THE UNIVERSITY OF READING

# Fast algorithms for setting up the stiffness matrix in $hp$-FEM: a comparison

T. Eibner J.M. Melenk

**Abstract**

We analyze and compare different techniques to set up the stiffness matrix in the $hp$-version of the finite element method. The emphasis is on methods for second order elliptic problems posed on meshes including triangular and tetrahedral elements. The polynomial degree may be variable. We present a generalization of the Spectral Galerkin Algorithm of [7], where the shape functions are adapted to the quadrature formula, to the case of triangles/tetrahedra. Additionally, we study on-the-fly matrix-vector multiplications, where merely the matrix-vector multiplication is realized without setting up the stiffness matrix. Numerical studies are included.

## I. Introduction

The $hp$-version of the finite element method ($hp$-FEM) (see, e.g., the monographs [13], [11], [5] and the references therein) as well as the closely related spectral method (see, e.g., the survey article [1] and the references there) are well-established tools in computational structural and fluid mechanics. Typically, the bulk of the alphanumerical cost in the $hp$-FEM is in the numerical quadratures used to set up the stiffness matrix; this is in sharp contrast to the standard low order FEM ($h$-FEM), where most of the computational effort is spent on the solution of the resulting (linear) system. The present paper therefore considers different techniques for the rapid computation of the stiffness matrix in high order methods. As is standard in most FEM, all quadratures are done on a reference element $\widehat{K}$, which, in 2D, is either the reference square or the reference triangle; in 3D, the most important ones are the hexahedron, the tetrahedron, the prism, and the pyramid. Also the shape functions are defined on the reference element. We will discuss the generation of the element stiffness matrices for the following situation that is typical of scalar valued second order elliptic equations:

$$(S_K)_{ij} := a(\ _j, \ _i); \qquad a(u, v) := \int_{\widehat{K}} (A(x) \nabla u) \cdot \nabla v \, d\ ; \tag{1}$$

where the shape functions $\ _i \in \ := \{\ _i \mid i = 1, \dots,$ order

## II. SOME KEY IDEAS ILLUSTRATED IN 2D

### A. The idea of sum factorization

The idea of sum factorization can be motivated by the following problem, which mimics the computation of a mass or a stiffness matrix in 2D: Compute, for all double indices $(i_1, i_2), (j_1, j_2) \in \{1, \ldots, p\}^2$ the field $M$ with entries

$$M_{(i_1,i_2),(j_1,j_2)} := \sum_{k_1=0}^{q} \sum_{k_2=0}^{q} \varphi_{i_1}(k_1)\, \psi_{i_2}(k_2)\, \tilde{\varphi}_{j_1}(k_1)\, \tilde{\psi}_{j_2}(k_2)\, g(k_1, k_2) \tag{2}$$

where the functions $\varphi_i$, $\psi_i$, $\tilde{\varphi}_i$, $\tilde{\psi}_i$, and $g$ are given. The naive evaluation of all entries of $M$ requires $O(p^4(1+q)^2)$ floating point operations. Since in our applications $q = O(p)$, we arrive at a total cost of $O(p^6)$ to compute the $p^4$ entries of $M$. By rerranging the sums, this work can be reduced:

$$M_{(i_1,i_2),(j_1,j_2)} = \sum_{k_1=0}^{q} \varphi_{i}$$

We note that the bilinear form $\mathbf{a}$ is well-defined on $\widetilde{\mathbf{Q}}_{\mathbf{p}}$ because the polynomials $\mathbf{u}\ 2\ \widetilde{\mathbf{Q}}_{\mathbf{p}}$ are constant on the line $\ _{\mathbf{2}} = 1$ so that the term $\frac{\mathbf{1}}{\mathbf{1}\ \ _2} @\ _1 \mathbf{u}$, which seemingly has a singularity at $\ _{\mathbf{2}} = 1$, is in fact smooth there.

The basis on $\widehat{\mathbf{K}}$ is defined such that three goals are met: a) contains the space of $\mathbf{P}_{\mathbf{p}}$ of polynomial of degree $\mathbf{p}$ to ensure good approximation properties; b) leads to a fast evaluation of the stiffness matrix $\mathbf{S}_{\mathbf{K}}$; c) allows for an easy assembly of the element stiffness matrix $\mathbf{S}_{\mathbf{K}}$ into the global stiffness matrix. The last requirement effectively dictates that the set consist of “external shape functions” (which are typically split further into “vertex shape functions” and “edge shape functions”) and “internal shape functions”: the internal shape functions vanish on terms of the form $\frac{l_{i_2}^{2}(x)}{1-x}$ at $x = 1$ is then to be understood as taking the limit as $x \to 1$. This is merely a notational problem since the polynomials $l_{i_2}^{(2)}$, $l_{j_2}^{(2)}$ vanish at the endpoints $x = \pm 1$.

