Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 218 (2008) 192–200
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Static condensation, partial orthogonalization of basis functions, and ILU preconditioning in the *hp*-FEM

Tomáš Vejchodský[a,*,1], Pavel Šolín[b,c,2]

[a] *Institute of Mathematics, Czech Academy of Sciences, Žitná 25, 115 67 Prague 1, Czech Republic*
[b] *Institute of Thermomechanics, Czech Academy of Sciences, Dolejškova 5, 182 00 Prague 8, Czech Republic*
[c] *Department of Mathematical Sciences, University of Texas at El Paso, TX 79968, USA*

Received 1 February 2007

Abstract

Static condensation of internal degrees of freedom, partial orthogonalization of basis functions, and ILU preconditioning are techniques used to facilitate the solution of discrete problems obtained in the *hp*-FEM. This paper shows that for symmetric linear (not necessarily positive-definite) problems, under mild technical assumptions, these three techniques are completely equivalent. In fact, the same matrices can be obtained by the same arithmetic operations. The study can be extended to nonsymmetric problems naturally.
© 2007 Elsevier B.V. All rights reserved.

MSC: 65N30; 74S05

Keywords: *hp*-FEM; Static condensation; Orthogonal shape functions; ILU preconditioning

1. Introduction

The *hp*-FEM is a modern version of the finite element method (FEM) capable of achieving exceptionally fast convergence through optimal variation of the size and polynomial degree of elements. The presence of the so-called *bubble functions* (higher-degree basis functions local to element interiors) yields a special 2×2 block structure of the stiffness matrix which can be utilized to ease the solution of the discrete problem. This can be done, e.g., using the static condensation of internal degrees of freedom [2,3,5] or through partial orthogonalization of basis functions [6]. In addition to these two techniques, we also study the effect of incomplete LU preconditioning (see, e.g. [1]). It turns out that with a suitable ordering of unknowns, one obtains a 2×2 block matrix where the first diagonal block is the identity matrix, the off-diagonal blocks are zero, and the only nontrivial block is equal to the ILU-preconditioned reduced stiffness matrix obtained by the static condensation or by the partial orthogonalization of basis functions.

* Corresponding author.
E-mail addresses: vejchod@math.cas.cz (T. Vejchodský), solin@utep.edu (P. Šolín).

[1] The support of the Grant Agency of the Czech Academy of Sciences (Project No. IAA100760702) and of the Czech Academy of Sciences (Institutional Research Plan No. AV0Z10190503) is gratefully acknowledged.

[2] The second author acknowledges the financial support of the Czech Science Foundation (Projects No. 102/05/0629 and 102/07/0496) and of the Grant Agency of the Czech Academy of Sciences (Project No. IAA100760702).

0377-0427/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2007.04.044

The reader will find in the text a couple of easily verifiable technical assumptions which are satisfied obviously for most second-order symmetric linear elliptic problems, time-harmonic Maxwell's equations, and many other symmetric problems.

The outline of the paper is as follows: an overview of the *hp*-FEM with basic definitions and notations is provided in Section 2. Subsequent Sections 3–5 describe the static condensation of internal degrees of freedom, partial orthogonalization of basis functions, and ILU preconditioning. An illustrating numerical experiment is presented in Section 6.

2. Overview of *hp*-FEM

Let us consider a weak problem to find $u \in V$ such that

$$a(u, v) = \mathscr{F}(v) \quad \forall v \in V, \tag{1}$$

where V is a Hilbert space (usually a Sobolev space), $a : V \times V \mapsto \mathbb{R}$ is a symmetric bilinear form, and $\mathscr{F}$ stands for a linear functional on V. We assume that problem (1) admits a unique solution.

The *hp*-FEM solution to problem (1) is defined as an element $u_{hp} \in V_{hp}$ where V_{hp} is a finite dimensional (usually piecewise-polynomial) subspace of V, dim $V_{hp} = N < \infty$, satisfying

$$a(u_{hp}, v_{hp}) = \mathscr{F}(v_{hp}) \quad \forall v_{hp} \in V_{hp}. \tag{2}$$

The performance of the *hp*-FEM depends significantly on the choice of the subspace V_{hp} as well as on the choice of its basis $\mathscr{B} = \{\varphi_1, \varphi_2, ..., \varphi_N\}$. If we expand $u_{hp} \in V_{hp}$ as $u_{hp} = \sum_{j=1}^{N} y_j \varphi_j$ then the coefficients $\mathbb{Y} = (y_1, y_2, \ldots, y_N)^{\mathrm{T}}$ can be computed from a linear algebraic system of the form

$$\mathbb{A}\mathbb{Y} = \mathbb{F}, \tag{3}$$

where the stiffness matrix $\mathbb{A} \in \mathbb{R}^{N \times N}$ and the load vector $\mathbb{F}$ have the entries $\mathbb{A}_{ij} = a(\varphi_j, \varphi_i)$ and $\mathbb{F}_i = \mathscr{F}(\varphi_i)$, $i, j = 1, 2, \ldots, N$.

Let us have a closer look at the basis $\mathscr{B}$. By $\mathscr{T}_{hp}$ we denote a finite set of elements (the finite element mesh). Both the bilinear form $a(\cdot, \cdot)$ and the linear functional $\mathscr{F}(\cdot)$ can be split into the sums of element contributions,

$$a(u, v) = \sum_{K \in \mathscr{T}_{hp}} a_K(u, v) \quad \text{and} \quad \mathscr{F}(v) = \sum_{K \in \mathscr{T}_{hp}} \mathscr{F}_K(v) \quad \forall u, v \in V.$$

The forms $a_K(\cdot, \cdot)$ and $\mathscr{F}_K(\cdot)$ are assumed to be bilinear and linear, respectively. For future reference, we assume that the forms a_K are symmetric. This assumption is satisfied for most second-order symmetric linear elliptic problems, time-harmonic Maxwell's equations, and for many other problems.

