Applied Mathematics and Computation 279 (2016) 219–235

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

hp-discontinuous Galerkin method based on local higher order reconstruction☆

Vít Dolejší[a,*], Pavel Solin[b]

[a] Faculty of Mathematics and Physics, Charles University Prague, Sokolovska 83, 186 75 Prague, Czech Republic
[b] Department of Mathematics and Statistics, University of Nevada in Reno, USA

ARTICLE INFO

MSC:
65M50
65M60
65D05

Keywords:
hp-adaptivity
Error estimates
Higher-order reconstruction
Optimal re-meshing

ABSTRACT

We present a new adaptive higher-order finite element method (*hp*-FEM) for the solution of boundary value problems formulated in terms of partial differential equations (PDEs). The method does not use any information about the problem to be solved which makes it robust and equation-independent. It employs a higher-order reconstruction scheme over local element patches which makes it faster and easier to parallelize compared to *hp*-adaptive methods that are based on the solution of a reference problem on a globally *hp*-refined mesh. The method can be used for the solution of linear as well as nonlinear problems discretized by conforming or non-conforming finite element methods, and it can be combined with arbitrary a posteriori error estimators. The performance of the method is demonstrated by several examples carried out by the discontinuous Galerkin method.

1. Introduction

Adaptive methods are an efficient tool for the numerical solution of PDEs. Automatic mesh refinement or, more generally, an enhancement of the functional space where the approximate solution is sought, can significantly reduce the computational cost. A prominent place among adaptive methods has the *hp*-FEM which leads to unconditional exponential convergence [1–4].

Various approaches to automatic adaptivity include refining an element without increasing its polynomial degree (*h*-refinement), increasing the polynomial degree of an element without spatial subdivision (*p*-refinement), and performing refinements that combine spatial splitting of an element with various distributions of the polynomial degrees in subelements (genuine *hp*-refinement [5,6]).

To achieve exponential convergence, large higher-order elements must be used where the solution is smooth, and at the same time small low-order elements must be used where the solution exhibits non-smooth features such as singularities or internal/boundary layers. Therefore it would seem that looking at the smoothness of the solution is the best way to design a *hp*-adaptive method. However, *hp*-adaptive strategies based on smoothness estimation usually can only decide between *h*- and *p*-refinements because they do not have enough information to select optimal genuine *hp*-refinements [7–9].

☆ The research of V. Dolejší was supported by Grant no. 13-00522S of the Czech Science Foundation. The author acknowledges also the membership in the Nečas Center for Mathematical Modeling ncmm.karlin.mff.cuni.cz. The research of P. Solin was supported by Grant no. P102/11/0498 of the Grant Agency of the Czech Republic.

* Corresponding author. Tel.: +420 604239275.
E-mail addresses: dolejsi@karlin.mff.cuni.cz (V. Dolejší), solin@unr.edu (P. Solin).

http://dx.doi.org/10.1016/j.amc.2016.01.024
0096-3003/

To take full advantage of genuine *hp*-refinements, one has to obtain more information about the error – not only as an error estimate in the form of a number per element, but about its shape as a function that is defined inside an element. This can be done by solving a reference problem on a globally *hp*-refined mesh [5,6,10]. It leads to superior convergence rates in terms of degrees of freedom, but the reference problem tends to be huge and therefore the convergence is rather slow in terms of computing time, see [11] where a deep comparison of different *hp*-strategies were compared.

In this paper we propose a novel method for automatic *hp*-adaptivity that is guided by the approximation error, but it removes the tedious computation of the global reference solution. Instead, it calculates a more accurate approximation on local element patches constructed for each element separately. With the aid of a weighted least square reconstruction, we construct piecewise polynomial function which approximate the exact solution. The presented *hp*-adaptive strategy is based on comparing the higher-order reconstruction with the approximate solution u_h and its L^2-projections to lower-degrees polynomial spaces. Then, for each candidate, we directly estimate the number of degrees of freedom necessary to achieve a local tolerance for each element and propose a new (better) polynomial approximation degree and a new (better) element size. Consequently, a new triangular grid is constructed.

Although we originally developed this technique for the discontinuous Galerkin method, it can be simply modified to other types of finite element approximations including conforming finite elements, mixed finite elements, etc. Moreover, since this approach is based on the reconstruction of the approximate solution, we can employ it for arbitrary (linear as well as non-linear) boundary value problems. Finally, its extension to 3D problems is straightforward.

The outline of the paper is as follows: In Section 2, we introduce the governing equations and their discretization by the discontinuous Galerkin method. In Section 3, we introduce the higher-order reconstruction technique and formulate the saturation assumption. Its validity is numerically demonstrated in Section 4. The main novelty of this paper is presented in Section 5, where we present the *hp*-adaptive strategy. In Section 6, we present three numerical examples that demonstrate the performance and robustness of the proposed method.

2. Problem description

2.1. Governing equations

We consider the nonlinear *convection–diffusion problem*

$$\nabla \cdot \boldsymbol{f}(u) - \nabla \cdot (\boldsymbol{K}(u)\nabla u) = g(x), \tag{1a}$$

$$u|_{\partial\Omega_D} = u_D, \tag{1b}$$

$$\boldsymbol{K}(u)\frac{\partial u}{\partial \boldsymbol{n}}\Big|_{\partial\Omega_N} = g_N, \tag{1c}$$

where $u : \Omega \to \mathbb{R}$ is an unknown scalar function defined on $\Omega \in \mathbb{R}^2$. We assume that Ω is polygonal for simplicity. Moreover, $\boldsymbol{f}(u) = (f_1(u), f_2(u)) : \mathbb{R} \to \mathbb{R}^2$ and $\boldsymbol{K}(u) = \{K_{ij}(u)\}_{i,j=1}^2 : \mathbb{R} \to \mathbb{R}^{2\times 2}$ are nonlinear functions of their arguments, $\boldsymbol{n}$ is the unit outer normal to $\partial\Omega$ and $\emptyset \neq \partial\Omega_D \cup \partial\Omega_N = \partial\Omega$ are disjoint parts of the boundary of Ω. Symbols ∇ and $\nabla\cdot$ mean the gradient and divergence operators, respectively.

We assume that $f_s \in C^1(\mathbb{R})$, $f_s(0) = 0$, $s = 1, 2$, $\boldsymbol{K}$ is bounded and positively definite, $g \in L^2(\Omega)$, u_D is the trace of some $u^* \in H^1(\Omega) \cap L^\infty(\Omega)$ on $\partial\Omega_D$ and $g_N \in L^2(\partial\Omega_N)$. We use the standard notation for function spaces (see, e. g., [12]): $L^p(\Omega)$ denote the Lebesgue spaces, $W^{k,p}(\Omega)$, $H^k(\Omega) = W^{k,2}(\Omega)$ are the Sobolev spaces and $P^k(M)$ denotes the space of polynomial functions of degree $\le k$ defined on the domain $M \subset \mathbb{R}^2$. Let us note that any function from $P^k(M)$ can be interpreted as a polynomial function defined on $\mathbb{R}^2$ restricted to M. By $\phi|_M$ we denote the restriction of a function ϕ on M.

In order to introduce the weak solution, we define the spaces

$$V := \{v;\ v \in H^1(\Omega),\quad v|_{\partial\Omega_D} = 0\},\quad W := \{v;\ v \in H^1(\Omega),\quad v - u^* \in V\}. \tag{2}$$

We say that function u is the *weak solution* of (1), if the following conditions are satisfied

$$u \in W \cap L^\infty(\Omega), \tag{3a}$$

$$\int_\Omega [\nabla \cdot \boldsymbol{f}(u)\, v + (\boldsymbol{K}(u)\nabla u) \cdot \nabla v]\,\mathrm{d}x = \int_\Omega g v\,\mathrm{d}x + \int_{\partial\Omega_N} g_N\, v\,\mathrm{d}S \quad \forall v \in V. \tag{3b}$$

The assumption $u \in L^\infty(\Omega)$ in (3) guarantees the boundedness of functions $\boldsymbol{f}(u)$ and $\boldsymbol{K}(u)$ and therefore the existence of the integrals in (3a). This assumption can be weakened if functions $\boldsymbol{f}(u)$ and $\boldsymbol{K}(u)$ satisfy some growth conditions.

2.2. Discretization of the problem

Let $\mathscr{T}_h$ ($h > 0$) be a partition of the closure $\overline{\Omega}$ of the domain Ω into a finite number of triangles K with mutually disjoint interiors. We call $\mathscr{T}_h = \{K\}_{K\in\mathscr{T}_h}$ a *triangulation* of Ω and for simplicity, we assume that $\mathscr{T}_h$ satisfies the conforming properties from the finite element method, see, e.g, [13]. The diameter of $K \in \mathscr{T}_h$ is denoted by h_K.

By $\mathscr{F}_h$ we denote the set of all open edges of all elements $K \in \mathscr{T}_h$. Further, the symbol $\mathscr{F}_h^I$ stands for the set of all $\Gamma \in \mathscr{F}_h$ that are contained in Ω (inner edges). Moreover, we introduce notations $\mathscr{F}_h^D$ and $\mathscr{F}_h^N$ for the sets of all $\Gamma \in \mathscr{F}_h$ such that $\Gamma \subset \partial\Omega_D$ and $\Gamma \subset \partial\Omega_N$, respectively. Finally, for each $\Gamma \in \mathscr{F}_h$, we define a unit normal vector $\boldsymbol{n}_\Gamma$. We assume that for $\Gamma \in \mathscr{F}_h^D \cup \mathscr{F}_h^N$ the vector $\boldsymbol{n}_\Gamma$ has the same orientation as the outer normal of $\partial\Omega$. For $\boldsymbol{n}_\Gamma$, $\Gamma \in \mathscr{F}_h^I$, the orientation is arbitrary but fixed for each edge.

Over the triangulation $\mathscr{T}_h$ we define the so-called *broken Sobolev space* $H^s(\Omega, \mathscr{T}_h) := \{v; v|_K \in H^s(K)\ \forall K \in \mathscr{T}_h\}$, $s \geq 0$ with the seminorm $|v|_{H^s(\Omega,\mathscr{T}_h)} := \left(\sum_{K\in\mathscr{T}_h} |v|^2_{H^s(K)}\right)^{1/2}$, where $|\cdot|_{H^s(K)}$ denotes the seminorm of the Sobolev space $H^s(K)$, $K \in \mathscr{T}_h$.

Moreover, to each $K \in \mathscr{T}_h$, we assign a positive integer p_K (=local polynomial degree). Then we define the set $\mathsf{p} := \{p_K, K \in \mathscr{T}_h\}$ and the finite-dimensional subspace of $H^2(\Omega, \mathscr{T}_h)$ which consists of discontinuous piecewise polynomial functions associated with the vector p by

$$S_h^{\mathsf{p}} = \{v;\ v \in L^2(\Omega), v|_K \in P_{p_K}(K)\ \forall K \in \mathscr{T}_h\}, \tag{4}$$

where $P_{p_K}(K)$ denotes the space of all polynomials on K of degree $\leq p_K, K \in \mathscr{T}_h$. The dimension of S_h^{p} is called the number of *degrees of freedom* (DOF) which satisfies

$$\mathrm{DOF} := \dim S_h^{\mathsf{p}} = \sum_{K\in\mathscr{T}_h} (p_K + 1)(p_K + 2)/2. \tag{5}$$

The pair $\{\mathscr{T}_h, \mathsf{p}\}$ is called the *hp-mesh* and we use the notation $\mathscr{T}_h^{\mathsf{p}} := \{\mathscr{T}_h, \mathsf{p}\}$. The *hp*-mesh $\mathscr{T}_h^{\mathsf{p}}$ uniquely corresponds to the space S_h^{p} and vice-versa.