Upon expanding the matrix vector products inside the double sum we arrive at a sum with 4 four terms. For the technique of sum factorization, all four terms can be treated similarly. For example, for one of the "mixed" ones, we get

$$S^{II;mixed}_{(i_1;i_2);(j_1;j_2)} := \sum_{k=0}^{q}\sum_{l=0}^{q} \omega_k^{(1)} \omega_l^{(2)} l'_{i_1}(x_k^{(1)}) \frac{l_{i_2}(x_l^{(2)})}{1-x_l^{(2)}} A_{12}(x_k^{(1)}; x_l^{(2)}) l_{j_1}(x_k^{(1)}) \frac{l'_{j_2}(x_l^{(2)})}{1-x_l^{(2)}}.$$

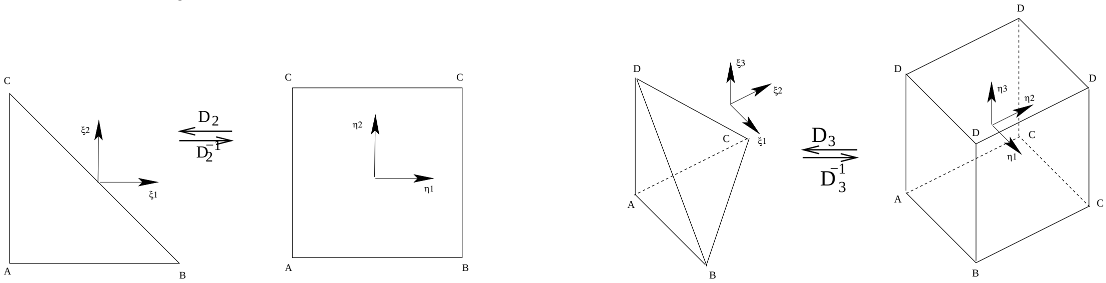
Using sum factorization techniques, the cost to evaluate all $(p-1)^4$ entries of $S^{II;mixed}$ is, as seen above, $O(p^4(1+q)+p^2(1+q)^2)$. Up to now, the fact that the functions $l_i^{(1)}$, $l_i^{(2)}$ are Lagrange interpolation polynomials with respect to some points was not relevant. If we choose the points $\{\xi_i^{(1)} \mid i = 0, \ldots, p\}$, $\{\xi_i^{(2)} \mid i = 0, \ldots, p\}$ to be subsets of the quadrature points, then the sets

$$\{k \in \{0, \ldots, q\} \mid l_{i_1}(x_k^{(1)}) \neq 0\}; \qquad \{l \in \{0, \ldots, q\} \mid l_{i_2}(x_l^{(2)}) \neq 0\};$$

have cardinality bounded by $1 + (q - p)$. Thus, from the ap]Tj 31 9.96264 Tf 1 0 0 1 300 686.86 Tm (1)Tj /R40 9.96264 Tf 7.68 0 Td Tj 4

*Theorem II.3: Let* $\widehat{\mathbf{K}} = \mathbf{T}^2$*, let* $\mathbf{A} \in \mathbf{L}^1(\widehat{\mathbf{K}})$ *be a matrix-valued function* $\mathbf{x} \mapsto \mathbf{A}(\mathbf{x}) \in \mathrm{R}^2$

Fig. 1. Transformations $D_2$ and $D_3$




employed. If the reference element $\widehat{K}$ is the reference triangle or tetrahedron, then it is natural to perform the quadrature on a square/hexahedron by a further transformation using the Duffy transformation. The following definition formalizes these notions:

*Definition III.1—reference elements:* We define the triangle, tetrahedron, and the $i$-th dimensional reference cube $Q^i$ by:

$$T^2 = \{(x;y) \mid \ 1 < x;y \ \wedge \ x+y < 0\};$$
$$T^3 = \{(x;y;z) \mid \ 1 < x;y;z \ \wedge \ x+y+z < \ 1\};$$
$$Q^i = ( \ 1;1)^i:$$

*Lemma III.2: The Duffy transformations* $D_2$, $D_3$ *are given by:*

$$D_2 \ : \ ( \ _1; \ _2) \mapsto \ \frac{1}{2}(1+ \ _1)(1 \ \ _2) \ \ 1; \ _2\Big); \tag{10}$$

$$D_3 \ : \ ( \ _1; \ _2; \ _3) \mapsto \ \frac{1}{4}(1+ \ _1)(1 \ \ _2)(1 \ \ _3) \ \ 1; \frac{1}{2}(1+ \ _2)(1 \ \ _3) \ \ 1; \ _3\Big): \tag{11}$$

*Then*

$$j \det D_2^0 j = \ \frac{1 \ \ _2}{2}\Big); \qquad j \det D_3^0 j = \ \frac{1 \ \ _2}{2}\Big) \ \frac{1 \ \ _3}{2}\Big)^2; \qquad \textit{and} \quad T^i = D_i(Q^i); \quad i = 2;3:$$

If $\widehat{K} = T^d$, then the chain rule allows us to rewrite the integral (1) as follows:

$$a(u;v) = \int_{T^d} (A(x) \nabla u) \ \nabla v \, d \ = \int_{Q^d} \ (\nabla u \ \ D_d); A(\nabla v \ \ D_d) \ j \det D_d^0 j d \ ; \qquad A := (D_d^0) \ ^1(A \ \ D_d)(D_d^0) \ ^T: \tag{12}$$

Once the integration over $T^d$ is rewritten as an integration over $Q^d$, we may employ standard tensor product quadrature techniques of Gauss, Gauss-Lobatto, or, more generally, of Gauss-Jacobi-Lobatto type.

