An easy treatment of hanging nodes in \(hp\)-finite elements

Paolo Di Stolfo\textsuperscript{a,*}, Andreas Schröder\textsuperscript{a}, Nils Zander\textsuperscript{b}, Stefan Kollmannsberger\textsuperscript{b}

\textsuperscript{a}Department of Mathematics, Universität Salzburg, Hellbrunner Str. 34, 5020 Salzburg, Austria
\textsuperscript{b}Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany

Abstract

We present an easy treatment of (multi-level) hanging nodes in \(hp\)-finite elements in two dimensions. Its simplicity is due to the fact that the connectivity matrices can directly be computed in a row-wise fashion. This is achieved by treating a global degree of freedom as a local degree of freedom on each element in an appropriate union of elements. This forms a type of support which is larger than the one used in the conventional approach and generalizes the recently presented multi-level \(hp\)-method.

Numerical experiments compare the new approach with respect to accuracy and the conditioning of the resulting linear system. We conclude that this new approach achieves nearly the same properties as the conventional approach in all investigated examples and provides an excellent trade-off between implementational complexity and solvability.

Keywords: high-order FEM, \(hp\) anisotropic refinements, arbitrary hanging nodes

1. Introduction

The \(hp\) version of the finite element method (\(hp\)-fem) aims at reducing the discretization error by a local adaption of the discretization in both the size (\(h\)) and the polynomial degree (\(p\)) of an element. The attractiveness of \(hp\)-fem lies in its ability to deliver exponential convergence rates for a wide variety of problems even if they contain singularities. Practically, this delivers high-fidelity results with a comparatively low number of degrees of freedom. A large body of literature treats \(hp\)-fem within which the classic books \cite{1, 2, 3, 4, 5}, among others, provide a good starting point for an introduction to the topic.

A wide class of problems discretized by \(hp\)-fem are partial differential equations of second order. The classical underlying (Bubnov- or Petrov-)Galerkin scheme then demands for the construction of \(H^1\)-conforming finite element spaces. Consequently, globally \(C^0\) continuous basis functions need to be constructed from the local shape functions defined on element level. However, the definition of the basis functions in the context of \(hp\)-adaptive methods is challenging. For \(p = 1\), the difficulties solely stem from hanging nodes, which occur naturally whenever an element is refined \(h\)-isotropically and at least one of its neighbors is not. For \(p > 1\), ensuring global \(C^0\) continuity is more demanding due to degrees of freedom associated to edges and faces and becomes essentially more difficult if arbitrary-level hanging nodes are permitted (\(n\)-irregular meshes).

In the presence of hanging nodes, edges and faces, global \(C^0\) continuity is usually ensured by constrained approximation which constrains the shape functions associated to these entities. The implementation of this approach bears challenges, so that some implementations only allow for 1-irregular meshes \cite{2, 3, 4, 5, 6, 7}. Only recent implementations allow for multi-level hanging nodes \cite{10, 11, 12}. Most of these are based on a local-to-global view on basis functions, as constraints are resolved locally: For each local shape function on an element, the constraints are used to define each global basis function having support on that element. This conventional approach directly leads to a support which spans a minimal number of elements and thus minimizes the coupling between the degrees of freedom.

*Corresponding author

Preprint submitted to Elsevier Monday 7th March, 2016
An alternative approach for the treatment of hanging nodes and edges is the multi-level \( hp \) enrichment presented in [13]. The approach is based on the idea of \( hp \)-\( d \)-refinement: A high-order base mesh is superposed with a finer \( h \)-overlay mesh in the domain of interest (see [14]) rather than replaced by finer elements. This alternative refinement strategy avoids the difficulties associated with arbitrary-level hanging nodes, edges and faces. However, refinement by overlaying leads to a larger support as compared to the conventional refine-by-replace approach, where an \( h \)-refined element gets replaced by child elements; this holds true since in the overlaying approach, (at least) the bilinear shape functions couple to all higher order modes generated hierarchically on the concerned base element.

In this work, we present a framework which takes a new global-to-local view on global basis functions in the context of multi-level hanging nodes in constrained approximation. Herein, global basis functions are determined by, firstly, defining their support and, afterwards, resolving the constraints with local shape functions on their support. We show that the framework is able to construct several supports of different sizes, namely, a large and a medium support.

The new global-to-local view reduces the implementational effort compared to conventional approaches substantially. In particular, the large-support approach allows for a very easy treatment of multi-level hanging nodes. Our implementation supports \( h \)-anisotropic refinements (via bisection in one direction) and \( p \)-anisotropic refinements (via increasing the polynomial degree in one direction). An educational version of the MATLAB code implementing the large-support approach is available at https://fcmlab.cie.bgu.tum.de/redmine/projects/simplehp. We emphasize that the presented concept may be applied to other kinds of refinements and support sizes as well since we employ the general technique of connectivity matrices (see, e.g., [11, 15]) which connect global basis functions and local shape functions on each element.

We investigate by numerical experiments the performance of the different supports. We demonstrate that the difference in terms of convergence in the energy norm, condition number and solvability of the resulting systems is marginal. This leads us to conclude that the simple algorithmic framework for the large support is a good trade-off between accuracy and implementational complexity.

This work is organized as follows. Section 2 explains a generic assembly procedure in the context of constrained approximation. After a general introduction, the new framework is derived and explained in detail in Section 3. We describe the computation of constraints coefficients and elaborate on the definition of an appropriate support, yielding three different support sizes. Section 4 demonstrates our new approach on four different examples: a problem with a non-smooth solution, a graded-material benchmark, an example with anisotropic \( h \)-refinements containing a cascade of multi-level hanging nodes, and an example with anisotropic \( p \)-refinements. We demonstrate that the convergence properties of the easier, large-support approach are comparable to the algorithmically more involved approaches, such as the conventional approach yielding the minimal support. These and other conclusions are drawn in the closing Section 5.

2. Generic assembly procedure

In the finite element method, the analytical solution \( u \) of a partial differential equation is approximated by a numerical solution \( u_h \) that can be represented as a linear combination of global basis functions \( \phi_i \), i.e.,

\[
  u_h = \sum_i \hat{u}_i \phi_i, \quad (1)
\]

where \( \hat{u}_i \) are the degrees of freedom that have to be computed.

Within this process, it is common practice to compose these global basis functions \( \phi_i \) by element shape functions \( \xi^E_j \) defined on the element \( E \) of a finite element mesh \( \hat{E} \). Typically, these element shape functions are defined using standard element shape functions \( \hat{\xi}_j \). These are defined on a reference element \( \hat{E} \) which is transformed into the physical space by a geometric mapping \( \Psi_E : \hat{E} \rightarrow E \). The local element shape function on element \( E \) is thus given by

\[
  \xi^E_j = \hat{\xi}_j \circ \Psi_E^{-1}. \quad (2)
\]
The mapping between the global and local shape functions is expressed by a connectivity matrix \( \pi^E \) of an element \( E \). In this matrix, every row corresponds to one global shape function \( \phi_i \) and every column to one local shape function \( \xi^E_{E_j} \) of the respective element. This allows to express each global basis function as a linear combination of local shape functions:

\[
\phi_i|_E = \sum_j \pi^E_{ij} \xi^E_{j}
\]  

In Fig. 1b this connectivity matrix is depicted for the third element of a 1D finite element example.

