COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 77 (1989) 181–212
NORTH-HOLLAND

TOWARD A UNIVERSAL *h-p* ADAPTIVE FINITE ELEMENT STRATEGY PART 3. DESIGN OF *h-p* MESHES

W. RACHOWICZ, J.T. ODEN and L. DEMKOWICZ
Texas Institute for Computational Mechanics, The University of Texas at Austin, Austin, TX 78712, U.S.A

Received 27 March 1989

In this third paper in our series, the issue of designing *h-p* meshes which are optimal in some sense is addressed. Criteria for *h-p* meshes are derived which are based on the idea of minimizing the estimated error over a mesh with a fixed number of degrees-of-freedom. An optimization algorithm is developed based on these criteria and is applied to several model one-dimensional and two-dimensional elliptic boundary-value problems. Numerical results indicate that the approach can lead to exponential rates-of-convergence.

1. Introduction

In previous works in this series [1, 2], we developed data structures, finite element formulations, and a posteriori error estimates for adaptive *h-p* finite element methods for boundary-value problems characterized by general elliptic systems of partial differential equations. We demonstrated that, by an appropriate choice of parameters, the method could also be applied to the numerical solution of time-dependent problems and, in particular, to parabolic systems and to convection–diffusion problems. We shall now address the question of how mesh sizes h and spectral orders p can be chosen throughout a finite element mesh.

There are (at least) two basic parameters in any finite element formulation of a boundary-value problem that affect accuracy and convergence: the size h of the mesh and the order p of the local polynomial approximations. One can expect that as either of these parameters are varied independently ($h \to 0$, $p \to \infty$) the approximation error should diminish in some sense. Thus, if an adaptive finite element method is designed for which the distributions h of element sizes and p of polynomial degrees (spectral orders) vary quite generally over the domain, one might expect it to be possible to obtain accuracies and rates-of-convergence superior to either pure h-versions or p-versions of the finite element method by judiciously changing both h and p. Indeed, some theoretical results of Babuška and coworkers (e.g. [3, 4, 5]) suggest that exponential rates-of-convergence can be attained with the proper combination of h- and p-refinements for special classes of one-dimensional problems, including problems with singularities, and also for certain two-dimensional cases. However, a systematic approach toward generating an optimal distribution of h and p for delivering solutions with a preset level of accuracy (or preset value of estimated error) is not available.

In the present study, we develop a simple, approximate, *h-p* mesh optimization technique

0045-7825/89/$3.50 © 1989, Elsevier Science Publishers B.V. (North-Holland)

that can be used in an attempt to construct *h-p* distributions that produce the best results, as measured by error estimates of the type studied in [2], for the least number of unknowns. For clarity, we restrict our discussion to model classes of one- and two-dimensional elliptic boundary-value problems. The finite element approximations to these problmes are constructed on 1-irregular meshes and local stiffness matrices are calculated using hierarchical polynomial shape functions in the manner discussed in [1]. The scheme is based fully on our *h-p* adaptive strategy in which the mesh is refined/unrefined and the spectral order is enriched/unenriched successively (following the rules in [1]), until the error in each element (measured by techniques in [2]) is below a preassigned tolerance, this resulting in a mesh with (approximately) the least possible number of degrees-of-freedom.

We note that the discrete optimization problem described here is a very complex one that can be formulated in several ways. It does not seem to be practical or possible at this point to seek a rigorously exact solution to it. Our aim is to offer a formulation of the problem and a practical and, hopefully, efficient approximate scheme that leads to a trajectory in space of *h-p* distributions that terminates at an *h-p* mesh close to the optimal. We are able to judge the quality of the approximate optimal mesh only through numerical experiments, but our *h-p* results are consistently superior to either the adaptive *h*- or adaptive *p*-methods for equal numbers of degrees of freedom. Importantly, results suggest that the approach also leads to exponential convergence rates for the *h-p* method.

Following this introduction, we review some essential details on the implementation of the *h-p* adaptive scheme introduced in [1] and we develop some general rules for *h-p* mesh optimization. Section 3 is devoted to an analysis of one-dimensional *h-p* mesh optimization for interpolation of smooth functions. There a precise formulation of an *h-p* optimization problem is given and a therorem is proved for a characterization of an optimal mesh under certain constraints. The interpretation of these results leads us ultimately to a scheme that proves to be effective in more general problems defined on two-dimensional domains. Section 4 contains a discussion of local *h-p* refinements. The *h-p* adaptive algorithm is finally presented in Section 5. The results of applying the method to several test problems are collected and discussed in Section 6.

2. Preliminaries on *h-p* methods and mesh optimization

Review of features of the h-p method

The *h-p* finite element method discussed in [1, 2] employs tensor products of hierarchical polynomial shape functions which, for the two-dimensional case, are *p*-unisolvent to degrees of freedom defined at nine nodes: four corner nodes, four midside nodes, and a node at the centroid. Generalizations to three dimensions are not discussed here, although the ideas are directly extendable to three dimensions and some numerical results on three-dimensional cases are available. The one-dimensional specialization is obvious.

One views the *h-p* adaptive strategy as starting from some initial mesh of elements K which are images of a master element $\hat{K} = \{\boldsymbol{\xi} = (\xi_1, \xi_2) \mid -1 \leq \xi_1, \xi_2 \leq 1\}$ under a quadratic map,

$$\boldsymbol{\xi} \mapsto \boldsymbol{x} \in \bar{\Omega}; \quad x^i = \sum_{j=1}^{9} x_j^i N_j(\boldsymbol{\xi}), \ i = 1, 2, \tag{2.1}$$

where $N_j(\xi)$ are quadratic Lagrange polynomials associated with the usual 9-node Q_2 element. For purposes of discussion, we consider cases in which the mesh $\bar{\Omega}_h = \cup_{K \in \mathcal{Q}_h} K$ over a partition $\mathcal{Q}_h$ of Ω is topologically equivalent to a uniform rectangular mesh. The process of enriching the polynomial basis functions by adding polynomials of larger degree and of subdividing elements into four subelements (sons) is referred to as an *h-p* refinement. For *h*-refinements, we restrict ourselves to those leading to 1-irregular meshes as indicated in Fig. 1a. Thus, an element is subdivided only if it has smaller or equal neighbors; otherwise, larger neighbors must be first subdivided before the element is refined. For *p*-enrichments, we adopt the maximum rule whereby the maximum polynomial degree dominates at the interface of two elements, i.e., when two neighboring elements are domains of shape functions of different polynomial degree, shape functions are added to those of the lower order element so as to make the global basis functions continuous across the interelement boundary (see Fig. 1b).

h-p mesh optimization; general rules

We now turn to the question of optimizing the distribution of mesh sizes h and polynomial degrees p over a finite element mesh. We begin with an idealized case in which Ω is a rectangle and the elements $K \subset \Omega_h$ have diameters h_K and shape functions of maximum spectral order p_K. For definiteness, we assume that the error indicators for various elements are of the interpolation error type and are functions of the mesh parameters h and p. Later, more general cases are considered. The distribution of mesh sizes and spectral orders over Ω_h is characterized by functions h and p:

$$h(x_1, x_2) = h_K \quad (x_1, x_2) \in K, \qquad p(x_1, x_2) = p_K \quad (x_1, x_2) \in K. \tag{2.2}$$

The error indicator θ_K for element K will depend on h_K and p_K and is represented by a local error density $\varphi_K = \varphi_K(h, p)$: $\theta_K = \int_K \varphi_K(h, p)\, dx$ (see [2]). Thus, an indication of the total

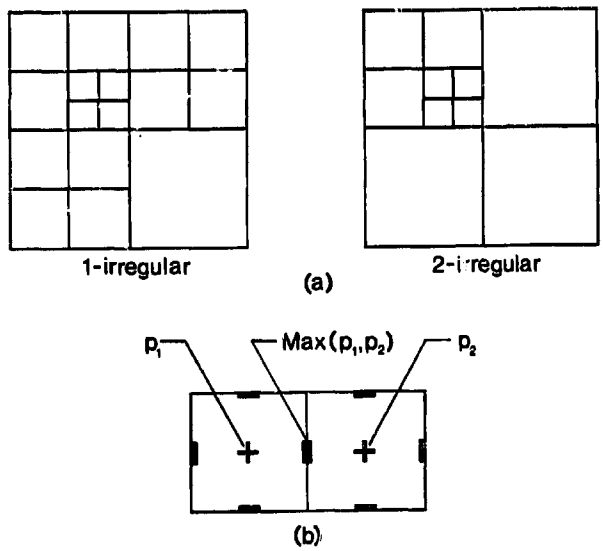


Fig. 1. Choices made in the *h-p* data structure. (a) Examples of 1-irregular and 2-irregular meshes; the *h-p* strategy uses only 1-irregular refinements. (b) The maximum rule is used at an interface of elements containing polynomials of different degrees.

error over Ω is given by the functional

$$J(h, p) = \sum_{K \in \Omega_h} \int_K \varphi_K(h, p)\, \mathrm{d}x\,. \tag{2.3}$$

Next, let n_K denote the total number of degrees of freedom of element K. We introduce a degree-of-freedom density $n = n(h, p)$ so that the total number of degrees-of-freedom M is given by

$$M = \int_\Omega n(h, p)\, \mathrm{d}x, \qquad n(h, p)|_{K=n_K/\mathrm{meas}(K)}\,. \tag{2.4}$$

The optimal mesh for a fixed number $M = M_0$ degrees-of-freedom will be defined by the h, p-distribution (h^*, p^*), such that

$$J(h^*, p^*) = \min_{\int_\Omega n(h, p)\, \mathrm{d}x = M_0} J(h, p)\,. \tag{2.5}$$

To resolve this problem by the method of Lagrange multipliers, we introduce the Lagrangian,

$$L(h, p; \lambda) = J(h, p) - \lambda\left(\int_\Omega n(h, p)\, \mathrm{d}x - M_0\right) \tag{2.6}$$

and arrive at the optimality conditions

$$\frac{\partial \varphi_K}{\partial h} - \lambda \frac{\partial n}{\partial h} = 0\,, \qquad \frac{\partial \varphi_K}{\partial p} - \lambda \frac{\partial n}{\partial p} = 0\,. \tag{2.7}$$

Thus, for the entire mesh, the optimal h-p distribution results when

$$\left.\frac{\mathrm{d}\varphi_K}{\mathrm{d}n}\right|_{p=\mathrm{const}} = \lambda = \mathrm{const}\,, \qquad \left.\frac{\mathrm{d}\varphi_K}{\mathrm{d}n}\right|_{h=\mathrm{const}} = \lambda = \mathrm{const}\,. \tag{2.8}$$

REMARK 2.1. The optimality conditions (2.7) cannot be fully exploited unless φ_K and n are known as functions of h and p, and this is not possible for general h-p refinements. However, some broad guidelines can be seen. For example, for the optimal h, p-distribution for fixed M_0, (2.8) suggest that the quantity,

$$\Delta\mathrm{error}/\Delta\,\mathrm{ndof}$$

should be equidistributed over Ω_h, Δerror being the change in error due to a change Δndof in the number of degrees-of-freedom. A refinement strategy designed to reduce the error as much as possible would make the change in error per change in ndof as large as possible. It would seem that if $\Delta\mathrm{error}/\Delta\mathrm{ndof}|_{h=\mathrm{const}}$ and $\Delta\mathrm{error}/\Delta\mathrm{ndof}|_{p=\mathrm{const}}$ are different, then the refinement resulting in the larger of these two should be adopted in searching for the optimum.

