Journal of Computational and Applied Mathematics 233 (2010) 3086–3094

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

PDE-independent adaptive *hp*-FEM based on hierarchic extension of finite element spaces

Pavel Solin [a,b,*], David Andrs [a,b], Jakub Cerveny [b], Miroslav Simko

[a] Department of Mathematics and Statistics, University of Nevada, Reno, USA
[b] Institute of Thermomechanics, Dolejskova 5, Prague, Czech Republic

ARTICLE INFO

Article history:
Received 17 March 2009

MSC:
65L50
68U20
74S05

Keywords:
Automatic adaptivity
Higher-order finite elements
hp-FEM
Hierarchic basis extension
Embedded stiffness matrices

ABSTRACT

We present a novel approach to automatic adaptivity in higher-order finite element methods (*hp*-FEM) which is free of analytical error estimates. This means that a computer code based on this approach can be used to solve adaptively a wide range of PDE problems. A posteriori error estimation is done computationally via hierarchic extension of finite element spaces. This is an analogy to embedded higher-order methods for ODE. The adaptivity process yields a sequence of embedded stiffness matrices which are solved efficiently using a simple combined direct-iterative algorithm. The methodology works equally well for standard low-order FEM and for the *hp*-FEM. Numerical examples are presented.

1. Introduction

The *hp*-FEM is a modern version of the finite element method capable of achieving exponential convergence rates by combining optimally finite elements of different sizes (h) and polynomial degrees (p) [1–8]. The goal of our work is to develop adaptive *hp*-FEM algorithms with controlled accuracy in both space and time for complex multiphysics engineering problems. Here we face two major complications:

- Very few analytical error estimates are available for arbitrary-order finite elements.
- Very few analytical error estimates exist for multiphysics problems.

Taking into account the complicated structure of PDE systems describing realistic multiphysics processes, and the challenges related to higher-order finite element discretizations, a question arises *if analytical error estimates for such problems will ever be available at all.*

During the past two decades, analytical error estimates have been derived mainly for single-physics problems solved by means of low-order methods (such as piecewise-linear FEM). Analytical error estimates differ a lot from one PDE to another, and are virtually impossible to combine into one universal methodology covering a wide range of multiphysics problems. Their practical application can be problematic, since sometimes they contain tuning parameters or dubious constants which need to be approximated using additional nontrivial mathematical tricks. The practitioner may not be skilled enough in mathematics or have enough time to do that. In the *hp*-FEM, an element can be refined in many different ways (typically around one hundred on 2D elements and several hundreds in 3D) [5], and thus an elementwise-constant error estimate is not enough — one needs to know the *shape of the error* on every element, as a function.

* Corresponding author at: Department of Mathematics and Statistics, University of Nevada, Reno, USA. Tel.: +1 775 848 7892.
E-mail address: solin@unr.edu (P. Solin).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.05.030

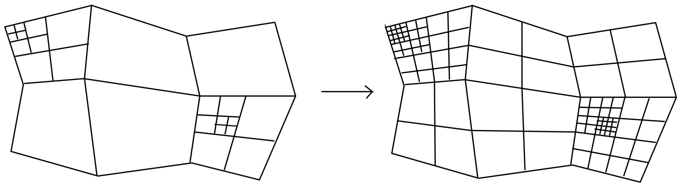

Fig. 1. Coarse mesh τ_0^c and the globally refined reference mesh τ_0^r.

In order to obtain a method which is applicable to a wide range of PDE including multiphysics problems, and which works for both the standard FEM and the *hp*-FEM, we use an a posteriori error estimate computed using *approximation pairs* (consisting of approximations with different orders of accuracy). The idea is analogous to embedded higher-order methods for ODE [9]. The adaptive strategy is described in Section 2. A novel technique for hierarchic extension of polynomial spaces on arbitrary-order quadrilateral and triangular finite elements is described in Section 3. In Section 4 we introduce a novel technique for efficient solution of the sequence of embedded discrete problems obtained during the adaptivity process. Section 5 uses a model problem to study the performance of the method.

2. PDE-independent adaptive strategy

For the reasons explained above, we need to replace traditional analytical error estimates with a more universal approach working for a wide range of PDE including multiphysics problems, as well as for the *hp*-FEM. We resort to an idea that has been used in the ODE community for a long time: In every time step, embedded adaptive higher-order ODE methods compute efficiently two approximations with different orders of accuracy (an approximation pair). The error is estimated using their difference. If the difference is large, then the last step is repeated with smaller time step size, otherwise the time step size is increased. Of course, the PDE case is technically more complicated:

Initial step — construction of the approximation pair

In the initial step, we construct a pair of approximations with different orders of accuracy. We begin with a coarse mesh τ_0^c, and construct a reference mesh τ_0^r using global refinement of τ_0^c. This is done by increasing the polynomial degrees of all elements in the mesh τ_0^c by one and subdividing them uniformly in space, as illustrated in Fig. 1.