We briefly explain how the assembly process is performed using the connectivity matrices. For this purpose, let the weak form of the considered problem read

\[
\text{Find } u \in V(\Omega) \text{ such that } a(v,u) = f(v) \quad \forall v \in V(\Omega),
\]

with \( V \) denoting the space spanned by the global shape functions \( \phi_i \), a bilinear form \( a(v,w) = \sum_{E \in \mathcal{E}} a_E(v,w) \), and a linear form \( f(v) = \sum_{E \in \mathcal{E}} f_E(v) \). Using the Einstein summation convention, the linear functional on the right hand side can then be written in the following form:

\[
f(\phi_i) = \sum_{E \in \mathcal{E}} f_E(\pi^E_{ij}\xi^E_{j}) = \sum_{E \in \mathcal{E}} \pi^E_{ij} f_E(\xi^E_{j})
\]

Thus, the global load vector \( F \) is assembled from the local element force vectors \( F^E \) by the matrix vector multiplication

\[
F = \sum_{E \in \mathcal{E}} \pi^E F^E,
\]

with \( F_i = f(\phi_i) \) and \( F^E_j = f_E(\xi^E_{j}) \). A similar transformation is applied to the bilinear form:

\[
a(\phi_i, \phi_j) = \sum_{E \in \mathcal{E}} a_E(\pi^E_{ik}\xi^E_{k}, \pi^E_{j\ell}\xi^E_{\ell}) = \sum_{E \in \mathcal{E}} \pi^E_{ik} a_E(\xi^E_{k}, \xi^E_{\ell}) \pi^E_{j\ell}
\]

Therefore, the global stiffness matrix \( K \) is assembled from the local element stiffness matrices \( K^E \) by the
following matrix multiplications:

\[ K = \sum_{E \in E} \pi^E K^E \left( \pi^E \right)^T, \quad (8) \]

with \( K_{ij} = a(\phi_i, \phi_j) \) and \( K_{k\ell}^E = a_E(\xi^E_k, \xi^E_{\ell}). \)

In the context of standard finite elements on regular meshes, the connectivity between the global shape function \( \phi_i \) and the local element shape function \( \xi^E_j \) is one-to-one. This means that on every element \( E \), each global shape function \( \phi_i \) is represented by exactly one local shape function \( \xi^E_j \), i.e.,

\[ \phi_i|_E = \xi^E_j, \quad (9) \]

whereby \( \phi_i|_E \) is the restriction of the global shape function \( \phi_i \) to the element \( E \). In the simple one-dimensional setting considered in Fig. [13] for example, the global shape function \( \phi_3 \) is equal to the first element shape function on the third element, i.e., \( \xi^E_3 \).

This simple structure of the connectivity is also reflected in the matrix \( \pi^E \), as the entries of the matrix can only be either one or zero. Furthermore, every row of the matrix can only contain one non-zero entry, as the mapping is one-to-one. These characteristics allow to transform the connectivity matrix into an equivalent, one-dimensional mapping array that is then used in the assembly processes of virtually any finite element program (see, e.g., [3]).

Unfortunately, this simple nature of the one-to-one connectivity is lost when the discretization also allows for hanging nodes. This becomes immediately evident when considering the simple scenario depicted in Fig. [14]. Here, the global shape function \( \phi_2 \) has no directly corresponding local shape function in the second (and third) element for which condition Eq. (9) holds. Instead, this global shape function has to be re-constructed by combining the first and fourth element shape functions of the second element with a weight of 1 and 1/2, respectively. Accordingly, the second row of the connectivity matrix reads:

\[ \left( \pi^E_{2j} \right)_{j=1, \ldots, 4} = \begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \end{pmatrix} \quad (10) \]

Clearly, this connectivity is no longer one-to-one. Thus, the usual simplification to an one-dimensional mapping array is not applicable in case of irregular meshes. Instead, the generic assembly procedure formalized in Eq. (3) has to be applied to ensure continuity of the global shape functions. Therefore, an essential task in case of irregular meshes is to determine the coefficients in \( \pi^E \).

There are numerous possibilities of computing \( \pi^E \) and the choice of which way to go depends on user-defined requirements such as the simplicity of the computation and data structures, the choice of local basis functions (see e.g. [6, 11, 16]), and, as we will discuss, the size of the support of the global basis functions \( \phi_i \).

3. An easy treatment of hanging nodes

Usually, the approaches for computing a connectivity matrix follow a local-to-global strategy, meaning that they determine all global shape functions to which one local shape function contributes. The connectivity matrix is thus filled column by column.

In this work, we propose a global-to-local approach instead which determines all local shape functions that re-construct one global shape function. Thus, the new approach we describe determines an entire row of the connectivity matrices at once and, therefore, we fill the connectivity matrix row by row.

An important idea of this new approach is to recreate a one-to-one correspondence between global and local shape functions by grouping multiple elements \( E_k \) into a quadrilateral \( Q_j \), i.e.,

\[ Q_j = \bigcup_k E_k. \quad (11) \]
(a) On the quadrilateral $Q_j$, it holds that $\phi_i|_{Q_j} = \xi^Q_{j\ell}$ for the local basis function $\xi^Q_{j\ell}$ corresponding to the lower left vertex of $Q_j$.

(b) After refining $Q_j$ with $E_k$, the equality $\phi_i|_{Q_j} = \xi^Q_k$ still holds.

Figure 2: Visualization of the global basis function $\phi_i$ (assigned to a vertex).

The global shape function is then chosen such that it can be represented by a single standard finite element shape function on $Q_j$:

$$\phi_i|_{Q_j} = \xi^Q_j$$

(12)

The union of $Q_j$ associated to one global basis function $\phi_i$ then defines the support of this function:

$$S = \bigcup_j Q_j$$

(13)

We call a quadrilateral $Q_j$ a sub-support of $S$. The corresponding situation is depicted in Fig. 2b. We emphasize that, although now the global basis function $\phi_i$ corresponds to exactly one local basis function $\xi^Q_{j\ell}$ on each $Q_j$, we need to express it in terms of shape functions on elements in the finite element mesh. Thus, constraining $\xi^Q_{j\ell}$ to $E_k$, we may write

$$\xi^Q_{j\ell}|_{E_k} = \sum_m \alpha_{\ell m} \xi^E_{km}.$$  

(14)

The coefficients $\alpha_{\ell m}$ of the linear combination are called constraints coefficients. Thus, the constraints coefficients express the connection of local shape functions $\xi^Q_{j\ell}$ and $\xi^E_{km}$. Note that these coefficients may differ from the entries in the connectivity matrices $\pi$ from Eq. (3), which express the connection of global shape functions and local shape functions. However, in the novel easy approach which we will present, the entries of $\pi^E_{km}$ are the same as the respective constraints coefficients $\alpha_{\ell m}$, i.e.,

$$\pi^E_{km} = \alpha_{\ell m}$$

(15)

This does not hold for other approaches, e.g., the conventional approach which will be described in Section 3.5 in Appendix B.

In a nutshell, the proposed approach is a two-stage algorithm: We have to find the quadrilaterals that form the support and we have to fill the connectivity matrices using the constraints coefficients. The next section describes how the constraints coefficients can be determined assuming a given quadrilateral $Q_j$. Section 3.3 then addresses different ways how the quadrilateral $Q_j$ can be found.