REMARK 2.2. Some special cases of mesh optimization can be obtained from the above arguments. Consider first only h-refinements on a rectangular mesh:

$$p = p_0 = \text{const.}, \qquad \theta_K(h, p_0) = \theta_K(h) = \int_K f(x_1, x_2) h^\sigma \, dx,$$

$$n(h, p_0) = n(h) = \frac{1}{h^2}, \tag{2.9}$$

where $f(x_1, x_2)h^\sigma$, $\sigma > 0$ is an error distribution function. Then (2.6) reduces to

$$\sigma f(x) h^{\sigma-1} + 2\lambda h^{-3} = 0,$$

and we have (if $\int_K h^{-2} \, dx = 1$),

$$\int_K f(x) h^\sigma \, dx = \theta_K = -2\lambda/\sigma = \text{const}. \tag{2.10}$$

Thus, we obtain the well-known result that the optimal mesh is obtained whenever the error (or error indicators) are equidistributed throughout the mesh.

For a hypothetical h-p mesh in which (approximately)

$$n(p, h) = p^2/h^2, \qquad M = \int_\Omega (p^2/h^2) \, dx, \tag{2.11}$$

conditions (2.6) yield the optimality conditions

$$h \frac{\partial \varphi_K}{\partial h} + p \frac{\partial \varphi_K}{\partial p} = 0 \qquad 2\lambda = -\frac{h^3}{p^2} \frac{\partial \varphi_K}{\partial h} = \frac{h^2}{p} \frac{\partial \varphi_K}{\partial p} = \text{const}. \tag{2.12}$$

The last condition suggests an equidistribution of the quantities shown equal to 2λ.

In general, the form of the error function φ_K is unknown and, in our h-p scheme, $n(h, p)$ may be a very complicated function of h and p or it may not exist at all. Thus, we can glean only general rules for seeking optimal meshes from these results. One principle may emerge from this calculation: choose a sequence of h-p refinements wherein the change in error per change in number of degrees-of-freedom is maximized. We shall attempt to derive a strategy based on this criterion in the next section.

3. An h-p refinement strategy for one-dimensional interpolation

We shall now describe an h-p refinement scheme for minimization of interpolation errors of a smooth function u defined on a one-dimensional domain $\Omega = (0, L) \subset \mathbb{R}$. The strategy is based on the following ideas.

(1) We begin with (say) an initial, continuous, piecewise linear ($p = 1$) approximation of u on

a uniform mesh of N elements Ω_k ($h_k = L/N$), $k = 1, 2, \ldots, N$. Later we admit the possibility of more general initial meshes having nonuniform distributions of h and p.

(2) Each of the subdomains Ω_k (the initial elements) is next subjected to an h-p refinement, a subdivision into smaller elements and a possible enrichment of spectral orders. The resulting structure of the approximation is defined by functions

$$\begin{aligned} &h = h(x) = \text{mesh distribution}, \quad h|_K = h_K = \operatorname{dia}(K), \\ &p = p(x) = \text{order distribution}, \quad p|_K = p_K = \text{order of shpae functions over } K, \\ &n = n(x) = \text{degrees-of-freedom distribution}, \quad n_k = n|_{\Omega_k}, \end{aligned} \tag{3.1}$$

where K is a subelement of Ω_k and n_k is the total number of degrees of freedom of Ω_k after the h-p refinement. Note that, with an appropriate numbering of nodes, n may be determined as a function of p/h, but there are many different choices of h- and p-refinement leading to the same total number of degrees of freedom.

(3) Consider the local finite element space

$$\begin{aligned} V^{h,p}(\Omega_k, n_k) = &\text{the finite-dimensional linear space of piecewise polynomials} \\ &\text{resulting from all possible } h\text{-}p \text{ refinements of } \Omega_k \text{ resulting in} \\ &\text{exactly } n_k \text{ degrees-of-freedom}. \end{aligned} \tag{3.2}$$

For example, if $\Omega_k = [x_{k-1}, x_k]$,

$$\begin{aligned} V^{h,p}(\Omega_k, 2) &= \mathcal{P}_1([x_{k-1}, x_k]), \\ V^{h,p}(\Omega_k, 3) &= \mathcal{P}_1([x_{k-1}, x_k - h_k/2]) \oplus \mathcal{P}_1([x_{k-1} + h_k/2, x_k]) \\ &\quad \oplus \{x^2 - (x_k + x_{k-1})x + x_k x_{k-1}\} \cap C^0(\Omega_k) \end{aligned}$$

and $V^{h,p}(\Omega_k, 5)$ is the space of piecewise linear, quadratic, cubic and quartic functions corresponding to the eleven possible refinements of Ω_k resulting in five degrees-of-freedom, shown in Fig. 2.

(4) We define as an error indicator θ_k for subdomain Ω_k the local interpolation error in a suitable norm corresponding to n_k degrees-of-freedom, e.g.,

$$\theta_k(n_k) = \inf_{v_{hp} \in V^{h,p}(\Omega_k, n_k)} \|u - v_{hp}\|^2_{H^s(\Omega_k)}, \quad s = 0 \text{ or } 1. \tag{3.3}$$

Thus, the calculation of $\theta_k(n_k)$ has resulted from the particular h-p refinement of Ω_k resulting in the smallest error for n_k degrees-of-freedom. The dependence of the error on h and p over Ω_k has thus been taken care of by the calculation of $\theta_k(n_k)$.

(5) The total error measured by such indicators is given by the functional

$$J(n) = \sum_{k=1}^{N} \theta_k(n_k) \tag{3.4}$$

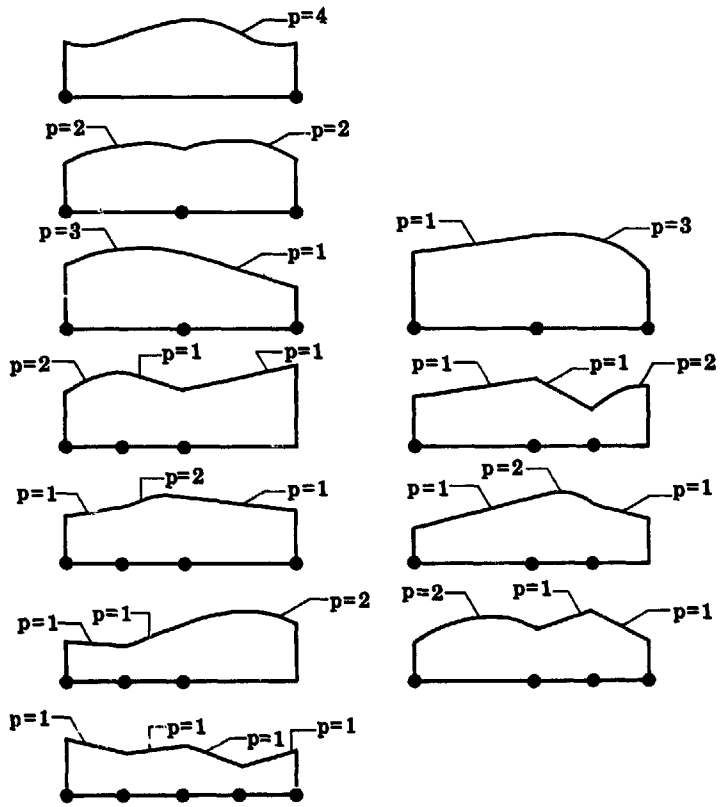


Fig. 2. Various combinations of h- and p that result in a five degree-of-freedom element and form a basis for $V^{h,p}(\Omega_k, 5)$.

Following the plan of the previous section, we seek to minimize J subject to the constraint that the total number of degrees-of-freedom is constant, i.e., we are led to the discrete optimization problem:

$$\text{Find } \boldsymbol{n}^* \text{ (or, equivalently, } n_1^*, n_2^*, \ldots, n_N^*), \quad \text{such that}$$

$$J(\boldsymbol{n}^*) = \min_{\boldsymbol{n} \in \mathbb{R}^N} J(\boldsymbol{n}), \quad \text{subject to the constraint } \sum_{k=1}^{n} n_k = M = \text{const}. \tag{3.5}$$

Alternatively, let

$$\mathcal{M}_M = \left\{ \boldsymbol{n} \in \mathbb{R}^n \mid \boldsymbol{n} = (n_1, n_2, \ldots, n_N),\ n_i = \text{integer} > 0,\ 1 \leq i \leq N, \right.$$

$$\left. \sum_{i=1}^{N} n_i = M = \text{const} \right\}. \tag{3.6}$$

Then (3.5) can also be written:

$$\text{Find } \boldsymbol{n}^* \in \mathcal{M}_M, \quad \text{such that} \quad J(\boldsymbol{n}^*) = \min_{\boldsymbol{n} \in \mathcal{M}_M} J(\boldsymbol{n}). \tag{3.7}$$

Characterization of the minimizer n^*

The solution n^* of the minimization problem (3.6) can be characterized by an elementary argument. We know that

$$J(n) \geq J(n^*) \quad \forall n \in \mathcal{M}_M ,$$

so that

$$\begin{aligned} &\theta_1(n_1) + \theta_2(n_2) + \cdots + \theta_i(n_i) + \cdots + \theta_N(n_N) \\ &\quad \geq \theta_1(n_1^*) + \cdots + \theta_i(n_i^*) + \cdots + \theta_N(n_N^*) . \end{aligned} \tag{3.8}$$

Since $\Sigma_k n_k^* = M = \text{const}$, we can set

$$n_1 = n_1^*, n_2 = n_2^*, \ldots, n_i = n_i^* + 1, \ldots, n_j = n_j^* - 1, \ldots, n_N = n_N^* ,$$

so that the constraint is satisfied. But (3.8) yields

$$\theta_i(n_i^* + 1) + \theta_j(n_j^* - 1) \geq \theta_i(n_i^*) + \theta_j(n_j^*) . \tag{3.9}$$

Similarly,

$$\theta_i(n_i^* - 1) + \theta_j(n_j^* + 1) \geq \theta_i(n_i^*) + \theta_j(n_j^*) . \tag{3.10}$$

Introducing the discrete differences

$$\partial_+ \theta_i(n_i) \overset{\text{def}}{=} \theta_i(n_i + 1) - \theta_i(n_i) , \qquad \partial_- \theta_i(n_i) \overset{\text{def}}{=} \theta_i(n_i) - \theta_i(n_i - 1) , \tag{3.11}$$

the relations (3.9) and (3.10) can be written

$$\partial_+ \theta_i(n_i^*) \geq \partial_- \theta_j(n_j^*), \quad \partial_+ \theta_j(n_j^*) \geq \partial_- \theta_i(n_i^*) \quad \forall i, j, 1 \leq i, j \leq N . \tag{3.12}$$

Let $\partial\theta_i(n_i)$ denote the interval

$$\partial\theta_i(n_i) \overset{\text{def}}{=} [\partial_- \theta_i(n_i), \partial_+ \theta_i(n_i)] \subset \mathbb{R} . \tag{3.13}$$

Then we have the following result on the characterization of n^*.

THEOREM 3.1. A solution n^* of the discrete optimization problem (3.7) is such that (3.12) holds for any $i, j, 1 \leq i, j \leq N$. In turn, this condition implies that there exists a $\lambda \in \mathbb{R}$, such that

$$\lambda \in \partial\theta_i(n_i^*) \quad \forall i, i = 1, 2, \ldots, N . \tag{3.14}$$

PROOF. Condition (3.12) follows immediately from the calculations preceding it. To prove (3.14), assume, contrary to the assertion, that $\exists i, j$ such that

$$[\partial_- \theta_i(n_i^*), \partial_+ \theta_i(n_i^*)] \cap [\partial_- \theta_j(n_j^*), \partial_+ \theta_j(n_j^*)] = \emptyset ,$$

But then we would have

$$\partial_+ \theta_j(n_j^*) < \partial_- \theta_i(n_i^*) \quad \text{or} \quad \partial_+ \theta_i(n_i^*) < \partial_- \theta_j(n_j^*)$$

in contradiction to (3.12). □

REMARK 3.1. If the error indicators θ_i are convex functions of n_i, then the minimizer $\boldsymbol{n}^*$ is unique. In general, uniqueness of $\boldsymbol{n}^*$ cannot be guaranteed.

