Constrained approximation in hp-FEM: Unsymmetric subdivisions and multi-level hanging nodes

Andreas Schröder

Abstract In conform hp-finite element schemes on irregular meshes, one has to ensure the finite element functions to be continuous across edges and faces in the presence of hanging nodes. A key approach is to constrain the appropriate shape functions using so-called connectivity matrices. In this work the connectivity matrices for hierarchical tensor product shape functions are explicitly determined. In particular, the presented approach includes both unsymmetric subdivsions and multlilevel hanging nodes *not* using hierarchical or multi-level information of subdivions. Moreover, the problem of edge and face orientations is considered.

1 Introduction

In adaptive finite element schemes, local refinements are typically realized by subdivisions of mesh elements. Using conform finite element schemes, one has to ensure the finite element functions to be continuous across edges and faces. In the presence of hanging or irregular nodes, this is done through contraint of the local basis functions associated to them and to adjacent irregular edges and faces, which is known as constrained approximation. A natural approach is to use connectivity matrices in the assembly process. Let $\mathscr{T} := \{T_0, T_1, \ldots,\}$ be a mesh subordinate to $\Omega \subset \mathbb{R}^k$, $k \in \{2,3\}$, where $\overline{T}_i \cap \overline{T}_j$ is empty or a vertex, an edge or a face of T_i or T_j, $i \neq j$. Furthermore, let $\Psi_T : \hat{T} \to T \in \mathscr{T}$ be a bijective and sufficiently smooth mapping for some reference element $\hat{T}$, e.g., $\hat{T} := [-1,1]^k$ for quadrangles or hexahedrons, and let $\mathscr{P}_T$ be a finite polynomial space on $\hat{T}$. Thus, the space of piecewise continuous polynomials is defined as $\mathscr{S} := \{v \in C^0(\Omega) \mid \forall T \in \mathscr{T} : \; v_{|T} \circ \Psi_T \in \mathscr{P}_T\}$. We denote the global basis functions of $\mathscr{S}$ by $\{\varphi_i\}_{0 \leq i < n}$ and the local basis functions of $\mathscr{P}_T$ by $\{\eta_{T,i}\}_{0 \leq i < n_T}$. The matrices $\pi_T \in \mathbb{R}^{n \times n_T}$, $T \in \mathscr{T}$, connecting the

Andreas Schröder
Department of Mathematics, Humboldt-Universität zu Berlin, D-10099 Berlin, e-mail: andreas.schroeder@mathematik.hu-berlin.de

local and global basis functions are called *connectivity matrices* and are given by $\varphi_{i|T} = \sum_{j=0}^{n_T-1} \pi_{T,ij}\eta_{T,j} \circ \Psi_T^{-1}$. The assembly of the stiffness matrix K and the load vector F is thus given by $K := \sum_{T\in\mathscr{T}} \pi_T K_T \pi_T^\top$ and $F := \sum_{T\in\mathscr{T}} \pi_T F_T$ for the local stiffness matrices $K_T \in \mathbb{R}^{n_T\times n_T}$ and local load vectors $F_T \in \mathbb{R}^{n_T}$, $T \in \mathscr{T}$.
A fundamental problem in finite element implementations is to provide connectivity matrices through suitable data structures as their computation is highly dependent on the choice of shape functions and refinement patterns. Moreover, the edge and face orientations have to be taken into account. If mesh elements containing hanging nodes are subdivided, multi-level hanging nodes occur. This significantly complicates the computation of the connectivitiy matrices and, in particular, their implementation. Therefore, most finite element codes do not allow for more than one hanging node per edge or face.
In the literature, connectivity matrices, their calculation and several data structures are described. In [1], the constraints are stated for integrated Legendre shape functions on quadrangles. Also, the extension to multi-level *hp*-refinement is discussed. The constraints are inserted via data structures representing a sparse data format for connectivity matrices. In [3], some data structure arrays for quadrangles storing the constraint information are proposed which also describe connectivity matrices in sparse data format. Similar approaches are suggested in [2, 4, 7, 11, 12]. A broad overview on data structures and algorithms for constrained approximation in two and three dimensions is given in the comprehensive monographs by Demkowicz et. al. [5, 6].
The aim of this work is to compute the connectivity matrices for hierarchical tensor product shape functions including both unsymmetric subdivisions and multi-level hanging nodes. The basic idea is to consider an irregular face as a subset of a regular face regardless of whether it results from a multi-level, symmetric or unsymmetric subdivision and to compute the entries of the connectivity matrices from this information only. Hence, no hierarchical or multi-level information of the subdivisions is needed. This simplifies the implementation greatly. A further emphasis of this work is on edge and face orientations and on implementation aspects based on some simple data structures for the storage of mesh elements.

