Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **2**, No. 4, pp. 518-532

DOI: 10.4208/aamm.10-m1012
August 2010

Adaptive hp-FEM with Arbitrary-Level Hanging Nodes for Maxwell's Equations

Pavel Solin[1,2,*], Lenka Dubcova[2] and Ivo Dolezel[2]

[1] *Department of Mathematics and Statistics, University of Nevada, Reno, 89557, USA*
[2] *Institute of Thermomechanics, Academy of Sciences of the Czech Republic, Narodni 3, 11720 Praha 1, Czech Republic*

Received 22 February 2010; Accepted (in revised version) 13 April 2010

Available online 28 May 2010

Abstract. Adaptive higher-order finite element methods (hp-FEM) are well known for their potential of exceptionally fast (exponential) convergence. However, most hp-FEM codes remain in an academic setting due to an extreme algorithmic complexity of hp-adaptivity algorithms. This paper aims at simplifying hp-adaptivity for $\boldsymbol{H}$(curl)-conforming approximations by presenting a novel technique of arbitrary-level hanging nodes. The technique is described and it is demonstrated numerically that it makes adaptive hp-FEM more efficient compared to hp-FEM on regular meshes and meshes with one-level hanging nodes.

AMS subject classifications: 35Q60, 65N30, 65N50, 78M10

Key words: hp-FEM, arbitrary-level hanging nodes, irregular meshes, higher-order edge elements, Maxwell's equations.

1 Introduction

Nowadays, vector-valued finite elements with continuous tangential components on element interfaces (edge elements) are a standard tool for the solution of Maxwell's equations in various cavity devices such as waveguides, resonators, microwave ovens, and other models. Edge elements are based on differential forms introduced in late 1950s by H. Whitney [19], in the context of differential geometry. Apparently, the first link between the Whitney forms and computational electromagnetics was made in 1984 by P. R. Kotiuga in his thesis [9]. A nice monograph on this subject is [3].

Adaptive higher-order finite element methods (hp-FEM) based on higher-order edge elements belong to the youngest topics in computational electromagnetics (see,

*Corresponding author.
URL: `http://hpfem.math.unr.edu/people/pavel/work/solin.tar.gz`.
Email: solin@unr.edu (P. Solin), dubcova@gmail.com (L. Dubcova), doleze@fel.cvut.cz (I. Dolezel)

e.g., [9, 10, 16] and the references therein). Especially for problems involving important small-scale phenomena such as singularities or steep gradients along internal or boundary layers, the efficiency gap between adaptive hp-FEM and standard adaptive low-order FEM can be impressive. On the other hand, these methods are not used widely by practitioners yet due to their high algorithmic complexity. From this point of view, the design of simple hp-adaptivity algorithms is of crucial importance.

It is worth mentioning that hp-adaptivity is profoundly different from h-adaptivity due to a large number of element refinement options per element (around 100 in 2D and several hundred in 3D). This number depends on multiple factors such as whether one allows anisotropic refinements in space and anisotropic (directionally different) polynomial degrees in quadrilateral/hexahedral elements, how much the polynomial degree is allowed to vary in subelements after an element is refined in space, etc. Standard a-posteriori error estimates used for h-adaptivity, that only provide an information about the magnitude of error in elements, do not help to select an optimal element refinement in hp-adaptivity. For that, one needs a much better information about the error, namely *its shape in every element*. In principle, this information might be reconstructed from suitable a-posteriori estimates of higher derivatives of the solution, but this would be extremely difficult and the authors are not aware of any such work. Currently, the two major approaches to guiding adaptivity in higher-order finite element methods are:

1. *Computing a reference solution* on a globally refined mesh [11, 14]. This approach is computationally expensive but on the other hand it works for *any equation including multiphysics coupled problems where no standard a-posteriori error estimates are available* [7, 15, 17, 18].
2. *Estimating analyticity* of the solution in every element in order to decide whether an h- and p-refinement should be done [8]. This technique requires additional equation-dependent tuning parameters, and it does not allow variable polynomial degrees in subelements when an element is refined in space.

In this paper we use the former approach, and extend a novel technique of arbitrary-level hanging nodes [13] from standard H^1-conforming (continuous scalar) approximations to vector-valued approximations in $\boldsymbol{H}$(curl). This technique is a valuable addition to existing adaptivity algorithms since it makes it possible to refine any element in the mesh locally, without affecting its neighbors. In turn one can design simple hp-adaptivity algorithms that work in an element-by-element fashion. In other words, when refining an element, one never has to refine neighboring mesh elements to keep the mesh regular. Note that this is impossible with algorithms employing regular meshes such as [4] or meshes containing one-level hanging nodes [6], since in these cases one has to deal with unwanted, regularity-enforced additional refinements.

There exist several implementations of the technique of multiple-level hanging nodes for second-order elliptic problems [6, 12, 13], but to our best knowledge, the technique [13] is the only one to work independently of the underlying higher-order shape functions.

The extension of the results [13] to $\boldsymbol{H}$(curl)-conforming approximations was non-trivial because of different conformity requirements in the space $\boldsymbol{H}$(curl) and different structure and properties of higher-order shape functions (there are no vertex functions in the $\boldsymbol{H}$(curl)-conforming case and the higher-order shape functions are constructed in a different way). Our implementation of inter-element constraints is compatible with the well-known De Rham diagram [16] that connects the spaces H^1 and $\boldsymbol{H}$(curl) through the gradient operator. Otherwise, the message of this paper is similar to the message of [13]: arbitrary-level hanging nodes are not so difficult to implement as one would expect, they simplify hp-adaptivity algorithms substantially, and they make the computer code more efficient.