Next we build the stiffness matrix on the reference mesh τ_0^r, its LU factorization (using sophisticated multifrontal algorithms of the direct solver UMFPACK [10]), and solve on τ_0^r. By u_0^r let us denote the reference solution on τ_0^r. The initial algorithmic step is finished by projecting the reference solution u_0^r onto the coarse mesh τ_0^c. Note that an orthogonal projection yields the best approximation of u_0^r on the coarse mesh τ_0^c, and moreover, it is easier and faster to perform than to solve the finite element problem on the coarse mesh τ_0^c. The finite element problem is never solved on the coarse mesh. The multi-mesh *hp*-FEM [6] described above is used to perform efficiently all operations involving the coarse and reference meshes.

One step of the adaptive algorithm — update of the approximation pair

Let us set $k := 0$. The difference $e_k = u_k^r - u_k^c$ is used as an error function on the mesh τ_k^c. Note that such an error estimator does not depend on the underlying equation and that it works without any limitations both for standard FEM and *hp*-FEM. If the global norm of the function e_k is less than a prescribed tolerance *TOL*, the computation stops. Otherwise, one marks for refinement elements in the coarse mesh τ_k^c with largest approximation error, until the number of newly added degrees of freedom (DOF) reaches a user-defined number N_{ref}. After performing these local refinements, one obtains a new coarse mesh τ_{k+1}^c. Analogous local refinements are done to the reference mesh τ_k^r so that the new reference mesh τ_{k+1}^r corresponds to a global refinement of the new coarse mesh τ_{k+1}^c. The finite element basis on τ_k^r is extended hierarchically to a basis on τ_{k+1}^r (to be described in Section 3). Hence, the stiffness matrix S_k^r corresponding to the mesh τ_k^r is subset of the new stiffness matrix S_{k+1}^r on the mesh τ_{k+1}^r. This fact is used to optimize the solution of the discrete problem on the mesh τ_{k+1}^r (to be described in Section 4).

3. Hierarchic basis extensions on higher-order elements

Let us consider a triangular element $K_{h,p}$ of degree $p \geq 1$. The corresponding polynomial space $P^p(K_{h,p})$ of dimension $(p+1)(p+2)/2$ contains 3 vertex functions associated with the vertices, $3(p-1)$ edge functions associated with the edges, and $(p-1)(p-2)/2$ bubble functions which are local to the element interior [5]. By the symbol $K_{h/2,p+1}$ let us denote the enriched element obtained by increasing the polynomial degree of $K_{h,p}$ by one and splitting it uniformly in space, as illustrated in Fig. 2.

It is easy to calculate that the dimension of the enriched space on $K_{h/2,p+1}$ is $2p^2 + 7p + 6$: The standard *hp*-FEM basis on $K_{h/2,p+1}$ contains 6 vertex functions (one per vertex), $9p$ edge functions (p per edge) and $4p(p-1)/2$ bubble functions [$p(p-1)/2$ per subelement]. In order to create a new basis on $K_{h/2,p+1}$ containing the original basis on $K_{h,p}$ as a subset, let us begin with the $(p+1)(p+2)/2$ basis functions from $K_{h,p}$. Next we add 3 vertex functions associated with the new vertices (denoted by black dots in Fig. 3).

Next we add $p-1$ edge functions of degrees $2, 3, \ldots, p$ to each of the six edges highlighted in Fig. 4. The set of edge functions is completed by adding one edge function of degree $p+1$ to each of the 9 edges of $K_{h/2,p+1}$. Next, we pick any

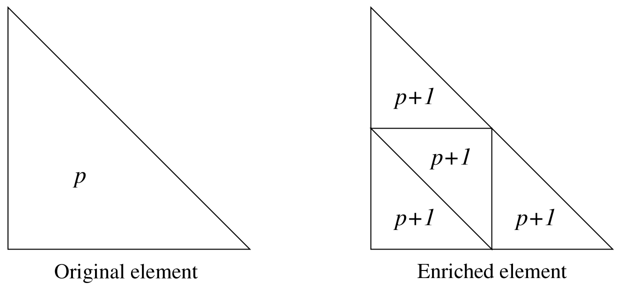

Fig. 2. Triangular element $K_{h,p}$ and the corresponding enriched element $K_{h/2,p+1}$.