2 Tensor product shape functions of Legendre type

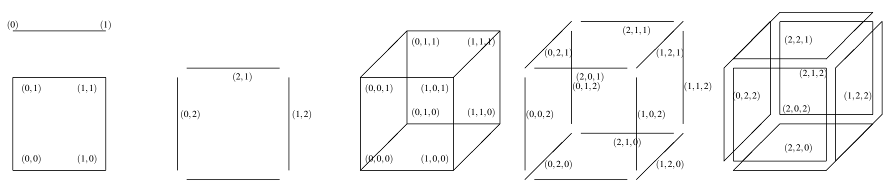


Fig. 1 Index tuple identifying nodes, edges and faces of the reference cube.

Tensor product shape functions based on integrated Legendre or Gauss-Lobatto polynomials are a widely used family of shape functions for higher-order FEM. Using Gegenbauer polynomials $\{G_i^\rho\}_{i\in\mathbb{N}_0}$ defined as $(i+1)G_{i+1}^\rho(x) = 2(i+\rho)xG_i^\rho(x) - (i+2\rho-1)G_{i-1}^\rho(x)$ with $\rho \in \mathbb{R}$, $G_0^\rho(x) := 1$ and $G_1^\rho(x) := 2\rho x$, we obtain integrated Legendre ($\beta_i := 1$) or Gauss-Lobatto ($\beta_i := \sqrt{(2i-1)/2}$) polynomials $\xi_0(x) := \frac{1}{2}(1-x)$, $\xi_1(x) := \frac{1}{2}(1+x)$, and $\xi_i(x) := \beta_i G_i^{-1/2}(x)$ for $i = 2,\dots,p$. Tensor product shape functions are constructed on the unit cube $[-1,1]^k$ via

$$\eta_\alpha(x) := \prod_{r=0}^{k-1} \xi_{\alpha_r}(x_r),\ x \in \mathbb{R}^k$$

for a k-tuple α with $\alpha_r \in \{0,\dots,p_r\}$, $0 \le r < k$ and local polynomial degrees $p_0,\dots,p_{k-1} \ge 1$, cf. [8, Ch.3]. Usually, the shape functions are separated into nodal, edge, face and inner modes. For this purpose, we associate a node, an edge or a face to a k-tuple with values in $\{0,1,2\}$ as shown in Figure 1 and the unit cube itself to the k-tuple $(2,\dots,2)$. In the following, let b be such a k-tuple. Typically, one also introduces additional local polynomial degrees for edges and faces, for instance, to ensure the minimum rule, cf. [11]. We denote these degrees by $p_r^b \in \{1,\dots,p_r\}$ for all $r = 0,\dots,k-1$ with $b_r = 2$. With these preparations at hand, the modes associated to b are $\{\eta_\alpha\}_{\alpha\in I^b}$ with

$$I^b := \{\alpha \mid \alpha_r := b_r \text{ if } b_r \in \{0,1\},\ \text{otherwise } \alpha_r \in \{2,\dots,p_r^b\}\}.$$