For the purpose of the presented *hp*-adaptation method, we also define the space

$$S_h^{\mathsf{p}+1} := \{v_h \in L^2(\Omega);\ v_h|_K \in P^{p_K+1}(K)\ \forall K \in \mathscr{T}_h\}. \tag{6}$$

Obviously, $S_h^{\mathsf{p}} \subset S_h^{\mathsf{p}+1} \subset H^2(\Omega, \mathscr{T}_h)$.

Furthermore, for $\Gamma \in \mathscr{F}_h^I$ and $v \in H^2(\Omega, \mathscr{T}_h)$, we denote by $v|_\Gamma^{(+)}$ and $v|_\Gamma^{(-)}$ the trace of v on Γ in the same and the opposite direction of $\boldsymbol{n}_\Gamma$, respectively. Moreover, by $\langle v\rangle_\Gamma$ and $[\![v]\!]_\Gamma$ we denote the mean value and the jump of v on Γ, respectively. In case that $\boldsymbol{n}_\Gamma$, $[[\cdot]]_\Gamma$ and $\langle\cdot\rangle_\Gamma$ are arguments of $\int_\Gamma \ldots \mathrm{d}S$, $\Gamma \in \mathscr{F}_h$, we omit the subscript Γ and write simply $\boldsymbol{n}$, $[[\cdot]]$ and $\langle\cdot\rangle$, respectively.

We discretize Eq. (1a) with the aid of the interior penalty Galerkin (IPG) variant of the discontinuous Galerkin (DG) method in the same way as in [14,15]. For $u, v \in H^2(\Omega, \mathscr{T}_h)$ we define the form $a_h : H^2(\Omega, \mathscr{T}_h) \times H^2(\Omega, \mathscr{T}_h) \to \mathbb{R}$ by

$$\begin{aligned} a_h(u,v) :=& \sum_{\Gamma\in\mathscr{F}_h} \int_\Gamma H(u|_\Gamma^{(+)}, u|_\Gamma^{(-)}, \boldsymbol{n})\,[\![v]\!]\,\mathrm{d}S - \sum_{K\in\mathscr{T}_h} \int_K \boldsymbol{f}(u)\cdot\nabla v\,\mathrm{d}x \\ &+ \sum_{K\in\mathscr{T}_h} \int_K \boldsymbol{K}(u)\nabla u\cdot\nabla v\,\mathrm{d}x + \sum_{\Gamma\in\mathscr{F}_h^I} \int_\Gamma \left(-\langle \boldsymbol{K}(u)\nabla u\rangle\cdot\boldsymbol{n}[\![v]\!] + \theta\langle \boldsymbol{K}(u)\nabla v\rangle\cdot\boldsymbol{n}[\![u]\!] + \sigma[\![u]\!][\![v]\!]\right)\mathrm{d}S \\ &+ \sum_{\Gamma\in\mathscr{F}_h^D} \int_\Gamma \left(-\boldsymbol{K}(u)\nabla u\cdot\boldsymbol{n}\,v + \theta\boldsymbol{K}(u)\nabla v\cdot\boldsymbol{n}\,(u-u_D) + \sigma(u-u_D)\,v\right)\mathrm{d}S - \int_\Omega g v\,\mathrm{d}x - \int_{\partial\Omega_N} g_N v\,\mathrm{d}S, \end{aligned} \tag{7}$$

where $\theta = -1, 0$ and 1 for SIPG, IIPG and NIPG variants of discontinuous Galerkin method, respectively, the penalty parameter σ is chosen by $\sigma|_\Gamma = \varepsilon C_W h_\Gamma^{-1}$, $\Gamma \in \mathscr{F}_h$, where $h_\Gamma = \mathrm{diam}(\Gamma)$, $\Gamma \in \mathscr{F}_h$, ε denotes the amount of diffusion (we put $\varepsilon|_\Gamma := \langle \boldsymbol{K}(u)\rangle_\Gamma$) and $C_W > 0$ is a suitable constant which guarantees the convergence of the method. Numerical analysis implies (see, e.g., [32]) that C_W can be constant for all polynomial approximation degrees p for the NIPG variant whereas it has to be chosen proportionally to p for the SIPG and IIPG variants. E.g., in [33], the relation $C_W \sim p^2$ was employed for the SIPG variant. However, our numerical experiments show that the IIPG scheme is stable for the value C_W independent on p. The numerical examples in the paper were obtained by the IIPG method with $C_W = 20$ for all polynomial degrees.

The function H in (7) is the *numerical flux*, well-known from finite volume methods (see, e.g., [16], Section 3.2), which approximates the convective flux by $\boldsymbol{f}(u)\cdot\boldsymbol{n} \approx H(u|_\Gamma^{(+)}, u|_\Gamma^{(-)}, \boldsymbol{n})$ on an element edge. On $\partial\Omega_D$ the value $u|_\Gamma^{(-)}$ is taken from the boundary conditions (1b) and on $\partial\Omega_N$ the value $u_\Gamma^{(-)}$ is extrapolated from the interior of Ω. We shall assume that the numerical flux is *conservative* and *consistent*, i.e., $H(u,v,\boldsymbol{n}) = -H(v,u,-\boldsymbol{n})$ and $H(u,u,\boldsymbol{n}) = \boldsymbol{f}(u)\cdot\boldsymbol{n}$, respectively, see [14] for details.

We say that function $u_h \in S_h^{\mathsf{p}}$ is an *approximate solution* of (3), if

$$a_h(u_h, v_h) = 0 \quad \forall v_h \in S_h^{\mathsf{p}}. \tag{8}$$

The problem (8) represents a system of nonlinear algebraic equations, which we solve by a *damped Newton-like method*, where the Jacobi matrix is replaced by the *flux matrix* that arises by a partial linearization of the form a_h. More details, including the used stopping criteria, can be found in [17].

The time-dependent variant of (1) was analyzed in [18], see also [19], where quasilinear elliptic boundary value problems were considered. Hence, the approximate solution satisfies the following *a priori error estimate*

$$\|u-u_h\|_{\mathrm{DG}} \le C \sum_{K\in\mathscr{T}_h} \frac{h_K^{2\mu_K-2}}{p_K^{2s_K-3}} \|u\|^2_{H^{\mu_K}(K)}, \tag{9}$$

where u and u_h are the exact and the approximate solution, respectively, $C>0$ is a constant independent of h_K and $p_K, K\in\mathscr{T}_h$, p_K is the polynomial approximation degree on K, s_k denotes the *local Sobolev regularity* of the exact solution on K (i.e., $u|_K \in H^{s_K}(K), K\in\mathscr{T}_h$), $\mu_K=\min(p_K+1, s_K)$ and the DG-norm is defined by

$$\|v\|_{\mathrm{DG}} := \left(|v|^2_{H^1(\Omega,\mathscr{T}_h)} + \|v\|^2_{\mathrm{J}}\right)^{1/2}, \quad \text{where} \quad \|v\|^2_{\mathrm{J}} := \sum_{\Gamma\in\mathscr{F}_h^I}\int_\Gamma \sigma[\![v]\!]^2\,\mathrm{d}S + \sum_{\Gamma\in\mathscr{F}_h^D}\int_\Gamma \sigma v^2\,\mathrm{d}S. \tag{10}$$

In the following we are going to estimate the error in the broken H^1-seminorm, i.e., $|u-u_h|_{H^1(\Omega,\mathscr{T}_h)}$. It will be more rigorous to estimate the error in the DG-norm but since $[\![u_h-u]\!]|_\Gamma = [\![u_h]\!]|_\Gamma, \Gamma\in\mathscr{F}_h^I$ and $u|_\Gamma = u_D, \Gamma\in\mathscr{F}_h^D$, we have

$$\|u-u_h\|^2_{\mathrm{DG}} = |u-u_h|^2_{H^1(\Omega,\mathscr{T}_h)} + \|u-u_h\|^2_{\mathrm{J}} = |u-u_h|^2_{H^1(\Omega,\mathscr{T}_h)} + \sum_{\Gamma\in\mathscr{F}_h^D}\int_\Gamma \sigma(u_D-u_h)^2\,\mathrm{d}S.$$

The last term can be evaluated directly.

3. Error estimates based on a higher-order reconstruction

In [20–22], the authors compute a *reference solution* $u^{\mathrm{ref}}\in S^{\mathsf{p}+1}_{h/2}$ on a globally refined mesh with an increased polynomial degree of elements, to guide automatic *hp*-adaptivity. The reference solution is defined by

$$a_{\frac{h}{2}}(u^{\mathrm{ref}}, v_h) = 0 \quad \forall v_h \in S^{\mathsf{p}+1}_{h/2}, \tag{11}$$

where $a_{\frac{h}{2}}$ is defined by (7) on the triangular mesh $\mathscr{T}_{h/2}$, which arises by the refinement of $\mathscr{T}_h$, where each $K\in\mathscr{T}_h$ is split onto 4 sub-triangles by connecting the middles of its edges, and

$$S^{\mathsf{p}+1}_{h/2} := \{v_h\in L^2(\Omega);\ v_h|_{K'}\in P^{p_K+1}(K')\ \forall K'\in\mathscr{T}_{h/2}\}. \tag{12}$$

Then the *error indicator* is defined by

$$\tilde{\eta}_K := \|u_h - u^{\mathrm{ref}}\|_K \quad \forall K\in\mathscr{T}_h. \tag{13}$$

This approach is superior to many other techniques from the point of view of the rate of the convergence, which was clearly demonstrated in the paper [11], where 13 different *hp*-strategies were compared. However, this approach is very time-consuming due to the necessity to compute the reference solution (11).

Our idea is to obtain an analogue to the reference solution without the necessity to solve the large global problem (11). We employ a *high-order piecewise polynomial* reconstruction for local element patches.

3.1. Higher-order reconstruction

Let $u_h\in S_h^{\mathsf{p}}$ be the approximate solution given by (8). Our aim is to construct for each $K\in\mathscr{T}_h$ a function $\tilde{u}_K\in P^{p_K+1}(K)$ such that it approximate the exact solution on K better than u_h, i.e.,

$$|\tilde{u}_K - u|_K| \ll |u_h|_K - u|_K|, \quad K\in\mathscr{T}_h, \tag{14}$$

where $u|_K$ and $u_h|_K$ are the restriction of u and u_h on K, respectively. In order to construct $\tilde{u}_K$, we use a *weighted least square approximation* from the elements sharing at least a vertex with K. Let $K\in\mathscr{T}_h$, we denote by $\mathscr{D}_K$ the patch

$$\mathscr{D}_K := K\cup\{K'\in\mathscr{T}_h;\ K' \text{ share at least a vertex with } K\} \tag{15}$$

and $w_{K'}>0$, $K'\in\mathscr{D}_K$ be the *weights*. Then we define a function $\tilde{U}_K\in P^{p_K+1}(\mathscr{D}_K)$ by

$$\tilde{U}_K := \arg\min_{U_h\in P^{p_K+1}(\mathscr{D}_K)} \sum_{K'\in\mathscr{D}_K} w_{K'}\|U_h-u_h\|^2_{H^1(K')}. \tag{16}$$

The existence and uniqueness of $\tilde{U}_K$ follows from the fact that $P^{p_K+1}(\mathscr{D}_K)$ is a complete finite-dimensional space. Fig. 1 illustrates the higher order reconstruction in 1D. From the piecewise polynomial function $u_h\in S_h^{\mathsf{p}}$, we construct the polynomial function U_h over the whole patch $\mathscr{D}_K$ by the weighted least-square technique (16) and consequently we restrict $\tilde{U}_K$ just to element K and obtain $\tilde{u}_K$. Examples of 2D patches $\mathscr{D}_K$ corresponding to interior and boundary elements are shown in Fig. 2.