3.1. Computation of constraints coefficients

The task in this section is to derive a methodology to compute the coefficients of $\alpha_{\ell m}$ assuming a given sub-support $Q_j$. However, we face the problem that the global shape function $\phi_i$ is not explicitly known. As we will show, an explicit definition is not necessary to derive the connectivity constraints. To this end, we consider the situation depicted in Fig. 3. It describes the generalization of the composition depicted in Fig. 2b.
Figure 3: Exemplary depiction of the support of a vertex basis function in an irregular mesh

We distinguish between three spaces: The global mesh is defined in the physical space $\Omega$ where the global basis functions $\phi_i$ are defined on a support $S$. As discussed above, this support is composed of several quadrilateral sub-supports $Q_j$ as given by Eq. (13). It is assumed that each quadrilateral is defined in terms of a parameter space of the sub-support $\hat{Q}$ and a corresponding mapping into $\Psi_{Q_j} : \hat{Q} \rightarrow Q_j$. Likewise, every element $E_k$ is defined in terms of a parameter space of the element $E$ called $\hat{E}$ and a mapping $\Psi_{E_k} : \hat{E} \rightarrow E_k$. Using the parameter space of the sub-support $Q_j$ and the mapping $\Psi_{Q_j}$, the left expression of Eq. (14) can be reformulated in terms of the standard element shape functions as follows:

$$\xi_{Q_j}^{|E_k} = \xi_{E_k} \circ \Psi_{Q_j}^{-1} \big|_{E_k}$$

(16)

In the same way, the right-hand side can be expressed in terms of the standard element shape functions using the mapping of the element

$$\xi_{E_k} = \xi_{E_k} \circ \Psi_{E_k}^{-1}.$$ 

(17)

It is now possible to rewrite Eq. (14) substituting Eq. (16) and Eq. (17):

$$\xi_{E_k} \circ \Psi_{Q_j}^{-1} \big|_{E_k} = \sum_m \alpha_{lm} \xi_{E_k} \circ \Psi_{E_k}^{-1}$$

(18)

In point-wise notation for each $(x_1, x_2) \in E_k$, the above expression reads

$$\forall (x_1, x_2) \in E_k \quad \xi_{E_k} \circ \Psi_{Q_j}^{-1}(x_1, x_2) = \sum_m \alpha_{lm} \xi_{E_k} \circ \Psi_{E_k}^{-1}(x_1, x_2).$$

By substituting $(x_1, x_2)$ using the transformation $\Psi_{E_k}$ as $(\hat{x}_1, \hat{x}_2) = \Psi_{E_k}(x_1, x_2)$ and, equivalently, $(\hat{x}_1, \hat{x}_2) = \Psi_{Q_j}(x_1, x_2)$. 

6
\[ \Psi^{-1}_{E_k}(x_1, x_2), \text{ we can write} \]
\[ \forall(\hat{x}_1, \hat{x}_2) \in \hat{E}_k, \quad \xi_\ell \circ \Psi^{-1}_{Q_j}(\Psi_{E_k}(\hat{x}_1, \hat{x}_2)) = \sum_{m} \alpha_{\ell m} \hat{\xi}_m(\hat{x}_1, \hat{x}_2) \]

or, equivalently,
\[ \xi_\ell \circ \Psi^{-1}_{Q_j} \circ \Psi_{E_k} = \sum_{m} \alpha_{\ell m} \hat{\xi}_m. \] (19)

It is essential to note that this last expression defines a condition on the sought-for constraints coefficients \( \alpha_{\ell m} \) without an explicit definition of the global shape function \( \phi_i \). Instead, the condition is stated completely in terms of standard element shape functions \( \xi \) and the two mappings \( \Psi_{Q_j} \) and \( \Psi_{E_k} \). By combining these two mappings as
\[ \Upsilon^{Q,E} = \Psi^{-1}_{Q_j} \circ \Psi_{E_k}. \] (20)

Eq. (19) can be rewritten in the more compact form
\[ \xi_\ell \circ \Upsilon^{Q,E} = \sum_{m} \alpha_{\ell m} \hat{\xi}_m. \] (21)

In the most general case, the constraints coefficients \( \alpha_{\ell m} \) can be determined by collocating the conditions from Eq. (21) at a sufficient number of points in \( \hat{E} \) and solving the resulting system of linear equations. However, we want to apply a more efficient and numerically stable approach which computes the constraints coefficients explicitly. To this end, two assumptions are sufficient: Firstly, we assume that the mapping \( \Upsilon^{Q,E} \) is paraxial and, secondly, that tensor product shape functions of integrated Legendre polynomials are used as element shape functions. We explain these two assumptions in the following.

The first assumption requires the image \( \Upsilon(\hat{E}) \) to be a paraxial subset of \( \hat{E} \), see Fig. 3. Thus, the mapping \( \Upsilon \) is given by
\[ \Upsilon^{Q,E}(\hat{x}) = \text{diag}(a^{Q,E}) \hat{x} + b^{Q,E}, \quad \text{where} \]
\[ a^{Q,E} := \frac{1}{2} \left( \Psi^{-1}_{Q_j}(\Psi_{E}(1,1)) - \Psi^{-1}_{Q_j}(\Psi_{E}(-1,-1)) \right), \]
\[ b^{Q,E} := \frac{1}{2} \left( \Psi^{-1}_{Q_j}(\Psi_{E}(1,1)) + \Psi^{-1}_{Q_j}(\Psi_{E}(-1,-1)) \right). \] (22)

It is clear that this definition of \( \Upsilon^{Q,E} \) imposes conditions on the element transformations \( \Psi_{Q_j}, \Psi_{E} \) and, thus, on the types of possible mesh refinements. In particular, refinements via bisection with the same division ratios on opposite edges are possible.

As a second assumption, we require a tensor-product structure for the element shape functions. For the local basis functions on the element \( E \) in \( \Omega \), we define
\[ \hat{\xi}_j^{E}(\hat{x}_1, \hat{x}_2) := h_{L_k,1}^{p_1}(\hat{x}_1) h_{L_k,2}^{p_2}(\hat{x}_2) \quad \text{for} \quad (\hat{x}_1, \hat{x}_2) \in [-1, 1]^{2}, \] (23)
with a suitable numbering \( L^E \in \mathbb{N}^{n_E \times 2} \), where \( n_E \) denotes the number of local shape functions on \( E \). By \( (p_1, p_2) \) we denote the polynomial degree assigned to \( E \). The polynomials \( \{\hat{\eta}_{m,i}^{E}\}_{0 \leq m \leq p} \) in Eq. (23) are based on the hierarchical integrated Legendre polynomials \( \{\ell_m\}_{m \in \mathbb{N}_0} \) defined by the recursion
\[ \ell_0(\hat{x}) := 1, \quad \ell_1(\hat{x}) := -\hat{x}, \]
\[ (m+1)\ell_{m+1}(\hat{x}) = 2(m-1/2)\hat{x}\ell_m(\hat{x}) - (m-2)\ell_{m-1}(\hat{x}), \quad m \geq 1. \]

7
Then, the polynomials are defined as
\[
\hat{\eta}_0(\hat{x}) := \frac{1}{2}(1 - \hat{x}), \quad \hat{\eta}_1(\hat{x}) := \frac{1}{2}(1 + \hat{x}), \quad \hat{\eta}_m(\hat{x}) := \ell_m(\hat{x}), \text{ for } m = 2, \ldots, p_i,
\]
see, e.g., [17]. Note that the definition of \(\hat{\eta}_m\) is independent of the polynomial degree \(p_i\) as the polynomials are hierarchical. Thus, we may omit the upper indices and write \(\hat{\eta}_{p_i}^m = \hat{\eta}_m\). We choose the integrated Legendre polynomials because they are orthonormal under the bilinear form \((\nabla \hat{\eta}_m, \nabla \hat{\eta}_n)\) for \(p > 1\) in the interval \([-1, 1]\) which leads to well conditioned systems of equations in the finite element setting. An in-depth discussion of their favorable properties may be found, for example, in [15].