Also serendipity shape functions with reduced number of face and inner modes (cf. [8]) can be captured using this notation. Let q^b be a polynomial degree which is assigned to b and let ℓ be the dimension of the object associated to b. With $p_r^b := q^b - 2(\ell-1)$, the index set is given by $\tilde{I}^b := \{\alpha \in I^b \mid \sum_{r=0,b_r=2}^{k-1} \alpha_r \in \{2\ell,\dots,q^b\}\}$. In most finite element implementations, a mesh element $T \in \mathscr{T}$ is respresented by a special data structure which enables the storage of information like coordinates, polynomial degrees, global numbering or to generate some information about the combinatorial structure of the mesh element. A simple data structure is given by the representation of T through $G_T = (G_T^0,\dots,G_T^{k-1}) \in (\mathscr{G}_0)^{\sigma_0} \times \dots \times (\mathscr{G}_{k-1})^{\sigma_{k-1}}$ where $\sigma_\ell := \sigma_\ell^k := 2^{k-\ell} k!/(\ell!(k-\ell)!)$ denotes the number of ℓ-dimensional adjacents objects in a k-dimensional cube. The set $\mathscr{G}_0 \subset \mathbb{R}^k$ represents the set of all nodes of $\mathscr{T}$, $\mathscr{G}_\ell \subset (\mathscr{G}_0)^{2^\ell}$ of all edges or faces of $\mathscr{T}$, $0 \le \ell < k$, respectively. For completeness, we define $\mathscr{G}_k := \{G_T^0 \mid T \in \mathscr{T}\}$. A natural orientation of edges and faces is shown in Figure 2(a), which is equivalently given by the matrices

$$\mathscr{I}^{1,2} := \begin{pmatrix} 0 & 1 & 3 & 0 \\ 1 & 2 & 2 & 3 \end{pmatrix},\ \mathscr{I}^{1,3} := \begin{pmatrix} 0 & 1 & 3 & 0 & 0 & 1 & 2 & 3 & 4 & 5 & 7 & 4 \\ 1 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 5 & 6 & 6 & 7 \end{pmatrix},\ \mathscr{I}^{2,3} := \begin{pmatrix} 0 & 1 & 3 & 4 & 0 & 0 \\ 1 & 2 & 2 & 5 & 3 & 1 \\ 2 & 6 & 6 & 6 & 7 & 5 \\ 3 & 5 & 7 & 7 & 4 & 4 \end{pmatrix}.$$

Here, the entries of the j-th column denotes the node indices of the edge or face with index j. We assume that for all $1 \le \ell < k$ and $0 \le \nu < \sigma_\ell$, there exists a unique

$0 \le i < 2^\ell$ such that

$$(G_T^0)_{\mathscr{I}_{0,\nu}^{\ell,k}} = ((G_T^\ell)_\nu)_i. \tag{1}$$

We denote this index by $g(G_T,\ell,\nu)$. Furthermore, we assume that for all $1 \le \ell < k$ and $0 \le \nu < \sigma_\ell$ there exists a unique $\delta \in \{-1,1\}$ such that

$$(G_T^0)_{\mathscr{I}_{i,\nu}^{\ell,k}} = ((G_T^\ell)_\nu)_{(g(G_T,\ell,\nu)+\delta i) \bmod 2^\ell} \tag{2}$$