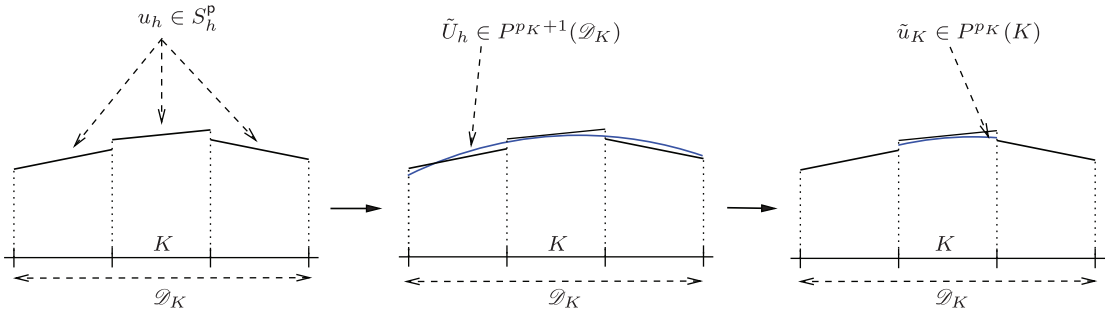


Fig. 1. 1D illustration of the higher order reconstruction on element $K \in \mathscr{T}_h$: original discontinuous piece-wise linear function u_h on K and its neighbors, the higher order (quadratic) reconstruction $\tilde{U}_h$ (blue) and its restriction $\tilde{u}_K =: \tilde{u}_h|_K$ to K. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

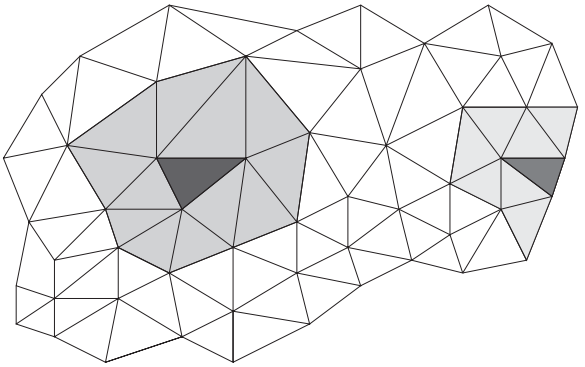


Fig. 2. Examples of patches $\mathscr{D}_K$ corresponding to interior and boundary elements.

The evaluation of the weighted least square approximation can be carried out in the following way. Let $\varphi_i, i = 1, \ldots, d_K$ be basis function of $P^{p_K+1}(K)$ which can be extended as polynomials to $P^{p_K+1}(\mathscr{D}_K)$. Obviously, $d_K := \dim P^{p_K+1}(\mathscr{D}_K) = (p_K + 2)(p_K + 3)/2$. Then problem (16) is equivalent to the linear algebraic system

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}, \quad \mathbb{A} = \{A_{i,j}\}_{i,j=1}^{d_K}, \quad A_{i,j} = \sum_{K' \in \mathscr{D}_K} w_{K'}(\varphi_i, \varphi_j)_{1,K'}, \quad i, j = 1, \ldots, d_K,$$

$$\boldsymbol{b} = \{b_i\}_{i=1}^{d_K}, \quad b_i = \sum_{K' \in \mathscr{D}_K} w_{K'}(u_h, \varphi_i)_{1,K'}, \quad i, j = 1, \ldots, d_K,$$

where $(\cdot,\cdot)_{1,K'}$ denotes the scalar product generating the H^1-norm on $K' \in \mathscr{D}_K$ defined by

$$(v, w)_{1,K'} = \int_{K'} (v w + \nabla v \cdot \nabla w)\, \mathrm{d}x.$$

Let $\boldsymbol{x} = (x_1, \ldots, x_{d_K})^{\mathrm{T}}$ be the solution of (17) then $\tilde{U}_K = \sum_{i=1}^{d_K} x_i \varphi_i$. Finally, we put

$$\tilde{u}_K := \tilde{U}_K|_K, \tag{18}$$

where $\tilde{U}_K$ is given by (16). Concerning the weights $w_{K'}, K' \in \mathscr{D}_K$, we use the values

$$w_{K'} := \begin{cases} 1 & \text{if } K' = K \text{ or } K \text{ and } K' \text{ are neighbors,} \\ \epsilon & \text{otherwise,} \end{cases} \tag{19}$$

where $\epsilon > 0$ is a small parameter. The role of ϵ is to stabilize the higher order reconstruction in situations, where the polynomial degrees are too varying on $\mathscr{D}_K$. Therefore, we put

$$\epsilon := \bar{\epsilon} \max(0, \quad \Delta p_K - 1), \quad \text{where } \Delta p_K = \max_{K' \in \mathscr{D}_K} p_{K'} - \min_{K' \in \mathscr{D}_K} p_{K'} \tag{20}$$

and $\bar{\epsilon} = 0.02$ is the empirical constant. Numerical experiments show that the results are not too much sensitive to the choice of $\bar{\epsilon}$, e.g., the values $\bar{\epsilon} \in (0.005, 0.1)$ give similar results. The relations (19) and (20) imply that if the polynomial degrees on $\mathscr{D}_K$ are varying at most one degree ($\Delta p_K \le 1$) then only neighboring elements are used for the higher order reconstruction.

Finally, we define the higher order reconstruction of $u_h \in S_h^{\mathsf{p}}$ by an element-wise compositions of the restrictions of $\tilde{u}_K$ on $K \in \mathscr{T}_h$, i.e.,

$$\tilde{u}_h \in S_h^{\mathsf{p}+1}: \quad \tilde{u}_h|_K := \tilde{u}_K \quad \forall K \in \mathscr{T}_h, \tag{21}$$

where $\tilde{u}_K$ is given by (18).

Remark 1. Let us mention that the presented higher order reconstruction technique is significantly faster than the standard approach of the computation of the reference solution mentioned above. The numerical experiments presented hereafter show that the reconstruction requires about 10% of the total computational time for linear problems and still less for nonlinear problems (since the solution of nonlinear PDEs is more time consuming). Therefore, the computational time for the higher order reconstruction is practically negligible in comparison to the computational time of the reference solution (11) on a globally refined grid.

3.2. Saturation assumption

By (21), we defined a higher-order polynomial reconstruction $\tilde{u}_h$ which should better approximate the exact solution than the approximate one u_h. This property is formulated as the *saturation assumption*, which will be verified numerically in Section 4 by two examples.

Assumption 2. Let $u \in V$ be the exact solution of (3) and $u_h \in S_h^p$ be the approximate solution given by (8). Let $\tilde{u}_h \in S_h^{p+1}$ be the higher-order reconstruction defined by (21). Then we assume that

$$e_h := u - u_h \approx \tilde{u}_h - u_h =: \mathscr{E}_h \quad \text{in } \Omega, \tag{22}$$

where e_h denotes the discretization error and $\mathscr{E}_h$ its approximation by the higher-order reconstruction.

The quantity $\mathscr{E}_h$ is used for the definition of the presented *hp*-adaptation strategy. Moreover, in this paper, we employ (22) also for the estimation of the error. Therefore, we define the *global* and *element error indicators* by

$$\eta := |\mathscr{E}_h|_{H^1(\Omega,\mathscr{T}_h)} \quad \text{and} \quad \eta_K := |\mathscr{E}_h|_{H^1(K)} \quad \forall K \in \mathscr{T}_h. \tag{23}$$

Obviously, $\eta^2 = \sum_{K\in\mathscr{T}_h} \eta_K^2$. However, generally, η and η_K can be replaced by another error estimators or indicators appearing in the literature, see, e.g., [23,24].

4. Verification of assumption (22)

In this section we numerically verify the saturation assumption (22) by two (linear and nonlinear) convection–diffusion equations: the first one gives a regular solution with two weak boundary layers and the second one contains a corner singularity. For both problems, we carried out computations using polynomial approximations P^p for $p = 1, \dots, 6$ on uniform triangular meshes with mesh spacing $h = 1/8$, $h = 1/16$, $h = 1/32$ and $h = 1/64$.

For each computation, we evaluate the error $e_h = u - u_h$ and its estimate $\mathscr{E}_h = \tilde{u}_h - u_h$ in the broken H^1-seminorm, and the *effectivity index* given by

$$i^{\text{eff}} := \frac{|\mathscr{E}_h|_{H^1(\Omega,\mathscr{T}_h)}}{|e_h|_{H^1(\Omega,\mathscr{T}_h)}}. \tag{24}$$

We also plot the local distribution of the error $|e_h|_{H^1(K)}, K \in \mathscr{T}_h$ and its estimator $|\mathscr{E}_h|_{H^1(K)}, K \in \mathscr{T}_h$ for selected cases. Moreover, we evaluate the *experimental order of convergence* (EOC) according to the formula

$$|e_h| \approx c\, h^{\text{EOC}}, \tag{25}$$

where $c > 0$ is a constant, $h = \max_{K\in\mathscr{T}_h} h_K$ and $\text{EOC} \in \mathbb{R}$ is the experimental order of convergence. Let $|e_{h_1}|$ and $|e_{h_2}|$ be computational errors of the numerical solutions obtained on two different meshes $\mathscr{T}_{h_1}$ and $\mathscr{T}_{h_2}$, respectively. Then from (25), eliminating the constant c, we obtain

$$\text{EOC} = \frac{\log(|e_{h_1}|/|e_{h_2}|)}{\log(h_1/h_2)}. \tag{26}$$

Similarly, we evaluate the EOC of the approximation $\mathscr{E}_h$.

Finally, hereafter we present the relative computational time of the higher order reconstruction (in percent) for each case, in the tables hereafter the columns %cpu.

4.1. Regular problem

We consider the scalar linear convection–diffusion equation (similarly as in [25,26])

$$-\varepsilon\triangle u - \frac{\partial u}{\partial x_1} - \frac{\partial u}{\partial x_2} = g \quad \text{in } \Omega = (0,1)^2, \tag{27}$$

where $\varepsilon = 0.1$ is a constant diffusion coefficient. We prescribe a Dirichlet boundary condition on $\partial\Omega$ and the source term g such that the exact solution has the form

$$u(x_1,x_2) = (c_1 + c_2(1-x_1) + e^{-x_1/\varepsilon})(c_1 + c_2(1-x_2) + e^{-x_2/\varepsilon}) \tag{28}$$

Table 1
Regular problem (27) and (28): the computational errors in the broken H^1-seminorm with their estimates, the effectivity indexes and the relative computational times (in percents) of the higher order reconstructions.

h	p	$\lvert e_h\rvert_{H^1(\Omega,\mathscr{T}_h)}$	EOC	$\lvert \mathscr{E}_h\rvert_{H^1(\Omega,\mathscr{T}_h)}$	EOC	i^{eff}	%cpu
1/8	1	5.02E−01	–	3.50E−01	–	0.70	1.3
1/16	1	2.68E−01	0.90	2.23E−01	0.65	0.83	0.9
1/32	1	1.37E−01	0.97	1.27E−01	0.81	0.93	0.6
1/64	1	6.89E−02	0.99	6.82E−02	0.90	0.99	0.6
1/8	2	9.41E−02	–	7.45E−02	–	0.79	2.9
1/16	2	2.62E−02	1.85	2.28E−02	1.71	0.87	1.9
1/32	2	6.76E−03	1.95	6.30E−03	1.86	0.93	0.9
1/64	2	1.71E−03	1.99	1.66E−03	1.92	0.97	0.9
1/8	3	1.55E−02	–	1.44E−02	–	0.93	4.8
1/16	3	2.21E−03	2.82	2.11E−03	2.77	0.96	6.5
1/32	3	2.85E−04	2.95	2.75E−04	2.94	0.97	1.1
1/64	3	3.58E−05	2.99	3.50E−05	2.97	0.98	1.1
1/8	4	2.29E−03	–	2.55E−03	–	1.11	8.5
1/16	4	1.64E−04	3.81	1.75E−04	3.86	1.07	6.5
1/32	4	1.05E−05	3.95	1.07E−05	4.03	1.01	1.4
1/64	4	6.62E−07	3.99	6.57E−07	4.03	0.99	1.4
1/8	5	2.78E−04	–	4.21E−04	–	1.51	11.6
1/16	5	9.97E−06	4.80	1.37E−05	4.94	1.37	10.3
1/32	5	3.21E−07	4.96	3.75E−07	5.19	1.17	1.5
1/64	5	1.01E−08	5.00	1.06E−08	5.15	1.05	5.5
1/8	6	2.83E−05	–	5.81E−05	–	2.05	14.8
1/16	6	5.09E−07	5.80	9.03E−07	6.01	1.77	9.3
1/32	6	8.20E−09	5.96	1.13E−08	6.32	1.38	10.8
1/64	6	1.28E−10	6.00	1.45E−10	6.28	1.13	6.8

with $c_1 = -e^{-1/\varepsilon}$, $c_2 = -1 - c_1$. The solution contains two weak boundary layers along $x_1 = 0$ and $x_2 = 0$, whose width is proportional to ε.