The algorithms presented in this paper are available on-line in the form of a modular C++ library Hermes (see, `http://hpfem.org/`).

The paper is organized as follows: the rest of Section 1 contains a model problem for time-harmonic Maxwell's equations. The technique of arbitrary-level hanging nodes for $\boldsymbol{H}$(curl)-conforming approximations is discussed in Section 2, and a simple element-by-element hp-adaptivity algorithm is described in Section 3. Numerical examples demonstrating the superiority of the novel hp-adaptivity algorithm over existing algorithms based on regular meshes are shown in Section 4. Conclusions and outlook are presented in Section 5.

1.1 Time-harmonic Maxwell's equations

Consider the problem of solving the normalized time-harmonic Maxwell's equation [10]

$$\nabla \times \left(\mu_r^{-1} \nabla \times \boldsymbol{E}\right) - \kappa^2 \epsilon_r \boldsymbol{E} = j\kappa\sqrt{\mu_0}\boldsymbol{J}_a, \tag{1.1}$$

in a bounded domain $\Omega \subset \mathbb{R}^2$ with piecewise-linear boundary. Here $\mu_r = \mu/\mu_0$ is the relative magnetic permeability, $\boldsymbol{E}$ the (complex) phasor of harmonic electric field strength, $\kappa = \omega/c$ the wave number, j the imaginary unit, $\epsilon_r = (\epsilon + j\gamma/\omega)/\epsilon_0$ the (complex) relative electric permittivity, $\boldsymbol{J}_a$ the (complex) phasor of the vector-valued density of conductive currents, ω the angular frequency and $c = (\sqrt{\epsilon_0\mu_0})^{-1}$ the speed of light in vacuum.

Eq. (1.1) may be equipped with various types of boundary conditions, such as, for example, the perfect conductor boundary condition

$$\boldsymbol{E} \cdot \boldsymbol{t} = 0, \quad \text{on } \partial\Omega, \tag{1.2}$$

or the impedance condition

$$\nabla \times \boldsymbol{E} - j\lambda \boldsymbol{E} \cdot \boldsymbol{t} = \boldsymbol{g} \cdot \boldsymbol{t}, \quad \text{on } \partial\Omega. \tag{1.3}$$

With (1.2), the weak formulation of (1.1) reads: find $\boldsymbol{E} \in \boldsymbol{Q}$, such that

$$\begin{aligned} &\int_\Omega \mu_r^{-1} (\nabla \times \boldsymbol{E}) \cdot (\nabla \times \overline{\boldsymbol{F}}) \mathrm{d}\boldsymbol{x} - \int_\Omega \kappa^2 \epsilon_r \boldsymbol{E} \cdot \overline{\boldsymbol{F}} \mathrm{d}\boldsymbol{x} \\ &= \int_\Omega j\kappa\sqrt{\mu_0} \boldsymbol{J}_a \cdot \overline{\boldsymbol{F}} \mathrm{d}\boldsymbol{x}, \end{aligned} \tag{1.4}$$

for all $\boldsymbol{F} \in \boldsymbol{Q}$, where $\boldsymbol{Q}$ is a complex vector space defined as

$$\boldsymbol{Q} = \Big\{ \boldsymbol{E} \in \boldsymbol{H}(\mathrm{curl}, \Omega);\ \ \boldsymbol{E} \cdot \boldsymbol{t} = 0,\ \ \text{on}\ \ \partial\Omega \Big\}. \tag{1.5}$$

The symbol $\overline{\boldsymbol{F}}$ stands for the complex-conjugate of $\boldsymbol{F}$. For completeness, let us mention the definitions

$$\boldsymbol{H}(\mathrm{curl}, \Omega) = \Big\{ \boldsymbol{E} \in [L^2(\Omega)]^2;\ \ \nabla \times \boldsymbol{E} \in L^2(\Omega) \Big\}, \tag{1.6a}$$

$$\nabla \times \boldsymbol{E} = \frac{\partial E_2}{\partial x_1} - \frac{\partial E_1}{\partial x_2}. \tag{1.6b}$$

The domain Ω is covered with a finite element mesh $\tau_{h,p}$ consisting of non-overlapping convex elements $K_1, K_2, \cdots, K_M$ (in practice triangles or quadrilaterals) equipped with polynomial degrees $1 \leq p_1, p_2, \cdots, p_M$. The finite element subspace of $\boldsymbol{Q}$ has the form

$$\boldsymbol{Q}_{h,p} = \Big\{ \boldsymbol{E}_{h,p} \in \boldsymbol{Q};\ \ \boldsymbol{E}_{h,p} \text{ is polynomial of degree } p_i \text{ in } K_i,\ i = 1, \cdots, M \Big\}. \tag{1.7}$$

By N we denote the dimension of $\boldsymbol{Q}_{h,p}$ (the number of degrees of freedom in the discrete problem). Recall that functions in the space $\boldsymbol{Q}_{h,p}$ are discontinuous but have continuous tangential components on element interfaces (see, e.g., [3, 10]). In the following we use a standard hierarchic basis on a reference square element $(-1, 1)^2$ that can be found, e.g., in paragraph 2.3.2 of [16].