Shape functions are associated to the vertices, edges, and the interior of \(\hat{E}\) and, therefore, are commonly referred to as vertex, edge, and inner modes, respectively. We use the numbering \(L_E\) from Eq. (23) to formalize this association and to enumerate the shape functions on each element \(E\). It is clear that this mapping depends on the definition of the polynomials \(\hat{\eta}_m\) and on the number \(n_E\) of shape functions on \(E\). If the full tensor product space with \(n_E = (p_1 + 1)(p_2 + 1)\) is used, then for the integrated Legendre polynomials \(\hat{\eta}_m\), the mapping \(L_E\) may be concisely written as the \(n_E \times 2\)-matrix \((V E I)^\top\), where

\[
V = \begin{pmatrix}
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{pmatrix},
E = \begin{pmatrix}
2 & \cdots & p_1 & 1 & \cdots & 1 & 2 & \cdots & p_1 & 0 & \cdots & 0 \\
0 & \cdots & 0 & 2 & \cdots & p_2 & 1 & \cdots & 1 & 2 & \cdots & p_2
\end{pmatrix},
I = \begin{pmatrix}
2 & \cdots & 2 & \cdots & p_1 & \cdots & p_1 \\
2 & \cdots & 2 & \cdots & p_2 & \cdots & p_2
\end{pmatrix}.
\]

The submatrices \(V\), \(E\), and \(I\) generate the vertex, edge, and inner modes, respectively, as visualized in Fig. 4.

We mention that this approach also works for the reduced set of shape functions, called trunk space [19] or serendipity shape functions. For details on the necessary modifications, see [17].

![Figure 4: Visualization of the association of shape functions to vertices, edges, and the interior of \(\hat{E}\) by the matrix \(L_E\).](image-url)

The tensor-product structure of the shape functions allows for a component-wise computation of constraint coefficients in the following manner: Let \(i \in \{1, 2\}\) and \(\beta_{rs}\) be the one-dimensional constraint coefficients satisfying

\[
\bar{\eta}_{L_{Q_i}} \circ \Upsilon_{Q,E} = \sum_{s=0}^{p_i} \beta_{L_{Q_i}^s} \hat{\eta}_s,
\]

where \(\Upsilon_{Q,E}^i\) denotes the \(i\)th component of \(\Upsilon_{Q,E}\) as defined by Eq. (22). Then, the constraint coefficients
$\alpha_{\ell m}$ for $\mathbf{Y}^{Q,E}$ are given by

$$\alpha_{\ell m} = \prod_{i=1}^{2} \beta_{L_{i}^{Q,E}_{\ell m}},$$  \hspace{1cm} \text{(26)}$$

where the values $\beta$ additionally depend on the mapping $\mathbf{Y}^{Q,E}$. This can be seen by observing

$$\xi_{\ell_{i}}^{Q} \circ \mathbf{Y}^{Q,E}(\hat{x}_{1}, \hat{x}_{2}) = \prod_{i=1}^{2} \eta_{L_{i}^{Q}_{\ell}} \circ \mathbf{Y}_{i}^{Q,E}(\hat{x}_{i})$$

$$= \prod_{i=1}^{2} \sum_{s=0}^{p_{i}} \beta_{L_{i}^{Q}_{\ell_{i}}^{\eta}} \hat{y}_{s}(x_{i})$$

$$= \sum_{s_{1}=0}^{p_{1}} \sum_{s_{2}=0}^{p_{2}} \left( \prod_{i=1}^{2} \beta_{L_{i}^{Q}_{\ell_{i}}^{\eta}} \right) \left( \prod_{i=1}^{2} \hat{y}_{s_{i}}(\hat{x}_{i}) \right)$$

$$= \sum_{m=1}^{n_{E}} \left( \prod_{i=1}^{2} \beta_{L_{i}^{Q}_{\ell_{i}}^{\eta}} \right) \xi_{L_{m}^{\eta}}(\hat{x}_{1}, \hat{x}_{2}).$$

where in the last step, we used the numbering $L_{E}$ to get $L_{E} = (s_{1}, s_{2})$.

This simplification now allows to collocate Eq. \text{(25)} for the one-dimensional constraints coefficients $\beta_{\ell_{i}}$ instead of the two-dimensional coefficients as in Eq. \text{(21)}. Together with the restriction to a paraxial mapping, this allows for an explicit computation of $\beta_{\ell_{i}}$.

To simplify the notation, let $i \in \{1, 2\}$ and $a := a_{i}^{Q,E}$, $b := b_{i}^{Q,E}$ as defined in Eq. \text{(22)}. Then, the one-dimensional constraints coefficients $\beta_{\ell_{i}}$ can be computed explicitly by the following recursion \textcolor{red}{[17]}:

$$\begin{align*}
\beta_{00} &= (1 + a - b)/2, & \beta_{01} &= (1 - a - b)/2, \\
\beta_{10} &= (1 - a + b)/2, & \beta_{11} &= (1 + a + b)/2, \\
\beta_{0s} &= \beta_{1s} = 0 & \text{for } s \in \{2, \ldots, p_{i}\}; \\
\beta_{20} &= \left(1 - (a - b)^{2}\right)/2, & \beta_{21} &= \left(1 - (a + b)^{2}\right)/2, & \beta_{22} &= a^{2}, \\
\beta_{2s} &= 0 & \text{for } s \in \{3, \ldots, p_{i}\}.
\end{align*}$$  \hspace{1cm} \text{(27)}

For $r \in \{2, \ldots, p_{i} - 1\}$, one obtains

$$\begin{align*}
\beta_{r+1,0} &= \frac{1}{r+1} ((2r - 1)\beta_{r,0}(b - a) - (r - 2)\beta_{r-1,0}), \\
\beta_{r+1,1} &= \frac{1}{r+1} ((2r - 1)\beta_{r,1}(a + b) - (r - 2)\beta_{r-1,1}), \\
\beta_{r+1,2} &= \frac{1}{r+1} \left( (2r - 1) \left( a\beta_{r,2} + \frac{1}{2}a\beta_{r,0} - \beta_{r,1} + b\beta_{r,2} \right) - (r - 2)\beta_{r-1,2} \right), \\
\beta_{r+1,s} &= \frac{1}{r+1} \left( (2r - 1) \left( a\beta_{r,s-1} + \frac{s}{2} \beta_{r,s} - \beta_{r,s-1} + s \beta_{r,s} - (s - 1)\beta_{r-1,s} \right) \right) & \text{for } s \in \{3, \ldots, r - 1\}, \\
\beta_{r+1,r} &= \frac{2r - 1}{r+1} \left( a\beta_{r,r-1} + \frac{s}{2r - 3} + b\beta_{r,r} \right), \\
\beta_{r+1,r+1} &= a\beta_{r,r}, \\
\beta_{tu} &= 0 & \text{for } t \in \{2, \ldots, p_{i} - 1\}, u \in \{m + 1, \ldots, p_{i}\}.
\end{align*}$$