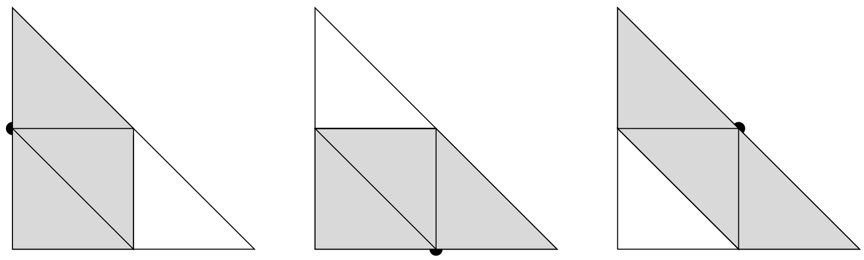

Fig. 3. New vertex functions added to the basis of $K_{h/2,p+1}$.

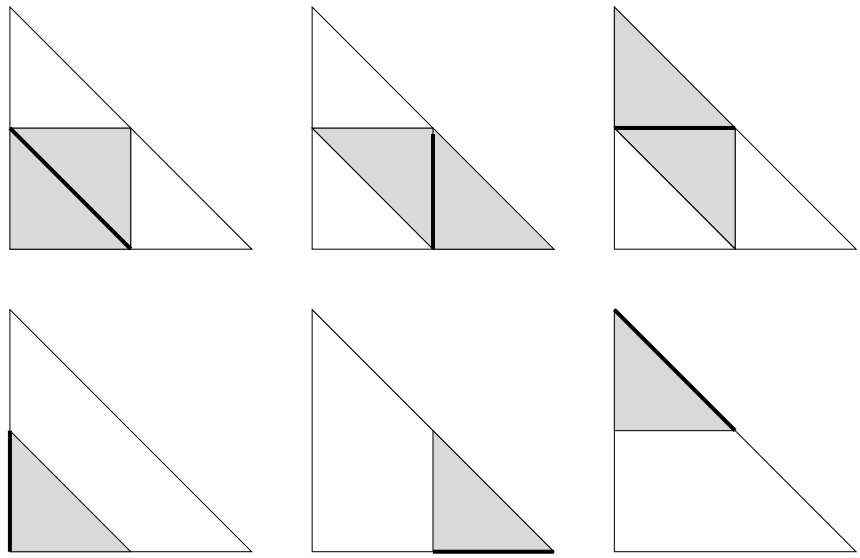

Fig. 4. New edge functions of degrees 2, 3, ..., p added to the basis of $K_{h/2,p+1}$.

three subelements (for example those highlighted in Fig. 5), and add to each of them $(p-1)(p-2)/2$ bubble functions of degrees $3, 4, \ldots, p$. In a final step, we add $p-1$ bubble functions of degree $p+1$ to each of the four subelements.

Proposition 3.1. *The* $2p^2+7p+6$ *functions on* $K_{h/2,p+1}$ *defined above are linearly independent, and they generate the same piecewise-polynomial space as the standard hp-FEM basis on* $K_{h/2,p+1}$.

The proof of this lemma is elementary — it is easy to check that the piecewise-polynomial spaces generated by the original and new basis are the same, that the number of basis elements in both cases is the same, and that the elements in the new basis are linearly independent.

Quadrilateral elements

The situation on quadrilaterals is even simpler than on triangles thanks to the product structure of the polynomial spaces. Consider two different directional polynomial degrees $p, q \geq 1$. A quadrilateral element $K_{h,p,q}$ and the corresponding enriched element $K_{h/2,p+1,q+1}$ are illustrated in Fig. 6.

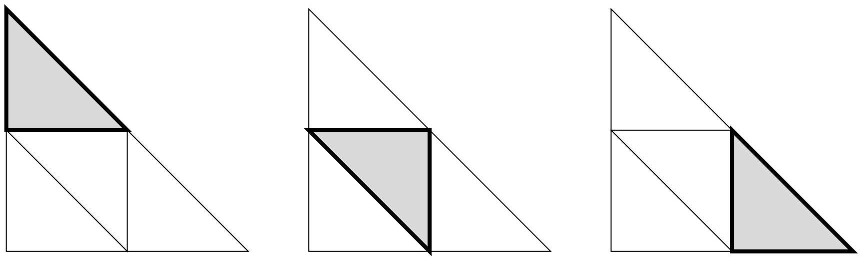

Fig. 5. Three subelements to which we add bubble functions of degrees 3, 4, . . . , p.