2 Arbitrary-level hanging nodes in H(curl)

Standard finite element mesh where every pair of adjacent elements shares either a vertex or an entire edge, is called *regular*. Every edge in a regular mesh is *unconstrained* since it carries one or more degrees of freedom. In a mesh with hanging nodes one has both unconstrained and *constrained* edges. Such a situation is depicted in Fig. 1. There, the edge AB of element K_1 is unconstrained but the edges CB of element K_2 and AC of element K_3 are constrained, since on each of them, the tangential component of the approximation is dictated by the tangential component of the approximation on the

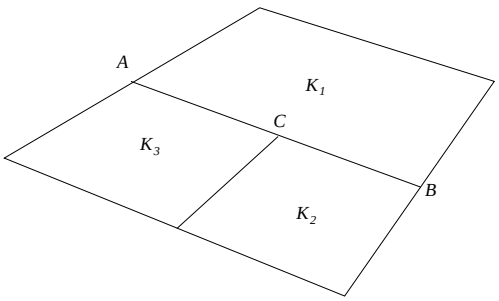

Figure 1: Example of a mesh with one-level hanging nodes.

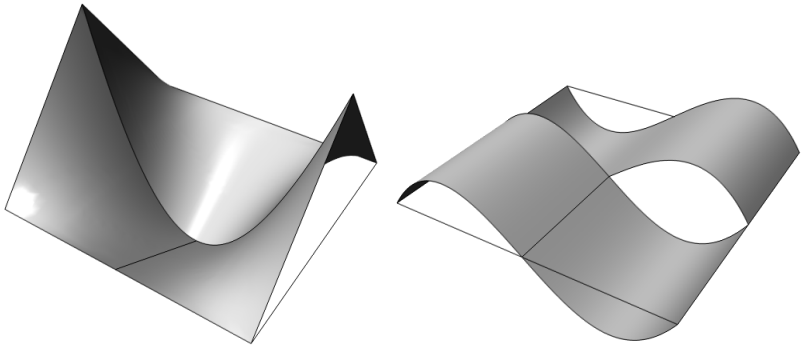

Figure 2: Quadratic basis function on the edge AB. Left: tangential component corresponding to AB, right: normal component corresponding to AB.

edge AB. Constrained edges do not carry degrees of freedom. We say that the edge AB is *constraining* the edges CB and AC. For illustration, a quadratic vector-valued edge function associated with the edge AB, with continuous tangential component and discontinuous normal component, is shown in Fig. 2.

By $\boldsymbol{E}_{K_i}$ we denote the restriction of the function from Fig. 2 to the element K_i, $i = 1, 2, 3$. The constraining function $\boldsymbol{E}_{K_1}$ is a standard edge function. The constrained functions $\boldsymbol{E}_{K_2}$ and $\boldsymbol{E}_{K_3}$ are linear combinations of standard edge functions on the corresponding elements such that

$$\boldsymbol{E}_{K_2} \cdot \boldsymbol{t}_{CB} \equiv \boldsymbol{E}_{K_1} \cdot \boldsymbol{t}_{AB}, \quad \text{on the edge } CB,$$

and

$$\boldsymbol{E}_{K_3} \cdot \boldsymbol{t}_{AC} \equiv \boldsymbol{E}_{K_1} \cdot \boldsymbol{t}_{AB}, \quad \text{on the edge } AC,$$

where $\boldsymbol{t}_{AB}$, $\boldsymbol{t}_{AC}$ and $\boldsymbol{t}_{CB}$ represent unit tangential vectors to the edges AB, AC and CB, respectively.

In general, let the polynomial degree of the edge AB be some $p_{AB} \geq 0$. Then there are $p_{AB} + 1$ constraining shape functions on K_1 associated with the edge AB, with polynomial degrees $p = 0, 1, \cdots, p_{AB}$. Interior shape functions (bubble functions) are never constraining nor constrained, and therefore they do not influence the calculation

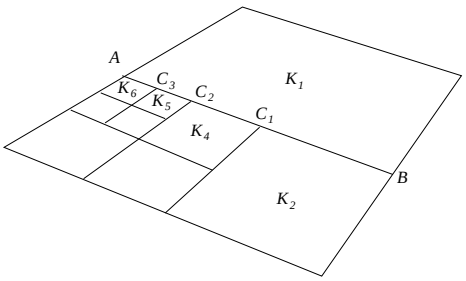

Figure 3: Example of a mesh with three-level hanging nodes.

of constraint coefficients. By definition, every constrained edge inherits its orientation and polynomial degree from the constraining one (even if this is in contradiction to the minimum rule [16]). Every edge function on K_1 of polynomial degree $0 \leq p \leq p_{AB}$ that is associated with the edge AB constrains $p+1$ edge functions on the element K_3 associated with edge AC and $p+1$ edge functions on the element K_2 associated with edge CB.

The case of multiple-level constraints is analogous. Consider, for illustration, a mesh with three-level hanging nodes shown in Fig. 3.

As in the previous case, there are $p_{AB}+1$ constraining edge functions on K_1 associated with the edge AB, and an edge function of polynomial degree p constrains $p+1$ edge functions on each of the elements K_2, K_4, K_5, K_6 (associated with edges $C_1B, C_2C_1, C_3C_2, AC_3$, respectively). Example of a quadratic basis function associated with the edge AB, which consists of a quadratic edge function $\boldsymbol{E}_{K_1}$ on K_1 constraining quadratic edge functions on the elements K_2, K_4, K_5, K_6, is shown in Fig. 4.

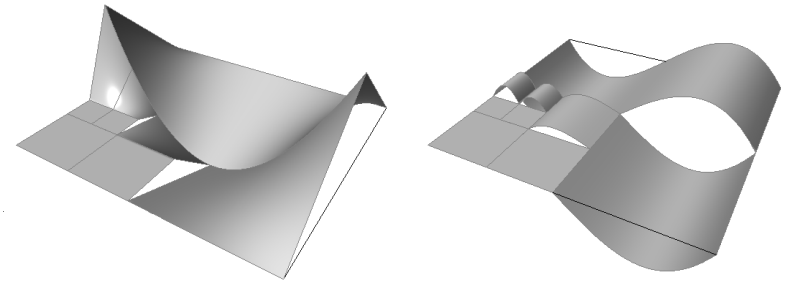

Figure 4: Quadratic basis function on the edge AB. Left: tangential component corresponding to AB, right: normal component corresponding to AB.