for all $0 \le i < 2^\ell$. Given $h(G_T,\ell,\nu) := \delta$, we obtain $g(G_T,1,4) = 1$, $h(G_T,1,4) = -1$, $g(G_T,2,5) = 2$ and $h(G_T,2,5) = -1$ in Figure 2(b). Conditions (1) and (2) ensure, that the edges and faces consist of the nodes given by G_T^0 and that they can be transfered to the reference edge or face by rotation or reflections, respectively.
The approximation space $\mathscr{S}$ is defined through a degree distribution which is given by the global polynomial degrees $p(G)_0,\ldots,p(G)_{\ell-1}$. Here, $G \in \mathscr{G}_\ell$, $1 \le \ell \le k$ represents a non-hanging or regular edge, face or a mesh element in $\mathscr{T}$. In the case that G represents a face, we associate $p(G)_0$ to the direction given by the nodes G_0 and G_1, and $p(G)_1$ to the direction given by G_1 and G_2. In the following, let $M(G,\beta)$ be a suitable global numbering where β is a ℓ-tuple with $\beta_r \in \{2,\ldots,p(G)_r\}$, $0 \le r < \ell$ which denotes the modes associated to G.
In the following, let $b(G)$ be the k-tuple associated to $G = (G_T^\ell)_\nu$ for some $0 \le \ell < k$, $0 \le \nu < \sigma_\ell$ or to $G = G_T^0$ with $\ell = k$. Furthermore, let $\alpha \in I^{b(G)}$. To construct continuous functions, we have to adjust the edge and face modes to the orientation of G given by the mappings g and h. This adjustment may be done switching the entries in α or using a sign number $\mu(\alpha)$. For this purpose, we specify the local polynomial degrees $p_r^{b(G)}$, the ℓ-tuple $\beta(\alpha)$ and the sign number $\mu(\alpha)$. In the case $\ell = 1$, we set $p_r^{b(G)} := p(G)_0$, $\beta(\alpha)_0 := \alpha_r$ and $\mu(\alpha) = h(G_T,1,\nu)^{\beta(\alpha)_0}$ for the unique $r \in \{0,\ldots,k-1\}$ with $b(G)_r = 2$. In the case $\ell = 2$, we have unique $r_0, r_1 \in \{0,\ldots,k-1\}$ with $b(G)_{r_0} = b(G)_{r_1} = 2$ and $r_0 < r_1$. Here, we distinguish four cases depending on the values of $f(G_T,\nu) := (g(G_T,2,\nu) + (h(G_T,2,\nu)-1)/2) \bmod 4 \in \{0,\ldots,3\}$. For $j = 0,1$, we define $p_{r_j}^{b(G)} := p(G)_j$, $\beta(\alpha)_j = \alpha_{r_j}$ if $f(G_T,\nu) \in \{0,2\}$, and $p_{r_{(j+1) \bmod 2}}^{b(G)} := p(G)_j$ and $\beta(\alpha)_{(j+1) \bmod 2} = \alpha_{r_j}$ otherwise. Furthermore, we set

$$\mu(\alpha) := (\lambda_0 h(G_T,2,\nu))^{\beta(\alpha)_0} \lambda_1^{\beta(\alpha)_1}$$

with $\lambda_i := 1$, $i = 0,1$, if $f(G_T,\nu) \in \{i,i+1\}$, and $\lambda_i := -1$ otherwise. For completeness, we define $p_j^{b(G)} := p(G)_j$, $j = 0,1,2$, and $\beta(\alpha) = \alpha$, if $\ell = 3$, and $\mu(\alpha) := 1$ if $\ell \in \{0,k\}$. Using all these preparations, the connectivity matrices are given by

$$\pi_{T,M(G,\beta(\alpha)),m_T(\alpha)} := \mu(\alpha) \tag{3}$$

where $m_T(\alpha)$ is a suitable local numbering, cf. [8, Ch.4.1.5.1]. All entries which are not captured by (3) are set to 0. Note that we implicitely assume that Ψ_T maps the vertices of the unit cube onto the nodes G_ν^0 in the same order as given in Figure 1. This is, e.g., done by $\Psi_T := \sum_{0 \le \nu < 2^k} \eta_{b((G_T^0)_\nu)} (G_T^0)_\nu$.

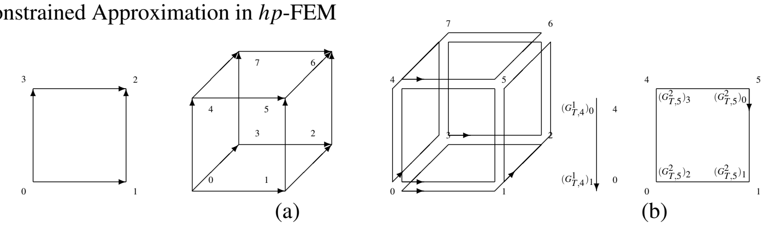


Fig. 2 **(a)** Edge and face orientations in the reference element, **(b)** non matching orientations.