Further, by $\mathscr{B}^K$ we denote the set of all basis functions supporting the form a_K and by $\mathscr{B}_0^K$ the set of all *bubble functions* associated with an element K,

$$\mathscr{B}^K = \{\varphi_j \in \mathscr{B} : \exists\, \varphi_i \in \mathscr{B} \text{ such that } a_K(\varphi_j, \varphi_i) \neq 0\}, \tag{4}$$

$$\begin{aligned} \mathscr{B}_0^K = \{&\varphi_j \in \mathscr{B}^K : a(\varphi_j, \varphi_i) = a_K(\varphi_j, \varphi_i) \quad \forall \varphi_i \in \mathscr{B} \\ &\text{and } a_{K^*}(\varphi_j, \varphi_i) = 0 \quad \forall \varphi_i \in \mathscr{B} \forall K^* \in \mathscr{T}_{hp},\ K \neq K^*\}. \end{aligned} \tag{5}$$

Remark 1. Using the standard *hp*-FEM terminology we see that $\varphi_j \in \mathscr{B}^K$ if and only if $K \subset \text{supp}(\varphi_j)$ and that φ_j is a bubble function if $\overline{K} = \text{supp}(\varphi_j)$.

Remark 2. Formally, in certain situations (on elements adjacent to Neumann/Newton boundary), the above definition can be satisfied also by basis functions associated with vertices or edges. For the sake of algorithmic simplicity, such basis functions usually are treated as vertex or edge functions, not as bubble functions (although the theory works fine with both options).

For every finite element $K \in \mathscr{T}_{hp}$, let N^K and M^K represent the number of elements of $\mathscr{B}^K$ and $\mathscr{B}_0^K$, respectively. We enumerate the basis functions in $\mathscr{B}$ as follows: $\varphi_1, \varphi_2, \ldots, \varphi_{M^{K_1}}$ are the bubble functions in the element K_1, $\varphi_{M^{K_1}+1}, \varphi_{M^{K_1}+2}, \ldots, \varphi_{M^{K_1}+M^{K_2}}$ are the bubble functions in the element $K_2 \in \mathscr{T}_{hp}$, etc. This means that φ_M stands for the last bubble function of the last element in $\mathscr{T}_{hp}$. By $\varphi_{M+1}, \varphi_{M+2}, \ldots, \varphi_N$ we denote the remaining basis functions.

We define the index sets $I(K) = \{j : 1 \leqslant j \leqslant N,\ \varphi_j \in \mathscr{B}^K\}$ and $I_0(K) = \{j : 1 \leqslant j \leqslant N,\ \varphi_j \in \mathscr{B}_0^K\}$. For every element K we also define a one-to-one connectivity mapping $\imath_K : \{1, 2, \ldots, N^K\} \mapsto I(K)$ such that both restrictions $\imath_K : \{1, 2, \ldots, M^K\} \mapsto I_0(K)$ and $\imath_K : \{M^K + 1, \ldots, N^K\} \mapsto I^K \setminus I_0(K)$ are one-to-one. The connectivity mapping is an essential part of every *hp*-FEM code, see [5].

Using the connectivity mappings, we can easily define for every element $K \in \mathscr{T}_{hp}$ a local stiffness matrix $\mathbb{A}^K \in \mathbb{R}^{N^K \times N^K}$ and local load vector $\mathbb{F}^K \in \mathbb{R}^{N^K}$ with the entries $\mathbb{A}^K_{\ell m} = a_K(\varphi_{\imath_K(m)}, \varphi_{\imath_K(\ell)})$ and $\mathbb{F}^K_\ell = \mathscr{F}_K(\varphi_{\imath_K(\ell)})$, $\ell, m = 1, 2, \ldots, N^K$. The above enumeration of basis functions (bubble functions first) yields a 2×2 block structure of both global and local stiffness matrices and load vectors

$$\mathbb{A} = \begin{pmatrix} A & B^{\mathrm{T}} \\ B & D \end{pmatrix}, \quad \mathbb{F} = \begin{pmatrix} F \\ G \end{pmatrix}, \quad \mathbb{A}^K = \begin{pmatrix} A^K & (B^K)^{\mathrm{T}} \\ B^K & D^K \end{pmatrix}, \quad \mathbb{F}^K = \begin{pmatrix} F^K \\ G^K \end{pmatrix}. \tag{6}$$

Here, $A \in \mathbb{R}^{M \times M}$ and $A^K \in \mathbb{R}^{M^K \times M^K}$ correspond to the products of bubble functions with bubble functions, $B \in \mathbb{R}^{(N-M)\times M}$ and $B^K \in \mathbb{R}^{(N^K - M^K)\times M^K}$ correspond to the products of non-bubbles with bubbles, etc.

For future reference, we impose another technical assumption: let both $\mathbb{A}$ and A be nonsingular and let $\mathbb{A}_{jj} = a(\varphi_j, \varphi_j) \neq 0$, $j = 1, 2, \ldots, M$. Again, this assumption is satisfied for most symmetric problems.