Remark 3. Let us note that the source term g in (27) (and in the other problems in this paper) is not given by a simple formula. In practice, we have to differentiate the relation for the exact solution and then multiply and sum the appropriate first and second order derivatives. E.g., the source term g in (27) can be expressed by

$$g = -U_1 - U_2 - \varepsilon(U_{11} + U_{22}),$$

where

$$U_1 = -\left(c_2 + \frac{e^{-x_1/\varepsilon}}{\varepsilon}\right) d_2(x_2), \quad U_2 = -\left(c_2 + \frac{e^{-x_2/\varepsilon}}{\varepsilon}\right) d_1(x_1), \quad U_{11} = \frac{e^{-x_1/\varepsilon}}{\varepsilon^2} d_2(x_2), \quad U_{22} = \frac{e^{-x_2/\varepsilon}}{\varepsilon^2} d_1(x_1),$$

$$d_1(x_1) = (c_1 + c_2(1 - x_1) + e^{-x_1/\varepsilon}), \quad d_2(x_2) = (c_1 + c_2(1 - x_2) + e^{-x_2/\varepsilon}).$$

Finally, the Dirichlet boundary condition is evaluated from the exact solution by $u_D := u|_{\partial\Omega}$.

Table 1 shows the achieved results. We observe that the index i^{eff} is not far from 1 and we detect an asymptotic exactness, i.e., $i^{\text{eff}} \to 1$ for $h \to 0$ (except the P_6-approximation on the finest grid caused probably by the limits of the finite precision arithmetic). Furthermore, Fig. 3 shows the distribution of the computational error in the broken H^1-seminorm and its estimate in the computational domain for selected computations. We observe that $|e_h|_{H^1(K)}$ and $|\mathscr{E}_h|_{H^1(K)}$ have similar distribution for $K \in \mathscr{T}_h$. Therefore, $\mathscr{E}_h$ approximates e_h well also locally.

4.2. Singular problem

We consider the scalar nonlinear convection–diffusion equation

$$\begin{aligned} &-\nabla\cdot(\boldsymbol{K}(u)\nabla u) - \frac{\partial u^2}{\partial x_1} - \frac{\partial u^2}{\partial x_2} = g \qquad \text{in} \quad \Omega = (0,1)^2, \\ &u = u_D \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{29}$$

where $\boldsymbol{K}(u)$ is the nonsymmetric matrix given by

$$\boldsymbol{K}(u) = \varepsilon \begin{pmatrix} 2 + \arctan(u) & (2 - \arctan(u))/4 \\ 0 & (4 + \arctan(u))/2 \end{pmatrix}. \tag{30}$$

The parameter $\varepsilon > 0$ plays a role of an amount of diffusion and we put $\varepsilon = 10^{-3}$. We prescribe a Dirichlet boundary condition u_D on $\partial\Omega$ and set the source term g such that the exact solution is

$$u(x_1, x_2) = (x_1^2 + x_2^2)^{\alpha/2} x_1 x_2 (1 - x_1)(1 - x_2), \quad \alpha \in \mathbb{R}. \tag{31}$$

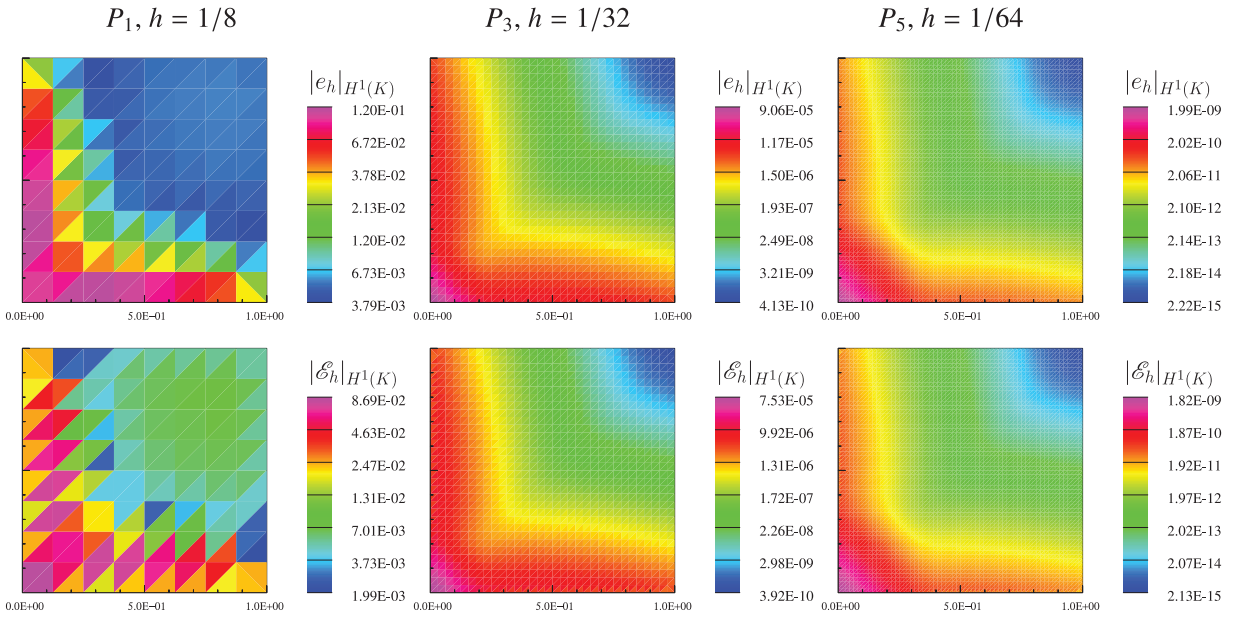


Fig. 3. Regular problem (27) and (28): local distribution of the computational error $|e_h|_{H^1(K)}, K \in \mathscr{T}_h$ (top) and its estimate $|\mathscr{E}_h|_{H^1(K)}, K \in \mathscr{T}_h$ (bottom) for the selected computations.

Table 2
Singular problem (29)–(31): the computational errors in the broken H^1-seminorm with their estimates, the effectivity indexes and the relative computational times (in percents) of the higher order reconstructions.

h	p	$\lvert e_h\rvert_{H^1(\Omega,\mathscr{T}_h)}$	EOC	$\lvert\mathscr{E}_h\rvert_{H^1(\Omega,\mathscr{T}_h)}$	EOC	i^{eff}	%cpu
1/8	1	3.93E−01	–	2.87E−01	–	0.73	0.8
1/16	1	2.82E−01	0.48	2.16E−01	0.41	0.76	1.0
1/32	1	2.00E−01	0.49	1.57E−01	0.46	0.78	0.8
1/64	1	1.42E−01	0.50	1.12E−01	0.48	0.79	0.6
1/8	2	1.86E−01	–	2.38E−01	–	1.28	1.8
1/16	2	1.32E−01	0.50	1.73E−01	0.46	1.31	2.3
1/32	2	9.38E−02	0.49	1.24E−01	0.48	1.32	1.3
1/64	2	6.65E−02	0.50	8.85E−02	0.49	1.33	0.8
1/8	3	1.68E−01	–	2.24E−01	–	1.33	2.4
1/16	3	1.20E−01	0.49	1.62E−01	0.47	1.36	2.2
1/32	3	8.50E−02	0.49	1.16E−01	0.48	1.37	1.6
1/64	3	6.04E−02	0.49	8.29E−02	0.49	1.37	1.0
1/8	4	1.29E−01	–	2.08E−01	–	1.61	4.5
1/16	4	9.24E−02	0.49	1.50E−01	0.47	1.63	3.5
1/32	4	6.58E−02	0.49	1.08E−01	0.48	1.64	2.5
1/64	4	4.67E−02	0.49	7.69E−02	0.49	1.65	1.4
1/8	5	1.23E−01	–	1.96E−01	–	1.59	5.8
1/16	5	8.77E−02	0.49	1.41E−01	0.47	1.61	4.9
1/32	5	6.25E−02	0.49	1.01E−01	0.48	1.62	2.8
1/64	5	4.44E−02	0.49	7.22E−02	0.49	1.62	1.6
1/8	6	1.27E−01	–	1.95E−01	–	1.53	7.2
1/16	6	9.10E−02	0.48	1.41E−01	0.47	1.55	5.3
1/32	6	6.49E−02	0.49	1.01E−01	0.48	1.56	3.6
1/64	6	4.62E−02	0.49	7.20E−02	0.49	1.56	2.6

We put $\alpha = -3/2$. It is possible to show (see [27]) that $u \in H^\kappa(\Omega), \kappa \in (0, 3/2)$. where $H^\kappa(\Omega)$ denotes the Sobolev–Slobodetskii space of functions with "non-integer derivatives".

Table 2 shows the achieved results. We observe reasonable values of the index i^{eff} close to 1. They indicate h-independence and a weak p-dependence. Moreover, Fig. 4 shows the distribution of the computational error in the H^1-seminorm and its estimate in the computational domain for selected computations. Again, the distributions of $|e_h|_{H^1(K)}$ and $|\mathscr{E}_h|_{H^1(K)}$ are similar for $K \in \mathscr{T}_h$. Some exception is the case P_6 and $h = 1/64$, when the error is many time underestimate in regions far from the singularity. However, the errors arising in these regions contribute negligibly to the total error.