Next let us show how the constraint coefficients are calculated. The algorithm requires a unique enumeration of basis functions $\boldsymbol{E}_1, \boldsymbol{E}_2, \cdots, \boldsymbol{E}_N$ of the finite element space $\boldsymbol{Q}_{h,p}$ as well as a unique local enumeration of shape functions on the reference domain $\hat{K}$. Let e be an unconstrained edge of a mesh element K_i, and let p_e be the polynomial degree of e. The element K_i is mapped onto the reference domain $\hat{K}$ via a reference map $\boldsymbol{x}_{K_i} : \hat{K} \to K_i$. Let $\hat{e}$ be the edge of $\hat{K}$ such that

$$\boldsymbol{x}_{K_i}(\hat{e}) = e,$$

and by $\boldsymbol{\varphi}^e_0, \boldsymbol{\varphi}^e_1, \cdots, \boldsymbol{\varphi}^e_{p_e}$ let us denote the edge functions on $\hat{K}$ associated with the edge $\hat{e}$. The edge e is equipped with an edge node

$$\boldsymbol{d}^e = \{m^e_0, m^e_1, \cdots, m^e_{p_e}\}, \tag{2.1}$$

where m^e_j are indices of the basis functions of the space $\boldsymbol{Q}_{h,p}$ that are related to the shape functions $\boldsymbol{\varphi}^e_0, \boldsymbol{\varphi}^e_1, \cdots, \boldsymbol{\varphi}^e_{p_e}$ via the standard transformation relation

$$\boldsymbol{\varphi}^e_j(\boldsymbol{\xi}) = \Big(\frac{\mathrm{D}\boldsymbol{x}_{K_i}}{\mathrm{D}\boldsymbol{\xi}}\Big)^T(\boldsymbol{\xi})\ \boldsymbol{E}_{m^e_j}\big(\boldsymbol{x}_{K_i}(\boldsymbol{\xi})\big), \quad j = 0, 1, \cdots, p_e. \tag{2.2}$$

Next, let K_i be an element in the mesh whose edge e is subset of another mesh edge AB of polynomial degree p_e. In addition to the standard (unconstrained) edge node $\boldsymbol{d}^e$, we define a *constrained edge node*

$$\boldsymbol{c}^e = \{r,\, q^e\},$$

where r is a reference to the standard node associated with the constraining edge AB and the index q^e identifies uniquely the geometrical position of the constrained edge e within AB (see Fig. 5). Note that AB is oriented uniquely through the indices of the vertices A and B, and e inherits its orientation.

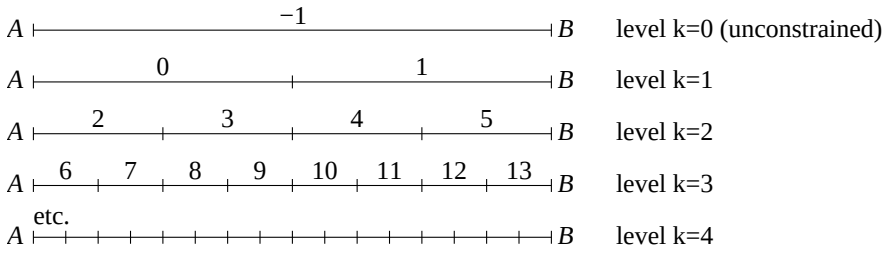

Figure 5: Values of the index q^e for various positions of a constrained edge e within a constraining edge AB.

Finally, assume a basis function $\boldsymbol{E}_k$ of the space $\boldsymbol{Q}_{h,p}$, whose tangential component $\boldsymbol{E}_k \cdot \boldsymbol{t}_{AB}$ on the edge AB is a polynomial of degree p. Restricted to the edge $e \subset AB$, the tangential component $\boldsymbol{E}_k \cdot \boldsymbol{t}_{AB}$ determines the constraint coefficients $\alpha_0^{e,p}, \alpha_1^{e,p}, \cdots, \alpha_p^{e,p}$ corresponding to the edge functions $\boldsymbol{\varphi}_0^e, \boldsymbol{\varphi}_1^e, \cdots, \boldsymbol{\varphi}_p^e$ on K_i. The values of these coefficients are obtained by solving a system of $p+1$ linear algebraic equations of the form

$$\sum_{j=0}^{p} \alpha_j^{e,p} \boldsymbol{\varphi}_j^e(y_i^p) = \psi_{e,AB}(y_i^p), \quad 0 \le i \le p,$$

where $y_i^p \in [-1,1]$ are the $p+1$ Chebyshev points of degree p on the edge e, and $\psi_{e,AB}$ is the tangential component $\boldsymbol{E}_k \cdot \boldsymbol{t}_{AB}$ transformed linearly from e to $[-1,1]$. Then,

$$\boldsymbol{\varphi}^e = \sum_{j=0}^{p} \alpha_j^{e,p} \boldsymbol{\varphi}_j^e,$$

is a new edge function of degree p on $\hat{K}$. After transforming $\boldsymbol{\varphi}^e$ to the element K_i through (2.2), its tangential component on the edge $e \subset AB$ matches exactly the tangential component of $\boldsymbol{E}_k \cdot \boldsymbol{t}_{AB}$.