Lemma 1. *Let $1 \leqslant j, k \leqslant N$. Unless there exists an element $K \in \mathscr{T}_{hp}$ such that both $j \in I(K)$ and $k \in I(K)$, it is $\mathbb{A}_{jk} = 0$.*

Proof. It follows from (4) and it is left to the reader as an easy exercise. □

The preceding Lemma 1 is used to define the sparsity structure of the matrix $\mathbb{A}$: if at least one such element K exists, $\mathbb{A}_{jk}$ is treated as nonzero.

Lemma 2. *Let φ_k, $1 \leqslant k \leqslant M$ be a bubble function in $K \in \mathscr{T}_{hp}$, i.e., $k \in I_0(K)$. Then $\mathbb{A}_{jk} = \mathbb{A}^K_{\imath_K^{-1}(j), \imath_K^{-1}(k)}$ for all $j \in I(K)$.*

Proof. By (5) we have $A_{jk} = a(\varphi_k, \varphi_j) = a_K(\varphi_k, \varphi_j)$. If $j \in I(K)$ then by definition of local stiffness matrices $a_K(\varphi_k, \varphi_j) = \mathbb{A}^K_{\imath_K^{-1}(j), \imath_K^{-1}(k)}$. □

3. Static condensation of internal degrees of freedom

Let us recall the block structure (6) of system (3),

$$\mathbb{A}\mathbb{Y} = \begin{pmatrix} A & B^{\mathrm{T}} \\ B & D \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix} = \mathbb{F}. \tag{7}$$

By Lemmas 1 and 2, $A = \mathrm{blockdiag}\{A^K,\ K \in \mathscr{T}_{hp}\} \in \mathbb{R}^{M \times M}$. The static condensation of internal degrees of freedom is based on the following idea: express $\mathbf{x}$ from the first row of (7)

$$\mathbf{x} = A^{-1}(F - B^{\mathrm{T}}\mathbf{y}) \tag{8}$$

and substitute the result into the second row of (7). This yields a system of the form

$$S\mathbf{y} = \widetilde{G}, \tag{9}$$

where $S = D - BA^{-1}B^{\mathrm{T}}$ and $\widetilde{G} = G - BA^{-1}F$. Hence the vector $\mathbf{y} \in \mathbb{R}^{N-M}$ can be computed from a smaller system (9) and finally, $\mathbf{x} \in \mathbb{R}^M$ is obtained by solving the block-diagonal system (8).

Lemma 3. *The matrix S obtained using local Schur complements S^K on elements is identical to the Schur complement of A in $\mathbb{A}$, i.e., $S = D - BA^{-1}B^{\mathrm{T}}$. Moreover, S has the same sparsity structure (in view of Lemma 1) as the original block D in (7).*

Lemma 3 will be proven in the next section using Lemmas 5–7. Before we do that, let us present an efficient static condensation algorithm:

1. Build the local stiffness matrices $\mathbb{A}^K$ and the local load vectors $\mathbb{F}^K$.
2. Compute the local Schur complements $S^K = D^K - B^K(A^K)^{-1}(B^K)^{\mathrm{T}}$ and local complement loads $\widetilde{G}^K = G^K - B^K(A^K)^{-1}F^K$.
3. Run the standard *hp*-FEM assembling algorithm, see, e.g. [4], to build the global Schur complement $S \in \mathbb{R}^{(N-M)\times(N-M)}$ and the global load vector $\widetilde{G} \in \mathbb{R}^{N-M}$ from the local complements $S^K \in \mathbb{R}^{(N^K-M^K)\times(N^K-M^K)}$ and from the local vectors $\widetilde{G}^K \in \mathbb{R}^{N-M}$, respectively.
4. Solve the Schur complement system $S\mathbf{y} = \widetilde{G}$.
5. Disassemble the global coefficient vector $\mathbf{y} \in \mathbb{R}^{N-M}$ to local contributions $\mathbf{y}^K \in \mathbb{R}^{N^K-M^K}$, where $\mathbf{y}^K_\ell = \mathbf{y}_{\iota_K(M^K+\ell)}$, $\ell = 1, 2, \ldots, N^K - M^K$.
6. Compute $\mathbf{x}^K = (A^K)^{-1}(F_K - (B^K)^{\mathrm{T}}\mathbf{y}^K)$ element by element.
7. Assemble vector $\mathbf{x} \in \mathbb{R}^M$ by $\mathbf{x}_j = \mathbf{x}^K_{\iota_K^{-1}(j)}$, $j = 1, 2, \ldots, M$, where $K \in \mathscr{T}_{hp}$ is the unique element such that $j \in I_0(K)$.

4. Partial orthogonalization of basis functions

In this paper, "orthogonality" is understood in the sense of the bilinear form $a(\cdot,\cdot)$. This notation, however, is not exact because $a(\cdot,\cdot)$ may not represent an inner product. The idea proposed in [6] was to define new basis functions $\widetilde{\varphi}_1, \widetilde{\varphi}_2, \ldots, \widetilde{\varphi}_N$ so that $a(\widetilde{\varphi}_j, \widetilde{\varphi}_i) = 0$ whenever $i \leqslant M$ & $j > M$ or $j \leqslant M$ & $i > M$. This means that in the new stiffness matrix $\widetilde{\mathbb{A}}$ the off-diagonal blocks $\widetilde{B}$ and $\widetilde{B}^{\mathrm{T}}$ are zero. Let us remark that in [6] the authors moreover orthonormalized bubble functions in all elements so that the new block $\widetilde{A}$ in $\widetilde{\mathbb{A}}$ was an identity matrix. We will not orthonormalize the bubble functions in this study to clearly demonstrate the connection with static condensation.