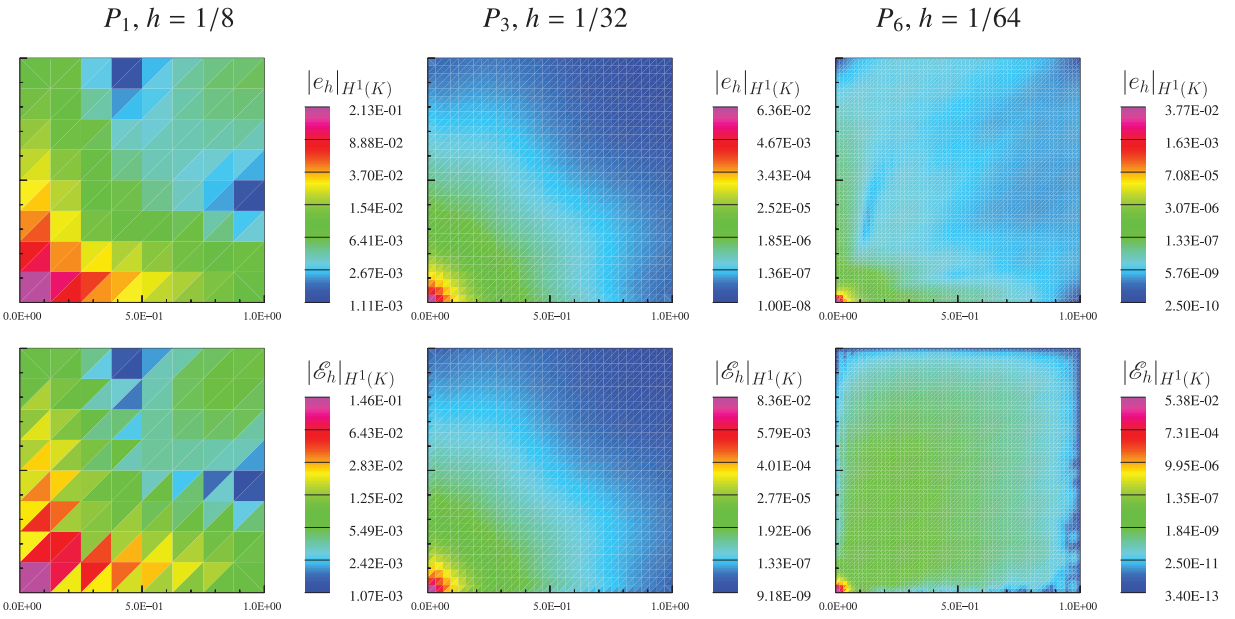


Fig. 4. Singular problem (29)–(31): local distribution of the computational error $|e_h|_{H^1(K)}, K \in \mathscr{T}_h$ (top) and its estimate $|\mathscr{E}_h|_{H^1(K)}, K \in \mathscr{T}_h$ (bottom) for the selected computations.

5. *hp*-adaptive strategy

5.1. Isotropic mesh adaptation

The ultimate goal of the computation is to *achieve* the prescribed *error tolerance* with the *smallest possible* number of *degrees of freedom* (DOF), cf. (5). More precisely, we need to minimize DOF of the *hp*-mesh $\mathscr{T}_h^{\mathsf{p}}$ such that

$$\eta \le \omega, \tag{32}$$

where η is a posteriori error estimate given by (23) (or any other estimator) and $\omega > 0$ is the given tolerance. In order to fulfill (32), we require

$$\eta_K \le \omega_K := \omega\sqrt{1/\#\mathscr{T}_h} \quad \forall K \in \mathscr{T}_h, \tag{33}$$

where η_K is given by (23) and $\#\mathscr{T}_h$ denotes the number of elements of $\mathscr{T}_h$. Obviously, due to (23),

$$\eta_K \le \omega_K \ \forall K \in \mathscr{T}_h \implies \eta \le \omega,$$

hence, the local condition (33) is stronger than the global one (32). However, the great advantage of (33) is the possibility to adapt the whole mesh at once (and not only the elements with the highest error estimates) which saves the computational time. Let us note that in order to achieve the faster convergence in practical computations, we set $\omega_K := C_F\omega\sqrt{1/\#\mathscr{T}_h}$ where $C_F = 1/2$ is the security factor.

Based on the higher-order reconstruction $\tilde{u}_h \in S_h^{\mathsf{p}+1}$ introduced in Section 3, we define the adaptive process which modifies the given *hp*-mesh and creates a new (better) one. Particularly, let h_K be the diameter of $K \in \mathscr{T}_h$ and p_K be the corresponding polynomial approximation degree. The idea is to set the better degree $p_K^{\text{new}} \in \{p_K - 1, p_K, p_K + 1\}$ and the better diameter $h_K^{\text{new}} > 0$ in such a way that a *local tolerance condition* (33) will be satisfied with the smallest possible number of degrees of freedom.

Finally, when p_K^{new} and h_K^{new} are set for each $K \in \mathscr{T}_h$, we employ the framework of the *isotropic mesh adaptation*, which is a simplification of technique developed in [28,29]. We define an isotropic Riemann metric (= Euclidean metric scaled by $1/h_K^{\text{new}}$). Then, with the aid of our in-house code ANGENER [30], we create a new triangulation $\mathscr{T}_h^{\text{new}}$ which is not defined by refining the original grid. Moreover, from p_K^{new}, $K \in \mathscr{T}_h$, we define piecewise constant function on $\mathscr{T}_h$, which is interpolated into the space of integer-valued piece-wise constant functions on $\mathscr{T}_h^{\text{new}}$.

5.2. Main idea of the hp-adaptation

In agreement with (21), $\tilde{u}_K = (\tilde{u}_h)|_K, K \in \mathscr{T}_h$ denotes the restriction of the higher order reconstruction to the mesh elements. We define the value

$$\vartheta_{K,0} := |\tilde{u}_K - u_h|_{H^1(K)}, \quad K \in \mathscr{T}_h. \tag{34}$$

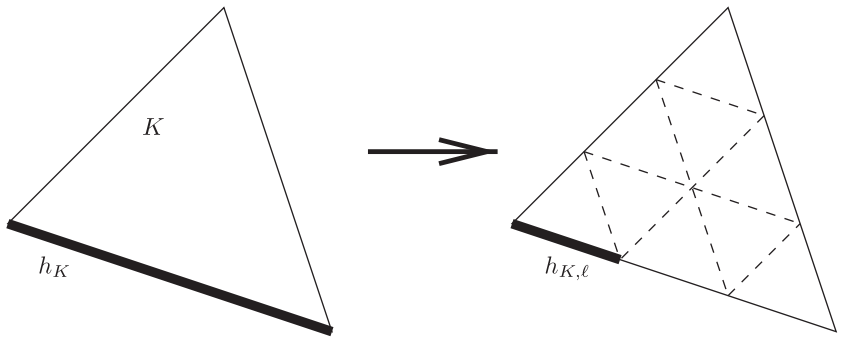


Fig. 5. Illustration of the element splitting for $q_\ell = 3$ in (41).

Let us note that due to (23), we have $\vartheta_{K,0} = \eta_K$. However, we use this duplication in the notation since η_K represents the estimation of the error (which can be replaced by any other indicator) whereas $\vartheta_{K,0}$ is employed for a decision about p-adaptation.

Obviously, the quantity $\vartheta_{K,0}$ approximate the discretization error achieved on K having the diameter h_K and the polynomial approximation degree p_K. Now we are interested in the following question:

(Q1) What happens if we locally decrease the polynomial approximation degree p_K to $p_K - \ell$, $\ell = 1, 2$?

It is natural to expect that the decrease of the polynomial approximation degree leads to an increase of the computational error as well as its estimate $\vartheta_{K,0}$. On the other hand, the number of degrees of freedom on K decreases. Therefore, in order to keep the local error estimate $\vartheta_{K,0}$, it is necessary to refine the element K into several *sub-elements*, see Fig. 5.

Hence, we replace question (Q1) by a new one which is more quantitative:

(Q2) Let $\vartheta_{K,0}$ be the estimate of the discretization error achieved on K with the polynomial approximation degree p_K. If we replace p_K by $p_K - \ell$, $\ell = 1, 2$, onto how many sub-elements we need to split K such that the corresponding error will be again $\vartheta_{K,0}$?

In order to find the answer to question (Q2), we define, similarly as in (6), the space of piecewise-polynomial functions

$$S_h^{\mathsf{p}-\ell} := \{v_h \in L^2(\Omega);\ v_h|_K \in P^{p_K-\ell}(K)\ \forall K \in \mathscr{T}_h\}, \quad \ell = 1, 2, \tag{35}$$

where $P^{p_K-\ell}(K)$ denotes the space of polynomials of degree $\le p_K - \ell$ on K (for simplicity we assume that $p_K - \ell \ge 0$). Obviously, $S_h^{\mathsf{p}-\ell} \subset S_h^{\mathsf{p}}$ for $\ell = 1, 2$. Moreover, we denote by $\Pi_h^{\mathsf{p}-\ell}$, $\ell = 1, 2$ the L^2-projection from S_h^{p} to $S_h^{\mathsf{p}-\ell}$, i.e.,

$$\Pi_h^{\mathsf{p}-\ell} u_h \in S_h^{\mathsf{p}-\ell}: \quad \left(\Pi_h^{\mathsf{p}-\ell} u_h, v_h\right)_{0,\Omega} = (u_h, v_h)_{0,\Omega} \quad \forall v_h \in S_h^{\mathsf{p}-\ell}, \quad \ell = 1, 2 \tag{36}$$

Let us note that if we consider a hierarchical basis of S_h^{p} then the projection (36) is straightforward, we simply remove the basis coefficients corresponding to the basis functions of degree greater than $p_K - \ell$ for each $K \in \mathscr{T}_h$.

Now, for $K \in \mathscr{T}_h$, we define two additional quantities

$$\vartheta_{K,\ell} := \left|\tilde{u}_K - \Pi_h^{\mathsf{p}-\ell} u_h\right|_{H^1(K)}, \quad \ell = 1, 2,\ K \in \mathscr{T}_h, \tag{37}$$

which approximate the local discretization errors in the H^1-seminorm for the used polynomial approximation degrees $p_K - \ell$ on K. Their evaluation is very simple since $\tilde{u}_K$ is already available due to (21).

Usually, the quantities $\vartheta_{K,\ell}$, $\ell = 0, 1, 2$ satisfy the inequalities

$$\vartheta_{K,0} \le \vartheta_{K,1} \le \vartheta_{K,2}, \quad K \in \mathscr{T}_h, \tag{38}$$

which corresponds to the fact that a higher degree polynomial approximation gives at most the same error as a lower degree one. In the following, we present the main idea of our algorithm for the sufficiently regular exact solution. Later we explain that the same technique works also for a non-regular solution.

5.2.1. Regular solution

Let us assume that the exact solution is sufficiently regular. Then, in agreement with (9), we expect that

$$\vartheta_{K,\ell} \approx C h_K^{p_K-\ell}, \quad \ell = 0, 1, 2, \tag{39}$$

where C is a unknown constant independent of h_K. In order to find the answer to question (Q2), we seek the *diameter* of *sub-elements* of K, denoted by $h_{K,\ell}$, such that the resulting error will by equal to $\vartheta_{K,0}$, i.e.,

$$\vartheta_{K,0} \approx C(h_{K,\ell})^{p_K-\ell}, \quad \ell = 0, 1, 2. \tag{40}$$

The above together with (39) gives the ratios

$$q_\ell := \frac{h_K}{h_{K,\ell}} = \left(\frac{\vartheta_{K,\ell}}{\vartheta_{K,0}}\right)^{\frac{1}{p_K-\ell}}, \quad \ell = 0, 1, 2. \tag{41}$$

Obviously, $q_0 = 1$. Therefore, (theoretically), if we split K onto q_ℓ^2 sub-triangles then the error estimate $\vartheta_{K,\ell}$ reduces to $\vartheta_{K,0}$, see Fig. 5 for an illustration. Generally, the ratios q_1 and q_2 are non-integer but this does not present any obstacle in the framework of the isotropic mesh adaptation.

If the element K is split onto q_ℓ^2 sub-elements and on each of them the polynomial approximation degree $p_K - \ell$ is used, then corresponding number of degrees of freedom is equal to

$$d_{K,\ell} = q_\ell^2 (p_K - \ell + 1)(p_K - \ell + 2)/2, \quad \ell = 0, 1, 2. \tag{42}$$

Therefore, we have three candidates, $\ell = 0, 1, 2$, which achieve theoretically the level $\vartheta_{K,0}$ with $d_{K,\ell}$ degrees of freedom. Hence, we choose this one which has minimal value $d_{K,\ell}$, i.e., we put

$$\bar{\ell} = \arg \min_{\ell \in \{0,1,2\}} d_{K,\ell}. \tag{43}$$

Let us discuss the values $\bar{\ell} = 0$ and $\bar{\ell} = 2$.