REMARK 3.2. The intervals $\partial\theta_i(n_i)$ are regarded as discrete subgradients in analogy with the subdifferentials of nondifferentiable continuous functions. Since

$$\theta_i(n_i + 1) - \theta_i(n_i) = \partial_+ \theta_i(n_i) = \partial_- \theta_i(n_i + 1) ,$$

the discrete subgradients of $\theta_i(n_i)$ can be viewed as a sequence of subintervals connected at endpoints.

Some implications of the characterization of $\boldsymbol{n}^*$

To interpret (3.12) and (3.14), consider two adjacent subdomains Ω_k and Ω_{k+1} in the initial mesh, such as those separated in Fig. 3. Various paths, shown by dashed lines in the figure, indicate the reduction in error indicators θ_k and θ_{k+1} due to various choices of h and p—and

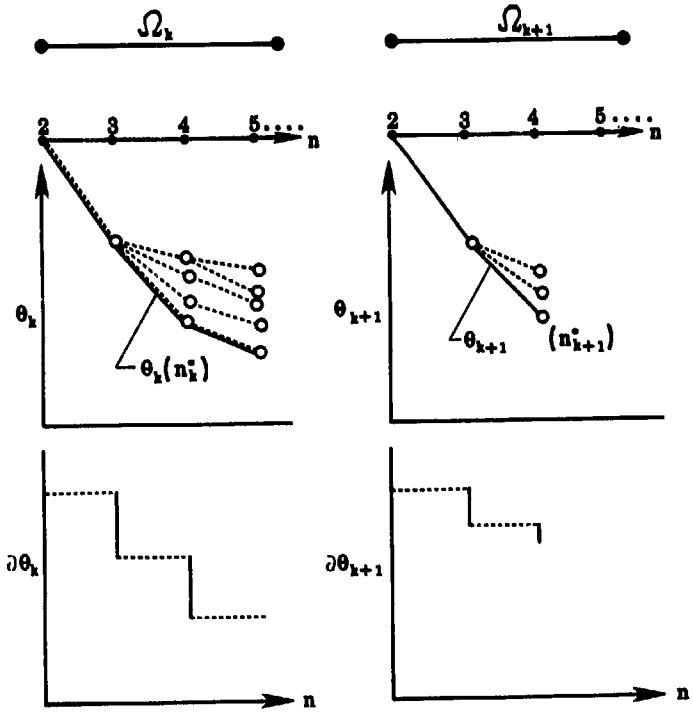


Fig. 3. Geometrical interpretation of the variation in θ_k, $\partial\theta_k$ with n.

hence $\boldsymbol{n}$. The particular local optimal error reductions are the solid lines $\theta_k(n_k^*)$, $\theta_{k+1}(n_{k+1}^*)$ shown, and the discrete subgradients are the intervals shown in solid lines below the θ-graphs.

Now let us plot these intervals versus the number of degrees of freedom n for each of the N subdomains. For example, for $N = 7$ the situation is represented in Fig. 4. Consider for this case a trial solution $\boldsymbol{n} \in \mathcal{M}_M$ of degrees-of-freedom corresponding to some arbitrary distribution of h and p resulting in θ_k's and $\partial\theta_k$'s indicated in the figure. Since this trial degree-of-freedom distribution is not optimal, we will not have $\lambda \in \partial\theta_k(\boldsymbol{n})$, as shown. However, the optimal mesh will be one for which the line $\lambda \in \partial\theta_k(\boldsymbol{n}^*)$ passes through all intervals, as indicated in the figure.

It is clear from these observations that to modify a trial mesh so that one achieves an optimal, requires that one refines those elements with $|\partial_+\theta_k(n_k)|$ below λ and unrefines those for which $|\partial_-\theta_k(n_k)|$ is above λ. Or, to avoid unrefinements, we can assume $\lambda > |\partial_+\theta_k(n_k)|$ and refine these elements with the largest subgradients $\partial_+\theta_k(n_k)$. In short, to achieve a solution to (3.7), we must refine those subdomains for which the anticipated decrease in error is the largest. If the change in error due to an increase in n_i by unity is unavailable, we employ linear interpolation, as suggested in Fig. 5. Then the discrete subgradients for these artificial n_i's reduce to points and can be neglected. On the other hand, the $\partial\theta_k$'s for n_i's must be scaled by 1/(actual increment of n_i). In other words, our criterion for optimal refinements is: refine elements for which the anticipated decrease of the error per unit new degree-of-freedom is the largest.

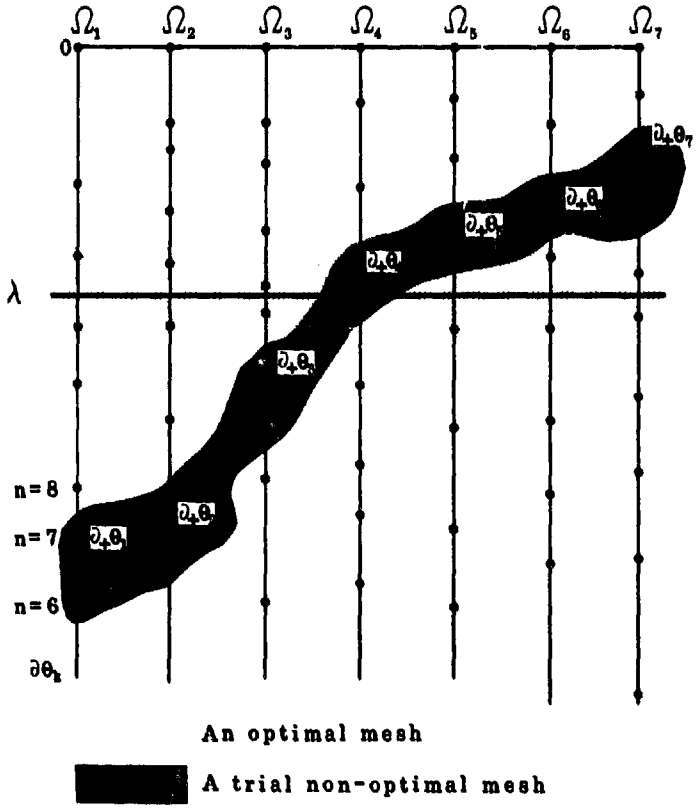


Fig. 4. Illustration of $\lambda \in \partial\theta_k(n^*)$ for an optimal h-p mesh.

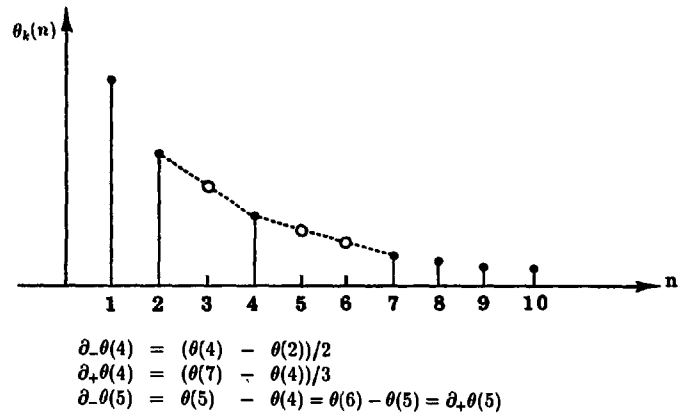


Fig. 5. Definition of $\theta(n)$ and $\partial_\pm\theta(n)$ for cases of $\Delta n > 1$ by linear interpolation.

REMARK 3.3. The proposed strategy is different from the equidistribution of errors strategy. However, as we have shown in Section 2, for some special types of error functions $\theta(n)$ these strategies coincide.

REMARK 3.4. The assumed property that each function $\theta_k(n_k)$ depends only on refinements of the kth element is not satisfied in most problems if errors other than interpolation errors are considered. However, for a wide class of problems, the influence of refinements is local in the sense that the refinement of a given element affects accuracy within this element more than in neighboring elements.

4. The *h-p* strategy in two dimensions

Some preliminary observations

The two-dimensional case is fundamentally different from the one-dimensional case discussed above. First of all, most of the elementary refinements cannot be restricted to only one element: p-enrichment of the midside nodes affects the approximation inside two neighboring elements; h-refinement of a given element may cause subdivision of some of its neighbors in order to preserve the 1-irregularity of the mesh. Since any refinement of an element affecting its boundary affects its neighbor, we cannot formulate the optimization problem as a sum of elementary noninterfering problems on initial elements as in (3.7) even for interpolation errors [2]. However, the general relations derived in Section 2 and the details of the one-dimensional case in Section 3 suggest a simplified approach for the two-dimensional case which is a straightforward extrapolation of the one-dimensional strategy: perform refinements for which the anticipated decreases of the error per new degree of freedom are as large as possible.

Some technical details of two-dimensional strategies

p-refinements

Recall our maximum convention: if two elements of different degrees have a common side, then we supplement the space of shape functions of the lower order element by the missing

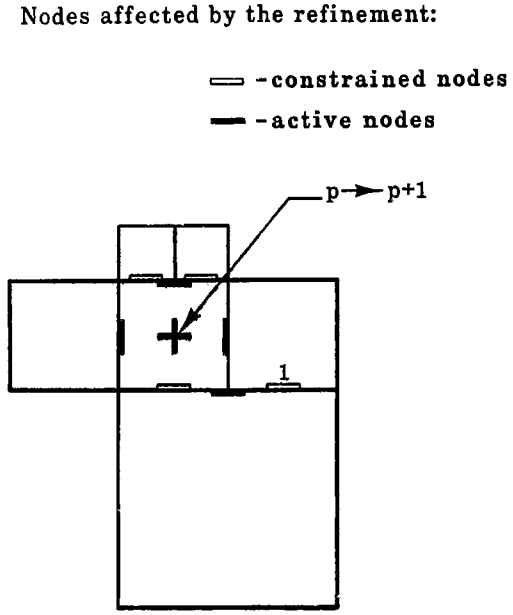


Fig. 6. Possible cases of enriching nodes of neighbors caused by raising p in a given element.

higher order functions. Since hierarchical shape functions are used, this means in practice that the order of the element is identified with the order of its central node and for two elements with spectral orders p_1 and p_2, the order of the common boundary node is $\max(p_1, p_2)$. Thus enriching a given element may affect the boundary nodes of all its neighbors. What is in some way unexpected is that the affected nodes may not necessarily belong to the common side: in the case of constrained elements, the order of approximation may be increased along the common constrained leg so that a constrained node not belonging to the common side becomes enriched (node 1 in Fig. 6). We present all possible cases of enriching nodes of neighbors caused by raising p in a given element in Fig. 6. Note also that if, according to the results of Section 3, we attempt to examine changes of the error due to various possible refinements, then the trial enrichments of the element itself and of its neighbors may lead to some of the situations displayed in Fig. 7.

h-refinements

In the case of h-refinements, the restriction to 1-irregular meshes limits the local character of refinements: whenever a constrained element is subdivided, we must first bisect its neighbor. Of course, this procedure may cause a sequence of subdivisions of neighbors.

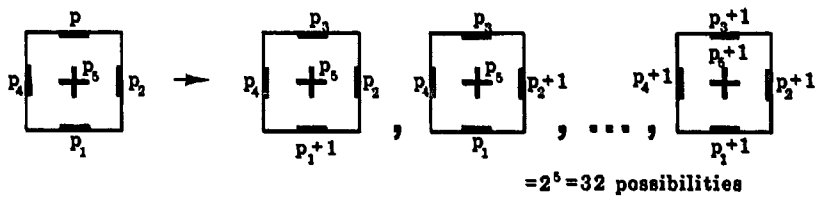


Fig. 7. All possible ways of enriching an element.