## B. Shape functions on $T^2$ and $T^3$

The stiffness matrix $S_K$ is completely fixed once the shape functions are chosen. We will discuss two different choices of this set below. Some key considerations that determine the construction are:

- Since the quadrature over $T^d$ is reformulated as a quadrature over $Q^d$ in (12), it is advantageous to define the shape functions explicitly as functions on $Q^d$ and thereby only implicitly on $T^d$ by means of the Duffy transformation. In view of the fact that sum factorization techniques will employed, the shape functions (as defined on $Q^d$) have product structure.
- In order to facilitate variable polynomial degree distributions in $hp$-FEM, a (possibly different) polynomial degree is associated with each of the topological entities, i.e., each edge $e$, face $f$ (in 3D), and the element has its own degree.

We will discuss two sets of shape functions. The first set $^{(KS)}$ is taken from [5] and contains shape functions with a tensor product structure suitable for applying sum factorization. The second set $^{(Lag)}$ is a modification of $^{(KS)}$ that contains shape functions where the so-called internal shape functions are adapted to the quadrature rules. First some abbreviations:

*Abbreviations III.3:* Let 0f dif

(

a55 d0 ali 7 6733 05f d (5 ( ) 6264 f i1 45
i

where

$$^{(KS)}_{B} := {}^{(KS)}_{B} \quad D_2{}^{1} = \quad D_2{}^{1} j \quad 2 \quad {}^{(KS)}_{B} \quad ; \qquad {}^{(Lag)}_{B} := {}^{(Lag)}_{B} \quad D_2{}^{1} = \quad D_2{}^{1} j \quad 2 \quad {}^{(Lag)}_{B}$$

and the sets $_{B}$ are sets of functions defined on $Q^2$ given by

$$
\begin{aligned}
{}^{(KS)}_{0} = {}^{(Lag)}_{0} = {}_{0} &:= ff_3(\ {}_2)g;\\
{}^{(KS)}_{1} = {}^{(Lag)}_{1} = {}_{1} &:= ff_2(\ {}_1)f_2(\ {}_2); f_3(\ {}_1)f_2(\ {}_2)g;\\
{}^{(KS)}_{2} &:= \quad f_1(\ {}_1)f_2^{p+1}(\ {}_2)P_{p\ \ 1}^{(1;1)}(\ {}_1)\ j\ p = 1; :::; p_{AB} \quad 1 \quad ;\\
{}^{(Lag)}_{2} &:= \quad f_1(\ {}_1)f_2^{2}(\ {}_2)P_{p\ \ 1}^{(1;1)}(\ {}_1)\ j\ p = 1; :::; p_{AB} \quad 1 \quad ;\\
{}^{(KS)}_{3} = {}^{(Lag)}_{3} = {}_{3} &:= \quad f_2(\ {}_1)f_1(\ {}_2)P_{q\ \ 1}^{(1;1)}(\ {}_2)\ j\ q = 1; :::; p_{AC} \quad 1 \quad ;\\
{}^{(KS)}_{4} = {}^{(Lag)}_{4} = {}_{4} &:= \quad f_3(\ {}_1)f_1(\ {}_2)P_{q\ \ 1}^{(1;1)}(\ {}_2)\ j\ q = 1; :::; p_{BC} \quad 1 \quad ;\\
{}^{(KS)}_{5} &:= \quad f_1(\ {}_1)f_1(\ {}_2)f_2^{p}(\ {}_2)P_{p\ \ 1}^{(1;1)}(\ {}_1)P_{q\ \ 1}^{(2p+1;1)}(\ {}_2)\ \begin{matrix} 1 & p & p_K & 2 & \\ 1 & q & p_K & p & 1 \end{matrix} \quad ;\\
{}^{(Lag)}_{5} &:= \quad f_1(\ {}_1)f_1(\ {}_2)f_2(\ {}_2)C_p l_p^{(N_1)}(\ {}_1)C_q l_q^{(N_2)}(\ {}_2)\ \begin{matrix} 1 & p & p_K & 1 \\ 1 & q & p_K & 2 \end{matrix} \quad ;
\end{aligned}
$$

where the constants $C_p$ are scaling parameters.