- Let $\bar{\ell} = 0$, i.e. the highest order tested approximation is the best. Hence, it would be useful to try to increase the polynomial approximation degree p_K.
- Let $\bar{\ell} = 2$, i.e. the lowest order tested approximation is the best. Therefore we should decrease the polynomial approximation degree p_K.

Based on this argumentation, we propose the following choice of p_K^{new}

$$\begin{aligned} &\bar{\ell} = 0 \Rightarrow p_K^{\text{new}} := p_K + 1, \\ &\bar{\ell} = 1 \quad \Rightarrow p_K^{\text{new}} := p_K, \\ &\bar{\ell} = 2 \Rightarrow p_K^{\text{new}} := p_K - 1. \end{aligned} \tag{44}$$

Remark 4. The setting (44) disagrees somewhat with the contents of question (Q2). We performed experiments with the higher order reconstruction of degree $p_K + 2$ which should better suits to the setting (44) but this strategy did not work properly. We do not claim that the setting (44) is optimal but the experiments presented hereafter demonstrate its potential.

5.2.2. Non-regular solution

Now, we consider the case when the exact solution is not regular, particularly when the local polynomial approximation degree is greater than the local Sobolev regularity. Let $u|_K \in H^{s_K}(K)$ such that $s_K < p_K$. In this case, instead of (38), the quantities $\vartheta_{K,\ell}, \ell = 0, 1, 2$ satisfy the relations

$$\vartheta_{K,0} \approx \vartheta_{K,1} \approx \vartheta_{K,2}, \tag{45}$$

which is in agreement with (9) and it corresponds to the observation than an increase of the polynomial approximation degree does not bring any essential increase of the accuracy when the solution is not regular. Therefore, relations (41) and (45) imply $q_1 \approx 1$ and $q_2 \approx 1$. Consequently, relation (42) gives $d_{K,2} < d_{K,1} < d_{K,0}$ and according (44), the algorithm automatically decreases p_K.

5.3. hp-adaptive algorithm

Based on the consideration formulated in Section 5.2, we formulate the *hp*-adaptive algorithm. Let $\omega > 0$ be the prescribed tolerance and ω_K be the local error tolerance defined by (33). The presented algorithm is based on the following two steps:

(S1) according (44), we set the better polynomial degree,
(S2) we propose new (more optimal) size of K according

$$h_K^{\text{new}} = h_K/\rho, \quad \text{where} \quad \rho := (\eta_K/\omega_K)^{1/p_K}. \tag{46}$$

The formulas (46) follow from relations $\eta_K \approx C h_K^{p_K}$ and $\omega_K \approx C(h_K^{\text{new}})^{p_K}$, cf. (39). In order to avoid a strong refinement in one level of mesh adaptation, we restrict $\rho \le 5$. Furthermore, the algorithm contains two branches depending if the local tolerance condition (33) is satisfied or not. If (33) is violated and if we increase p_K according (44) then we do not change element size according to (46) since then we avoid a needless h-refinement at the beginning of mesh adaptations. This setting follows from the observation that the computation starts usually from a very coarse grid and a low polynomial approximation degree. Then a common h- and p-refinement at the first iterative loops leads to a too fine h-refinement (and still relatively low approximation degrees) which is gradually de-refined when polynomial approximation degrees are increased. This leads to (almost) the same *hp*-grid but it requires more iterative loops.

Hence, we define the following algorithm:

Algorithm 1 setting of h_K^{new} and p_K^{new} for $K \in \mathscr{T}_h$.

let $u_h \in S_h^{\text{p}}$ be given
for all $K \in \mathscr{T}_h$ **do**
  let $h_K = \text{diam}(K)$ and p_K be the corresponding polynomial approximation degree,
  compute the higher-order reconstruction $\tilde{u}_K$ by (16) and (18),
  set η_K by (23) (or by another error estimate),
  set ω_K by (33),
  set $\vartheta_{K,\ell}, \quad \ell = 0, 1, 2$ by (37),
  set $d_{K,\ell}, \quad \ell = 0, 1, 2$ by (42),
  set $\bar{\ell}$ for which $d_{K,\ell}$ is the minimal
  if $\eta_K > \omega_K$ **then**
    if $\bar{\ell} = 0$ **then**
      $p_K^{\text{new}} := p_K + 1, \quad \rho := 1$
    else if $\bar{\ell} = 1$ **then**
      $p_K^{\text{new}} := p_K\,, \qquad \rho := (\eta_K/\omega_K)^{1/p_K}$
    else if $\bar{\ell} = 2$ **then**
      $p_K^{\text{new}} := p_K - 1, \qquad \rho := (\eta_K/\omega_K)^{1/(p_K-1)}$
    end if
  else
    if $\bar{\ell} = 0$ **then**
      $p_K^{\text{new}} := p_K + 1\,, \quad \rho := (\eta_K/\omega_K)^{1/(p_K+1)}$
    else if $\bar{\ell} = 1$ **then**
      $p_K^{\text{new}} := p_K, \quad \rho := (\eta_K/\omega_K)^{1/p_K}$
    else if $\bar{\ell} = 2$ **then**
      $p_K^{\text{new}} := p_K - 1, \quad \rho := (\eta_K/\omega_K)^{1/(p_K-1)}$
    end if
  end if
  $h_K^{\text{new}} := \frac{h_K}{\min(\rho,5)}$
end for

this algorithm works locally for each $K \in \mathscr{T}_h$, hence it can be simply parallelized. However, even for one computer core, the total computational time of the mesh adaptation including the higher-order reconstruction and the creation of the new *hp*-grid is shorter than the computational time necessary for assembling and solving the nonlinear algebraic system (8). Further advantage of the presented algorithm is that it does not contain any empirical constant.

The use of Algorithm 1 is obvious: we start with an initial *hp*-mesh, solve the approximate problem (8) and by Algorithm 1 we generate a new *hp*-mesh where we solve problem (8) again. The combination of the problem solution and mesh adaptation is repeated until the condition (32) is achieved.

6. Numerical examples

In this section we demonstrate the computational performance of the proposed *hp*-adaptive algorithm. We present three examples which are (except the first one) modifications of problems from [31] where a collection of 2D elliptic problems for testing adaptive grid refinement algorithms was published. Our aim is to demonstrate the exponential rate of the convergence of the error with respect to DOF and also to show a reasonable ability of the error estimator (23) to approximate the error. Moreover, the selection of the examples (Laplace problem, linear convection–diffusion equations, quasilinear elliptic equation) indicate the robustness of this approach.

For each case, we present the convergence of the adaptive process, namely the values of triangles $\#\mathscr{T}_h$ of the generated grids, the corresponding DOF given by (5), the computational errors in the broken H^1-seminorm, their estimates $|\mathscr{E}_h|_{H^1(\Omega,\mathscr{T}_h)}$ and the effectivity index i^{eff} given by (24).

Since the meshes are locally refined, we investigate the *experimental order of convergence* (EOC) with respect to DOF, i.e., instead of (25), we expect that the error behaves according to the formula

$$|e_h| \approx c\,\text{DOF}^{-\text{EOC}}, \tag{47}$$

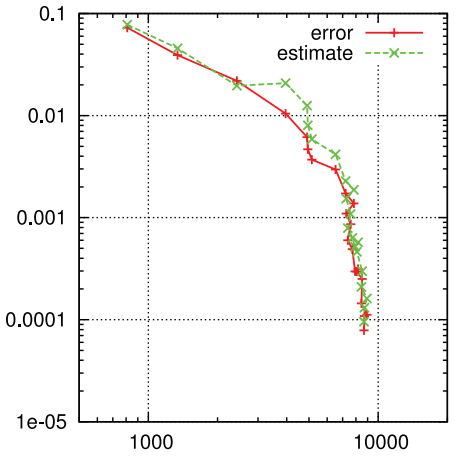
where $c > 0$ is a constant. Then, similarly as in (26), we derive

$$\text{EOC} = -\frac{\log(|e_{h_1}|/|e_{h_2}|)}{\log(\text{DOF}_1/\text{DOF}_2)}, \tag{48}$$

where $|e_{h_1}|$ and $|e_{h_2}|$ are computational errors obtained on two different *hp*-meshes $\mathscr{T}_{h_1}$ and $\mathscr{T}_{h_2}$ having DOF_1 and DOF_2 degrees of freedom, respectively. In some situation, the EOC may be negative, namely when the algorithm decreases DOF as well es the error and its estimates. This is in fact the advantage of our approach.

Table 3
Case 1, *hp*-adaptive computation: the errors in the broken H^1-seminorm, their estimates and corresponding EOC and i^{eff}, numerical values (left) and the corresponding plot (right).

lev	$\#\mathcal{T}_h$	DOF	$\lvert e_h\rvert_{H^1(\Omega,\mathcal{T}_h)}$	EOC	$\lvert\mathscr{E}_h\rvert_{H^1(\Omega,\mathcal{T}_h)}$	EOC	i^{eff}	%cpu
0	135	810	7.26E-02	–	7.78E-02	–	1.07	3.2
1	134	1340	3.91E-02	1.23	4.56E-02	1.06	1.17	4.4
2	165	2430	2.19E-02	0.97	1.96E-02	1.42	0.89	6.0
3	197	3957	1.05E-02	1.51	2.08E-02	-0.12	1.98	7.5
4	192	4907	6.15E-03	2.48	1.25E-02	2.36	2.04	8.9
5	166	4939	4.67E-03	42.17	8.04E-03	67.95	1.72	10.8
6	184	5148	3.70E-03	5.62	5.87E-03	7.61	1.58	11.4
7	235	6525	2.96E-03	0.94	4.17E-03	1.44	1.41	11.2
8	250	7224	1.73E-03	5.30	2.28E-03	5.92	1.32	10.8
9	263	7846	1.38E-03	2.73	1.87E-03	2.39	1.36	10.7
10	242	7260	1.10E-03	-2.91	1.54E-03	-2.56	1.40	10.6
11	258	7595	8.62E-04	5.38	1.09E-03	7.70	1.26	10.4
12	240	7381	6.01E-04	-12.63	7.89E-04	-11.16	1.31	10.4
13	254	7717	4.91E-04	4.55	6.33E-04	4.93	1.29	10.3
14	259	7936	2.97E-04	17.98	5.19E-04	7.09	1.75	10.2
15	260	8172	3.14E-04	-1.97	5.74E-04	-3.43	1.83	10.2
16	257	8084	2.96E-04	-5.59	4.63E-04	-19.83	1.57	10.2
17	280	8515	2.51E-04	3.16	2.99E-04	8.43	1.19	10.2
18	280	8455	1.44E-04	-78.25	2.10E-04	-50.23	1.45	10.2
19	295	8936	1.12E-04	4.61	1.62E-04	4.71	1.44	10.1
20	285	8709	1.08E-04	-1.22	1.31E-04	-8.12	1.21	10.1
21	285	8672	7.86E-05	-75.76	9.55E-05	-74.34	1.22	10.1



6.1. Case 1: re-entrant corner singularity

We consider the Poisson problem

$$-\Delta u = 0 \quad \text{in } \Omega = (-1,1)^2 \setminus \{(x_1,x_2);\ -2x_1 < x_2 < 0\}, \qquad u = u_D \quad \text{on } \partial\Omega, \tag{49}$$

where u_D is chosen such that the exact solution is

$$u(r,\varphi) = r^{2/3}\sin(2\varphi/3). \tag{50}$$

Here (r, φ) are the polar coordinates. This problems has a corner singularity at the origin.