The matrix $A = \mathrm{blockdiag}\{A^K,\ K \in \mathscr{T}_{hp}\}$ is invertible and by Lemma 2, $A^{-1} = \mathrm{blockdiag}\{(A^K)^{-1},\ K \in \mathscr{T}_{hp}\}$. We define matrices $Q = BA^{-1} \in \mathbb{R}^{(N-M)\times M}$ and $Q^K = B^K(A^K)^{-1} \in \mathbb{R}^{(N^K-M^K)\times M^K}$. The following Lemma 4 shows the relation of Q and Q^K and it implies that the matrices Q and B have the same sparsity structure.

Lemma 4. *Let φ_k be a bubble function in an element $K \in \mathscr{T}_{hp}$, i.e., $k \in I_0(K)$. For every $1 \leqslant j \leqslant N - M$ it holds: $Q_{jk} = Q^K_{\iota_K^{-1}(M+j),\iota_K^{-1}(k)}$ if $M + j \in I(K)$ and $Q_{jk} = 0$ otherwise.*

Proof. We consider $k \in I_0(K)$ and an arbitrary index $r \in \{1, 2, \ldots, M\}$. First, we assume $M + j \in I(K)$. There are two possibilities. If (a) $r \in I_0(K)$ then Lemma 2 shows that $B_{jr} = B^K_{\iota_K^{-1}(M+j),\iota_K^{-1}(r)}$ and $A_{rk} = A^K_{\iota_K^{-1}(r),\iota_K^{-1}(k)}$. Thanks to block-diagonal structure of A we obtain $A^{-1}_{rk} = (A^K)^{-1}_{\iota_K^{-1}(r),\iota_K^{-1}(k)}$. On the other hand if (b) $r \notin I_0(K)$ then from Lemma 1 follows $A_{rk} = 0$ and hence $A^{-1}_{rk} = 0$ due to the block-diagonal structure of A. Summarizing possibilities (a) and (b) we obtain

$$Q_{jk} = \sum_{r\in I_0(K)} B_{jr}A^{-1}_{rk} = \sum_{s=1}^{M^K} B^K_{\iota_K^{-1}(M+j),s}(A^K)^{-1}_{s,\iota_K^{-1}(k)} = Q^K_{\iota_K^{-1}(M+j),\iota_K^{-1}(k)}.$$

Second, we assume $M + j \notin I(K)$ then either (c) $r \in I_0(K)$ or (d) $r \notin I_0(K)$. In case (c) we have $B_{jr} = 0$ while in case (d) $A_{rk} = 0$. Both variants follow from Lemma 1. Moreover, due to the block-diagonal structure of A we conclude in case (d) that $A^{-1}_{rk} = 0$. Cases (c) and (d) imply $Q_{jk} = \sum_{r=1}^{M} B_{jr}A^{-1}_{rk} = 0$. □

Finally, we define the new basis functions

$$\widetilde{\varphi}_j = \varphi_j \quad \text{for } j = 1, 2, \ldots, M,$$

$$\widetilde{\varphi}_{M+j} = \varphi_{M+j} - \sum_{k=1}^{M} Q_{jk}\varphi_k \quad \text{for } j = 1, 2, \ldots, N - M. \tag{10}$$

One can verify easily that the new basis functions (10) are linearly independent. The new stiffness matrices $\widetilde{\mathbb{A}}$ and $\widetilde{\mathbb{A}}^K$ have a 2×2 block structure analogous to the original matrices $\mathbb{A}$ and $\mathbb{A}^K$, respectively. We denote the blocks in $\widetilde{\mathbb{A}}$ by $\widetilde{A}$, $\widetilde{B}^{\mathrm{T}}$, $\widetilde{B}$, $\widetilde{D}$, and the blocks in $\widetilde{\mathbb{A}}^K$ by $\widetilde{A}^K$, $(\widetilde{B}^K)^{\mathrm{T}}$, $\widetilde{B}^K$, $\widetilde{D}^K$ analogously to (6). The following Lemmas 5–7 explain the relation of the new blocks to the corresponding blocks in the original matrices $\mathbb{A}$ and $\mathbb{A}^K$.

Lemma 5. *The stiffness matrix $\widetilde{\mathbb{A}} \in \mathbb{R}^{N\times N}$ with the entries $\widetilde{\mathbb{A}}_{jk} = a_K(\widetilde{\varphi}_k, \widetilde{\varphi}_j)$ and the load vector $\widetilde{\mathbb{F}} \in \mathbb{R}^N$ with the entries $\widetilde{\mathbb{F}}_j = \mathscr{F}(\widetilde{\varphi}_j)$, $j, k = 1, 2, \ldots, N$, have the following block structure*:

$$\widetilde{\mathbb{A}} = \begin{pmatrix} \widetilde{A} & \widetilde{B}^{\mathrm{T}} \\ \widetilde{B} & \widetilde{D} \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix} \quad \textit{and} \quad \widetilde{\mathbb{F}} = \begin{pmatrix} \widetilde{F} \\ \widetilde{G} \end{pmatrix} = \begin{pmatrix} F \\ G - QF \end{pmatrix}, \tag{11}$$

where $S = D - BA^{-1}B^{\mathrm{T}}$.