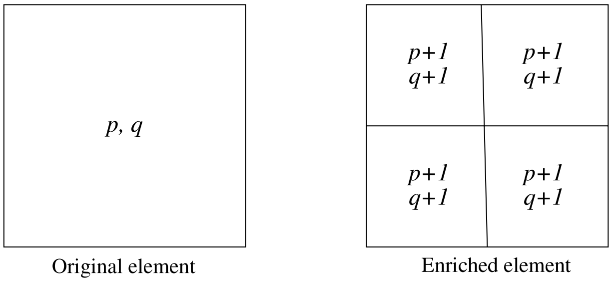

Fig. 6. Quadrilateral element $K_{h,p,q}$ and the corresponding enriched element $K_{h/2,p+1,q+1}$.

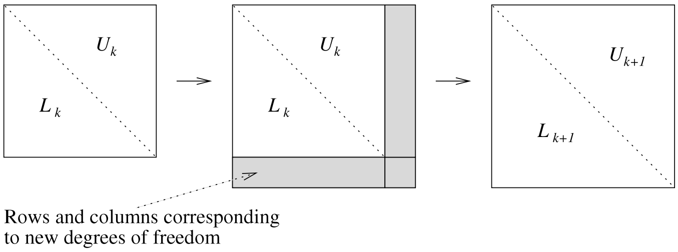

Fig. 7. Extension of the LU factorization of the stiffness matrix $S_k^r = L_k U_k$ to the LU factorization $S_{k+1}^r = L_{k+1}U_{k+1}$ corresponding to the next reference mesh τ_{k+1}^r.

Analogously to the triangular case, we calculate that the dimensions of the original and enriched spaces are $(p+1)(q+1)$ and $9+6p+6q+4pq = (2p+3)(2q+3)$, respectively. Due to the product character of the basis, it is sufficient to describe the extension of basis for a one-dimensional interval of degree p: The original basis has $p+1$ elements (two vertex functions and $p-1$ bubble functions). The enriched basis contains one additional vertex function associated with its midpoint, $p-1$ additional bubble functions of degrees 2, 3, . . . , p (each being nonzero in one half of the interval only), and two additional bubble functions of degree $p+1$ (each being nonzero in one half of the interval only). Thus the enriched space on the one-dimensional interval has dimension $(p+1)+1+(p-1)+2 = 2p+3$.

4. Solution of embedded discrete problems

Thanks to the hierarchic extensions of finite element spaces during the adaptivity process, the reference meshes $\tau_0^r, \tau_1^r, \tau_2^r, \ldots$ yield a sequence of embedded stiffness matrices $S_0^r \subset S_1^r \subset S_2^r \subset \cdots$. This fact can be used to optimize the solution of the discrete problem in every adaptivity step. In this section, we describe two different methods that we designed for this purpose. Since our target applications are in multiphysics problems, both of them are based on direct sparse solvers. The performance of the methods will be compared in Section 5.

M1: Hierarchic extension of LU factorizations

During the initial step, we create the LU factorization of the stiffness matrix S_0^r corresponding to the initial reference mesh τ_0^r (more precisely, we use UMFPACK [10] for this purpose). In every other adaptivity step, the LU factorization from the previous step is extended by adding new columns and rows corresponding to the newly added basis functions, as illustrated in Fig. 7.

The algorithm is easy to design as it only involves elementary matrix operations. With the LU factorization of the new stiffness matrix S_{k+1}^r in hand, the new reference solution u_{k+1}^r is computed instantly. By projecting the new reference solution on the mesh τ_{k+1}^c, we obtain its best representant u_{k+1}^c on τ_{k+1}^c and thus also the new error function $e_{k+1} = u_{k+1}^r - u_{k+1}^c$. Hence, one step of the adaptivity algorithm is completed.