*Definition III.5—shape functions on* $T^3$*:* Let $T^3$ be the reference tetrahedron and let $\mathrm{p} = (p_{AB}; :::; p_{CD}; p_{ABC}; :::; p_{BCD}; p_K)$ be a degree vector. As in the 2D case, the subscripts $AB; :::; BCD$ represent edges and faces of the tetrahedron $T^3$ with vertices $A$, $B$, $C$, $D$. For $i = 1; 2; 3$ let $N_i = f\ {}^{(i)}_{k} jk = 1; :::; p_K \quad 3g$ be a nodal set with and $l_k^{(N_i)}$ the $k$-th Lagrange interpolation polynomial with respect to $N_i$. Then we define

$$^{(KS)} = \bigcup_{B=0}^{13} {}^{(KS)}_{B} \quad \text{and} \quad {}^{(Lag)} = \bigcup_{B=0}^{13} {}^{(Lag)}_{B};$$

where

$$^{(KS)}_{B} := {}^{(KS)}_{B} \quad D_3{}^{1} = \quad D_2{}^{1} j \quad 2 \quad {}^{(KS)}_{B} \qquad {}^{(Lag)}_{B} := {}^{(Lag)}_{B} \quad D_3{}^{1} = \quad D_2{}^{1} j \quad 2 \quad {}^{(Lag)}_{B}$$

and the sets $_{B}$ are sets of functions defined on $Q^3$ given by:

$$^{(KS)}_{0} = {}^{(Lag)}_{0} = {}_{0} := ff_3(\ {}_3)g;; \quad )$$

D f 1 0 Tf 1 Tf -1 1 Tf 1 f 2

$${}^{(KS)}_{11} = {}^{(Lag)}_{11} = {}_{11} := f_2(\;_1) f_1(\;_2) P_q^{(1;1)}{}_1(\;_2) f_1(\;_3) f_2^{q}(\;_3) P_r^{(2q+1;1)}{}_1(\;_3)$$

$$1 \quad q \quad p_{ACD} \quad 2; \, 1 \quad r \quad p_{ACD} \quad q \quad 1 \; ;$$

$${}^{(KS)}_{12} = {}_{\mathrm{K}}\,\frac{\mathrm{Soag}}{11}\,)_{11} =$$

*where*

$$\mathbf{r}\;{}_{\mathbf{i}}^{(\mathbf{K})} := \begin{cases} \qquad\qquad \frac{\mathbf{1}}{(\mathbf{1}\quad{}_2)}\frac{\partial}{\partial\;{}_1};\frac{\partial}{\partial\;{}_2}\lrcorner^{\mathbf{T}} & : \quad \mathbf{d} = 2 \\ \frac{\mathbf{1}}{(\mathbf{1}\quad{}_2)(\mathbf{1}\qquad)}\frac{\partial}{\partial\;{}_1}; & \end{cases}$$

### B. Sum factorization

Considering Definition III.4 and Definition III.5 we observe that after transformation

*Remark V.4:* For each pair $(B; B')$ a separate quadrature rule could be chosen. In particular for blocks with low polynomial degree this might lead to further savings.
*Remark V.5:* The precomputing of the shape functions and coefficient matrix in step 3 and 4 is done due to the fact that evaluations of the shape functions and coefficient matrix can be very expensive, especially for large polynomial degrees or coefficients with a complicated structure. The precomputations of step 3 and 4 lead to a considerable speed-up, since we have to evaluate the shape functions and coefficient matrix just once for all quadrature points.
*Remark V.6:* It is not necessary to perform the precomputations of step 3 for each element of a meshing $\mathcal{T}$. Provided the quadrature rules depend only on the internal polynomial degree and due to the fact that $p_e, p_f \leq p_K$ for all edges and faces of the element K, it suffices to compute the arrays of step 3 just once for each $p_K \in \{1, \dots, p_{max}\}$ and assumed uniform polynomial degree distribution, that is $p_e = p_f = p_K$ for all edges and faces.

### C. Spectral Galerkin method

In the last subsection we already exploited the tensor product structure of the shape functions. If, however, the shape functions and the quadrature rules are adapted to each other, then a further reduction of the complexity is possible. To that end we consider in the following quadrature rules of the form

$$\mathrm{QR}^i = S^{(i)} \times W^{(i)} = \{(\xi_0^{(i)}, \omega_0^{(i)}), \dots, (\xi_{q_i}^{(i)}, \omega_{q_i}^{(i)})\}, \qquad \mathrm{QR} = \mathrm{QR}^1 \otimes \dots \otimes \mathrm{QR}^d$$

which incorporate the $\det |D_d'|$ terms of (13) in conjunction with the modified shape functions $\varphi^{(Lag)}$, where the nodal sets $N^{(i)}$ are subsets of the quadrature points, that is $N^{(i)} \subseteq S^{(i)}.$ Considering the shape functions of $\varphi^{(Lag)}$, we additionally observe a simpler tensor product stucture as in the general cases (13) and (14)

$$\varphi^{(Lag)} = \varphi_{(B;k_1;k_2)}(\xi) = g^{(1)}_{B;k_1}(\xi_1) g^{(2)}_{B;k_2}(\xi_2) \mid 1 \leq k_i \leq K_i(B)$$