We mention that one may also use Lagrange polynomials as the shape functions. The constraints coefficients may then be determined numerically stably using a simple duality relation. For details, we refer
3.2. Example

We illustrate the computation of the constraints coefficients by a simple example. Consider the quadrilateral $Q := [-1,1]^2$ and the element $E := [0,1/2] \times [-1/2,0] \subseteq Q$ in the finite element mesh depicted in Fig. 5 which consists of seven elements. We set the polynomial degrees of $Q$ and $E$ equal to $(2,2)$. Thus, the number of shape functions on $E$ is $n_E = 9$ and the mappings $L_Q$ and $L_E$ result in the $9 \times 2$-matrix

$$L_Q = \begin{pmatrix}
0 & 1 & 1 & 0 & 2 & 1 & 2 & 0 & 2 \\
0 & 0 & 1 & 1 & 0 & 2 & 1 & 2 & 2
\end{pmatrix}^\top = L_E.$$  

The transformation $Y^{Q,E}$ can be written as

$$Y^{Q,E}(\hat{x}) = \text{diag}(a^{Q,E})\hat{x} + b^{Q,E} = \begin{pmatrix}
\frac{1}{4} & 0 \\
0 & \frac{1}{4}
\end{pmatrix} \hat{x} + \begin{pmatrix}
\frac{1}{4} \\
\frac{1}{4}
\end{pmatrix} = \begin{pmatrix}
\frac{1}{4}\hat{x}_1 + \frac{1}{4} \\
\frac{1}{4}\hat{x}_2 - \frac{1}{4}
\end{pmatrix}.$$  

Assuming that the global basis function $\phi$ is assigned to the lower-left vertex of $Q$, we expect $\phi|_E = \frac{3}{8}\xi_1^E + \frac{3}{16}\xi_2^E + \frac{1}{8}\xi_3^E + \frac{1}{4}\xi_4^E$. We aim to find coefficients $\alpha_{1j}$ such that

$$\phi|_E = \xi_1^Q|_E = \sum_{j=1}^9 \alpha_{1j}\xi_j^E.$$  

Figure 5: Visualization of the global basis function $\phi = \xi_1^Q$ to $[11]$. 

10
To compute $\alpha_{1j}$, we calculate $\beta_{L_i^0 L_j^0}$ for $j \in \{1, \ldots, 9\}$, which yields

$$
\alpha_{11} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{00}(a_1, b_1) \cdot \beta_{00}(a_2, b_2) = \frac{1}{2} \left( 1 + \frac{1}{4} - \frac{1}{4} \right) \cdot \frac{1}{2} \left( 1 + \frac{1}{4} + \frac{1}{4} \right) = \frac{3}{8},
$$

$$
\alpha_{12} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{01}(a_1, b_1) \cdot \beta_{00}(a_2, b_2) = \frac{1}{2} \left( 1 - \frac{1}{4} - \frac{1}{4} \right) \cdot \frac{1}{2} \left( 1 + \frac{1}{4} + \frac{1}{4} \right) = \frac{3}{16},
$$

$$
\alpha_{13} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{00}(a_1, b_1) \cdot \beta_{01}(a_2, b_2) = \frac{1}{2} \left( 1 - \frac{1}{4} - \frac{1}{4} \right) \cdot \frac{1}{2} \left( 1 - \frac{1}{4} + \frac{1}{4} \right) = \frac{1}{8},
$$

$$
\alpha_{14} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{01}(a_1, b_1) \cdot \beta_{01}(a_2, b_2) = \frac{1}{2} \left( 1 + \frac{1}{4} - \frac{1}{4} \right) \cdot \frac{1}{2} \left( 1 - \frac{1}{4} + \frac{1}{4} \right) = \frac{1}{4},
$$

$$
\alpha_{15} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{12}(a_1, b_1) \cdot \beta_{00}(a_2, b_2) = 0,
$$

$$
\alpha_{16} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{01}(a_1, b_1) \cdot \beta_{02}(a_2, b_2) = 0,
$$

$$
\alpha_{17} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{02}(a_1, b_1) \cdot \beta_{01}(a_2, b_2) = 0,
$$

$$
\alpha_{18} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{00}(a_1, b_1) \cdot \beta_{02}(a_2, b_2) = 0,
$$

$$
\alpha_{19} = \prod_{i=1}^{2} \beta_{L_i^0 L_j^0}(a_i, b_i) = \beta_{02}(a_1, b_1) \cdot \beta_{02}(a_2, b_2) = 0,
$$

where $\beta_{03} = 0$ according to Eq. (27). These coefficients match the $\alpha_{1j}$ expected from Fig. 5.

### 3.3. A new approach for constructing the support

The previous sections were dedicated to the computation of the constraints coefficients $\alpha_{\ell m}$ assuming a given sub-support $S_j$. This section is devoted to the construction of the sub-support $S_j$, which, according to Eq. (13), composes the support $S$ of the global shape functions. Naturally, the aim is to minimize the support of the shape functions in order to minimize the coupling of different basis functions and, therefore, the number of entries in the stiffness matrix. Nevertheless, easier approaches with larger support sizes are possible.

In this section, we present our new approach, which leads to two different supports: a large and a medium support as depicted in Fig. 6 for a global basis function for an edge. We describe the connection of our new approach to the multi-level $hp$ approach from [13], and to the conventional approach, which yields the minimal support of a global basis function, in the subsequent sections.

![Figure 6: Visualization of a mesh with several refinements and a global basis function associated to an edge. In this case, the two supports resulting from the approach differ.](image)
Our approach for constructing a support $S$ of the global shape function $\phi_i$ is to group elements $E$ into sub-supports $Q_j$ that do not overlap: In this case, (i) pairs of $Q_j$ are either disjoint or share a vertex $v$ or an edge $e$, and (ii) the topological component $t_i$ to which the global shape function $\phi_i$ is associated is part of each $Q_j$. We call a support that fulfills the conditions (i) and (ii) an admissible quadrilateral support of the shape function $\phi_i$, see Fig. 7 for an example.

Therefore, the one-to-one correspondence between the global shape function $\phi_i$ and a local shape function $\xi_{Q_j}$ in the sense of Eq. (12) is re-established. Thus, the constraints coefficients $\alpha_{\ell m}$ can be computed for each $E_k \in \mathcal{E}$ with $E_k \subseteq Q_j$ and are exactly the entries in the connectivity matrix in the sense that $\pi_{\ell m} = \alpha_{\ell m}$.

In the following paragraphs, we describe two different supports resulting from applying the new approach.

**Large support.** The simplest support consists of the set of quadrilaterals $Q$ formed by the union of the original, non-refined quadrilaterals $Q$ attached to the topological component associated to the global basis function $\phi_i$. A more concise definition may be given by use of the refinement history $\mathcal{H}$, which is the set of all elements created at any point of time during the refinement process, as follows: For a basis function $\phi_i$ associated to a topological component $t_i$ let $\tilde{S}$ be the union of elements $E$ in $\mathcal{H}$ that contain $t_i$ and have no ancestor in $\mathcal{H}$ that also contains $t_i$. Then, we define the support $S$ as all in-mesh descendants of elements in $\tilde{S}$. See Fig. 8 for an example of a global basis function for an edge.