M2: Combined direct-iterative method

The first discrete problem on the initial reference mesh τ_0^r has the form $S_0^r Y_0 = F_0$. We solve it via UMFPACK and keep the LU factorization of S_0^r. After the first mesh refinement, we obtain a new mesh τ_1^r. Due to the hierarchic extension of the finite element space, the corresponding stiffness matrix S_1^r consists of four blocks S_0^r, B_1, C_1 and S_1. The right-hand side of the new discrete problem consists of the original vector F_0 and a new vector F_1 corresponding to the newly added basis functions. The situation looks as follows:

$$\begin{pmatrix} S_0^r & C_{1,1} \\ B_{1,1} & S_1 \end{pmatrix} Y = \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}. \tag{1}$$

We can assume that the matrix S_1 is nonsingular since it corresponds to the solution of the original PDE problem in a subdomain equipped with homogeneous essential boundary conditions. We employ UMFPACK to construct the LU factorization of the matrix S_1 and solve efficiently the system $S_1 Y_1 = F_1$. The solution of the system (1) can be written in the form

$$Y = \begin{pmatrix} Y_0 + \Delta Y_0 \\ Y_1 + \Delta Y_1 \end{pmatrix}. \tag{2}$$

By substituting (2) into (1), we obtain

$$\begin{aligned} &S_0^r Y_0 + S_0^r \Delta Y_0 + C_{1,1} Y_1 + C_{1,1} \Delta Y_1 = F_0, \\ &B_{1,1} Y_0 + B_{1,1} \Delta Y_0 + S_1 Y_1 + S_1 \Delta Y_1 = F_1. \end{aligned} \tag{3}$$

Application of $S_0^r Y_0 = F_0$ and $S_1 Y_1 = F_1$ simplifies this to

$$\begin{aligned} &S_0^r \Delta Y_0 + C_{1,1} Y_1 + C_{1,1} \Delta Y_1 = 0, \\ &B_{1,1} Y_0 + B_{1,1} \Delta Y_0 + S_1 \Delta Y_1 = 0. \end{aligned} \tag{4}$$

The unknown vectors ΔY_0 and ΔY_1 are computed using the following iterative method:

$$\begin{aligned} S_0^r \Delta Y_0^{(k+1)} &= -C_{1,1} Y_1 - C_{1,1} \Delta Y_1^{(k)}, \\ S_1 \Delta Y_1^{(k+1)} &= -B_{1,1} Y_0 - B_{1,1} \Delta Y_0^{(k)}, \end{aligned} \tag{5}$$

which starts with $\Delta Y_0^{(0)} = 0$ and $\Delta Y_1^{(0)} = 0$. When this process converges, we know the solution u_1^r on the mesh τ_1^r.

After n refinement steps, the discrete problem has the form

$$\left(\begin{array}{c|c|c|c|c} S_0^r & C_{1,1} & C_{1,2} & \dots & C_{1,n} \\ \hline B_{1,1} & S_1 & C_{2,2} & \dots & C_{2,n} \\ \hline B_{1,2} & B_{2,2} & S_2 & \dots & C_{3,n} \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline B_{1,n} & B_{2,n} & B_{3,n} & \dots & S_n \end{array}\right) \left(\begin{array}{c} \overline{Y}_0 + \Delta Y_{0,n} \\ \hline \overline{Y}_1 + \Delta Y_{1,n} \\ \hline \overline{Y}_2 + \Delta Y_{2,n} \\ \hline \vdots \\ \hline \overline{Y}_n + \Delta Y_{n,n} \end{array}\right) = \left(\begin{array}{c} F_0 \\ \hline F_1 \\ \hline F_2 \\ \hline \vdots \\ \hline F_n \end{array}\right). \tag{6}$$

Here, $\overline{Y}_j = Y_j + \sum_{k=j}^{n-1} \Delta Y_{j,k}$, $j = 0, 1, \dots, n$. The $(j+1)$th equation in the system (6) has the form

$$\sum_{k=1}^{j} B_{k,j} (\overline{Y}_{k-1} + \Delta Y_{k-1,n}) + S_j (\overline{Y}_j + \Delta Y_{j,n}) + \sum_{k=j+1}^{n} C_{j+1,k} (\overline{Y}_k + \Delta Y_{k,n}) = F_j. \tag{7}$$

The vector $(\overline{Y}_0, \overline{Y_1}, \dots, \overline{Y}_{n-1})^{\mathrm{T}}$ is the solution of the system obtained after $n-1$ refinement steps. In other words,

$$\sum_{k=1}^{j} B_{k,j} \overline{Y}_{k-1} + S_j \overline{Y}_j + \sum_{k=j+1}^{n-1} C_{j+1,k} \overline{Y}_k = F_j, \quad j = 0, 1, \dots, n-1. \tag{8}$$