3 Adaptive hp-FEM based on arbitrary-level hanging nodes

In contrast to standard adaptive FEM (h-FEM), automatic adaptivity in the hp-FEM requires more information about the behavior of the error in element interiors (see,

e.g., [6, 8, 16] and the references therein). Some authors investigate numerically the analyticity of the solution in every element in order to decide between p- and h-refinement [8]. Such approach uses two refinement candidates per element, as illustrated in Fig. 6 (the numbers in elements stand for their polynomial degrees). According to our experience, at least for elliptic problems this strategy yields exponential convergence as expected.

We prefer a different approach motivated by the work of Demkowicz et al. [6], where more refinement candidates are considered, as shown in Fig. 7.

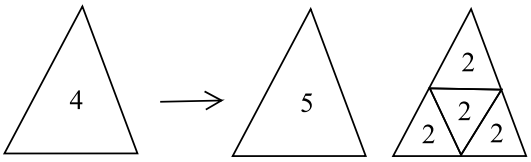

Figure 6: hp-adaptivity with two refinement candidates (p and h refinement).

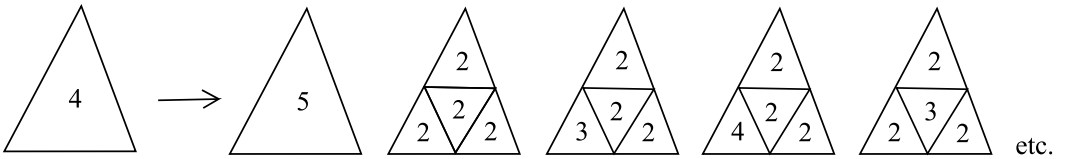

Figure 7: hp-adaptivity with multiple refinement candidates.

Typically, we vary the polynomial degrees in the subelements by two, which for a triangular element yields 3^4=81 h-refinement candidates. The strategy was described in detail in [13]. Since in the latter case every refinement candidate can be reproduced using several steps with the pair of candidates of the former strategy, it is not surprizing that usually the convergence curves are almost identical when the estimated error is plotted as a function of the number of degrees of freedom. However, according to our experience, computations with the latter approach usually take less CPU time since fewer adaptivity steps are needed and thus the discrete problem is solved less frequently.

Obviously, the latter strategy requires even more information about the error than the level of its analyticity. In order to select an optimal refinement candidate, we need to know the approximate *shape* of the error function

$$\epsilon_{h,p} = \boldsymbol{E} - \boldsymbol{E}_{h,p}.$$

In principle, this information could be recovered from suitable estimates of higher derivatives of the solution, but such approach is not very practical and it has not been used by anyone to our best knowledge. In practice, we employ the technique of *reference solutions* [6]. The reference solution $\boldsymbol{E}_{ref}$ is sought in an enriched finite element space $\boldsymbol{Q}_{ref}$, and the error function is approximated as

$$\epsilon_{h,p} \approx \boldsymbol{E}_{ref} - \boldsymbol{E}_{h,p}.$$

The reference space $\boldsymbol{Q}_{ref}$ is constructed in such a way that all elements in the mesh are subdivided uniformly and their polynomial degree is increased, i.e.,

$$\boldsymbol{Q}_{ref} = \boldsymbol{Q}_{\frac{h}{2},p+1}.$$

The method for selecting the optimal refinement candidate will be described in the following.

3.1 Element-by-element adaptivity algorithm

With an a-posteriori error estimate of the form

$$\epsilon_{h,p} \approx \boldsymbol{E}_{ref} - \boldsymbol{E}_{h,p}, \tag{3.1}$$

the outline of our hp-adaptivity algorithm is as follows:

---

1. Assume an initial coarse mesh $\tau_{h,p}$ consisting of (usually) quadratic elements. Besides other technical data, user input includes a prescribed tolerance $TOL > 0$ for the $\boldsymbol{H}(\text{curl})$ norm of the approximate error function (3.1) and the number D_{DOF} of degrees of freedom to be added in every hp-adaptivity step,
2. Compute coarse mesh approximation $\boldsymbol{E}_{h,p} \in \boldsymbol{Q}_{h,p}$ on $\tau_{h,p}$,
3. Find reference solution $\boldsymbol{E}_{ref} \in \boldsymbol{Q}_{ref}$, where $\boldsymbol{Q}_{ref}$ is obtained by dividing all elements and increasing the polynomial degrees by one,
4. Construct the approximate error function (3.1), calculate its norm

$$ERR_i^2 = \|\epsilon_{h,p}\|_A^2 = (\nabla \times \epsilon_{h,p}, \nabla \times \epsilon_{h,p})_{K_i} + \kappa^2 (\epsilon_{h,p}, \epsilon_{h,p})_{K_i},$$

   on every element K_i in the mesh, $i = 1, 2, \cdots, M$. Calculate the global error

$$ERR^2 = \sum_{i=1}^{M} ERR_i^2,$$

5. If $ERR \leq TOL$, stop computation and proceed to postprocessing,
6. Sort all elements into a list L according to their ERR_i values in decreasing order,
7. While the number of newly added degrees of freedom in this step is less than D_{DOF} do:
   (a) Take next element K from the list L,
   (b) Perform hp-refinement of K (to be described in more detail in Paragraph 3.2). Note that the refinement of K may introduce new hanging nodes on its edges, but the surrounding mesh elements are not affected,
8. Adjust polynomial degrees on unconstrained edges using the so-called *minimum rule* (every unconstrained edge is assigned the minimum of the polynomial degrees on the pair of adjacent elements),
9. Continue with Step 2.

Here $\kappa = \omega/c$ is the wave number. Our experience shows that for large κ, the adaptive process converges better when guided by the $\|\cdot\|_A$-norm than with the standard $\boldsymbol{H}(\text{curl})$-norm.