**Remark.** The above description of the large support simply treats all topological components in the same way. As a consequence, the large support of a vertex remains the same during the refinement process, which leads to unnecessarily large supports in, e.g., regular meshes, as visualized in Fig. 9a. To overcome this issue, we may define the support of a vertex as the union of all large supports of incident regular edges, see Fig. 9b.

Figure 7: Visualization of an admissible quadrilateral support for a global basis function for $e$ consisting of two quadrilaterals

Figure 8: The large support constructed for the global basis function for $e$ consists of all in-mesh descendants of the quadrilaterals $Q_1, Q_2$.

Figure 9: Visualization of two variants of the large support of a vertex
Medium support. The large support presented previously constructs supports with a unnecessarily high number of mesh elements, however, it is easy to implement since the elements chosen from $\mathcal{H}$ automatically form an admissible quadrilateral support.

Let $t_i$ be a topological component associated to the global shape function $\phi_i$. The medium support of this global basis function $\phi_i$ is defined as the smallest admissible quadrilateral support $S$ for $t_i$ in the sense that there is no other admissible quadrilateral support $S'$ for $t_i$ with $S' \subseteq S$. Obviously, this will lead to a support that is at most as large as the large support.

Whereas the computation of the large support should be very easy in any finite element codes where the refinement history is available, a suitable algorithm for computing the medium support is more involved. In Appendix A, we provide an algorithm that supports refinements stemming from paraxial mappings as given by Eq. (22).

3.4. Multi-level hp method

The large support constructed above is the same as the support constructed by the multi-level hp method as presented in [13], where only $h$-isotropic and $p$-isotropic refinements are supported. The underlying idea of this approach is to prevent hanging nodes by construction. To this end, $h$-refinement is achieved by superposing the concerned element hierarchically with new elements in a tree-like fashion as depicted in Fig. 10. $C^0$-compatibility is ensured by completely deactivating degrees of freedom on the boundary of the overlays. Linear independence is guaranteed by deactivating all degrees of freedom associated to topological components with active sub-components. The multi-level hp method and the large-support approach presented in this work create the same global basis functions associated with the same topological components and, thus, create the same finite element spaces.

![Figure 10: Visualization of the large support for an edge, consisting of all active faces as depicted in Fig. 8, but constructed by the multi-level hp method [13]](image)

3.5. Conventional approach (minimal support)

In our new approach, we confine ourselves to admissible quadrilateral supports as this allows for a direct computation of the rows of the connectivity matrices. However, the size of the support can be reduced further if the restriction to admissible quadrilateral supports is lifted.
The conventional approach (see, e.g., [10, 11]) computes global basis functions with minimal support and, thus, minimizes the coupling between the degrees of freedom. See Fig. 11 for a visualization of the minimal support for an edge basis function. While the conventional approach is usually described and implemented from a bottom-up point of view, it is possible to fit it into the top-down view employed in this work using the techniques of Section 3.1 and Section 3.3 where the connectivity matrices are filled row by row. A description of the top-down version of the algorithm is given in Appendix B.

4. Numerical experiments

In this section, the numerical behavior of our approach, which yields large and medium support sizes as explained in Section 3.3, compared to the conventional approach yielding minimal support sizes is examined. To this end, a MATLAB implementation of the approaches has been developed which is used for solving the problems in the following subsections. The attached code contains a simple implementation of the approach yielding the large support.

The first problem, discussed in Section 4.1, demonstrates the applicability of the three approaches to the Poisson problem on an L-shaped domain which is a standard non-smooth benchmark problem. A second non-smooth example is then presented in Section 4.2 which has relevance to a practical application as it imitates the typical situation in welding or laser sintering processes. To this end, strong local gradients internal to the computational domain have to be resolved around which the material properties change rapidly as well. The third problem features $h$-anisotropy and multi-level hanging nodes. In the fourth problem, $p$-anisotropy is demonstrated.

In the first three examples, we compare the effect of the different support sizes with respect to key figures, namely, the error in the energy norm, the condition number, the number of non-zeros in the resulting linear system, and the number of iterations a standard solver has to perform. The fourth example demonstrates the capabilities of the large support in the context of $hp$-adaptive refinements.

4.1. Non-smooth problem

The aim of this first example is to analyze the convergence properties of the proposed refinement methods in the context of non-smooth problems. For this purpose, the well-known L-shaped domain problem is considered (see e.g. [20]).
Problem definition. The Poisson problem

$$\Delta u = 0 \quad \forall (r, \theta) \in \Omega$$

defined on the L-shaped domain depicted in Fig. 12a, together with the boundary conditions defined as

$$u = 0 \quad \forall (r, \theta) \in \Gamma_D$$

$$\nabla u \cdot n = \frac{2}{3} r^{-\frac{2}{3}} \left( x \sin \left( \frac{2}{3} \theta \right) - y \cos \left( \frac{2}{3} \theta \right) \right) \cdot n \quad \forall (r, \theta) \in \Gamma_N.$$

Therein, $\Gamma_D$ and $\Gamma_N$ define the Dirichlet and Neumann boundaries, respectively, such that

$$\Gamma_D \cup \Gamma_N = \partial \Omega \quad \text{and} \quad \Gamma_D \cap \Gamma_N = \emptyset.$$

As shown in e.g. [2], the analytical solution $u$ can be given in polar coordinates $r$ and $\theta$ as

$$u(r, \theta) = r^\frac{2}{3} \sin \left( \frac{2}{3} \theta \right) \quad \forall (r, \theta) \in \Omega.$$

Numerical results. For computing an approximation to the solution $u$ using $hp$ finite elements, the domain is discretized using three square mesh elements of size 1. To this base mesh, 14 steps of refinement are applied. As a refinement strategy, a geometric mesh in combination with decreasing element degrees towards the singularity is used, as depicted in Fig. 13a. The three elements sharing the origin are assigned the polynomial degree 1. The strategy is known to yield exponential convergence of the relative energy error

$$\|e\| = \sqrt{E_{\text{EX}} - E_{\text{FE}} / E_{\text{EX}}},$$

where $E_{\text{EX}}$ and $E_{\text{FE}}$ denote the exact and the approximated energy, respectively [4].

The same refinement strategy is applied to a second mesh consisting of three elements, where the refinements are not paraxial, yielding a skewed mesh consisting of general convex quadrilaterals, see Fig. 13b. In case of this skewed mesh, we provide the results only for the large- and minimum-support approach since the algorithm for the medium-support approach as described in Appendix A is not applicable since the refinements are not paraxial.

![Visualization of the distribution of polynomial degrees on the orthogonal mesh after 4 refinements](image1)

(a) Visualization of the distribution of polynomial degrees on the orthogonal mesh after 4 refinements

![Visualization of the skewed mesh](image2)

(b) Visualization of the skewed mesh

Figure 13: Visualization of the refinement strategy on the L-shaped domain

Indeed, the experimental results confirm the convergence characteristics predicted by theory, as the relative energy error shows exponential convergence: The refinement strategy can be shown to yield error bounds of the form $C \exp \left(-bN^{1/3}\right)$, where $N$ denotes the number of degrees of freedom [4]. Thus, the error decrease is exponential if the error depicts a straight line when plotted logarithmically against the third root of the number of degrees of freedom, as can be seen in Fig. 14a. The three approaches show no visible difference in approximation quality even though the basis has a different support.