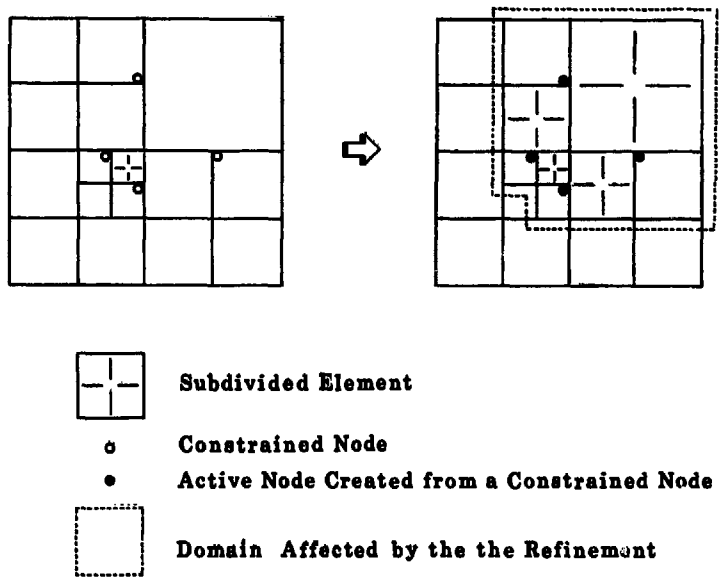


Fig. 8. Modifications of the mesh caused by subdivision of a constrained element.

Another modification of elements occurring during this process consists of changing constrained nodes into active nodes: if a constrained node of a given element belongs to a side of the larger neighbor which is bisected, then the element-son resulting from the bisection has the same size as its previously constrained neighbor, so there is no longer need for constraints and all the nodes along the common side become active.

The modifications of elements discussed above are illustrated in Fig. 8.

Analyzing various types of *h*-refinements, we must keep in mind the objective of determining changes of errors caused by subdividing elements. From this point of view, subdivision of an element may be performed in one of the 16 ways shown in Fig. 9a while subdividing its neighbors may remove constrained nodes, resulting in one of the three situations shown in Fig. 9b.

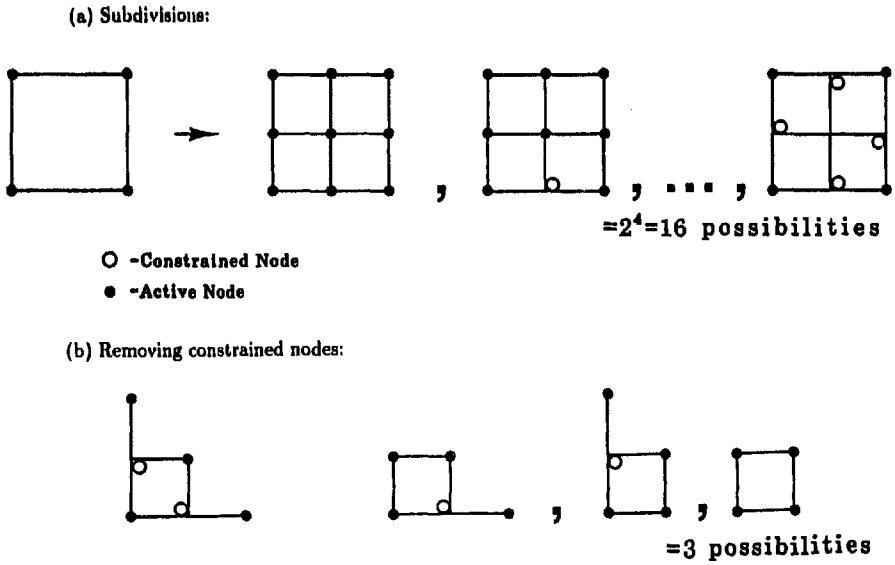


Fig. 9. Possible ways of subdividing an element and removing its constrained nodes.

5. An *h-p* mesh refinement algorithm

In constructing the algorithm two important questions arise:

(1) Is it sufficient to examine changes of the error caused only by single elementary refinements: raising p by 1, or subdividing the element only once?

(2) Is it necessary to introduce unrefinements in the process of optimization?

That the answer to the first question may be negative is suggested by the following example: adding symmetric degrees of freedom may not decrease the error if the solution is antisymmetric. Thus, the element may never by adequately refined if only one step forward is employed.

It is also easy to construct examples for which unrefinements are necessary: analyzing a one-dimensional singular solution (such as $u(x) = x^{\alpha}$), we find that raising p gives a larger decrease of the interpolation error than subdividing the element, even though, by considering a number of combined h- and p-refinements, the best mesh is found to be that which is geometrically graded towards the singularity with low p. If we raise p early in the process, it may be impossible to follow a refinement path toward a true optimal without unrefinement.

These observations suggest that multilevel checking and unrefinement may be important in a robust optimization algorithm. However, because of the complexity in implementing these procedures, we do not consider them in designing the present algorithm, except for the one-dimensional algorithm and for two-dimensional cases in problems with singularities, where limited versions of multilevel checking are introduced.

In the description of the algorithm below, the following notations are used:

$\Delta\theta_h, \Delta\theta_p$, decreases of the error indicators caused by h- and p-refinements,
$\Delta n_h, \Delta n_p$, the corresponding changes in the numbers of degrees of freedom,
$\alpha_1 \leq 1$, a parameter characterizing the algorithm,
α_2, the required accuracy tolerance (the algorithm stops when accuracy α_2 is attained).

The algorithm of h-p refinements

(1) For all elements in an initial mesh, compute the anticipated decreases of errors for all of the refinements displayed in Figs. 7, 9 which may actually occur.
(2) For every element evaluate $\Delta\theta_p/\Delta n_p, \Delta\theta_h/\Delta n_h$. (For interpolation errors, use the procedure outlined below.)
Set $\Delta\theta/\Delta n = \max(\Delta\theta_p/\Delta n_p, \Delta\theta_h/\Delta n_h)$ and store for every element.
(3) Scan the mesh and identify the largest value of $\Delta\theta/\Delta n$ in the mesh: $(\Delta\theta/\Delta n)_{\max}$.
(4) Create a list of elements for which $\Delta\theta/\Delta n \geq \alpha_1 \cdot (\Delta\theta/\Delta n)_{\max}$.
(5) Perform refinements of elements on the list. The types of refinements corresponds to information stored in (2).
(6) Update the solution: solve the problem on the new mesh.
(7) Estimate the global error $J = \Sigma_k \theta_k$. If $J \leq \alpha_2$, stop, otherwise go to (1).

The procedure for computing $\Delta\theta_p/\Delta n_p$ and $\Delta\theta_h/\Delta n_h$ for interpolation error indicators in the two-dimensional case

(2.1) Subroutine for determining $\Delta\theta_p/\Delta n_p$:
1. Evaluate new orders of nodes of a given element K_i and its neighbors.
2. Compute $\Delta\theta_p$ using errors for K_i and its neighbors in the situations shown in Fig. 7.
3. Determine the number of the new degrees of freedom Δn_p and compute $\Delta\theta_p/\Delta n_p$.

(2.2) Subroutine for determining $\Delta\theta_h/\Delta n_h$:
1. Evaluate the sequence of elements that must be subdivided due to refining the given element K_i and two-to-one rule (see Fig. 8).
2. Determine elements with disappearing constrained nodes (Fig. 8).
3. Compute $\Delta\theta_h$ using errors for all the subdividing elements found in 1 and elements listed in 2; appropriate situations from Figs. 9a and 9b must be considered.
4. Count the number of new degrees of freedom Δn_h and compute $\Delta\theta_h/\Delta n_h$.

6. Numerical examples

We shall now present several numerical examples designed to illustrate the *h-p* refinement strategy proposed in previous sections. In all of these examples, we construct *h-p* approximations of functions which exhibit some type of irregular behavior due to singularities or high gradients since, for very smooth functions, our optimization scheme generally leads to only *p*-refinements. All examples are analyzed using the algorithms presented in Section 5. As a first model error estimate, we use the interpolation estimates as an error indicator, assuming that the interpolated function is known explicitly. However, we also consider cases driven by interpolation error estimates of solutions of model boundary value problems obtained by post-processing computed values and gradients of the finite element approximations. Our post-processing method is based on extraction formulas and a brief discussion of the method is given in [2].

In the examples that follow, we attempt to analyze the accuracy of approximations on optimal meshes and make comparisons of results and rates of convergence obtained by other refinement techniques. We also verify numerically that the optimal rate of convergence is exponential: $\|e\| \leq C \exp(-\beta \sqrt[3]{N})$, N being the number of degrees-of-freedom [6]. Details concerning issues of numerical integration which are crucial in these studies are collected in Appendix A.

EXAMPLE 1. Consider the function

$$u = x^{\alpha}, \quad x \in (0, 1), \quad \text{with} \quad \alpha = 0.6 .$$

The initial mesh is uniform and consists of 10 linear elements. Figure 10 shows the optimal mesh obtained after a sequence of refinements. Our algorithm leads to a mesh geometrically graded towards the singularity. Only a small number of elements were enriched to second order and only one element with a node at singularity was enriched to third order.

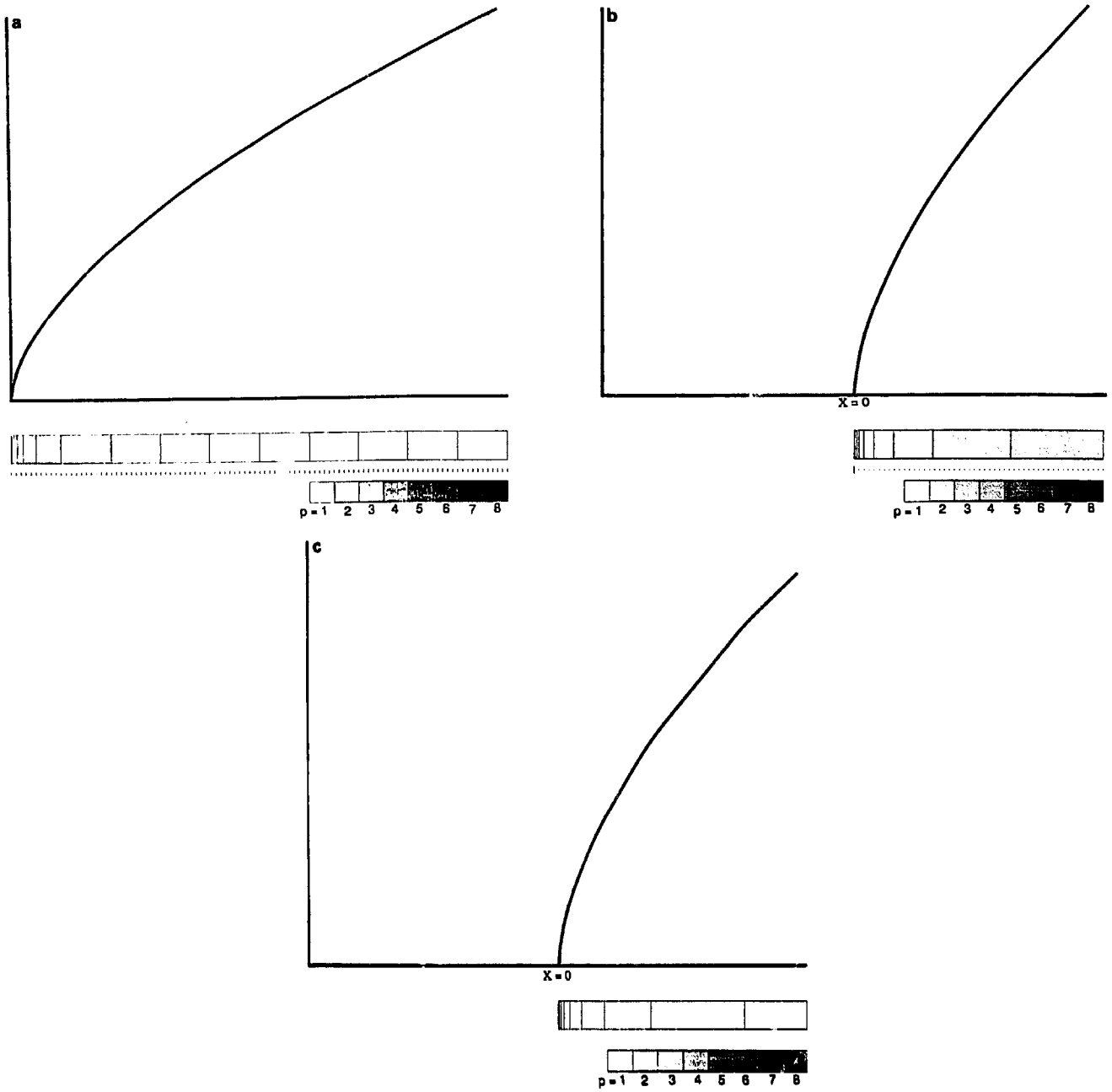


Fig. 10. (a) An optimal h-p mesh for $u = x^{\alpha}$, $\alpha = 0.6$, for error indicators obtained by exact interpolation of u. (b) Zoom of (a) centered at the singularity. Magnification 100. (c) Zoom of (a) centered at the singularity. Magnification 10^9.

