport problems. Computer Methods Appl. Mech. and Engrg., 253(1):186–198, 2013.
- [19] J.-L. Guermond and B. Popov. Invariant domains and second-order continuous finite element approximation for scalar conservation equations. SIAM J. Numer. Anal., 55(6):3120–3146, 2017.
- [20] W. Höhn and H. Mittelmann. Some remarks on the discrete maximum principle for finite elements of higher order. Computing, 27(2):145–154, 1981.
- [21] W. Hundsdorfer, D. I. Ketcheson, and I. Savostianov. Error analysis of explicit partitioned Runge-Kutta schemes for conservation laws. J. Sci. Comput., 63:633–653, 2015.
- [22] C. E. Kees, M. W. Farthing, A. Dimakopoulos, T. de Lataillade, I. Akkerman, A. Ahmadi, A. Bentley, G. Cozzuto, Y. Yang, M. Quezada de Luna, A. Zhang, N. Neri, T. Povich, S. Mattis, S. Patty, B. Chambers, P. Otiñar Morillas, T. J. Corona, M. Malej, M. Saha, N. J. Wani, A. Song, L. Maurel, H. Collins, E. Tovar, Y. Lin, R. Sawko, M. Ragan-Kelley, L. Jenkins, J. Brown, and G. Scovazzi. erdc/proteus: Release for hp-FEM paper, Jan. 2019. URL <https://doi.org/10.5281/zenodo.2546985>.
- [23] D. I. Ketcheson, C. B. MacDonald, and S. J. Ruuth. Spatially partitioned embedded Runge-Kutta methods. SIAM J. Numer. Anal., 51(5):2887–2910, 2013.
- [24] R. C. Kirby. Fast simplicial finite element algorithms using Bernstein polynomials. Numer. Math., 117(4):3087–3109, 2011.
- [25] P. Knabner and L. Angermann. Numerical Methods for Elliptic and Parabolic Partial Differential Equations. Springer, 2003.
- [26] D. Kuzmin and F. Schieweck. A parameter-free smoothness indicator for high-resolution finite element schemes. Central European Journal of Mathematics, 11(8):1478–1488, 2013.

- [27] D. Kuzmin, M. J. Shashkov, and D. Svyatskiy. A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems. Journal of Computational Physics, 228(9):3448–3463, 2009.
- [28] D. Kuzmin, R. Löhner, and S. Turek. Flux-Corrected Transport: Principles, Algorithms, and Applications. Scientific Computation. Springer, 2012.
- [29] C. Lohmann, D. Kuzmin, J. Shadid, and S. Mabuza. Flux-corrected transport algorithms for continuous Galerkin methods based on high order Bernstein finite elements. J. Comput. Phys., 344(1):51–186, 2017.
- [30] R. Löhner. Applied CFD Techniques: An Introduction Based on Finite Element Methods. John Wiley & Sons, 2008.
- [31] S. May and M. Berger. An explicit implicit scheme for cut cells in embedded boundary meshes. Journal of Scientific Computing, 71(3):919–943, 2017.
- [32] J. M. Melenk and I. Babuška. The Partition of Unity Finite Element Method: Basic theory and applications. Computer Methods in Applied Mechanics and Engineering, 139(1-4):289–314, 1996.
- [33] P. Solin, K. Segeth, and I. Dolezel. Higher-Order Finite Element Methods. Chapman and Hall / CRC Press, 2003.
- [34] P. Solin, J. Cervený, and I. Dolezel. Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM. Math. Comput. Simul., 77:117 – 132, 2008.
- [35] P. van Slingerland, M. Borsboom, and C. Vuik. A local theta scheme for advection problems with strongly varying meshes and velocity profiles. Technical Report 08-17, Department of Applied Mathematical Analysis, Delft University of Technology, 2008.