3.2 Selection of suitable hp-refinement

Let $K \in \tau_{h,p}$ be an element of polynomial degree p_K that was marked for refinement. Without loss of generality, assume that K is a triangle, the procedure for refinement of quadrilateral elements is analogous. We consider the following $N_{ref}=k+(k+1)^4$ refinement options, where $k \geq 0$ is a user input parameter:

1. Increase the polynomial degree of K by $1, 2, \cdots, k$ without spatial subdivision. This yields k refinement candidates,
2. Split K into four similar triangles K_1, K_2, K_3, K_4. Define p_0 to be the integer part of $p_K/2$ (largest integer number that is less or equal to $p_K/2$). For each K_i, $1 \leq i \leq 4$ consider $k+1$ polynomial degrees $p_0 \leq p_i \leq p_0 + k$. This yields additional $(k+1)^4$ refinement candidates. In this case, edges lying on the boundary of K inherit the polynomial degree p_j of the adjacent interior element K_j. Polynomial degrees on interior edges are determined using the minimum rule.

For each of these N_{ref} options, we perform a standard $\boldsymbol{H}$(curl)-projection of the reference solution $\boldsymbol{E}_{ref}$ onto the corresponding vector-valued piecewise-polynomial space on the refinement candidate. The candidate with minimum projection error relative to the number of added degrees of freedom is selected. In practice, and also in the following Section 4 we use the value k=2.

Remark 3.1. Note that if the technique of arbitrary-level hanging nodes is not in effect, hp-refinements involving spatial subdivision can be more costly than p-refinement candidates, since the latter never cause forced refinements. If the selection of the optimal element refinement is done locally, i.e., without taking the forced refinements into account, the hp-adaptive algorithm may make wrong decisions.

4 Numerical examples

Let us compare the performance of our algorithm with hp-adaptivity based on one-level hanging nodes and regular meshes.

Consider a square waveguide $\Omega=(-0.125,\ 0.125)^2$ filled with air, containing a spherical load of radius r=0.015625m and relative permittivity ϵ_r=5.5 (permittivity of porcelain). The situation is depicted in Fig. 8.

We solve the normalized time-harmonic Maxwell's equations

$$\nabla \times (\mu_r^{-1} \nabla \times \boldsymbol{E}) - \kappa^2 \epsilon_r \boldsymbol{E} = \boldsymbol{F},$$

where

$$\mu_r = \frac{\mu}{\mu_0}, \quad \kappa = \frac{\omega}{c}, \quad \text{and} \quad \epsilon_r = \frac{\epsilon}{\epsilon_0} + \frac{j\gamma}{\omega \epsilon_0}.$$

By $L=c/f$ we denote the wavelength. The frequency is chosen to be f=1.799GHz, therefore the wavelength L=1/6m and the domain contains three halves of the wave.

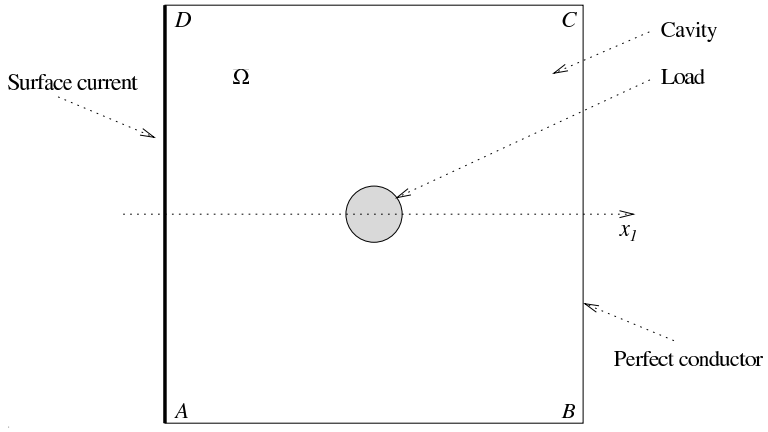

Figure 8: Computational domain.

In the waveguide, a horizontal wave is generated by time-harmonic current along the edge DA, using the Neumann boundary condition

$$\boldsymbol{n} \times (\mu_r^{-1} \nabla \times \boldsymbol{E}) = -j\omega \boldsymbol{J}_a.$$

We use time-harmonic exciting current $\boldsymbol{J}_a = 10^{-7}$A. The rest of the boundary is equipped with perfect conductor boundary conditions $\boldsymbol{E} \cdot \boldsymbol{t} = 0$. The problem was solved three-times: using adaptive hp-FEM with arbitrary-level hanging nodes, adaptive hp-FEM with one-level hanging nodes, and adaptive hp-FEM with regular meshes. In Figs. 9–13 we show the approximate solution and the corresponding finite element meshes for the three cases. In the meshes, the numbers inside the elements stand for their polynomial degrees. The presence of two numbers (colors) inside an element means that polynomial degrees in the horizontal and vertical direction are

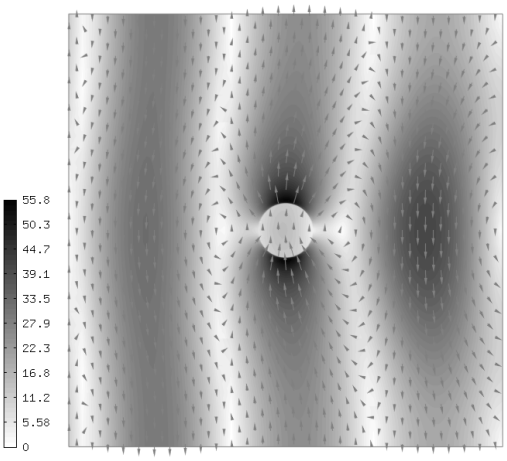

Figure 9: Approximate solution $\boldsymbol{E}$ (est. rel. error 0.1% in the $\boldsymbol{H}$(curl)-norm).