Proof. It is easy to see that $\widetilde{A} = A$ and $\widetilde{F} = F$, and it follows from a straightforward calculation that $\widetilde{B} = 0$, $\widetilde{D} = S$, and $\widetilde{G} = G - QF$. For illustration let us explain why $\widetilde{B} = 0$: for $1 \leqslant j \leqslant N - M$ and $1 \leqslant k \leqslant M$ we calculate

$$\widetilde{B}_{jk} = a(\widetilde{\varphi}_k, \widetilde{\varphi}_{M+j}) = a(\varphi_k, \varphi_{M+j}) - \sum_{r=1}^{M} Q_{jr}a(\varphi_k, \varphi_r) = B_{jk} - \sum_{r=1}^{M} Q_{jr}A_{rk}.$$

Thus, $\widetilde{B} = B - QA = B - BA^{-1}A = 0$. □

Lemma 6. *Let $K \in \mathscr{T}_{hp}$. The local stiffness matrix $\widetilde{\mathbb{A}}^K \in \mathbb{R}^{N^K\times N^K}$ with the entries $\widetilde{\mathbb{A}}^K_{\ell m} = a_K(\widetilde{\varphi}_{\iota_K(m)}, \widetilde{\varphi}_{\iota_K(\ell)})$ and the local load vector $\widetilde{\mathbb{F}}^K \in \mathbb{R}^{N^K}$ with the entries $\widetilde{\mathbb{F}}^K_\ell = \mathscr{F}_K(\widetilde{\varphi}_{\iota_K(\ell)})$, $\ell, m = 1, 2, \ldots, N^K$, have the following block structure*:

$$\widetilde{\mathbb{A}}^K = \begin{pmatrix} \widetilde{A}^K & (\widetilde{B}^K)^{\mathrm{T}} \\ \widetilde{B}^K & \widetilde{D}^K \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & S^K \end{pmatrix} \quad \textit{and} \quad \widetilde{\mathbb{F}}^K = \begin{pmatrix} \widetilde{F}^K \\ \widetilde{G}^K \end{pmatrix}, \tag{12}$$

where $S^K = D^K - B^K(A^K)^{-1}(B^K)^{\mathrm{T}}$, $\widetilde{F}^K = F^K$, and $\widetilde{G}^K = G^K - Q^K F^K$.

Proof. The proof is analogous to the proof of Lemma 5 except that we work with local element matrices. The transfer to the local level is justified by Lemma 4. For illustration let us explain why $\widetilde{D}^K = S^K$: let $1 \leqslant \ell, m \leqslant N^K - M^K$, and let $M + j = \iota_K(\ell)$ and $M + k = \iota_K(m)$. Using definitions of the local stiffness matrices $\mathbb{A}^K$ and Lemma 4, we obtain

$$\begin{aligned} \widetilde{D}^K_{\ell m} &= a_K(\widetilde{\varphi}_{M+k}, \widetilde{\varphi}_{M+j}) = a_K(\varphi_{M+k}, \varphi_{M+j}) - \sum_{r\in I_0(K)} Q_{jr}a_K(\varphi_{M+k}, \varphi_r) \\ &\quad - \sum_{p\in I_0(K)} a_K(\varphi_p, \varphi_{M+j})Q_{kp} + \sum_{r\in I_0(K)}\sum_{p\in I_0(K)} Q_{jr}a_K(\varphi_p, \varphi_r)Q_{kp} \\ &= D^K_{\ell m} - \sum_{s=1}^{M^K} Q^K_{\ell s}(B^K)^{\mathrm{T}}_{sm} - \sum_{q=1}^{M^K} B^K_{\ell q}(Q^K)^{\mathrm{T}}_{qm} + \sum_{s=1}^{M^K}\sum_{q=1}^{M^K} Q^K_{\ell s}A^K_{sq}(Q^K)^{\mathrm{T}}_{qm}. \end{aligned} \tag{13}$$

Notice that $\sum_{r=1}^{M} a_K(v, \varphi_r) = \sum_{r\in I_0(K)} a_K(v, \varphi_r)$ for all $v \in V_{hp}$ due to (4) and (5). Thus, relation (13) implies $\widetilde{D}^K = D^K - Q^K(B^K)^{\mathrm{T}} - B^K(Q^K)^{\mathrm{T}} + Q^K A^K(Q^K)^{\mathrm{T}} = D^K - B^K(A^K)^{-1}(B^K)^{\mathrm{T}}$.

Lemma 7. *Let* $1 \leqslant j, k \leqslant N-M$. *If* $M+j \notin I(K)$ *or* $M+k \notin I(K)$ *for any* $K \in \mathscr{T}_{hp}$ *then* $\widetilde{D}_{jk} = S_{jk} = a(\widetilde{\varphi}_{M+k}, \widetilde{\varphi}_{M+j}) = 0$.

Proof. We use $\widetilde{D}_{jk} = \sum_{K \in \mathscr{T}_{hp}} \widetilde{D}^K_{\iota_K^{-1}(M+j), \iota_K^{-1}(M+k)}$ and similarly to (13) we express

$$\widetilde{D}_{jk} = a(\varphi_{M+k}, \varphi_{M+j}) - \sum_{K \in \mathscr{T}_{hp}} \sum_{r \in I_0(K)} Q_{jr} a_K(\varphi_{M+k}, \varphi_r) - \sum_{K \in \mathscr{T}_{hp}} \sum_{p \in I_0(K)} a_K(\varphi_p, \varphi_{M+j}) Q_{kp} + \sum_{K \in \mathscr{T}_{hp}} \sum_{r \in I_0(K)} \sum_{p \in I_0(K)} Q_{jr} a_K(\varphi_p, \varphi_r) Q_{kp}.$$