for $d = 2$ and

$$\varphi^{(Lag)}_{13} = \varphi_{(13;k_1;k_2;k_3)}(\xi) = g^{(1)}_{13;k_1}(\xi_1) g^{(2)}_{13;k_2}(\xi_2) g^{(3)}_{13;k_3}(\xi_3) \mid 1 \leq k_i \leq K$$

for $d = 3$. Evaluating the gradient of the interior bubble shape functions at the quadrature points, we obtain, due to the adaption of shape functions and quadrature rules, a considerable number of zeros. Thus, we can replace the sums in Algorithm V.2 by the following pattern:

$$\sum_{l_1=0}^{q_1} g^{(1)}_{(B;r;k_1)} g^{(1)}_{(B';r';k_1')} C_{r';r}(\xi^1_{l_1}, \xi^2_{l_2}, \dots) \rightarrow \sum_{l_i \in NZ^1_{r,r'}[k_1;k_1']} g^{(1)}_{(B;r;k_1)} g^{(1)}_{(B';r';k_1')} C_{r';r}(\xi^1_{l_1}, \xi^2_{l_2}, \dots);$$

where the sets of relevant indices

$$NZ^{(1)}_{(r;r')}[k_1; k_1'] := \{l_1 \in \{0, \dots, q^{(1)}\} \mid g^{(1)}_{(B;r;k_1)} g^{(1)}_{(B'}$$

gp

3) Find a permutation $(i_1; i_2; i_3)$ with $W_{(i_1;i_2;i\;)} \quad W_{(i'_1;i'_2;i')}$ for all $(i^0_1; i^0_2; i^0_3)$.
4) For $1 \quad k_i \quad K_i(B)$, $1 \quad k^0_i \quad K_i(B^0)$ and $0 \quad l_i \quad q_i$ compute the auxiliary arrays

$$H^{(1)}[k_i\;; k^0_i\;; l_{i_1}; l_{i_2}] = \sum_{l_i\;\; 2NZ\; i} F^{(i\;)}[k_i\;; k^0_i\;; l_i\;] C(r^0; r; l_1; l_2; l_3) !^{(i\;)}_{l_i}$$

$$H^{(2)}[l_{i_1}; k_i\;; k^0_i\;; k_{i_2}; k^0_{i_2}] = \sum_{l_{i_2}\, 2NZ\; i_2} F^{(i_2)}[k_{i_2}; k^0_{i_2}; l_{i_2}] H^{(1)}[l_{i_1}; l_{i_2}; k_i\;; k^0_i\;] !^{(i_2)}_{l_{i_2}}$$

5) For all $1 \quad k_i \quad K_i(B)$, $1 \quad k^0_i \quad K_i(B^0)$ add

$$S_K[(B; k_1; k_2; k_3)][(B^0; k^0_1; k^0_2; k^0_3)] \quad += \sum_{l_{i_1}\, 2NZ\; i_1} F^{(i_1)}[k_{i_1}; k^0_{i_1}; l_{i_1}] H^{(2)}[l_{i_1}; k_i\;; k^0_i\;; k_{i_2}; k^0_{i_2}] !^{(i_1)}_{l_{i_1}}$$

Assuming quadradure rules of order

$$q_i = \begin{cases} p_K + q & : \; d = 2; \; i = 1 \\ p_K \quad 1 + q & : \; d = 2; \; i = 2 \\ p_K + q & : \; d = 3 \end{cases} \tag{15}$$

with $q \quad 0$ and $q = O(1)$, we obtain, completely analogously to [7], a complexity of $O(p^{2d}_K)$ for setting up the element stiffness matrix $S_K$. Thus, asymptotically Algorithm V.7 is superior to Algorithm V.2 if we consider the computing time for setting up $S_K$. However, the critical point is that due to the increased number of internal shape functions for large polynomial degrees $p_K$ the advantage in setting up the stiffness matrix will be offset if we consider an $hp$-implementation making use of static condensation, since, at least asymptotically, the cost of static condensation dominates the total cost per element. Only numerical tests, which we present below, can tell whether there exists a range $f p_0; :::; p_N g$ of polynomial degrees where it is preferable to use Algorithm V.7.