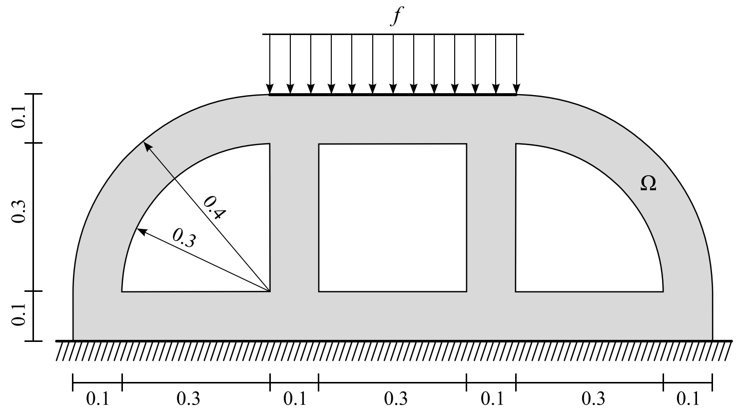

Fig. 8. Computational domain.

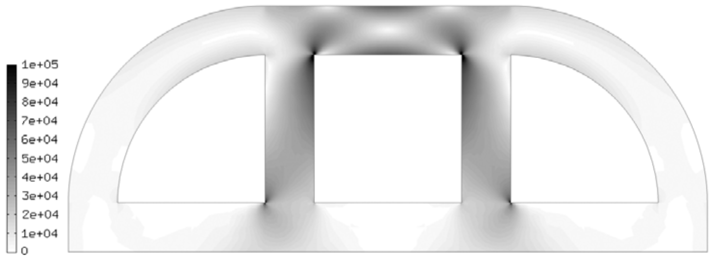

Fig. 9. Stress distribution.

Substituting this along with $S_n Y_n = F_n$ into (7), and using the fact that $\overline{Y}_n = Y_n$, we obtain

$$\sum_{k=1}^{j} B_{k,j} \Delta Y_{k-1,n} + S_j \Delta Y_{j,n} + C_{j+1,n} Y_n + \sum_{k=j+1}^{n} C_{j+1,k} \Delta Y_{k,n} = 0,$$
$$\text{where } j = 0, 1, \ldots, n-1, \quad \text{and}$$
$$\sum_{k=1}^{n} B_{k,n} (\overline{Y}_{k-1} + \Delta Y_{k-1,n}) + S_n \Delta Y_{n,n} = 0. \tag{9}$$

The unknown vectors $\Delta Y_{0,n}, \Delta Y_{1,n}, \ldots, \Delta Y_{n,n}$ are computed using the following iterative method:

$$S_j \Delta Y_{j,n}^{(k+1)} = -\sum_{k=1}^{j} B_{k,j} \Delta Y_{k-1,n}^{(k)} - C_{j+1,n} Y_n - \sum_{k=j+1}^{n} C_{j+1,k} \Delta Y_{k,n}^{(k)},$$
$$\text{where } j = 0, 1, \ldots, n-1, \quad \text{and} \tag{10}$$
$$S_n \Delta Y_{n,n}^{(k+1)} = -\sum_{k=1}^{n} B_{k,n} (\overline{Y}_{k-1} + \Delta Y_{k-1,n}^{(k)}).$$

After nth refinement step, we only construct and store the LU factorization of one relatively small matrix S_n. The above iterative process for the computation of u_n^r is based on the solution of n linear systems with the matrices $S_0^r, S_1, \ldots, S_n$ whose LU factorizations are known. Our preliminary results show that this iteration converges very fast both for symmetric positive definite matrices and for large ill-conditioned matrices. The convergence analysis is in progress.

5. Numerical example

Let us illustrate the methodology on a plane-strain model problem of linear elasticity describing the elastic behavior of a long hollow workpiece subject to vertical loading on its top surface. The workpiece lies on a rigid surface, as shown in Fig. 8.

Fig. 9 shows the resulting stress distribution. Piecewise-linear mesh and *hp*-FEM mesh corresponding to roughly the same relative error 0.5% are shown in Figs. 10 and 11, respectively. Convergence histories of standard FEM and *hp*-FEM in energy norm are compared in Fig. 12. Notice that the *hp*-FEM converges exponentially.

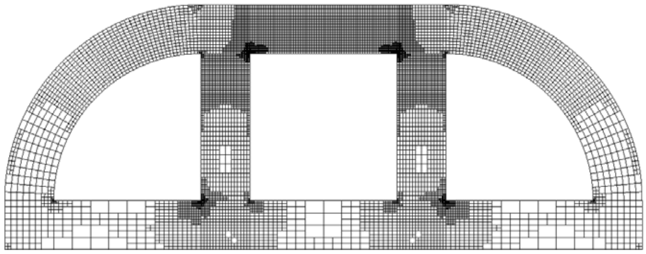

Fig. 10. Mesh consisting of lowest-order elements (approx. error 0.5% and 21 000 DOF).