We also considered $u = u(x)$ to be a solution of the boundary value problem

$$-u''(x) = -\alpha(\alpha - 1)x^{\alpha - 2}, \quad \alpha = 0.6, \quad u(0) = 0, \quad u(1) = 1 \tag{6.1}$$

and used the post-processed finite element approximation u_h of $u(x)$ instead of exact u in the optimization procedures. The resulting mesh is slightly different than in the first case: it is also geometrically graded but no p-enrichments appear.

Figure 11 shows comparisons of computed rates-of-convergence for various types of refinement strategies. Results of pure h-adaptive refinement are not shown as they were essentially coincident with the h-p refinements in this case. The pure p-adaptivity resulted in raising only the spectral order in the element containing the singularity.

EXAMPLE 2. Next we consider a function which is 'almost singular', namely a function with singularity slightly outside the domain:

$$u(x) = (x + 0.001)^{\alpha}, \quad \alpha = 0.6, \quad x \in (0, 1) .$$

The initial mesh is the same used in Example 1. The computed optimal mesh is shown in Fig. 12. As in the previous singular solution, the mesh is finer close to the left endpoint but now subdivisions of elements do not appear extensively in the refinements: beyond five elements past the origin only p-refinements are chosen and both subdivision and high p-enrichments are used only near the singularity.

The optimization results obtained using post-processed finite element solutions was conducted assuming that u is a solution of the boundary value problem,

$$-u''(x) = -\alpha(\alpha - 1)(x + 0.001)^{\alpha - 2}, \quad \alpha = 0.6, \quad u(0) = (0.001)^{\alpha}, \quad u(1) = (1.001)^{\alpha} . \tag{6.2}$$

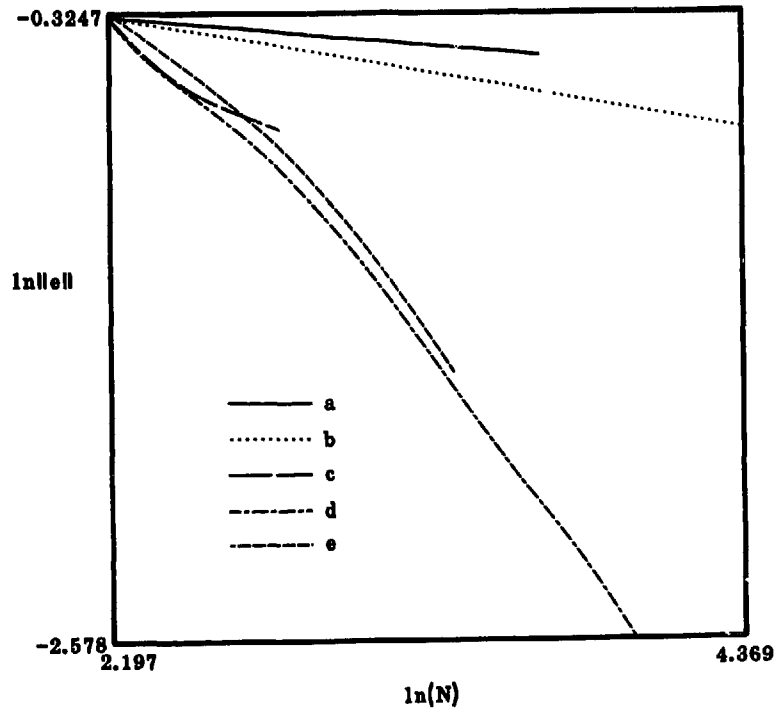


Fig. 11. Rates-of-convergence for various types of refinement strategies obtained for a singular function $u = x^{\alpha}$, $\alpha = 0.6$. (a) h-uniform, (b) p-uniform, (c) p-adaptive, (d) h-p adaptive with exact interpolation, (e) h-p adaptive with post-processing.

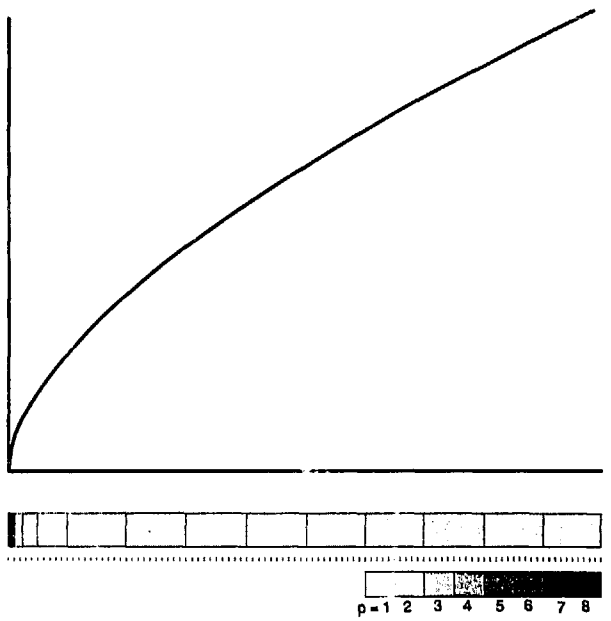


Fig. 12. An opitmal h-p mesh for $u = (x + 0.001)^{\alpha}$, $\alpha = 0.6$, for error indicators obtained by exact interpolation of u.

This approach resulted in meshes identical to those produced by interpolation.

Rates of convergence are shown in Fig. 13. We observe that the performance of p-adaptive methods in this problem is as good as the h-p scheme. The pure p-scheme, however, produces very high order p when a small number of elements is used.

EXAMPLE 3. The following two problems illustrate the performance of the h-p strategy in the case of smooth (even analytic) functions with high gradients in some small subdomains. The model function is

$$u(x) = (1 - x)[\tan^{-1}(\alpha(x - x_0)) + \tan^{-1}\alpha x_0], \quad x \in (0, 1). \tag{6.3}$$

Optimal meshes are computed for the cases $\alpha = 50$ and $\alpha = 200$ (α controls the magnitude of gradient at x_0) with $x_0 = 4/9$.

The optimal meshes are presented in Figs. 14 and 15. In both cases, combined subdivisions and enrichments are observed in the vicinity of x_0. In the case $\alpha = 200$, more h-refinements have been performed than for $\alpha = 50$.

Mesh optimization with post-processing was conducted assuming that $u(x)$ satisfies

$$\begin{aligned} &-(a(x)u'(x))' = f(x), \\ &u(0) = 0, \qquad u(1) = 0, \\ &a(x) = 1/\alpha + \alpha(x - x_0)^2, \\ &f(x) = 1 + [1 - 2\alpha(x - x_0)(\tan^{-1}\alpha(x - x_0) + \tan^{-1}\alpha x_0)]/a. \end{aligned} \tag{6.4}$$

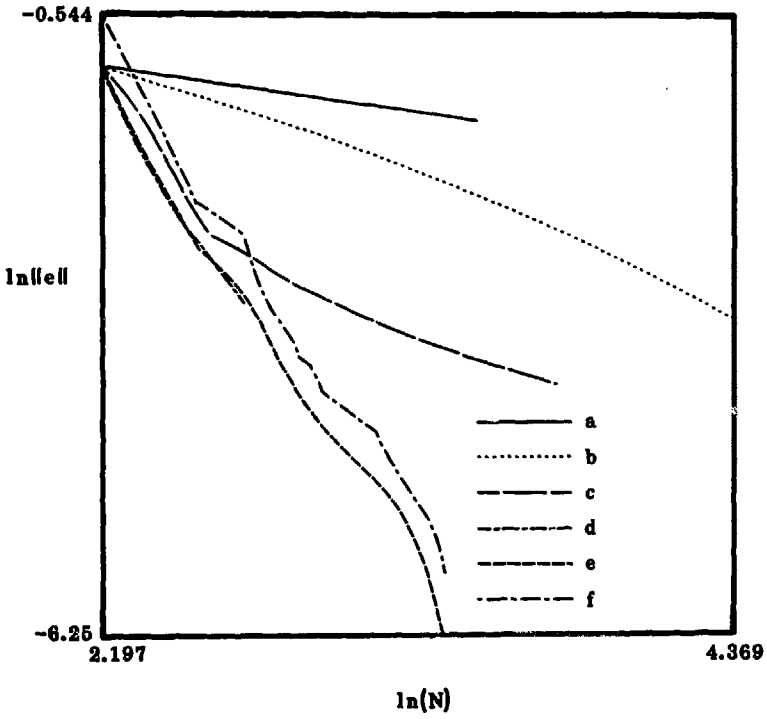


Fig. 13. Rates-of-convergence for various types of refinement strategies obtained for a function $u = (x + 0.001)^{\alpha}$, $\alpha = 0.6$. (a) h-uniform, (b) p-uniform, (c) h-adaptive, (d) p-adaptive, (e) h-p adaptive with exact interpolation, (f) h-p adaptive with post-processing.

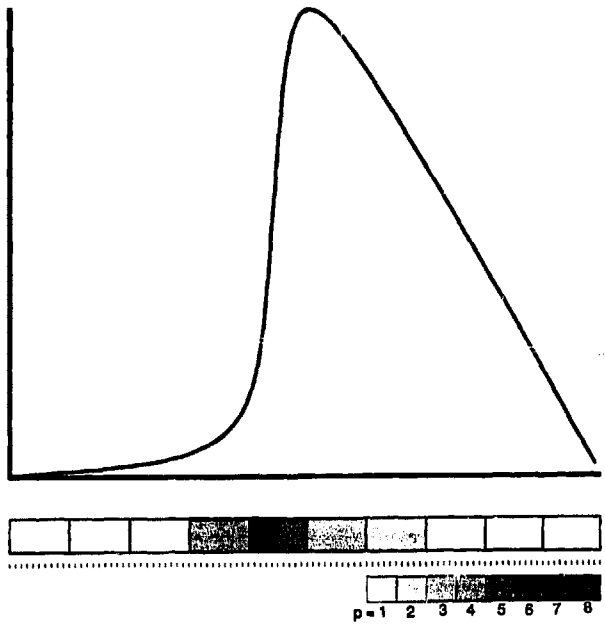


Fig. 14. An optimal h-p mesh for $u = (1 - x)[\tan^{-1}(\alpha(x - x_0)) + \tan^{-1}\alpha x_0]$, $\alpha = 50$, $x_0 = 4/9$, for error indicators obtained by exact interpolation of u.