The first term on the right-hand side $a(\varphi_{M+k}, \varphi_{M+j})$ is equal to D_{jk} and vanishes due to Lemma 1. The second term is zero because if $M + j \notin I(K)$ then $Q_{jr} = 0$ by Lemma 4 and if $M + k \notin I(K)$ then $a_K(\varphi_{M+k}, \varphi_r) = 0$ by (4). The same argument is used for the third term. The fourth term vanishes because if $M + j \notin I(K)$ or $M + k \notin I(K)$, then by Lemma 4 it is $Q_{jr} = 0$ or $Q_{kp} = 0$, respectively. □

Proof of Lemma 3. Lemmas 5 and 6 show that the standard assembling procedure with the new basis functions (10) leads to global and local stiffness matrices (11) and (12). However, the global and local blocks $\widetilde{D}$ and $\widetilde{D}^K$ in these matrices are identical to the global and local Schur complements S and S^K, respectively. This proves that the standard assembling algorithm applied to local Schur complements S^K gives the global Schur complement S. Moreover, Lemma 7 shows that the global Schur complement S has the same sparsity pattern (in the sense of Lemma 1) as the original block D in $\mathbb{A}$. □

In practice, the new expansion coefficients $\widetilde{\mathbb{Y}}^{\mathrm{T}} = (\widetilde{\mathbf{x}}^{\mathrm{T}}, \widetilde{\mathbf{y}}^{\mathrm{T}})$ are calculated from the systems $A\widetilde{\mathbf{x}} = F$ and $S\widetilde{\mathbf{y}} = \widetilde{G}$. The original vectors $\mathbf{x}$ and $\mathbf{y}$ are then expressed from $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ as follows:

$$u_{hp} = \sum_{j=1}^{M} \widetilde{x}_j \widetilde{\varphi}_j + \sum_{j=1}^{N-M} \widetilde{y}_j \widetilde{\varphi}_{M+j} = \sum_{k=1}^{M} \left(\widetilde{x}_k - \sum_{j=1}^{N-M} \widetilde{y}_j Q_{jk} \right) \varphi_k + \sum_{j=1}^{N-M} \widetilde{y}_j \varphi_{M+j}.$$

Thus, $\mathbf{x} = \widetilde{\mathbf{x}} - A^{-1} B^{\mathrm{T}} \widetilde{\mathbf{y}}$ and $\mathbf{y} = \widetilde{\mathbf{y}}$. Notice that these equalities together with the relations for $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ are identical to (8) and (9). This explains that the static condensation and the partial orthogonalization of basis functions are just two different interpretations of the same arithmetic procedure. It is possible to implement both approaches in such a way that the same arithmetic operations are performed.

5. ILU preconditioning

An incomplete LU preconditioning of system (7) is done as follows: (i) set $\widehat{\mathbb{A}} = \widehat{\mathbb{L}}^{-1} \mathbb{A} \widehat{\mathbb{U}}^{-1}$ and $\widehat{\mathbb{F}} = \widehat{\mathbb{L}}^{-1} \mathbb{F}$. (ii) Solve $\widehat{\mathbb{A}} \widehat{\mathbb{Y}} = \widehat{\mathbb{F}}$. (iii) Solve $\widehat{\mathbb{U}} \mathbb{Y} = \widehat{\mathbb{Y}}$. The preconditioners $\widehat{\mathbb{L}}$ and $\widehat{\mathbb{U}}$ are computed by incomplete LU factorization of $\mathbb{A}$. We also consider the exact factorization $\mathbb{A} = \mathbb{L}\mathbb{U}$. The preconditioners $\widehat{\mathbb{L}}$ and $\widehat{\mathbb{U}}$ and the LU factors $\mathbb{L}$ and $\mathbb{U}$ have the natural 2×2 block structure, cf. (6)

$$\widehat{\mathbb{L}} = \begin{pmatrix} \widehat{L}_A & 0 \\ \widehat{L}_B & \widehat{L}_D \end{pmatrix}, \quad \widehat{\mathbb{U}} = \begin{pmatrix} \widehat{U}_A & \widehat{U}_B \\ 0 & \widehat{U}_D \end{pmatrix}, \quad \mathbb{L} = \begin{pmatrix} L_A & 0 \\ L_B & L_D \end{pmatrix}, \quad \mathbb{U} = \begin{pmatrix} U_A & U_B \\ 0 & U_D \end{pmatrix}.$$

Lemma 8. *With the above notation we have* $\widehat{L}_A = L_A$, $\widehat{L}_B = L_B$, $\widehat{U}_A = U_A$, *and* $\widehat{U}_B = U_B$. *Moreover, the blocks* $\widehat{L}_D$ *and* $\widehat{U}_D$ *are the incomplete LU preconditioners of the Schur complement* $S = D - BA^{-1}B^{\mathrm{T}}$.

Proof. We show that the first M steps of ILU factorization does not lead to any fill-in. This is thanks to the fact that we first enumerate the M bubble functions and then the $N - M$ non-bubble basis functions.

Recall that the first M steps of the LU factorization algorithm are as follows: for every $k = 1, 2, \ldots, M$ and for every $i = k + 1, k + 2, \ldots, N$ multiply the kth row by $\mathbb{A}_{ik}/\mathbb{A}_{kk}$ and subtract the result from the ith row. In other words, the

entries $\mathbb{A}_{ij}$, $j = k, k+1, \ldots, N$, are replaced by $\mathbb{A}_{ij} - \mathbb{A}_{kj}\mathbb{A}_{ik}/\mathbb{A}_{kk}$. This step is omitted in the ILU algorithm if $\mathbb{A}_{ij} = 0$.