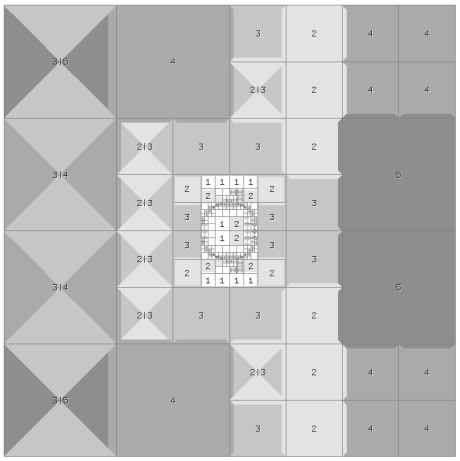

Figure 10: Mesh with arbitrary-level hanging nodes (est. rel. error 0.1%, 4335 DOF).

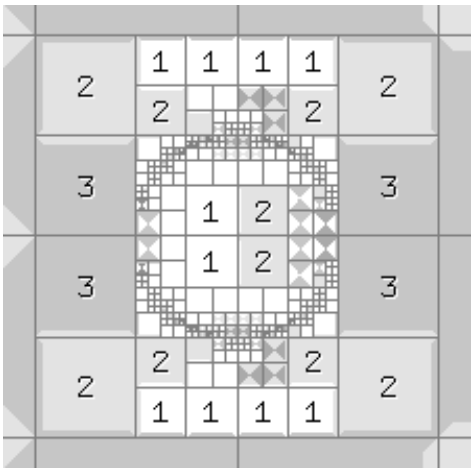

Figure 11: Detail of the central part of Fig. 10 showing fourth-level hanging nodes.

different.

The rate of convergence for all three cases is compared in Fig. 14. The reader can see there that the algorithm based on arbitrary-level hanging nodes was by far most efficient.

4.1 Extension to 3D

Let us comment briefly on the extension to 3D. In our 3D code (which was not discussed here) we use hexahedral elements. Every hexahedral element can be split either isotropically into 8 elements or anisotropically into 2 or 4 elements. Skipping, for simplicity, the anisotropic refinements and the fact that one can choose different poly-

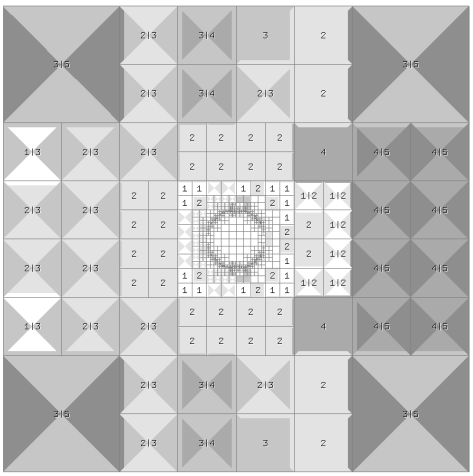

Figure 12: Mesh with one-level hanging nodes (est. rel. error 0.1%, 8438 DOF).

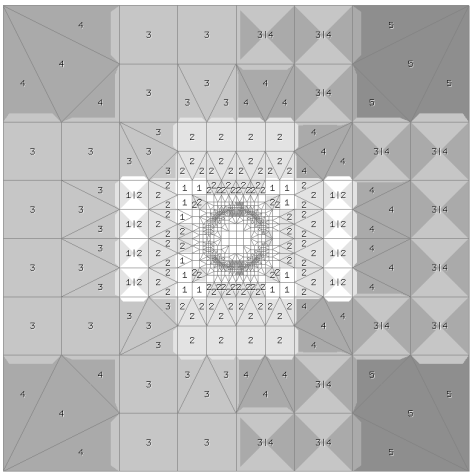

Figure 13: Regular mesh (est. rel. error 1.27%, 8752 DOF). The method did not converge to the est. rel. error level 0.1 % due to excessive memory requirements of the sparse direct solver.

nomial degrees in various directions, still one has $k+(k+1)^8$ candidates instead of $k+(k+1)^4$ in 2D. For illustration, with k=2, this is 83 in 2D and 6563 in 3D. Therefore, typically, we use k=2 in 2D but only k=1 in 3D.

Interestingly, however, it turns out that the parameter k has virtually no effect on the convergence speed in terms of degrees of freedom. The reason is that for two different values $k_1>k_2$, all refinement candidates obtained with k_1 can be reproduced in several refinement steps with k_2. With k_1, the mesh refinement takes longer but one needs fewer iterations while with k_2, the mesh refinement is faster but one needs more iterations. Since every mesh refinement is followed by a global solve, one also needs to consider the efficiency of the solver when choosing the parameter k.

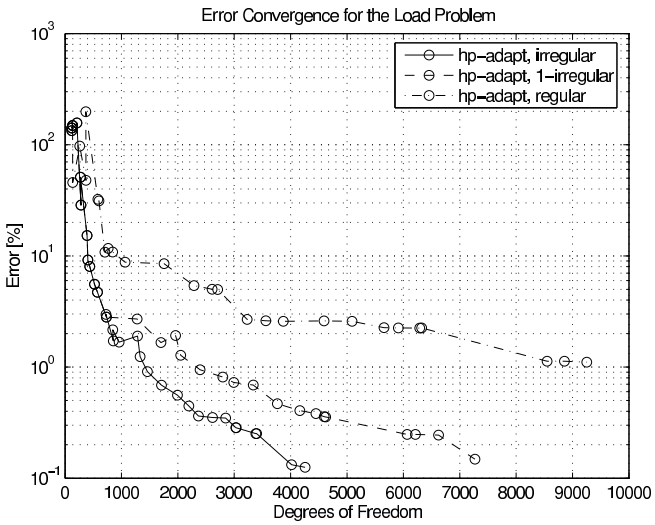

Figure 14: Convergence of adaptive hp-FEM with arbitrary-level hanging nodes, one-level hanging nodes, and regular meshes.