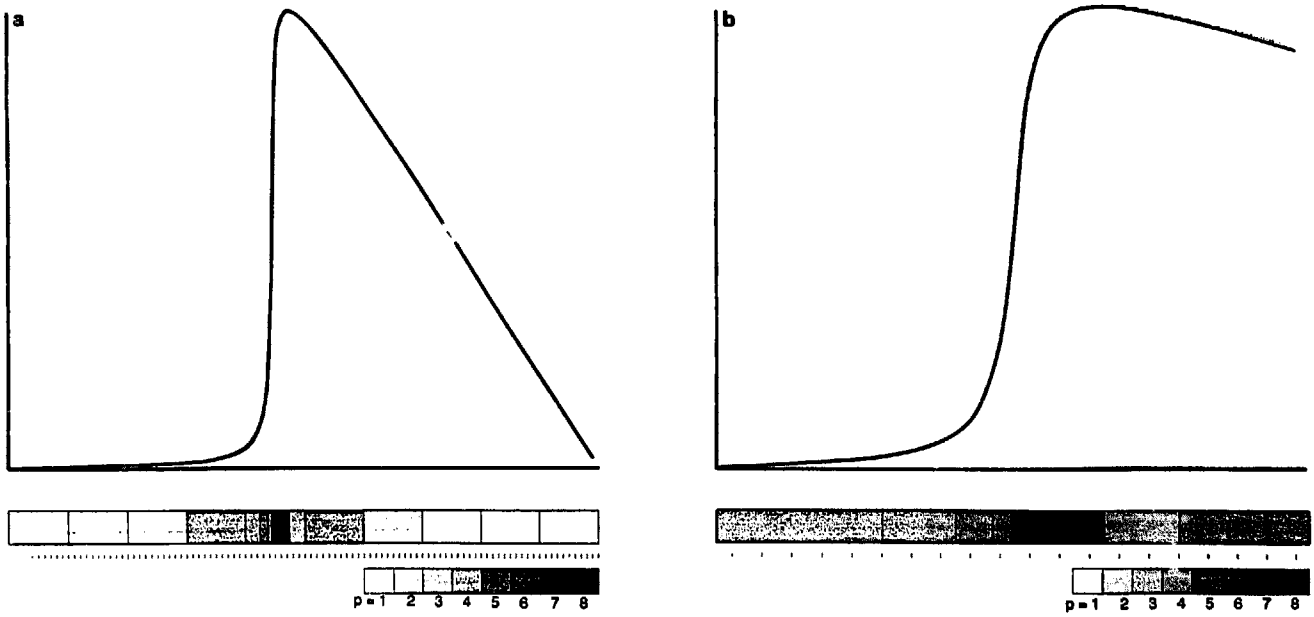


Fig. 15. (a) An optimal *h-p* mesh for $u=(1-x)[\tan^{-1}\alpha(x-x_0)+\tan^{-1}\alpha x_0]$, $\alpha=200$, $x_0=4/9$, for error indicators obtained by exact interpolation of u. (b) Zoom of (a) centered at $x_0=4/9$. Magnification 100.

The resulting meshes (Figs. 16 and 17) are not exactly the same as those obtained from pure interpolation; however, in general they have very similar features. Also, the accuracy for these meshes is only slightly inferior to interpolation (Figs. 18 and 19).

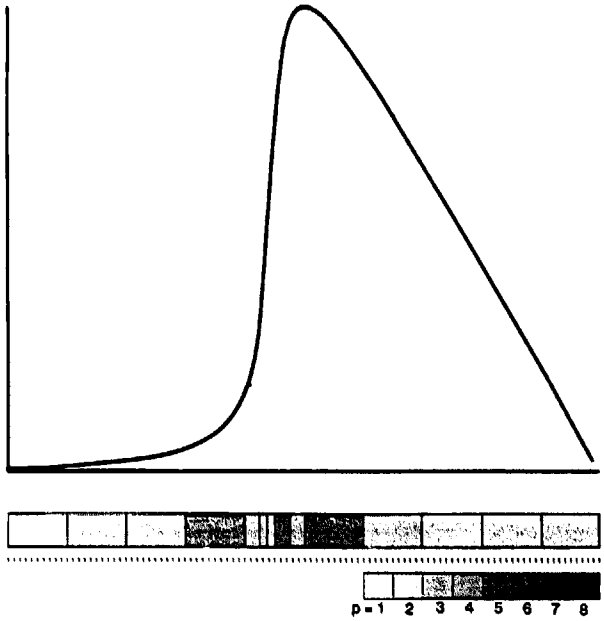


Fig. 16. An optimal *h-p* mesh for $u=(1-x)(\tan^{-1}\alpha(x-x_0)+\tan^{-1}\alpha x_0)$, $\alpha=50$, $x_0=4/9$, for error estimates obtained using extraction procedures.

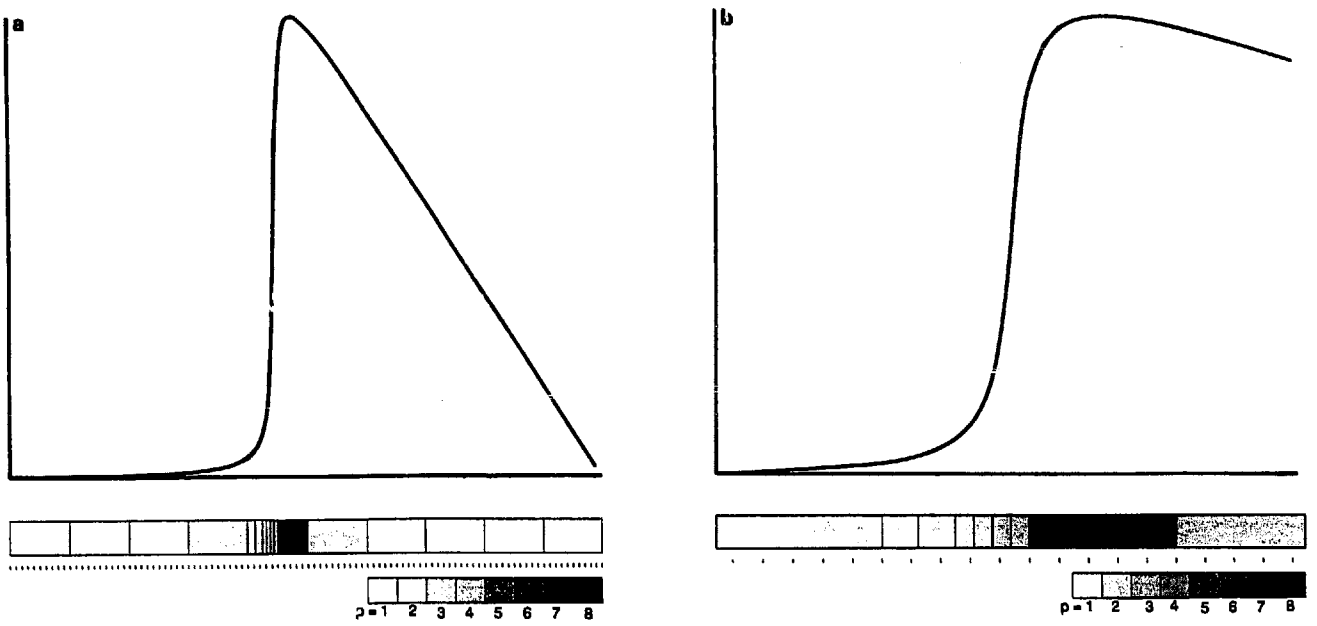


Fig. 17. (a) An optimal h-p mesh for $u=(1-x)(\tan^{-1}\alpha(x-x_0)+\tan^{-1}\alpha x_0)$, $\alpha=200$, $x_0=4/9$, for error estimates obtained using extraction formulas. (b) Zoom of (a) centered at $x_0=4/9$. Magnification 100.

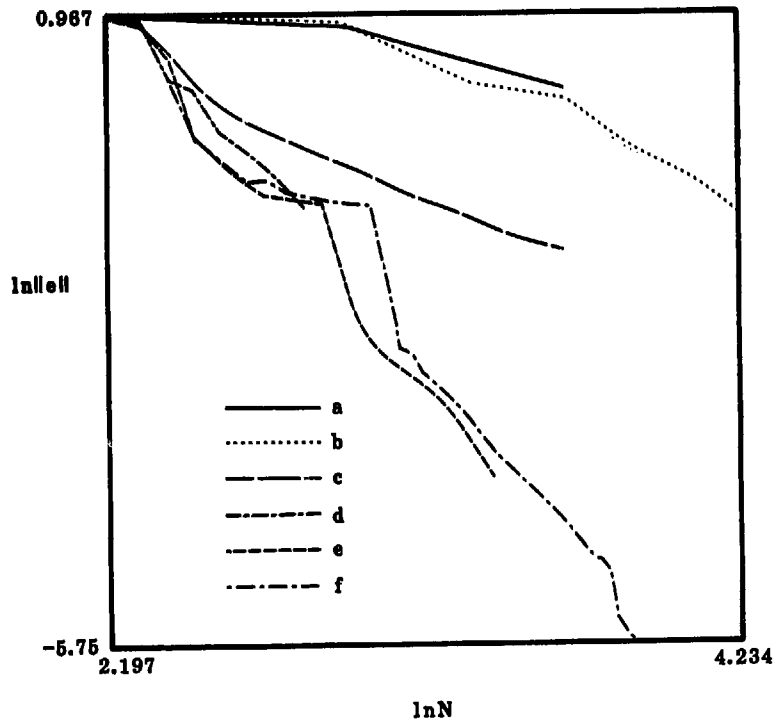


Fig. 18. Rates-of-convergence for various types of refinement strategies for $u=(1-x)(\tan^{-1}\alpha(x-x_0)+\tan^{-1}\alpha x_0)$, $\alpha=50$, $x_0=4/9$. (a) h-uniform, (b) p-uniform, (c) h-adaptive, (d) p-adaptive, (e) h-p adaptive with exact interpolation, (f) h-p adaptive with post-processing.

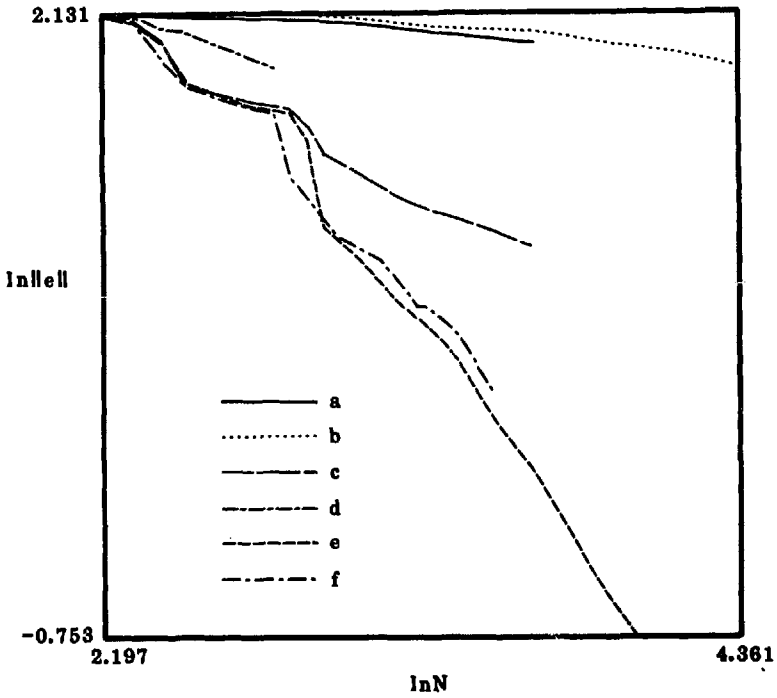


Fig. 19. Rates-of-convergence for various types of refinement strategies for $u=(1-x)(\tan^{-1}\alpha(x-x_0)+\tan^{-1}\alpha x_0)$, $\alpha=200$. $x_0=4/9$. (a) h-uniform, (b) p-uniform, (c) h-adaptive, (d) p-adaptive, (e) h-p adaptive with exact interpolation, (f) h-p adaptive with post-processing.