3 Constraints coefficients and multi-level hanging nodes

To calculate the connectivity matrices for elements with irregular nodes, edges or faces, we introduce a further data structure $G_F = (G_F^0, \ldots, G_F^{k-2}) \in \mathscr{G}_{k-1} \times \mathscr{G}_1^{\tilde{\sigma}_1} \ldots \times \mathscr{G}_{k-2}^{\tilde{\sigma}_{k-2}}$, $\tilde{\sigma}_\ell := \sigma_\ell^{k-1}$, which respresents an edge $F \subset \mathbb{R}^2$ or a face $F \subset \mathbb{R}^3$ of $\mathscr{T}$ for $k = 2,3$, respectively. Based on G_F, we define $\tilde{b}(G)$ as the $k-1$-tuple with values in $\{0,1,2\}$ which is associated to the node or edge $G = (G_F^\ell)_\nu$ as depicted in Figure 1 or to $G = G_F^0$ with $\tilde{b}(G) := (2,\ldots,2)$. Furthermore, let $\hat{F} \subset \mathbb{R}^3$ be the unique regular edge or face of $\mathscr{T}$ with $F \subset \hat{F}$. We assume that there exists numbers $v_r, w_r \in \mathbb{R}$, $0 \le r < k-1$, such that

$$\Phi(\Psi_F^{-1}((G_{\hat{F}}^0)_\nu)) = \Psi_F^{-1}((G_F^0)_\nu), \tag{4}$$

for all $0 \le \nu < 2^\ell$ with $\Psi_F := \sum_{0 \le \nu < 2^{k-1}} \eta_{\tilde{b}((G_F^0)_\nu)} (G_F^0)_\nu$ and $\Phi(x)_r = v_r x_r + w_r$, $v_r \in (0,1]$. Note that Ψ_F maps $[-1,1]^{k-1}$ onto F and that Φ is a compression. Furthermore, we assume that

$$g(G_F,\ell,\nu) = g(G_{\hat{F}},\ell,\nu), \quad h(G_F,\ell,\nu) = h(G_{\hat{F}},\ell,\nu) \tag{5}$$

for all $1 \le \ell < k-1$ and $0 \le \nu < \tilde{\sigma}_\ell$. The conditions (4) and (5) ensure that G_F and $G_{\hat{F}}$ have the same orientation and $\Psi_F^{-1}(\hat{F})$ is paraxial, cf. Figure 3(a).
Define $p(G_F^0)_r := p(G_{\hat{F}}^0)_r$, $0 \le r < k-1$, and $p((G_F^1)_\nu)_0 := \max\{p((G_{\hat{F}}^1)_\nu)_0, p(G_{\hat{F}}^0)_{\nu \bmod 2}\}$ if $(G_F^1)_\nu$ represents an irregular edge. Given assumption (4), the basic problem is to compute the so-called *constraints coefficients* $\kappa_{\hat{\gamma},\gamma}$, which are given by

$$\eta_{\hat{\gamma}} \circ \Phi = \sum_{G \in \mathrm{adj}(G_F),\ \gamma \in I^{\tilde{b}(G)}} \kappa_{\hat{\gamma},\gamma} \eta_\gamma$$

for $\hat{\gamma} \in I^{\tilde{b}(\hat{G})}$, $\mathrm{adj}(G_F) := \{G_F^0\} \cup \{(G_F^\ell)_\nu \mid 0 \le \ell < k-2,\ 0 \le \nu < \tilde{\sigma}_\ell\}$ and $\hat{G} \in \mathrm{adj}(G_{\hat{F}})$. Due to the tensor product structure and the properties of Φ, the coefficients are determined by $\kappa_{\hat{\gamma},\gamma} = \prod_{r=0}^{\ell-1} \bar{\kappa}_{\hat{\gamma}_r,\gamma_r}(v_r,w_r)$, where the coefficients $\bar{\kappa}_{ij}(v,w)$ solve the problem