## VI. Remarks on static condensation and precomputed arrays

### A. Static Condensation

In $hp$-FEM it is customary to perform static condensation. The partition of the shape functions into external E=$f$vertex, edge, face$g$ and I=internal shape functions implies a corresponding block structure of the local element stiffness matrices $S_K$ as well as of the global stiffness matrix $S_{glob}$. That is:

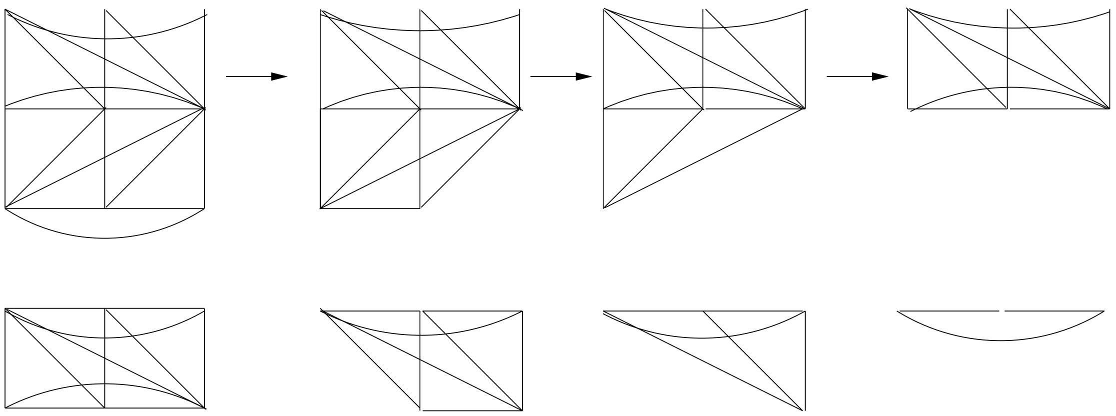
$$S_K = \begin{bmatrix} S^{EE}_K & S^{EI}_K \\ S^{IE}_K & S^{II}_K \end{bmatrix} \qquad S_{glob} = \begin{bmatrix} S^{EE} & S^{EI} \\ S^{IE} & S^{II} \end{bmatrix} :$$

Due to the support properties of the internal shape functions, the matrix $S^{II}$ is block diagonal with $S^{II} = \mathbf{diag}(S^{II}_K)$. In static condensation, the Schur complement is formed by eliminating the internal shape functions, $S^c = S^{EE} \quad S^{EI}(S^{II})\;\; ^1 S^{IE}$; which leads (at least foK least

5) Initialize $\mathbf{b} = 0$
6) Compute the auxiliary arrays

$$\mathbf{H}^{(1)}[\mathbf{r}^0;$$

Fig. 3. Proof of Lemma VII.3



The main idea to achieve a speed-up compared to Algorithm VII.1 is to exploit that the shape functions and the quadrature rules are adapted to each together with the simpler structure of the shape functions. Considering

$$b_{(B,k_1,k_2)} = \sum_{\{r,r',B'\}} \sum_{\{k_1',k_2',l_1,l_2\}} \omega^{(1)}_{l_1}\omega^{(2)}_{l_2} G^{(2)}_{B',r'}(k_2',l_2) G^{(1)}_{B',r'}(k_1',l_1) C_{r',r}(l_1,l_2) G^{(1)}_{B,r}(k_1,l_1) G^{(2)}_{B,r}(k_2,l_2) v_{(B',k_1',k_2')},$$

with $G^{(i)}_{B,r}(k_i,l_i)$, $C_{r',r}(l_1,l_2)$ and $v_{(B',k_1',k_2')}$ as in the previous section, we have 24 possible summation orders for $\{k_1',k_2',l_1,l_2\}$. Thus, we want to find and apply the cheapest, or at least a good summation order depending on $B, B', r, r'$. However, some of these permutations can be excluded a priori:

- $(k_i', l_j, \ , \ )$ since a calculation reveals that for all $i,j \in \{1,2\}$ $(l_j, k_i', \ , \ )$ is equal to or more efficient than $(k_i', l_j, \ , \ )$.
- $(k_i', k_{\bar{i}}', l_j, l_{\bar{j}})$ since this is equivalent to setting up a block of the stiffness matrix.
- $(\ , \ , k_i', l_j)$ du to an argumentation as in Remark VII.4.
- $(l_i, k_{\bar{i}}', l_{\bar{i}}, k_i')$ since an efficient implementation becomes equivalent to $(\ , \ , k_i', l_j)$.

Thus, it remain four permutations of type $(l_j, l_{\bar{j}}, k_i', k_{\bar{i}}')$ and two permutations of type $(l_j, k_j', l_{\bar{j}}, k_{\bar{j}}')$. Considering $(l_1, k_1', l_2, k_2')$, we have

$$b_{(B,k_1,k_2)} = \sum_{l_1} \omega^{(1)}_{l_1} G^{(1)}_{B,r}(k_1,l_1) \sum_{r',B',k_1'} G^{(1)}_{B',r'}(k_1',l_1) H^{(2)}_{r,r',B,B'}(k_2,k_1',l_1),$$

where

$$H^{(2)}_{r,r',B,B'}(k_2,k_1',l_1) = \sum_{l_2} H_{r',B'}(k_1',l_2) C_{r',r}(l_1,l_2) G^{(2)}_{B,r}(k_2,l_2) \tag{17}$$

$$H_{r',B'}(k_1',l_2) \quad (1) = \sum_{k_2'} \omega^{(2)}_{l_2} G^{(2)}_{B',r'}(k_2',l_2) v_{(B',k_1',k_2')}. \tag{18}$$