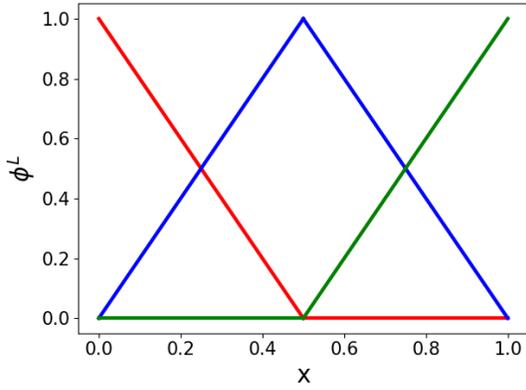
## A Numerical implementation

In this Appendix, we provide brief details about the numerical implementation of the adaptive space discretization proposed in §2. The implementation was done using PROTEUS (<https://proteustoolkit.org>), an open-source, freely downloadable Python toolkit for computational methods and simulations. We provide a release of PROTEUS that contains the implementation of the PUFEM method presented in this work along with all the numerical examples, see `proteus/proteus/tests/BlendedSpaces` in [22]. The space  $V_h(\alpha_h)$  is easy to implement in a standard finite element code if the same data structures are employed for all spaces. This natural implementation strategy implies that the contributions of  $S_{2h,2}$  and  $S_{h,1}$  use the same mappings, the same numbering of degrees of freedom etc. Note that the sparsity patterns of finite element matrices corresponding to the  $S_{2h,2}$  and  $S_{h,1}$  approximations are different. Indeed, the  $S_{2h,2}$  matrices have more non-zero entries than their  $S_{h,1}$  counterparts. For any nontrivial choice of the blending function  $\alpha_h$ , the sparsity pattern of the space  $V_h(\alpha_h)$  is the same

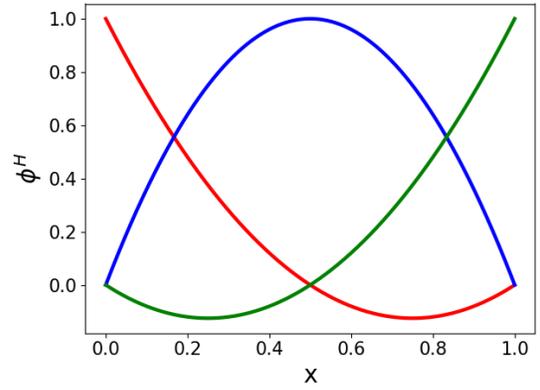
as that of  $S_{2h,2}$ . Therefore, it is worthwhile to use the extended  $S_{2h,2}$  connectivity matrix for the contribution of the space  $S_{h,1}$  as well. Then the size of the  $V_h(\alpha_h)$  element matrices and the data structures for indirect addressing will be the same as in the case of a pure  $S_{2h,2}$  approximation even if the compact-stencil  $S_{h,1}$  subcell approximation is employed in some cells.

The use of common data structures facilitates blending of the two spaces and makes it possible to perform operations with global degrees of freedom directly without any kind of index mapping. In the present 2D implementation, we use quadrilateral elements. The local basis functions for  $S_{h,1}$  and  $S_{2h,2}$  are defined as tensor products of piecewise-linear and quadratic polynomials on the common one-dimensional reference element, see Fig. 12. Using standard mappings from the reference element to physical cells, we construct 9 biquadratic functions  $\varphi_i^{e,H}$  and 9 piecewise-bilinear basis functions  $\varphi_i^{e,L}$  on each quadrilateral cell of the mesh  $\mathcal{T}_{2h}$ . Then standard finite element procedures are invoked to generate element matrices/vectors and perform element-by-element assembly.

When it comes to combining the spaces  $V_h^H = S_{2h,2}$  and  $V_h^L = S_{h,1}$  using a blending function  $\alpha_h \in V_h^L$ , we need to evaluate the partitioned basis functions  $\varphi_i^e = \alpha_h \varphi_i^{e,H} + (1 - \alpha_h) \varphi_i^{e,L}$  and their derivatives at the quadrature points. Note that the value of  $\varphi_i^e(\mathbf{x})$  is an easily computable convex combination of  $\varphi_i^{e,H}(\mathbf{x})$  and  $\varphi_i^{e,L}(\mathbf{x})$ . However, the partial derivatives of  $\varphi_i$  depend on the gradient of  $\alpha_h$  which is defined only on subcells, i.e., on elements of the submesh  $\mathcal{T}_h$  on which the basis functions  $\varphi_i^L$  are continuous bilinear polynomials. To avoid a loss of accuracy due to numerical integration of discontinuous functions on elements of the mesh  $\mathcal{T}_{2h}$ , we map each subcell to a subcell of the reference element, perform numerical integration on subcells using standard quadrature and the Jacobian of the subcell mapping, and add the resulting subcell contribution to the element matrix/vector in the same way in which element contributions are inserted into global matrices/vectors. The use of composite quadrature on subcells makes it possible to integrate the gradients of  $\varphi_i(\alpha_h)$  exactly and is a prerequisite for achieving optimal convergence with partitioned  $V_h(\alpha_h)$  spaces.



(a)



(b)

Figure 12: Definition of (a) piecewise-linear and (b) quadratic basis functions on the same one-dimensional reference element.