$$\xi_i(vx+w) = \sum_{j=0}^{p} \bar{\kappa}_{ij}(v,w)\xi_j(x),\ x \in \mathbb{R} \tag{6}$$

for $v, w \in \mathbb{R}$, cf. [9]. A simple method to calculate the coefficients in (6) is to solve the linear equation $\xi_i(vx_s + w) = \sum_{j=0}^{p} \bar{\kappa}_{ij}(v,w)\xi_j(x_s)$ with suitable test points $x_s \in (-1,1)$, $s = 0,\ldots,p$. In most finite element codes, the constraints coefficients are calculated for $v = 0.5$ and $w \in \{-0.5, 0.5\}$ describing symmetric subdivions, cf. [12]. In [9], an explicit and recursive formula for $\bar{\kappa}_{ij}(v,w)$ and arbitrary v and w is derived for the integrated Legendre and Gauss-Lobatto polynomials. This formula enables us to efficiently calculate the constraints coeffecents for arbitrary subdivsions fulfilling condition (4).

To calculate the entries of the connectivity matrices, two preprocessing steps have to be accomplished. The first step is to iterate through all faces F of $\mathscr{T}$, all $G \in \mathrm{adj}(G_F)$ and all $\gamma \in I^{\tilde{b}(G)}$. If G is associated to a regular node, edge or face, we set $\mathscr{B}(G,\beta(\gamma)) := \{(G,\beta(\gamma),1)\}$. Otherwise, we set

$$\mathscr{B}(G,\beta(\gamma)) := \left\{(\hat{G},\beta(\hat{\gamma}),\kappa_{\hat{\gamma},\gamma}) \mid \hat{G} \in \mathrm{adj}(G_{\hat{F}}),\ G \neq \hat{G},\ \hat{\gamma} \in I^{\tilde{b}(\hat{G})},\ \kappa_{\hat{\gamma},\gamma} \neq 0\right\}.$$

The second step is to combine the constraints coefficients through

$$\mathscr{C}(G,\beta) := \left\{(\hat{G},\hat{\beta},\kappa) \mid (\hat{G},\hat{\beta},\kappa) \in \mathscr{B}(G,\beta),\ \hat{G} \text{ regular}\right\} \biguplus_{\substack{(\hat{G},\hat{\beta},\kappa) \in \mathscr{B}(G,\beta),\\ \hat{G} \text{ irregular}}} \kappa\mathscr{C}(\hat{G},\beta)$$

with $\kappa\{(G_0,\beta_0,\kappa_0),(G_1,\beta_1,\kappa_1),\ldots\} := \{(G_0,\beta_0,\kappa\kappa_0),(G_1,\beta_1,\kappa\kappa_1),\ldots\}$ and

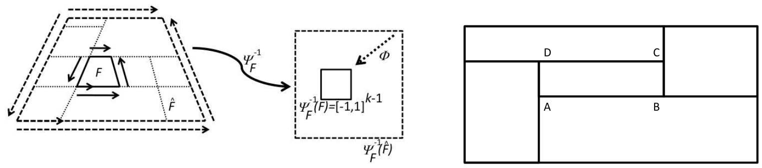
$$\begin{aligned}
\mathscr{C}_0 \uplus \mathscr{C}_1 :=& \{(G,\beta,\kappa) \mid (G,\beta,\kappa) \in \mathscr{C}_0,\ \nexists\kappa' :\ (G,\beta,\kappa') \in \mathscr{C}_1\} \\
&\cup \{(G,\beta,\kappa) \mid (G,\beta,\kappa) \in \mathscr{C}_1,\ \nexists\kappa' :\ (G,\beta,\kappa') \in \mathscr{C}_0\} \\
&\cup \{(G,\beta,\kappa+\kappa') \mid (G,\beta,\kappa) \in \mathscr{C}_0, \exists\kappa' :\ (G,\beta,\kappa') \in \mathscr{C}_1\}.
\end{aligned}$$

Using these sets, the entries of the connectivity matrix for a mesh element $T \in \mathscr{T}$ are computed using an extension of (3): For $G = (G_T^\ell)_\nu \in \mathscr{G}^\ell$ and $\alpha \in I^{b(G)}$, we set

$$\pi_{T,M(\hat{G},\hat{\beta}),m_T(\alpha)} := \mu(\alpha)\kappa$$

for all $(\hat{G},\hat{\beta},\kappa) \in \mathscr{C}(G,\beta(\alpha))$.