We applied Algorithm 1 with $\omega = 10^{-4}$. Table 3 shows the convergence of the adaptive process. We observe that Algorithm 1 gives exponential order of convergence, which that the decrease of the error is faster than any linear decrease in logarithmic scale, see the corresponding figure of data from Table 3. Moreover, the effectivity index i^{eff} is very close to 1 which means that the used "naive" error estimator η defined by (23) approximate error very well. Further, the last column shows that the computational time of the higher order reconstruction is approximately 10% of the total computational time.

Furthermore, Fig. 6 shows the *hp*-grids with several details for selected levels of adaptations. We observed a strong *h*-refinement in a small neighborhood of the corner, which is necessary for a decrease of the error under the given tolerance.

6.2. Case 2: wave front

We consider the convection–diffusion equation

$$-2\varepsilon\frac{\partial^2 u}{\partial x_1^2} - \frac{\varepsilon}{2}\frac{\partial^2 u}{\partial x_2^2} - x_2\frac{\partial u}{\partial x_1} + x_1\frac{\partial u}{\partial x_2} = g \quad \text{in } \Omega = (-1,1)^2, \tag{51}$$

$$u = u_D \quad \text{on } \partial\Omega, \tag{52}$$

where $\varepsilon = 10^{-2}$, the right-hand side g and the boundary condition u_D are prescribed such that the exact solution is

$$u(x_1,x_2) = \tan^{-1}(m(r - r_0)), \tag{53}$$

where $r = ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{1/2}$, $m = 50$ and $r_0 = 0.25$. The solution exhibits an asymmetric circular steep wave front in the interior of the domain, see [31]. Due to $\tan^{-1}$ function, there is also a mild singularity at the center $(\frac{1}{2},\frac{1}{2})$ of the circle.

We applied Algorithm 1 with $\omega = 10^{-2}$. Table 4 shows the exponential convergence of the adaptive process, the EOC is increasing with respect to levels of adaptation. On the other hand, the effectivity index i^{eff} is approximately 2, this is caused

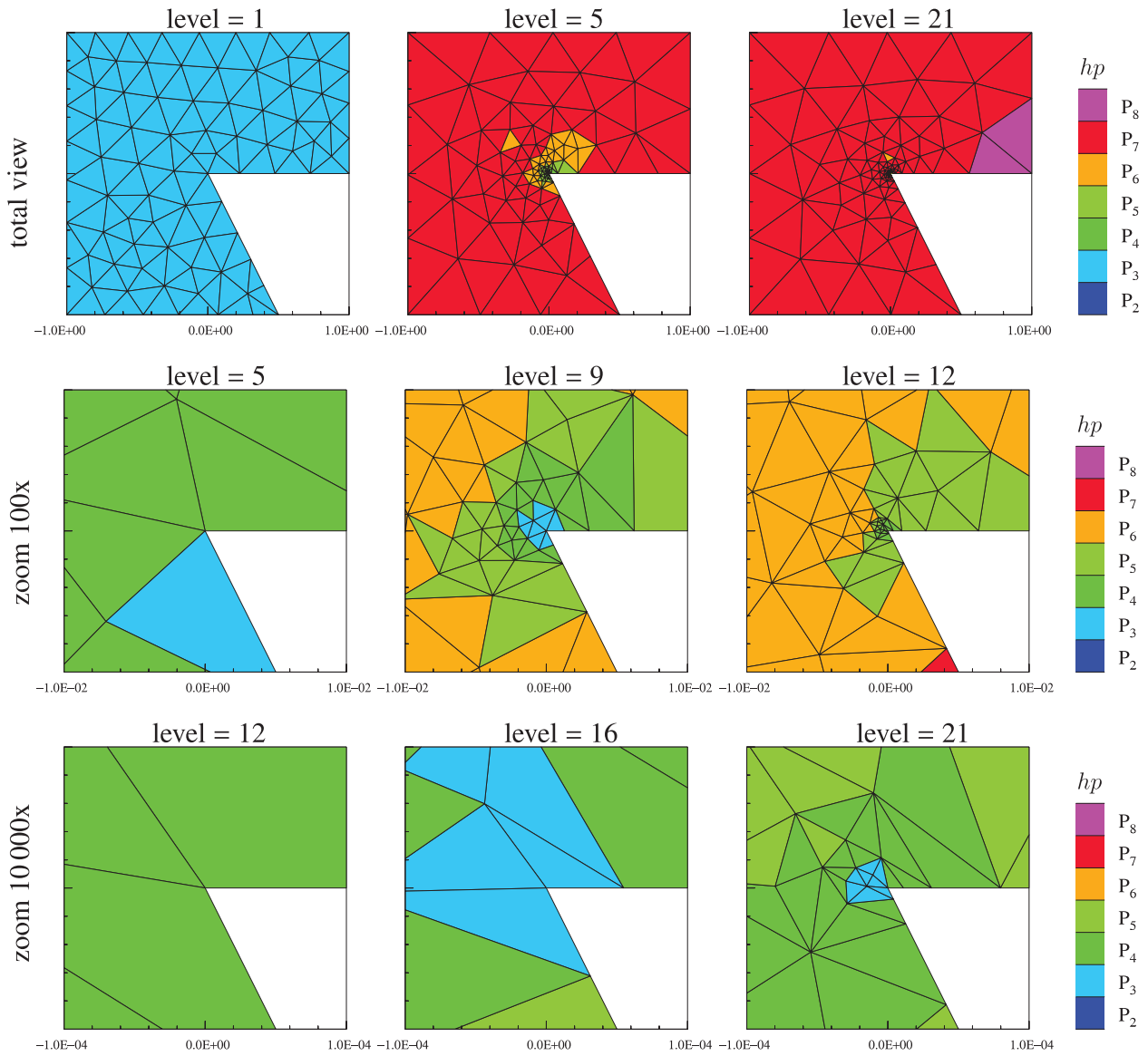
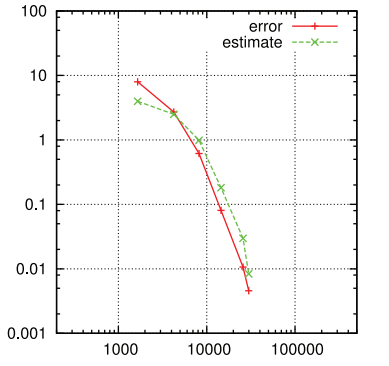


Fig. 6. Case 1: *hp*-grids with details around origin for selected levels of adaptations.

Table 4
Case 2, *hp*-adaptive computation: the errors in the broken H^1-seminorm, their estimates and corresponding EOC and i^{eff}, numerical values (left) and the corresponding plot (right).

lev	$\#\mathscr{T}_h$	DOF	$\lvert e_h\rvert_{H^1(\Omega,\mathscr{T}_h)}$	EOC	$\lvert \mathscr{E}_h\rvert_{H^1(\Omega,\mathscr{T}_h)}$	EOC	i^{eff}	%cpu
0	276	1656	7.98E+00	–	3.97E+00	–	0.50	2.7
1	457	4242	2.71E+00	1.15	2.50E+00	0.49	0.92	2.9
2	655	8185	6.16E-01	2.25	9.88E-01	1.41	1.60	3.2
3	919	14524	8.08E-02	3.54	1.81E-01	2.96	2.24	3.9
4	1269	25765	1.06E-02	3.54	2.98E-02	3.15	2.80	5.2
5	1218	29922	4.56E-03	5.67	8.34E-03	8.51	1.83	6.1



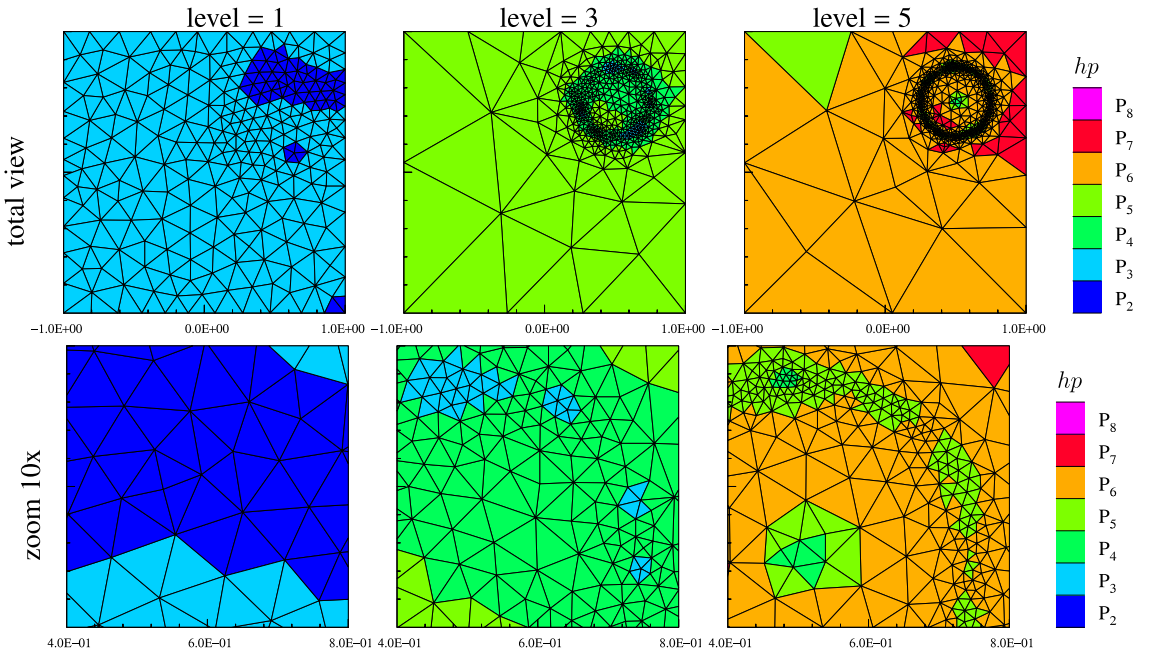


Fig. 7. Case 2: *hp*-grids with details containing the origin singularity and the part of the wave front for selected levels of adaptations.

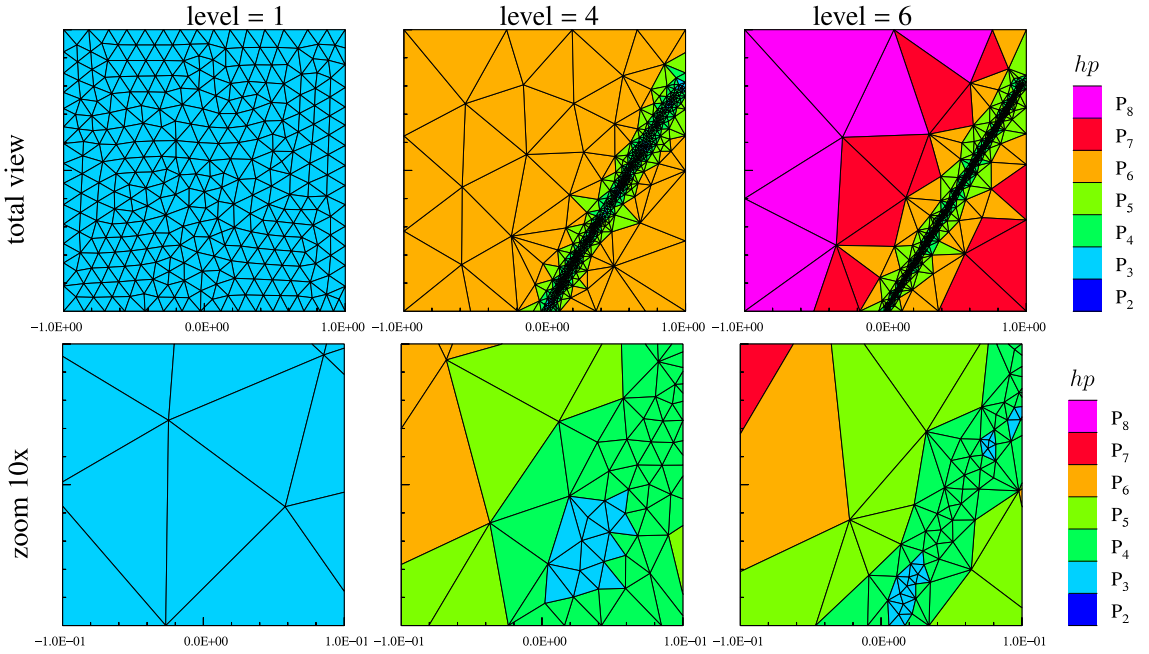


Fig. 8. Case 3: *hp*-grids with details containing detail along the interior layer for selected levels of adaptations.