EXAMPLE 4. A two-dimensional h-p optimization problem is now considered. We approximate the function

$$u(x, y)=\frac{3}{2}\, r^{2/3}\sin\frac{2\theta}{3}\,(1-x^2)(1-y^2),$$

$$r=\sqrt{x^2+y^2},\quad \theta=\tan^{-1}(-x/y),\quad (x, y)\in\Omega,$$

where Ω is the L-shaped domain shown in Fig. 20.

It turns out that the algorithm introduced in Section 6 fails in the case of the singular functions. One finds that after the first sequence of p-enrichments of elements surrounding the

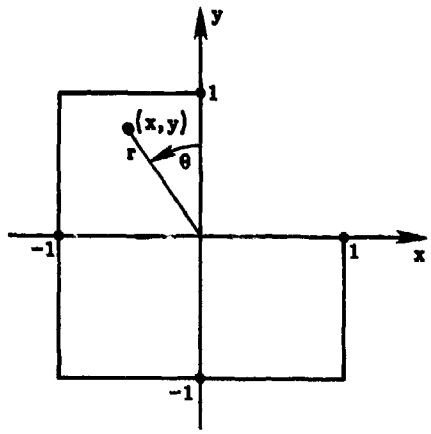


Fig. 20. L-shaped domain.

singularity, these elements are not subsequently refined any more. The explanation is as follows.

Unlike the one-dimensional case, the two-dimensional interpolation operator cannot be defined by a minimization principle that produces a function from the space of shape functions which minimizes the error in the H^1-seminorm (because its boundary values must be defined only by the boundary values of a given function to guarantee continuity of the approximation). For the two-dimensional case, we are not guaranteed that expanding the space of shape functions implies a decrease of the error, though in most cases this happens. It turns out that for functions like $r^{2/3} \sin \frac{2}{3}\theta$, raising p from 2 to higher orders causes an increase of interpolation error (at least for the kinds of quadratures we use, see Appendix A). Moreover, a single subdivision of only one of the elements surrounding the corner leaves its boundary stiff (i.e., introduces only constrained nodes) and this causes only an insignificant decrease of the error. As a result, the algorithm avoids refinements in zones where they are most needed.

To avoid these difficulties, we propose the following modifications of the algorithm.

(1) We exclude corner elements from p-refinements; this corresponds to the techniques of adaptivity around singularities suggested by Babuška and Guo [6], where corner elements are kept at fixed low order.

(2) We consider the combined bisection of all three corner elements together; this does not leave stiff element boundaries and results in a significant decrease of the interpolation error.

Tests suggest that, with these modifications, the performance of the two-dimensional algorithm is quite similar to the one-dimensional case.

Figure 21 shows the optimal mesh obtained with the modified algorithm. Like the one-dimensional case, the size of elements is graded geometrically towards the singularity. However, in elements far from the corner, more p-enrichments can be observed.

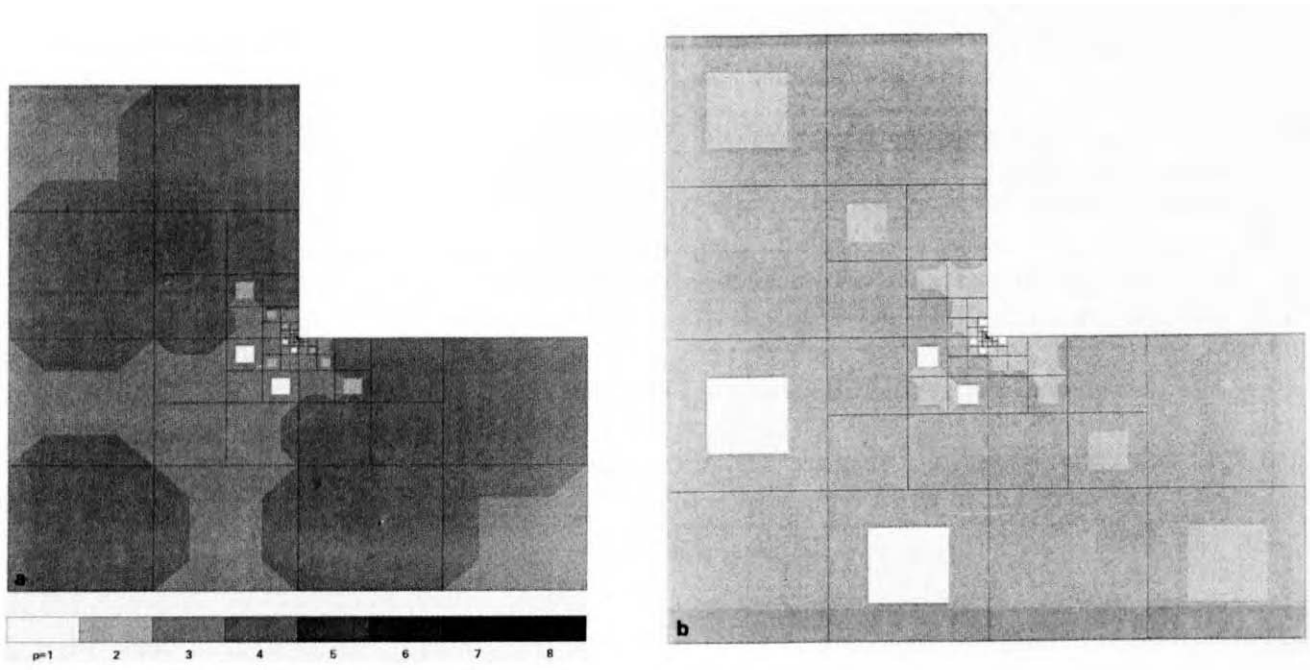
We next compare the accuracy of approximations obtained with our algorithm with those produced on meshes geometrically graded with p varying linearly with the number of layers of elements surrounding the corner. Figure 22 suggests that in both cases the convergence is better than polynomial order; however, we are unable to verify that the error is $O(\exp(\beta\sqrt[3]{N}))$.

EXAMPLE 5. In this case, we consider an almost singular function on an L-shaped domain obtained by shifting the singularity outside the domain to the point $(\varepsilon, \varepsilon)$, $\varepsilon = 0.001$:

$$u(x, y) = \frac{3}{2} r^{2/3} \sin \frac{2\theta}{3}, \qquad r = \sqrt{(x-\varepsilon)^2 + (y-\varepsilon)^2}.$$

In this case, we use the basic algorithm with only the second of modifications introduced for singularities: we subdivide three corner elements but allow them to acquire separate p enrichments. The mesh resulting from the optimization procedure is shown in Fig. 23. As in the one-dimensional case, a number of h-refinements of corner elements are performed and then p-enrichments follow and dominate the process, resulting in very high-order corner elements.

The accuracy of approximation is again compared with that of geometrically graded meshes with increasing p, Fig. 24. In this case, the asymptotic rate-of-convergence ($O(\exp\sqrt[3]{N})$) is easily verified.



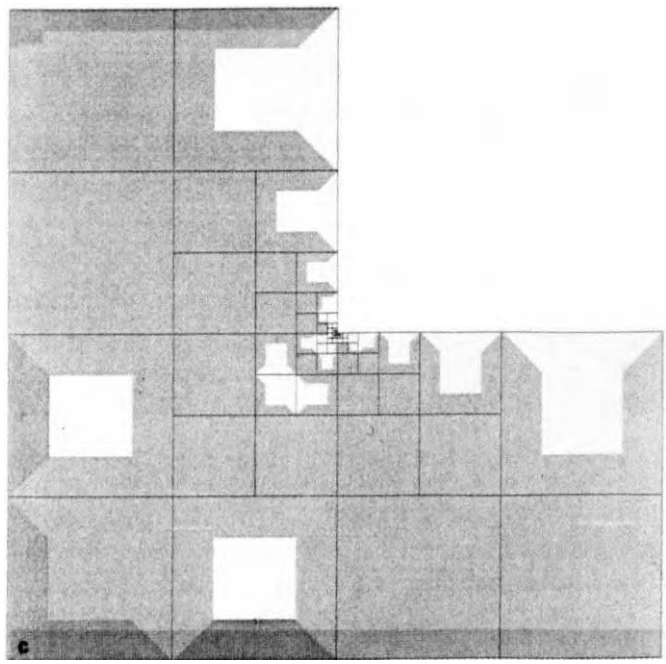


Fig. 21. (a) An optimal *h-p* mesh for $u = \frac{2}{3} r^{2/3} \sin \frac{2\theta}{3} (1 - x^2)(1 - y^2)$. (b) Zoom of (a) centered at the singularity. Magnification 16. (c) Zoom of (a) centered at the singularity. Magnification 256.

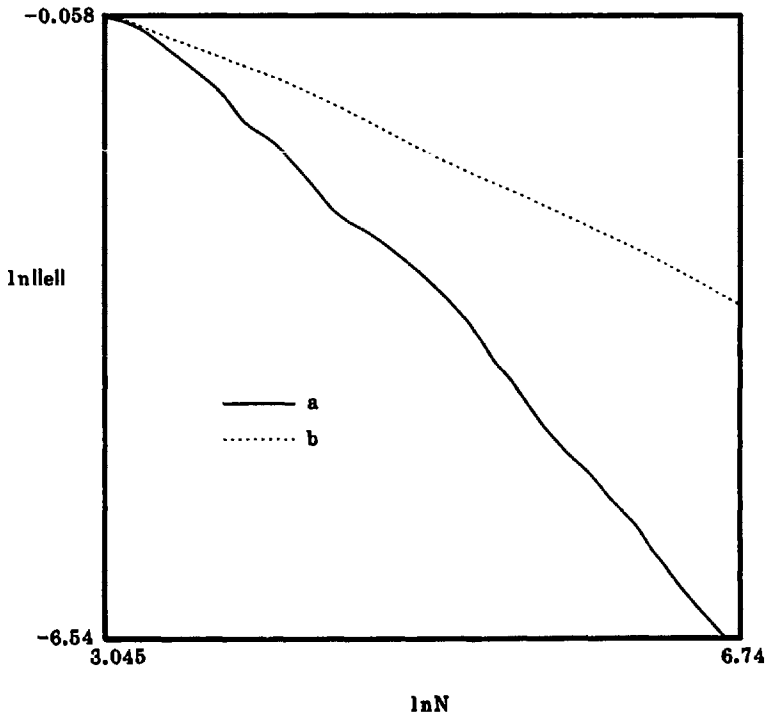


Fig. 22. Rates-of-convergence for $u = \frac{2}{3}r^{2/3}\sin\frac{2\theta}{3}(1-x^2)(1-y^2)$ for two kinds of meshes: (a) *h-p* adaptive, (b) geometrically graded with increasing p.