Note that there are some (possibly artificial) cases for which the recursive definition of $\mathscr{C}$ results in an infinite loop over the hanging nodes. A 2D-example for such a situation is given in Figure 3(b) for hanging nodes A,B,C and D. For implemention purposes, we need the data structures G_T and G_F to represent mesh elements and faces. Furthermore, we need a mapping which gives us the regular face $\hat{F}$ for an irregular face F with $F \subset \hat{F}$. Such a mapping is easily generated during the refinement process of a regular coarse mesh. The proposed approach may be extended to higher-dimensional mesh elements ($k \geq 4$), given an appropriate definition of $p_r^{b(G)}$, $\beta(\alpha)$ and $\mu(\alpha)$.



(a) (b)

Fig. 3 **(a)** Orientation of F, $\hat{F}$ and their edges, **(b)** irregular 2D mesh for which the generation of $\mathscr{C}$ results in an inifinite loop.

4 Numerical results

In this section, we give some numerical results on the application of unsymmetric subdivisions and multi-level hanging nodes in 2D and 3D. The problem under consideration is Poisson's problem $-\Delta u = f$ on a L-shaped domain and on a cube. The right-hand side f and the boundary conditions are chosen so that u has a corner singularity in the re-entrant corner of the L-shaped domain and at one corner of the cube, respectively. We use serendipity shape functions and adapt the finite element mesh with symmetric (symm.) as well as unsymmetric (unsymm.) subdivisions at the corner and an increasing polynomial degree distribution. Figure 4(d) shows such an unsymmetric refinement for a cube with polynomial degrees marked by grey scales. Moreover, we use an automatic *hp*-adaptive scheme based on two a posteriori error estimators η_T and $\tilde{\eta}_T$ which estimate the local discretization error on $T \in \mathscr{T}$ for different degree disctributions $p_T \leq \tilde{p}_T$. Using well-known a priori estimates, we estimate the local regularity ρ_T of u with $\rho_T \approx \frac{\log(\tilde{\eta}_T/\eta_T)}{\log(p_T/\tilde{p}_T)} + 1$. We increase the polynomial degree if $\rho_T \geq \tilde{p}_T$, and refine T otherwise. For more details, see [10]. We use this strategy for symmetric (Figure 4(a),(b) - adaptive) as well as unsymmetric subdivisions (Figure 4(c) - unsym.2). In Figure 4(c), only the polynomial degree is adapted whereas in Figure 4(a) und (e), both the polynomial degree and the mesh are adapted with multi-level hanging nodes. For all these *hp*-adaptive refinements, we obtain exponential convergence rates (Figure 4(f) for the L-shaped domain and (h) for the cube). Additional refinement of all mesh elements with multi-level hanging nodes can be applied to ensure 1-irregularity of the mesh. However, in our numerical experiments with automatic *hp*-adaptive schemes on the L-shaped domain, the exponential convergence is lost, see Figure 4(g). This is due to the fact that only mesh elements at the corner are refined in the first steps of the refinement so that multi-level hanging nodes do not occur; but thereafter some mesh elements are refined which are not at the re-entrant corner so that multi-level hanging nodes are generated. The additional refinements for the elimination of multi-level hanging nodes leads to further multi-level hanging nodes on the next layer and so on. In the end, an almost global refinement is performed, which results in the decrease of the convergence rate, see Figure 4(b). This underlines the benefit of schemes which are able to handle multi-level hanging nodes.