As discussed in Section 3.3, the large-support approach results in a larger support for vertex and edge
Figure 14: Convergence results and condition numbers of the three approaches, orthogonal mesh

Figure 15: Convergence results and condition numbers of the three approaches, skewed mesh
basis functions than medium- and minimum-support approach. This leads to a tighter coupling of degrees of freedom and, thus, to denser linear system with an increased bandwidth. Therefore, the solvability of the resulting linear system might be affected since typical solvers for positive definite symmetric systems operate best on sparse matrices with a small bandwidth.

To quantify the impact on the solvability of the resulting system for each approach, the condition numbers and the numbers of non-zero elements are compared. The condition number is presented in Fig. 14b. The large support yields a marginally higher condition number than the medium and minimal approach.

Figure 16a reveals that the number of non-zero elements in the system matrix is significantly higher for the large support than for the two other approaches. Thus, as predicted, the tighter coupling between the degrees of freedom is noticeable in the resulting systems. However, the larger number of non-zero elements has no visible impact on the solvability of the system. As suggested by Fig. 16a, MATLAB’s preconditioned conjugate gradient solver `pcg` with diagonal pre-conditioning (with termination at a residual error of at most $10^{-15}$) requires only a marginally higher number of iterations in case of the large support.

![Figure 16: Number of non-zero elements in the system and the number of solver iterations, orthogonal mesh](image)

![Figure 17: Number of non-zero elements in the system and the number of solver iterations, skewed mesh](image)
We conclude that support size has only a marginal influence on the resulting energy error and the solvability of the linear system.

4.2. Graded material

The aim of this second example is to compare the three approaches on a domain with heterogeneous material properties.

Problem definition. Consider the problem $-\text{div}(c \nabla u) = f$ in $\Omega = (-1, 1)^2$ with $u = 0$ on $\partial \Omega$. The heterogeneous material properties are modeled by the modulus $c(x, y) = \exp \left(-\left(x^2 + y^2\right)\right)$ on $\Omega$. The right-hand side $f$ is chosen such that

$$u(x, y) = \exp \left(-20(x^2 + y^2)\right)(x + 1)(x - 1)(y + 1)(y - 1)$$

is a solution to the problem.

![Figure 18: Setup and solution of the problem with heterogeneous material properties](image)

(a) Analytical solution $u$

(b) Material modulus $c$

(c) Right-hand side $f$

Figure 18: Setup and solution of the problem with heterogeneous material properties

The initial mesh consists of a single element. Similarly to the L-shaped domain, this base mesh is then refined geometrically towards the point $(0, 0)$ with decreasing polynomial degrees. Figure 19 visualizes the mesh after 3 refinement steps.

![Figure 19: Visualization of the mesh after 3 refinement steps](image)

Figure 19: Visualization of the mesh after 3 refinement steps

Numerical results. In order to compare the convergence properties of the three approaches, the reduction of the relative error in the energy norm is measured. As in the case of the L-shaped domain, the employed refinement approach succeeds in delivering an exponential convergence rate, although the significant decrease of the error occurs at a later refinement step than in case of the L-shaped domain example. The large-support approach exhibits the same convergence properties as all other approaches with almost the same energy error. The results are shown in Fig. 20a.

To evaluate the large-support approach with respect to the solvability of the resulting linear system, the condition number is examined. As Fig. 20b indicates, the large-support approach, again, leads to a
slightly higher condition number. Also, the number of non-zero elements in the stiffness matrix is increased compared to the other approaches, as visualized in Fig. 21a. However, the simple MATLAB `pcg` solver (with diagonal preconditioning) performs almost an identical number of iterations to reach a relative error of $10^{-15}$, regardless of which support is used. This is an interesting observation because at first sight, one might be inclined to think that the large support leads to a much tighter coupling between the basis functions. However, this effect is not very pronounced even though the integrated Legendre polynomials lose their orthogonality property under the given operator due to the spatially varying material coefficient $c(x, y)$.

We conclude that a non-constant material coefficient does not alter the convergence and solvability behavior of the large-support approach compared to the medium- and minimal-support approaches.

4.3. Example with multi-level hanging nodes

The aim of this third example is to demonstrate the capabilities of the three approaches in treating multi-level hanging nodes.
Problem definition. Consider the problem \(-\Delta u = f\) in \(\Omega = (0, 1)^2\) with \(u = 0\) on \(\partial \Omega\). The right-hand side \(f\) is chosen such that \(u\) is the smooth solution
\[
u(x, y) = \sin(\pi x) \sin(\pi y)
\]
to the problem. This function \(u\) is depicted in Fig. 22a.

Numerical results. The domain is discretized starting with four square elements that are subsequently refined in a spiral-wise manner, where anisotropic \(h\)-refinements are used. The resulting mesh consists of hanging nodes of at least 9 levels, as visualized in Fig. 22b. For the numerical experiment, the mesh is kept fixed while \(p\)-refinement is applied by increasing the degree of all elements isotropically from 1 up to 10. Since the solution is smooth, the \(p\)-strategy is expected to yield exponential convergence. The exponential decay of the norm of the energy error is depicted in Fig. 23a and machine precision is reached with \(p = 8\). For the condition number, the number of non-zeros, and the number of iterations, the minimal-support approach performs best, however, the difference to our easy approach creating the large- and medium support is negligible. The comparison is visualized in Fig. 23b, Fig. 24a, and Fig. 24b, respectively.

4.4. Example with anisotropic \(p\)-refinements

In this final example, we aim to demonstrate the capabilities of the large-support approach applied to meshes with anisotropic \(p\)-refinements.

Problem definition. We consider the problem \(-\Delta u = f\) in \(\Omega = (0, 1)^2\) with \(u = 0\) on \(\partial \Omega\). The right-hand side \(f\) is chosen such that \(u\) is the solution
\[
u(x, y) = \arctan(60(y - 2x)) - \pi/2
\]
to the problem. The solution is smooth, but it has steep gradients along a line in the interior of \(\Omega\) and, thus, is an interior-layer problem similar to the “shock” problem analyzed in, e.g., [1].

Numerical results. We discretize the domain using a base mesh of 16 square elements of isotropic degree 2. Then, a series of global refinements is applied, where each element is refined \(h\)-isotropically, \(p\)-isotropically or \(hp\)-anisotropically according to a regularity estimation depending on the decay rate of the coefficients in the Legendre expansion of the approximate solution on the element (cf. [21]). This refinement strategy creates
Figure 23: Convergence results and condition numbers of the three approaches

Figure 24: Number of non-zero elements in the system and the number of solver iterations
anisotropic $p$ and $h$ refinements and yields exponential convergence as visible in Fig. 26 which displays the reduction of the energy error using the large-support approach. We note that, since the solution is smooth, one may use isotropic $p$-refinements to achieve exponential convergence. Figure 26 shows the result of applying isotropic $p$-refinements to a fixed $16 \times 16$-mesh which implies that the convergence rate of this isotropic refinement is lower than in the case of the adaptive refinement strategy.

Figure 26: Convergence of the relative energy error using the large-support approach

5. Conclusions

This paper presented a framework to construct a high-order polynomial basis functions for the finite element method. To this end, it takes a novel global-to-local perspective and, additionally, utilizes the concept of connectivity matrices to cope with hanging nodes and edges. The framework is unified in the sense that, using its techniques, it is able to construct the large support of the multilevel-$hp$ method first presented in [13] as well as a minimal support which emerges using the conventional $hp$-version of the finite
element method. The framework also allows for the construction of other types of supports for hp-finite elements. As an example, a newly developed medium support was presented as well.