Since $G^{(1)}_{B,r}(k_1,l_1)$ depends on the same entities as the auxiliary array $H^{(2)}$, we can avoid setting up the array $H^{(2)}_{r,r',B,B'}(k_2,k_1',l_1)$ and evaluate the sum with almost the same or even less work as:

$$b_{(B,k_1,k_2)} = \sum_{l_1} \omega^{(1)}_{l_1} G^{(1)}_{B,r}(k_1,l_1) \sum_{r',B',k_1'} \sum_{l_2} G^{(1)}_{B',r'}(k_1',l_1) H_{r',B'}(k_1',l_2) C_{r',r}(l_1,l_2) G^{(2)}_{B,r}(k_2,l_2);$$

which in turn is equivalent to

$$b_{(B,k_1,k_2)} = \sum_{l_1} \omega^{(1)}_{l_1} G^{(1)}_{B,r}(k_1,l_1) \sum_{l_2} \sum_{r',B',k_1'} G^{(1)}_{B',r'}(k_1',l_1) H_{r',B'}(k_1',l_2) C_{r',r}(l_1,l_2) G^{(2)}_{B,r}(k_2,l_2).$$

Thus, since we can argue in the same way for $(l_2, k_2', l_1, k_1')$, we can restrict our algorithm to consider the four permutations of type $(l_j, l_{\bar{j}}, k_i', k_{\bar{i}}'\}$.

*Algorithm VII.6—spectral matrix vector multiplication on the fly:*

1) For $i = 1,2$ choose quadrature rules

$$\mathrm{QR}^{(i)} = \{(\ ^{(i)}_{l_i}, \omega^{(i)}_{l_i}) \mid l_i = 0,\dots,q_i\}$$

which incorporate the $\det |D_2'|$ term of (13).

2) For all $1 \ r \ 2$ and $0 \ B \ 5$ let

$$r_r \ {}_B = \ g^{(1)}_{(B,r,k_1)}(\ _1) g^{(2)}_{(B,r,k_2)}(\ _2) \quad 1 \ k_i \ K_i(B)$$

3) For $1 \ r \ 2$, $0 \ B \ 5$, $0 \ l_i \ q_i$, $1 \ k_i \ K_i(B)$ compute

$$G^{(1)}(B,r,k_1,l_1) = g^{(1)}_{B,r,k_1}(\ ^{(1)}_{l_1}) \qquad NZ^{(1)}(B,r,k_1) = \{l_1 \mid G^{(1)}(B,r,$$ Tj' /R43 6.97385 Tf r3:02253 2a 4.4396d (r)Tj

For 1 r

3)

as follows: For all 1 $\mathbf{r^0}$ 2, 0 $\mathbf{B^0}$ 5 compute

a) $\mathbf{S_1} := \sum_{\mathbf{k}_1'} \mathbf{N Z^{(1)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k}_1^0)$ and $\mathbf{S_2} := \sum_{\mathbf{k}_2'} \mathbf{N Z^{(2)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k}_2^0)$

b) If $[(\mathbf{q_2}+1)\mathbf{S_1}+\mathbf{K_1}(\mathbf{B}^0)\mathbf{S_2}]$ $[(\mathbf{q_1}+1)\mathbf{S_2}+\mathbf{K_2}(\mathbf{B}^0)\mathbf{S_1}]$

Initialize $\mathbf{H^{(1)}}[\mathbf{k}_1^0;\mathbf{l_2}] = 0$

Add $\mathbf{H^{(1)}}[\mathbf{k}_1^0;\mathbf{l_2}] += \mathbf{v}_{(\mathbf{B}';\mathbf{k}_1';\mathbf{k}_2')}\,!^{(2)}_{\mathbf{l}_2}\mathbf{G^{(2)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k}_2^0;\mathbf{l_2})$

for all 1 $\mathbf{k}_1^0$ $\mathbf{K_1}(\mathbf{B}^0)$, 1 $\mathbf{k}_2^0$ $\mathbf{K_2}(\mathbf{B}^0)$, $\mathbf{l_2}$ 2 $\mathbf{N Z^{(2)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k_2})$.

Add $\mathbf{H^{(2)}}[\mathbf{r}^0;\mathbf{l_2};\mathbf{l_1}] += \,!^{(1)}_{\mathbf{l}_1}\mathbf{H^{(1)}}[\mathbf{k}_1^0;\mathbf{l_2}]\mathbf{G^{(1)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k}_1^0;\mathbf{l_1})$

for all 1 $\mathbf{k}_1^0$ $\mathbf{K_1}(\mathbf{B}^0)$, $\mathbf{l_1}$ 2 $\mathbf{N Z^{(1)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k_1})$, 0 $\mathbf{l_2}$ $\mathbf{q_2}$.