Let us have a closer look at the first step ($k = 1$). Since $k \leqslant M$, there exists an element $K \in \mathscr{T}_{hp}$ such that the basis function φ_k is a bubble in K, i.e, $k \in I_0(K)$. First, we consider the case $\mathbb{A}_{ij} = 0$. In view of Lemma 1, this happens if $i \notin I(K)$ or $j \notin I(K)$. In these cases Lemma 1 implies that $A_{ik} = 0$ or $A_{kj} = 0$. In either case, both LU and ILU algorithms leave the zero element $\mathbb{A}_{ij}$ unchanged. Second, we consider $\mathbb{A}_{ij} \neq 0$ in the sense of Lemma 1. In this case both LU and ILU algorithms work identically.

Hence, after the first step of the ILU algorithm the sparsity structure of the remaining part of the matrix $\mathbb{A}$ remains unchanged and we can repeat the same analysis for $k = 2, 3, \ldots, M$ to conclude that during the first M steps no fill-in appears and both ILU and LU algorithms work identically.

After the elimination of the first M columns the resulting $(N-M)\times(N-M)$ block equals to the Schur complement S. Thus, the $\widehat{L}_D$ and $\widehat{U}_D$ blocks are the ILU preconditioners of S. □

Lemma 8 shows that

$$\widehat{\mathbb{L}}\widehat{\mathbb{U}} = \begin{pmatrix} L_A & 0 \\ L_B & \widehat{L}_D \end{pmatrix}\begin{pmatrix} U_A & U_B \\ 0 & \widehat{U}_D \end{pmatrix} = \begin{pmatrix} L_A U_A & L_A U_B \\ L_B U_A & L_B U_B + \widehat{L}_D \widehat{U}_D \end{pmatrix}.$$

Since $\mathbb{A} = \mathbb{L}\mathbb{U}$, we obtain $A = L_A U_A$, $B^{\mathrm{T}} = L_A U_B$, $B = L_B U_A$. A straightforward matrix computation reveals the structure of the preconditioned stiffness matrix $\widehat{\mathbb{A}}$ and of the preconditioned load vector $\widehat{\mathbb{F}}$:

$$\widehat{\mathbb{A}} = \widehat{\mathbb{L}}^{-1}\mathbb{A}\widehat{\mathbb{U}}^{-1} = \begin{pmatrix} I & 0 \\ 0 & \widehat{L}_D^{-1} S \widehat{U}_D^{-1} \end{pmatrix}, \quad \widehat{\mathbb{F}} = \widehat{\mathbb{L}}^{-1}\mathbb{F} = \begin{pmatrix} L_A^{-1} F \\ \widehat{L}_D^{-1}\widetilde{G} \end{pmatrix},$$

where $S = D - BA^{-1}B^{\mathrm{T}}$ and $\widetilde{G} = G - L_B L_A^{-1} F = G - BA^{-1}F$, cf. (11).

Hence, the solution $\widehat{\mathbb{Y}}^{\mathrm{T}} = (\widehat{\mathbf{x}}^{\mathrm{T}}, \widehat{\mathbf{y}}^{\mathrm{T}})$ to $\widehat{\mathbb{A}}\widehat{\mathbb{Y}} = \widehat{\mathbb{F}}$ is given by

$$\widehat{\mathbf{x}} = L_A^{-1} F \quad \text{and} \quad \widehat{L}_D^{-1} S \widehat{U}_D^{-1}\widehat{\mathbf{y}} = \widehat{L}_D^{-1}\widetilde{G}. \tag{14}$$

The final solution $\mathbb{Y}^{\mathrm{T}} = (\mathbf{x}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}})$ is then computed by $\widehat{\mathbb{U}}\mathbb{Y} = \widehat{\mathbb{Y}}$, i.e.,

$$U_A\mathbf{x} = \widehat{\mathbf{x}} - U_B\mathbf{y} = L_A^{-1}F - U_B\mathbf{y} \quad \text{and} \quad \widehat{U}_D\mathbf{y} = \widehat{\mathbf{y}}. \tag{15}$$

Thus, systems (14) and (15) for $\mathbf{x}$ and $\mathbf{y}$ are equal to suitably preconditioned static condensation systems (8) and (9). System (9) for $\mathbf{y}$ is preconditioned from both sides by incomplete LU factors $\widehat{L}_D$ and $\widehat{U}_D$. System (8) for $\mathbf{x}$ is preconditioned from the left by the factor L_A.

To summarize, the ILU preconditioner does the job and decouples the systems for bubble and non-bubble basis functions. However, the straightforward implementation, i.e., the application of the ILU preconditioner to system (3), is inefficient, because the block-diagonal subsystem for bubbles is solved superfluously in every iteration. This straightforward implementation does not see that the first M unknowns were already resolved by the preconditioner exactly. On the other hand, an efficient implementation stops the ILU algorithm for the matrix $\mathbb{A}$ after the first M steps and stores the resulting Schur complement S. Then it continues to compute the ILU preconditioners $\widehat{L}_D$ and $\widehat{U}_D$ of the complement S. The ILU-preconditioned systems (14) are then solved by a suitable iterative method. Finally, the coefficients $\mathbf{x}$ and $\mathbf{y}$ are obtained efficiently by (15). Nevertheless, notice that this efficient implementation is completely equivalent to the procedure of static condensation introduced in Section 3, where the Schur complement system $S\mathbf{y} = \widetilde{G}$ was preconditioned from both sides by ILU factors $\widehat{L}_D$ and $\widehat{U}_D$. Indeed, the first M elimination steps construct the Schur complement S in exactly the same way as the element by element procedure of static condensation.