The algorithmic complexity increases from the large support to the minimal support. We investigated the influence of the different supports on accuracy, non-zeros in the stiffness matrix, condition numbers and number of solver iterations for four benchmark problems. We found only a marginal computational advantage of the conventional, minimal support, over the algorithmically much simpler large support. Interestingly, this also holds in the case of breaking the orthogonality of the underlying basis by allowing for spatially varying material properties, skew-shaped quadrilaterals and anisotropic refinements.

An educational version of this new framework for hp-finite elements is available for download at https://fcmlab.cie.bgu.tum.de/redmine/projects/simplehp/files.

Acknowledgements

The first and the last author gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grant RA 624/27-1 and Grant SCHR 1244/4-1 of the priority programme SPP 1748.

Appendix A. Algorithm for determining the medium support

We describe an algorithm that determines the medium support $S$ for vertices and edges. Note that while the elements may be general convex quadrilaterals, the following algorithm permits only refinements described by the paraxial mapping from Eq. (22), i.e., refinements via bisection with the same division ratios on opposite edges are possible.

As the medium support consists of the union of quadrilaterals that form an admissible quadrilateral support, the basic idea is to start at a smallest possible set of such quadrilaterals and, if necessary, increase the set as little as possible.

Let $\phi_i$ be a global basis function for a vertex $v$ and let $S_L$ be the large support of $\phi_i$ from Section 3.3 consisting of quadrilaterals $Q_{Lr}$. In a similar manner, we create a list of quadrilaterals $Q_{Mr}$ that are initially empty. We determine all regular edges (i.e., edges that are associated a degree of freedom) incident to $v$ and include all their neighbors $E$ in the quadrilaterals $Q_{Mr}$ if $E \subseteq Q_{Lr}$, enlarging the quadrilaterals accordingly, as visualized in Fig. A.27a. Usually, the quadrilaterals determined so far do not automatically form an admissible quadrilateral support. Thus, we check whether each pair of quadrilaterals is either disjoint or shares an entire edge. As the quadrilaterals correspond to rectangles when transformed to the reference element, this check involves only rectangles. In case the quadrilaterals share only part of an edge, we need to enlarge the quadrilateral containing the shorter edge and include any new elements that have a non-empty intersection with the quadrilateral, as visualized in Fig. A.27b. Note that this may lead to a cascade of enlarge-and-include operations, which implies that the check for admissibility might have to be repeated.

If $\phi_i$ is a global basis function for an edge $e$, we start with two empty quadrilaterals sharing $e$ and perform enlarge-and-include operations with the neighbors of $e$. The two quadrilaterals are compatible by definition.
Appendix B. Computing the connectivity matrices using the conventional approach

We give an informal description of the top-down version of the conventional algorithm used for computing the rows of the connectivity matrices corresponding to a global basis function using similar techniques as developed in Section 3.

At first, we describe the computation of a global basis function for a vertex \( v \). Initially, consider a regular edge \( e_1 \) incident to \( v \) with its neighboring elements, i.e., a single element \( E_{\text{large1}} \) and a set of elements \( E_{11}, \ldots, E_{m1} \), as visualized in Fig. B.28a. We define \( \phi_i \) such that it equals a vertex shape function \( \xi_v \) along \( e_1 \). Then, \( \xi_v \) needs to be matched by linear combinations of shape functions at \( e_1 \) on the opposing elements \( E_{\text{large1}} \) and \( E_{11}, \ldots, E_{m1} \).

To enforce continuity along \( e_1 \) on each of its neighbors \( E \), we create the situation from Eq. (14) where a shape function on an element \( E \) is written as a linear combination of shape functions on its sub-elements \( E \subseteq Q \): For each neighbor \( E \) of \( e_1 \), we define \( Q \) to be the smallest quadrilateral containing \( e_1 \) as an edge and the element \( E \), implying that \( \xi_v := \xi_e \) is a local shape function on \( Q \), see Fig. B.28b for an example. Then, we compute the constraints coefficients \( \alpha_{\ell m} \) as in Eq. (21) and use them as temporary entries in the connectivity matrix \( \pi_E \), i.e., \( \pi_{E_{im}} = \alpha_{\ell m} \) for each \( m \). By definition of the hierarchical shape functions, only the entries \( \alpha_{\ell m} \) that correspond to shape functions along \( e_1 \) are non-zero.

Evidently, this ensures continuity of \( \phi_i \) along \( e_1 \), but not necessarily on edges shared by \( E_{11}, \ldots, E_{m1} \). Thus, we might have to modify the temporary entries in the connectivity matrices. Let \( e_2 \) be an edge that starts at a hanging node on \( e_1 \) and is shared by an element \( E_{\text{large2}} := E_{j1}, 1 \leq j1 \leq m1 \), and opposing elements \( E_{12}, \ldots, E_{m2} \), as visualized in Fig. B.28c. Consider the newly determined scaled shape function \( \alpha_v \xi_{e1} \) along \( e_2 \) from the first step. A crucial observation is that the possible discontinuity can be resolved by repeating the computational scheme for \( \alpha_v \xi_{e1} \) since \( \xi_{e1} \) is the only shape function at \( e_1 \) having support on both \( E_{\text{large2}} \) and one of the opposing elements \( E_{12}, \ldots, E_{m2} \) by definition. Moreover, for an element \( E \) at both \( e_1 \) and \( e_2 \), the entries in \( \pi_{E_{im}}^E \) affected by \( e_2 \) interfere with the entries affected by \( e_1 \) only at the entry for \( \xi_{e1} \), which implies that using the newly determined entries for \( e_2 \) does not destroy continuity at \( e_1 \).

It is noted that an edge \( e_j \) may be reached multiple times by this procedure. This may be the case, e.g., if \( e_j \) has two hanging nodes as endpoints, see Fig. B.28a for an example. As a consequence, applying the above algorithm would overwrite the coefficients computed for the neighbors when the edge is reached a second time. As a remedy, we define \( \phi_i \) on \( e_j \) to equal the sum of the two scaled vertex shape function computed at the endpoints. This has two advantages: Firstly, the corresponding constraints coefficients can be computed separately. Secondly, by definition of the shape functions, the values attained at the other endpoint remain unchanged by the addition, implying that continuity is not destroyed. As Fig. B.28b suggests, the scaled shape functions at subsequent hanging nodes change during the procedure due to the addition step.
For a vertex basis function $\phi_i$, each edge incident to $v$ has to be treated as above. Likewise, for an edge basis function, only $e$ has to be treated. Note that, in both cases, the minimal support of $\phi_i$ is constructed implicitly as the union of all mesh elements that are neighbors of any edge reached by the described procedure.

Figure B.29: Visualization of the algorithm treating an edge hanging on two vertices

For a vertex basis function $\phi_i$, each edge incident to $v$ has to be treated as above. Likewise, for an edge basis function, only $e$ has to be treated. Note that, in both cases, the minimal support of $\phi_i$ is constructed implicitly as the union of all mesh elements that are neighbors of any edge reached by the described procedure.

17. A. Schröder, Constraints coefficients in hp-fem, in: Numerical Mathematics and Advanced Applications: Proceedings of