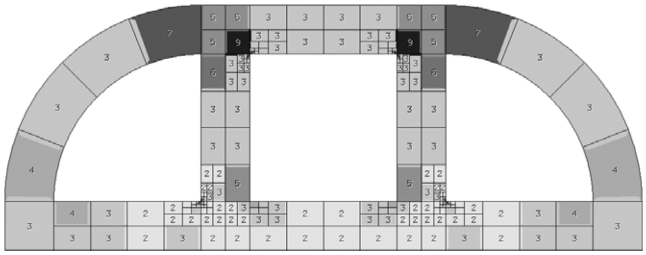

Fig. 11. Mesh consisting of higher-order elements (approx. error 0.5% and 2500 DOF).

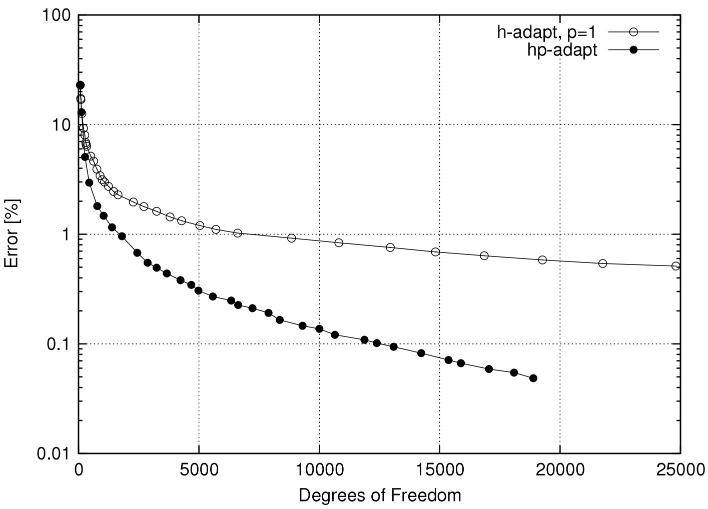

Fig. 12. Convergence of adaptive *h*-FEM (linear elements) and adaptive *hp*-FEM.

Performance of method M1

First we show that the technique M1 from Section 4 is not very suitable for practical use. To illustrate this, it is sufficient to consider a relatively small reference mesh τ_0^r with 16 129 DOF. UMFPACK is used to compute the LU factorization of the corresponding stiffness matrix S_0^r. We refine the mesh τ_0^r by adding 100, 1001, 2500, 5002 and 10 002 DOF, respectively (i.e., the new reference mesh τ_1^r has 16 229, 17 130, 18 629, 21 131 and 26 131 DOF). Table 1 shows the CPU times needed by UMFPACK to compute the LU factorization of the corresponding stiffness matrix S_1^r from scratch as well as the CPU time needed to construct the LU factorization of S_1^r by extending the LU factorization of the stiffness matrix S_0^r, using the method M1. All times are in milliseconds (ms).

The reader can see that the method M1 is not useful — in most cases the direct solver is more efficient.

Performance of method M2

To illustrate the performance of the method M2, let us consider two different reference meshes τ_0^r with 130 305 and 261 121 DOF, respectively. For each of them, we construct five different reference meshes τ_1^r by adding 1001, 2500, 5002

Table 1
Performance of the method M1.

Rank(S_0^r)	DOF added	Rank(S_1^r)	UMFPACK time	M1-time
16 129	100	16 229	479	411
16 129	1 001	17 130	545	4 212
16 129	2 500	18 629	640	11 403
16 129	5 002	21 131	720	24 515
16 129	10 002	26 131	1327	58 637

Table 2
Performance of the method M2, rank(S_0^r) = 130 305.

DOF	Rank(S_1^r)	UMFPACK time	M2-time, $eps = 10^{-2}$	M2-time, $eps = 10^{-4}$	M2-time, $eps = 10^{-6}$
1 001	131 306	4454	633	1750	3241
2 500	132 805	4156	648	1809	3316
5 002	135 307	4327	768	1874	3487
10 002	140 307	4420	863	2008	3631

Table 3
Performance of the method M2, rank(S_0^r) = 261 121.