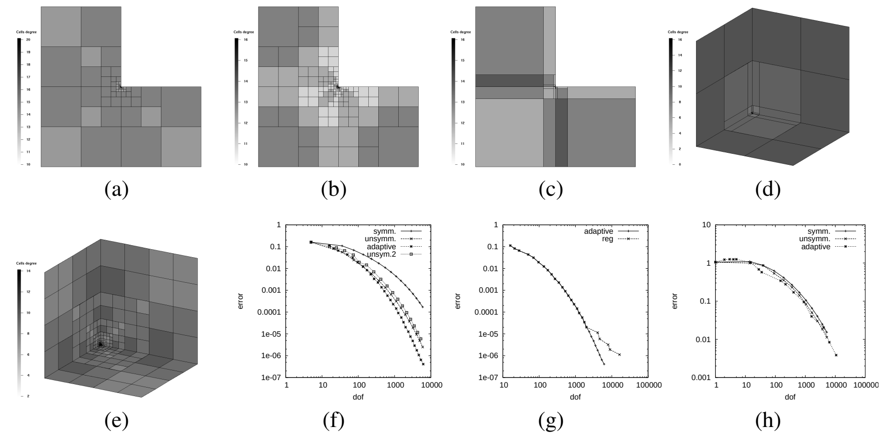


Fig. 4 *hp*-adaptive meshes in 2D and 3D and convergence rates.

References

1. Ainsworth, M., Senior, B.: Aspects of an adaptive *hp*-finite element method: Adaptive strategy, conforming approximation and efficient solvers. Comput. Methods Appl. Mech. Eng. **150**(1-4), 65–87 (1997)
2. Bangerth, W., Kayser-Herold, O.: Data structures and requirements for *hp* finite element software. ACM Transactions on Mathematical Software **36**(1), 4:1–4:31 (2009)
3. Demkowicz, L., Gerdes, K., Schwab, C., Bajer, A., Walsh, T.: HP90: A general and flexible Fortran 90 *hp*-FE code. Comput. Vis. Sci. **1**(3), 145–163 (1998)
4. Demkowicz, L., Oden, J.T., Rachowicz, W., Hardy, O.: Toward a universal h-p adaptive finite element strategy. i: Constrained approximation and data structure. Comp. Meth. Appl. Mech. Engrg **77**, 79–112 (1989)
5. Demkowicz, L.F.: Computing with *hp*-adaptive finite elements. Vol. 1: One- and two-dimensional elliptic and Maxwell problems. Chapman & Hall/CRC (2007)
6. Demkowicz, L.F., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W., Zdunek, A.: Computing with *hp*-adaptive finite elements. Vol. II: Frontiers: Three-dimensional elliptic and Maxwell problems with applications. Chapman & Hall/CRC (2008)
7. Frauenfelder, P., Lage, C.: Concepts – an object-oriented software package for partial differential equations. M2AN, Math. Model. Numer. Anal. **36**(5), 937–951 (2002)
8. Karniadakis, G.E., Sherwin, S.J.: Spectral/*hp* element methods for computational fluid dynamics. 2nd ed. Numerical Mathematics and Scientific Computation. Oxford: Oxford University Press. (2005)
9. Schröder, A.: Constraints coefficients in *hp*-FEM. Kunisch, Karl (ed.) et al., Numerical mathematics and advanced applications. Proceedings of ENUMATH 2007, Berlin: Springer, 183–190 (2008)
10. Schröder, A.: Error control in *h*- and *hp*-adaptive FEM for Signorini's problem. J. Numer. Math. **17**(4), 299–318 (2009)
11. Šolín, Segeth, K., Delezel, I.: Higher-order finite element methods. Studies in Advanced Mathematics. CRC Press, Boca Raton (2004)
12. Šolín, P., Červený, J., Doležel, I.: Arbitrary-level hanging nodes and automatic adaptivity in the *hp*-FEM. Math. Comput. Simul. **77**(1), 117–132 (2008)