5 Conclusions and outlook

We presented a novel technique of arbitrary-level hanging nodes for $\boldsymbol{H}$(curl)-conforming approximations. This technique eliminates regularity-enforced mesh refinements from the adaptive process, which in turn makes it possible to design very simple hp-adaptivity algorithms working in an element-by-element fashion. We have demonstrated numerically that the elimination of regularity-enforced refinements improves the performance of adaptivity algorithms, compared to algorithms employing regular meshes or meshes with one-level hanging nodes.

The simplification of hp-adaptivity algorithms is an important step towards our major goal–the development of adaptive hp-FEM for multiphysics coupled problems. In order to do this most efficiently, every physical field or solution component needs to be approximated on an individual mesh equipped with an autonomous adaptivity algorithm-we call this approach *multi-mesh hp-FEM*. Our first results related to the coupled problems of linear thermoelasticity and thermally-conductive flow [17, 18] are very promising.

The technique of arbitrary-level hanging nodes is an essential ingredient for the multi-mesh hp-FEM since it prevents conflicting refinements across multiple meshes. The simultaneous treatment of the electric field, temperature, flow, and possibly other quantities via the multi-mesh hp-FEM requires an ability to combine higher-order edge elements, standard continuous elements, discontinuous L^2-elements, and possibly other element types, always with arbitrary-level hanging nodes. With the results presented in this paper, we should now be able to extend the multi-mesh hp-FEM from thermoelasticity and thermally-conductive flow to coupled problems of electromagnetics.

References

[1] I. BABUSKA, W. GUI, *The h, p and hp-versions of the finite element method in 1 dimension-part III: the adaptive hp-version*, Numer. Math., 49 (1986), pp. 659–683.

[2] I. BABUŠKA, B. SZABO, I. N. KATZ, *The p-version of the finite element method*, SIAM. J. Numer. Anal., 18 (1981), pp. 515–545.

[3] A. BOSSAVIT, Computational Electromagnetism, Academic Press, 1998.

[4] C. CARSTENSEN, D. BRAESS AND R. W. HOPPE, *Convergence analysis of a conforming adaptive finite element method for an obstacle problem*, J. Numer. Math., 107(3) (2007), pp. 455–471.

[5] T. A. DAVIS, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM. T. Math. Software., 30(2) (2004), pp. 165–195.

[6] L. DEMKOWICZ, J. T. ODEN, W. RACHOWICZ AND O. HARDY, *Toward a universal hp-adaptive finite element strategy, part 1: constrained approximation and data structure*, Comput. Method. Appl. Math. Engrg., 77 (1989), pp. 79–112.

[7] L. DUBCOVA, P. SOLIN, J. CERVENY AND P. KUS, *Space and time adaptive two-mesh hp-FEM for transient microwave heating problems*, Electromagnetics., 30(1) (2010), pp. 23–40.

[8] T. EIBNER, J. M. MELENK, *An adaptive strategy for hp-FEM based on testing for analyticity*, Comp. Mech., 39 (2007), pp. 575–595.

[9] P. R. KOTIUGA, Hodge Decompositions and Computational Electromagnetics, Thesis, Department of Electrical Engineering, McGill University, 1984.

[10] P. MONK, Finite Element Methods for Maxwell's Equations, Clarendon Press, Oxford, 2002.

[11] W. RACHOWICZ AND L. DEMKOWICZ, *An hp-adaptive finite element method for electromagnetics, part II, a 3D implementation*, Internat. J. Numer. Methods. Engrg., 53 (2002), pp. 147–180.

[12] K. SCHMIDT AND P. FRAUENFELDER, *hp*-adaptive FE discretization for time-harmonic Maxwell equations in 2D, CSE Annual Report, ETH Zurich, 2004.

[13] P. SOLIN, J CERVENY AND I. DOLEZEL, *Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM*, Math. Comput. Simul., 77 (2008), pp. 117–132.

[14] P. SOLIN AND L. DEMKOWICZ, *Goal-oriented hp-adaptivity for elliptic problems*, Comput. Method. Appl. Mech. Engrg., 193 (2004), pp. 449–468.

[15] P. SOLIN, L. DUBCOVA AND J. KRUIS, *Adaptive hp-FEM with dynamical meshes for transient heat and moisture transfer problems*, J. Comput. Appl. Math. 233 (2010), pp. 3103–3112.

[16] P. SOLIN, K. SEGETH AND I. DOLEZEL, Higher-Order Finite Element Methods, Chapman & Hall/CRC Press, 2003.

[17] P. SOLIN, J. CERVENY AND L. DUBCOVA, *Adaptive multi-mesh hp-FEM for linear thermoelasticity*, J. Comput. Appl. Math., published online (doi 10.1016/j.cam.2009.08.092), 2009.

[18] P. SOLIN, J. CERVENY, L. DUBCOVA AND I. DOLEZEL, Multi-Mesh *hp*-FEM for Thermally Conductive Incompressible Flow, In: Proceedings of Eccomas Conference Couple Problems 2007 (M. Papadrakakis, E. Onate, B. Schrefler Eds.), CIMNE, Barcelona.

[19] H. WHITNEY, Geometric Integration Theory, Princeton University Press, Princeton, NJ, 1957.