DOF	Rank(S_1^r)	UMFPACK time	M2-time, $eps = 10^{-2}$	M2-time, $eps = 10^{-4}$	M2-time, $eps = 10^{-6}$
1 001	262 122	11 300	1262	2646	4133
2 500	263 621	10 355	1257	2771	4346
5 002	266 123	10 418	1301	2810	4370
10 002	271 123	11 432	1792	3077	4536

and 10 002 DOF, respectively. Tables 2 and 3 show the CPU times needed by UMFPACK (again in milliseconds) to construct the LU factorization of the stiffness matrix S_1^r in all cases, along with the CPU time needed by our method M2 to obtain the solution u_1^r on the mesh τ_1^r (using the existing LU factorization of S_0^r). The iterative method M2 stops when the l^2-norm of the difference of two successive approximations is less than *eps*.

The reader can see that the method M2 takes much less CPU time than if the enriched discrete problem is solved from scratch. The numerical experiment also shows that the performance gap becomes larger as the discrete problem size grows.

6. Conclusion and outlook

We described a method for hierarchic enrichment of finite element spaces in *hp*-FEM approximations, and investigated the performance of two different techniques for the solution of the resulting embedded matrix problems. The method M1 turned out to be useless. This was quite surprising to us, and we showed the results in order to warn the reader not to repeat our mistake.

On the other hand, the results obtained with the method M2 are very encouraging. This method can handle very large matrices including those which no longer can be LU-factorized. The method needs to be further studied and optimized, and applied to indefinite matrices arising in complex multiphysics problems. Our preliminary results indicate that the method converges very well also for strongly ill-conditioned indefinite matrices. Convergence of the iterative process needs to be analysed, as well as the influence of the parameter *eps* in the stopping criterion. The residual stopping criterion needs to be replaced with a more realistic convergence criterion based on the H^1-norm in the finite element space. We hope to report on our progress soon.

7. Free Adaptive *hp*-FEM Library *Hermes*

Hermes[1] (higher-order modular finite element system) is a C++/Python library for rapid prototyping of space and space–time adaptive *hp*-FEM solvers. Hermes is the first software capable of solving arbitrary multiphysics PDE problems via space–time adaptive *hp*-FEM on dynamical meshes. Sample applications presented on Hermes home page include heat transfer, thermoelasticity, electromagnetics, microwave heating, optics, incompressible and compressible flow, interface tracking in two-phase flow, flame propagation, the Black–Scholes equation of financial mathematics, the Schrödinger eigenvalue problem of quantum chemistry, and others. Thanks to the generality of the adaptive algorithms, Hermes even makes it possible to employ adaptive *hp*-FEM for image compression [11]. The library comes with an interactive web portal which allows the user to enter a large variety of PDE problems through a web browser and have them solved adaptively on HPC facilities at the University of Nevada, Reno.

1 Visit Hermes home page and interactive *hp*-FEM web portal http://hpfem.org/.

References

[1] I. Babuska, W. Gui, The *h*, *p* and *hp*-versions of the finite element method in one dimension — Part 3. The adaptive *hp*-version, Numer. Math. 49 (1986) 659–683.
[2] P. Solin, Partial Differential Equations and the Finite Element Method, J. Wiley & Sons, 2005.
[3] P. Solin, J. Cerveny, I. Dolezel, Arbitrary-level hanging nodes and automatic adaptivity in the *hp*-FEM, Math. Comput. Simulation 77 (2008) 117–132.
[4] P. Solin, L. Demkowicz, Goal-oriented *hp*-adaptivity for elliptic problems, Comput. Methods Appl. Mech. Engrg. 193 (2004) 449–468.
[5] P. Solin, K. Segeth, I. Dolezel, Higher-Order Finite Element Methods, Chapman & Hall/CRC Press, 2003.
[6] P. Solin, J. Cerveny, L. Dubcova, Adaptive multi-mesh *hp*-FEM for linear thermoelasticity problems, J. Comput. Appl. Math. (accepted).
[7] B. Szabo, I. Babuska, Finite Element Analysis, J. Wiley & Sons, 1991.
[8] T. Vejchodsky, P. Solin, Static condensation, partial orthogonalization of basis functions, and ILU preconditioning in the *hp*-FEM, J. Comput. Appl. Math. 216 (2008) 192–200.
[9] P. Deuflhard, F. Bornemann, Scientific Computing with Ordinary Differential Equations, Springer, 2001.
[10] T.A. Davis, Direct Methods for Sparse Linear Systems, SIAM, Philadelphia, 2006.
[11] P. Solin, D. Andrs, On scientific data and image compression based on adaptive higher-order FEM, Adv. Appl. Math. Mech. 1 (2009) 56–68.