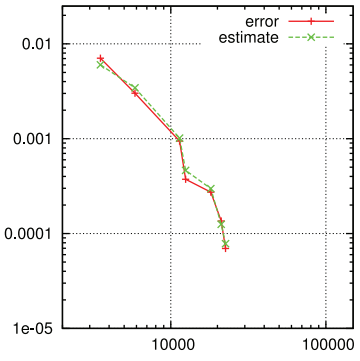
probably by the fact that patches $\mathscr{D}_K, K \in \mathscr{T}_h$ are larger in comparison with the width of the interior layer and then the higher-order reconstruction is smeared.

Further, the last column in Table 4 shows that the computational time of the higher order reconstruction is below 10% of the total computational time. This value is slightly increasing for the decreasing error. This is caused by the fact that the numerical solution on the last levels of adaptations are relatively fast since we have a good initial approximation from previous levels of adaptations. On the other hand, the reconstruction is computed directly without any previous information.

Moreover, Fig. 7 shows the *hp*-grids with several details for selected levels of adaptations. We observe a strong *h*-refinement along the steep wave front and also around the singularity at the center of the circle. On the other hand, the effectivity index i_{H^1} is slightly increasing.

Table 5
Case 3, *hp*-adaptive computation: the errors in the broken H^1-seminorm, their estimates and corresponding EOC and i^{eff}, numerical values (left) and the corresponding plot (right).

lev	$\#\mathcal{T}_h$	DOF	$\lvert e_h\rvert_{H^1(\Omega,\mathcal{T}_h)}$	EOC	$\lvert\mathscr{E}_h\rvert_{H^1(\Omega,\mathcal{T}_h)}$	EOC	i^{eff}	%cpu
0	586	3516	7.06E-03	–	6.01E-03	–	0.85	4.5
1	587	5870	3.02E-03	1.66	3.44E-03	1.09	1.14	4.7
2	760	11395	9.43E-04	1.75	1.02E-03	1.84	1.08	5.8
3	770	12462	3.73E-04	10.36	4.63E-04	8.76	1.24	6.1
4	1155	18125	2.73E-04	0.83	2.98E-04	1.18	1.09	8.0
5	1421	21128	1.36E-04	4.56	1.25E-04	5.66	0.92	7.8
6	1515	22547	6.97E-05	10.26	7.81E-05	7.24	1.12	8.7



6.3. Case 3: interior line singularity

We consider the scalar nonlinear diffusion equation

$$\begin{aligned} -\nabla\cdot(\boldsymbol{K}(u)\nabla u) &= g \quad \text{in } \Omega=(-1,1)^2,\\ u &= u_D \quad \text{on } \partial\Omega, \end{aligned} \tag{54}$$

where $\boldsymbol{K}(u)$ is given by (30) with $\varepsilon=1$. We prescribe a Dirichlet boundary condition u_D on $\partial\Omega$ and set the source term g such that the exact solution is

$$u(x_1,x_2)=\begin{cases}\cos(\pi x_2/2) & \text{for } x_1\le\beta(x_2-1),\\ \cos(\pi x_2/2)+(x_1-\beta(x_2-1))^\alpha & \text{for } x_1>\beta(x_2-1),\end{cases} \tag{55}$$

where we put $\alpha=2$ and $\beta=0.6$. The solution satisfies $u\in H^{\alpha+1/2-\epsilon}(\Omega)$ and possess a weak singularity along the line $x_1-\beta(x_2-1)=0$. This line singularity is difficult to capture since it is very weak. However, without a sufficient refinement along this line, it is not possible to decrease the computational error under the given tolerance.

We applied Algorithm 1 with $\omega=10^{-4}$. Table 5 shows the convergence of the adaptive process, however, it is not exponential but the given tolerance is achieved within few adaptive levels. On the other hand, the effectivity index i^{eff} indicates an accurate estimation of the error. Further, the last column shows that the computational time of the higher order reconstruction is below 10% of the total computational time.

Moreover, Fig. 8 shows the *hp*-grids with several details for selected levels of adaptations. We observe a *h*-refinement along the line singularity, which is spread only to few elements in the direction perpendicular to this line.

7. Conclusion

We presented a new *hp*-adaptive technique for the numerical solution of boundary value problems. The method is based on higher-order reconstruction over local element patches which makes it faster and easy to parallelize. The presented algorithm is free of user-defined parameters, it can be generalized for any numerical method using polynomial approximation, and combined with any a posteriori error estimator. Several numerical experiments were performed to demonstrate the outstanding performance and robustness of the presented method.

References

[1] W. Gui, I. Babuška, The *h*, *p* and *h-p* versions of the finite element method in 1 dimension. III. The adaptive *h-p* version., Numer. Math. 49 (1986) 659–683.
[2] I. Babuška, T. Strouboulis, K. Copps, *hp* optimization of finite element approximations: Analysis of the optimal mesh sequences in one dimension., Comput. Methods Appl. Mech. Eng. 150 (1–4) (1997) 89–108.
[3] I. Babuška, T. Strouboulis, The Finite Element Methods and its Reliability., Clarendon Press, Oxford, 2001.
[4] C. Schwab, *p*- and *hp*-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics, in: Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1998.
[5] L.F. Demkowicz, Computing with *hp*-Adaptive Finite Elements. Volume 1: One- and Two-Dimensional Elliptic and Maxwell Problems. With CD-ROM., Applied Mathematics and Nonlinear Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2007.
[6] P. Šolín, K. Segeth, I. Doležel, Higher-Order Finite Element Methods, Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, FL, 2004.
[7] T. Eibner, J. Melenk, An adaptive strategy for *hp*-FEM based on testing for analyticity., Comput. Mech. 39 (5) (2007) 575–595.
[8] W. Dörfler, V. Heuveline, Convergence of an adaptive *hp* finite element strategy in one space dimension., Appl. Numer. Math. 57 (10) (2007) 1108–1124.
[9] T.P. Wihler, An *hp*-adaptive strategy based on continuous Sobolev embeddings., J. Comput. Appl. Math. 235 (8) (2011) 2731–2739.
[10] L.F. Demkowicz, J. Kurtz, D. Pardo, M. Paszyński, W. Rachowicz, A. Zdunek, Computing with *hp*-Adaptive Finite Elements. Volume II: Frontiers: Three-Dimensional Elliptic and Maxwell Problems with Applications., Applied Mathematics and Nonlinear Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2008.

[11] W.F. Mitchell, M.A. McClain, A comparison of *hp*-adaptive strategies for elliptic partial differential equations, ACM Trans. Math. Softw. 41 (1) (2014) 39. Article no. 2
[12] A. Kufner, O. John, S. Fučík, Function Spaces, Function Spaces, Academia, Prague, 1977.
[13] P.G. Ciarlet, The Finite Elements Method for Elliptic Problems, North-Holland, Amsterdam, New York, Oxford, 1979.
[14] V. Dolejší, M. Feistauer, V. Sobotíková, Analysis of the discontinuous Galerkin method for nonlinear convection–diffusion problems, Comput. Methods Appl. Mech. Eng. 194 (2005) 2709–2733.
[15] V. Kučera, Optimal $L^\infty(L^2)$-error estimates for the DG method applied to nonlinear convection–diffusion problems with nonlinear diffusion, Numer. Funct. Anal. Optim. 31 (3) (2010) 285–312.
[16] M. Feistauer, J. Felcman, I. Straškraba, Mathematical and Computational Methods for Compressible Flow, Clarendon Press, Oxford, 2003.
[17] V. Dolejší, *hp*-DGFEM for nonlinear convection–diffusion problems, Math. Comput. Simul. 87 (2013) 87–118.
[18] V. Dolejší, Analysis and application of IIPG method to quasilinear nonstationary convection–diffusion problems, J. Comput. Appl. Math. 222 (2008) 251–273.
[19] P. Houston, J. Robson, E. Süli, Discontinuous Galerkin finite element approximation of quasilinear elliptic boundary value problems I: The scalar case, IMA J. Numer. Anal. 25 (2005) 726–749.
[20] P. Šolín, L. Demkowicz, Goal-oriented *hp*-adaptivity for elliptic problems, Comput. Methods Appl. Mech. Eng. 193 (2004) 449–468.
[21] L. Demkowicz, W. Rachowicz, P. Devloo, A fully automatic *hp*-adaptivity, J. Sci. Comput. 17 (1–4) (2002) 117–142.
[22] P. Kus, P. Solin, D. Andrs, Arbitrary-level hanging nodes for adaptive *hp*-FEM approximations in 3D, J. Comput. Appl. Math. 270 (2014) 121–133.
[23] R. Verfürth, A Posteriori Error Estimation Techniques for Finite Element Methods, Numerical Mathematics and Scientific Computation, Oxford University Press, 2013.
[24] M. Vohralík, A Posteriori Error Estimates, Stopping Criteria and Inexpensive Implementation, Université Pierre et Marie Curie–Paris 6, 2010 (Habilitation thesis).
[25] C. Clavero, J.L. Gracia, J.C. Jorge, A uniformly convergent alternating direction (HODIE) finite difference scheme for 2D time-dependent convection–diffusion problems, IMA J. Numer. Anal. 26 (2006) 155–172.
[26] V. Dolejší, H.G. Roos, BDF-FEM for parabolic singularly perturbed problems with exponential layers on layer-adapted meshes in space, Neural Parallel Sci. Comput. 18 (2) (2010) 221–235.
[27] I. Babuška, M. Suri, The *p*- and *hp*- versions of the finite element method. An overview, Comput. Methods Appl. Mech. Eng. 80 (1990) 5–26.
[28] V. Dolejší, Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes, Comput. Vis. Sci. 1 (3) (1998) 165–178.
[29] V. Dolejší, Anisotropic mesh adaptation technique for viscous flow simulation, East-West J. Numer. Math. 9 (1) (2001) 1–24.
[30] V. Dolejší, ANGENER – software package, Charles University Prague, Faculty of Mathematics and Physics, 2000. www.karlin.mff.cuni.cz/dolejsi/angen.html.
[31] W.F. Mitchell, A collection of 2D elliptic problems for testing adaptive grid refinement algorithms, Appl. Math. Comput. 220 (2013) 350–364.
[32] V. Dolejší, M. Feistauer, Discontinuous Galerkin method – analysis and applications to compressible flow, Springer International Publishing Switzerland, 2015.
[33] S. Giani, D. Schötzau, L. Zhu, An a-posteriori error estimate for *hp*-adaptive DG methods for convection-diffusion problems on anisotropically refined meshes, Comput. Math. Appl. 67 (2014) 869–887.