6. Numerical experiment

In this section we perform a simple numerical experiment to compare the performance of the static condensation and of the straightforward (inefficient) implementation of the ILU preconditioner to solve the linear system (3). The tested linear system comes from *hp*-FEM discretization of the Poisson equation with homogenous Dirichlet boundary conditions in a square domain. The first row in Table 1 corresponds to a coarse *hp*-FEM mesh sketched in Fig. 1. The

Table 1
The first column shows the number of successive *h*-refinement steps of the original mesh, M and N stand for the sizes of the matrices S and $\mathbb{A}$, nnz(S) and nnz($\mathbb{A}$) represent the number of their nonzero entries, and N_{iter} denotes the number of ILU-PCG iterations. Finally, we indicate CPU times needed to solve system (3)

Refin. step	Static condensation				ILU-PCG			
	M	nnz(S)	N_{iter}	CPU time (s)	N	nnz($\mathbb{A}$)	N_{iter}	CPU time (s)
0	34	202	3	0.004	50	1180	3	0.005
1	136	1983	5	0.012	225	7095	5	0.017
2	544	10 795	7	0.049	953	33 847	7	0.130
3	2176	49 419	11	0.389	3921	147 039	11	1.665
4	8704	210 667	21	4.697	15 905	612 175	21	26.10
5	34816	869 163	40	71.10	64 065	24 97 455	40	415.5

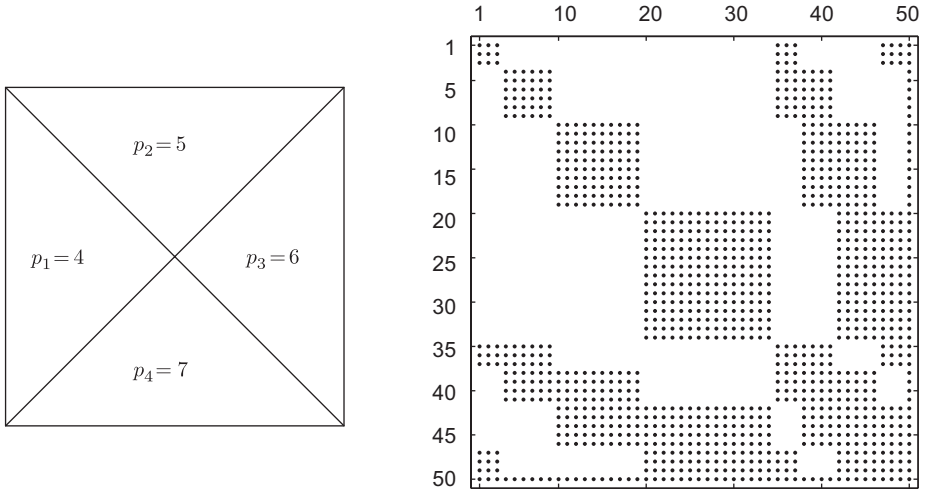


Fig. 1. The rough mesh with four elements of polynomial degrees 4, 5, 6, and 7 (left) and the sparsity pattern of the corresponding stiffness matrix $\mathbb{A}$, cf. (6).

subsequent rows in Table 1 correspond to a uniform *h*-refinement of this mesh. In each refinement step we split every triangle into four similar triangles with the same polynomial degree.

In case of static condensation the CPU times in Table 1 measure steps 2–7 of the algorithm presented in Section 3, i.e., the inversion of local blocks A^K, assembling of the global Schur complement S and of the complement load $\widetilde{G}$, solving system (9) by ILU preconditioned conjugated gradients, and computation of the condensed unknowns by (8). While in the case of ILU-PCG the presented CPU times measure the computation of ILU preconditioner of $\mathbb{A}$ and the PCG iterations to solve (3). The test was done in Matlab with its built-in ILU preconditioner and PCG routines. The relative residual tolerance for PCG iterations was set to 10^{-5} in all cases.

As we discussed above, the static condensation with ILU-preconditioned Schur complement system (9) and the ILU preconditioning of the original system (3) are two equivalent arithmetic procedures and therefore we observe the same number of iterations in both cases. However, the static condensation performs faster because the iteration matrix has less nonzero elements.

References

[1] A.M. Bruaset, A Survey of Preconditioned Iterative Methods, Longman Scientific & Technical, Harlow, 1995.
[2] R.D. Cook, Concepts and Applications of Finite Element Analysis, Wiley, New York, 1981.
[3] G.E. Karniadakis, S.J. Sherwin, Spectral/*hp* Element Methods for Computational Fluid Dynamics, Oxford University Press, New York, 2005.

[4] P. Šolín, Partial Differential Equations and the Finite Element Method, Wiley, Hoboken, 2006.
[5] P. Šolín, K. Segeth, I. Doležel, Higher-Order Finite Element Methods, Chapman & Hall/CRC Press, Boca Raton, 2003.
[6] P. Šolín, T. Vejchodský, M. Zítka, Orthogonal *hp*-FEM for elliptic problems based on a non-affine concept, in: A. Bermúdes, D. Gómez, P. Quintela, P. Salgado (Eds.), Numerical Mathematics and Advanced Applications, ENUMATH 2005, Springer, Berlin, 2006, pp. 683–690.