c) If $[(\mathbf{q_2}+1)\mathbf{S_1}+\mathbf{K_1}(\mathbf{B}^0)\mathbf{S_2}] > [(\mathbf{q_1}+1)\mathbf{S_2}+\mathbf{K_2}(\mathbf{B}^0)\mathbf{S_1}]$

Initialize $\mathbf{H^{(1)}}[\mathbf{k}_2^0;\mathbf{l_1}] = 0$

Add $\mathbf{H^{(1)}}[\mathbf{k}_2^0;\mathbf{l_1}] += \mathbf{v}_{(\mathbf{B}';\mathbf{k}_1';\mathbf{k}_2')}\,!^{(1)}_{\mathbf{l}_1}\mathbf{G^{(1)}}(\mathbf{B}^0;\mathbf{r}^0;\mathbf{k}_1^0;\mathbf{l_1})$

for all 1 $\mathbf{k}_1^0$ $\mathbf{K_1}(\mathbf{B}^0)$, 1 $\mathbf{k}_2^0$ $\mathbf{K_2}(\mathbf{B}^0)$, $\mathbf{l}$ $_2)\,!^{(1)}_{\mathbf{l}}$

Add $\mathbf{H}$

with

$$\mathbf{W_H}(\mathbf{B}^0;\mathbf{r}^0) = \min \mathbf{f}\ (\mathbf{q_2}+1)\mathbf{S_1}(\mathbf{B}^0;\mathbf{r}^0) + \mathbf{K_1}(\mathbf{B}^0)\mathbf{S_2}(\mathbf{B}^0;\mathbf{r}^0); (\mathbf{q_1}+1)\mathbf{S_2}(\mathbf{B}^0;\mathbf{r}^0) + \mathbf{K_2}(\mathbf{B}^0)\mathbf{S_1}(\mathbf{B}^0;\mathbf{r}^0)\ \mathbf{g};$$
$$\mathbf{W_b}(\mathbf{B};\mathbf{r}) = \min \mathbf{f} 2(\mathbf{q_2}+1)\mathbf{S_1}(\mathbf{B};\mathbf{r}) + \mathbf{K_1}(\mathbf{B})\mathbf{S_2}(\mathbf{B};\mathbf{r}); 2(\mathbf{q_1}+1)\mathbf{S_2}(\mathbf{B};\mathbf{r}) + \mathbf{K_2}(\mathbf{B})\mathbf{S_1}(\mathbf{B};\mathbf{r})\ \mathbf{g}$$

and $\mathbf{S_i}(\mathbf{B};\mathbf{r}) = \ _{\mathbf{k}_i}\ \mathbf{N}\,\mathbf{Z}^{(\mathbf{i})}(($

Fig. 5. Setting up the element stiffness matrix - computing time - blockwise - 2D

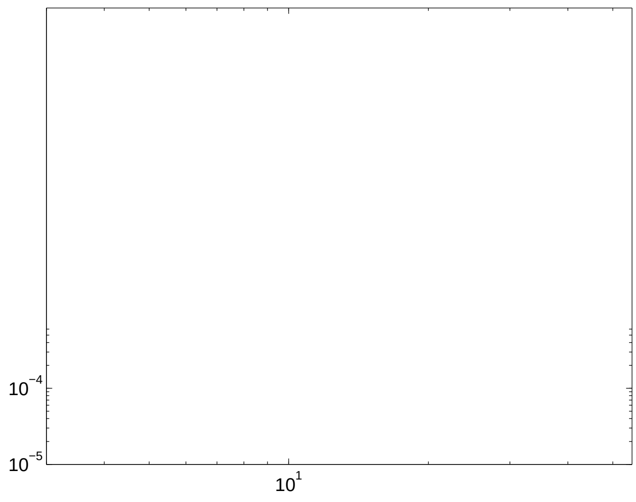


TABLE II

Fig. 6. Setting up the element stiffness matrix - computing time - 3D

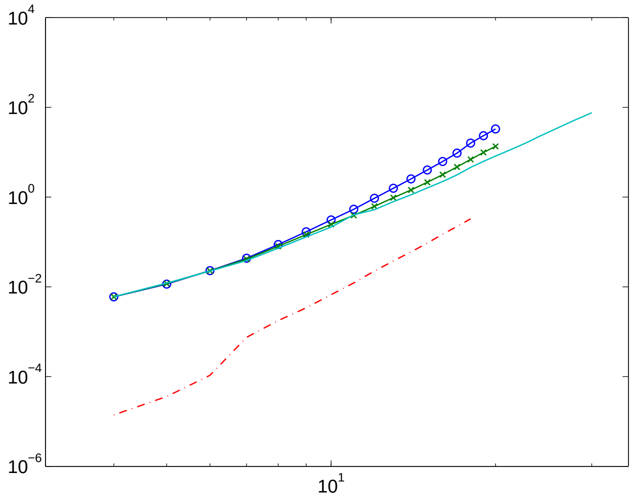


Fig. 9. Different quadrature rules - assemblation and matrix vector multiplication - computing time - 3D