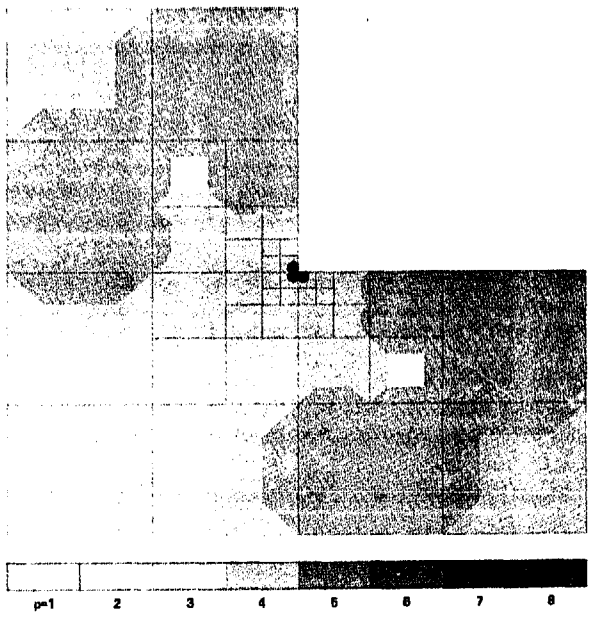


Fig. 23. Optimal *h-p* mesh for a function $u = \frac{3}{2}r^{2/3}\sin\frac{2\theta}{3}$, $r = \sqrt{(x-\varepsilon)^2 + (y-\varepsilon)^2}$, $\varepsilon = 0.001$.

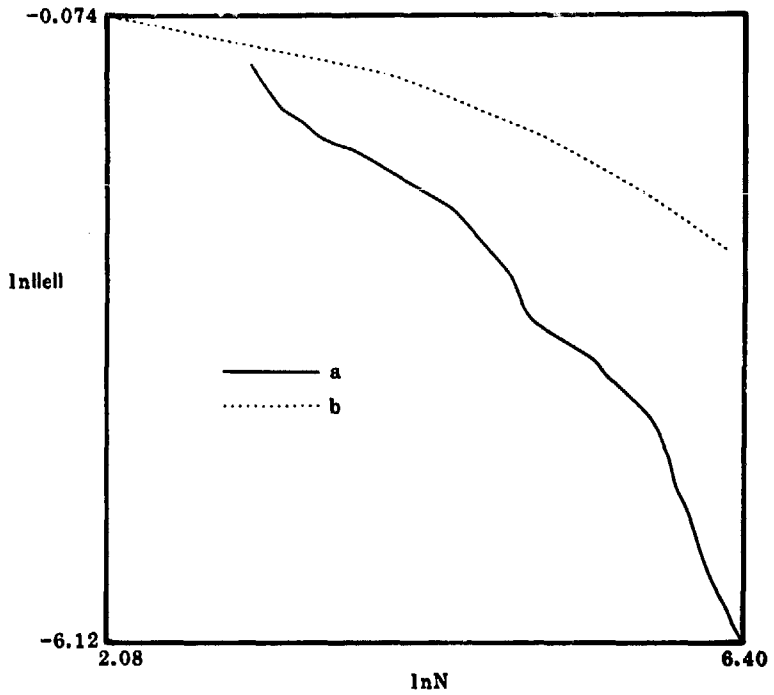


Fig. 24. Rates-of-convergence for $u = \frac{3}{2}r^{2/3}\sin\frac{2\theta}{3}$, $r = \sqrt{(x-\varepsilon)^2 + (y-\varepsilon)^2}$, $\varepsilon = 0.001$, for two types of meshes: (a) h-p adaptive, (b) geometrically graded with increasing p.

EXAMPLE 6. In this calculation, the performance of the optimization algorithm for the case of a smooth function with large concentrated gradients was tested. The function approximated in the refinement process is

$$u(x, y) = \tan^{-1}(\alpha(r - r_0)),$$
$$r^2 = (x - x_0)^2 + (y - y_0)^2, \quad (x, y) \in \Omega = (0, 1) \times (0, 1),$$
$$x_0 = 1.25, \quad y_0 = -0.25, \quad \alpha = 60, \quad r_0 = 1.0.$$

This function is very flat throughout most of the domain but exhibits a rapid jump along the circle $r = r_0$.

The optimal meshes observed with h-p, h- and p-adaptive procedures are shown in Fig. 25. In the h-p approach, both h- and p-refinements were chosen by the optimization algorithm. Most of the h-refinements occurred in the beginning of the adaptive process, then p-enrichments dominated.

A comparison of accuracy of the three adaptive strategies and of results obtained by uniform h and p refinements are presented in Fig. 26. There the superiority of the h-p approach is seen and results confirm that an exponential asymptotic behavior of the error is attained. At some stage of the h-p optimization process, we observe a deterioriation of the convergence rate for several steps, after which it rapidly improves. This phenomenon is analogous to that discussed in Example 4 and is attributed to the effect of the process delaying refinements in some elements due to a momentary increase in the local interpolation error caused by the constrained refinement strategy. Fortunately, refinements of neighboring

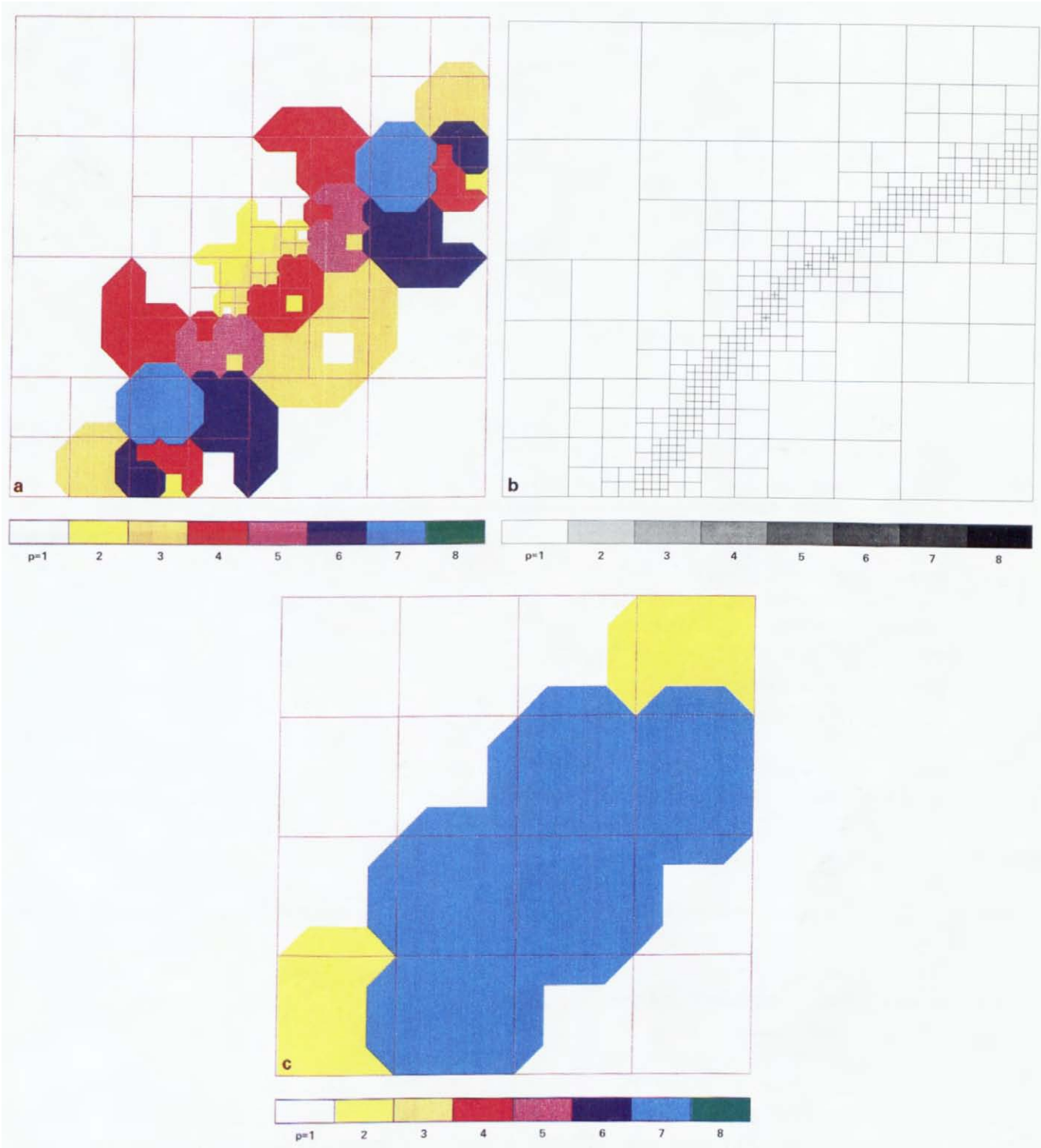


Fig. 25. (a) An optimal h-p mesh for a function $u = \tan^{-1} \alpha(r - r_0)$, $r^2 = (x - x_0)^2 + (y - y_0)^2$, $x_0 = 1.25$, $y_0 = 0.25$, $r_0 = 1.0$, $\alpha = 60$, for error estimates obtained by exact interpolation. (b) An optimal h-adaptive mesh ($p = 1$) for a function $u = \tan^{-1} \alpha(r - r_0)$, $\alpha = 60$, for error estimates obtained by exact interpolation. (c) An optimal p-adaptive mesh for a function $u = \tan^{-1} \alpha(r - r_0)$, $\alpha = 60$, for error estimates obtained by exact interpolation.

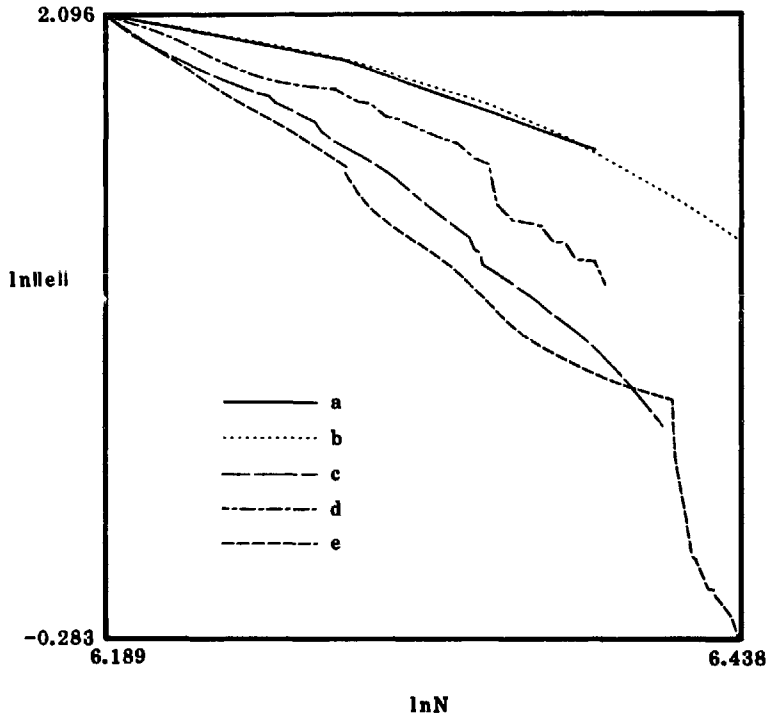


Fig. 26. Rates-of-convergence for a function $u = \tan^{-1} \alpha(r - r_0)$, $\alpha = 60$, for various types of refinement strategies. (a) h-uniform, (b) p-uniform, (c) h-adaptive, (d) p-adaptive, (e) h-p adaptive.

elements ultimately effect the element itself, and this finally leads to its refinement with the result being a very rapid decrease of the local error.

The performance of the h-adaptive approach appears to be only slightly inferior to the h-p method. Asymptotically, however, the h-method can lead to a global rate-of-convergence no better than that of uniform refinements [7].

EXAMPLE 7. We next consider the case of a function possessing both a point singularity and a zone of large but finite gradients:

$$u(x, y) = \frac{3}{2} r^{2/3} \sin \frac{2\theta}{3} + \tan^{-1}(\alpha(\xi - \xi_0)),$$

$$\xi = (x + y)/\sqrt{2}, \quad \alpha = 30, \quad \xi_0 = -0.75, \quad (x, y) \in \Omega.$$

Here r and θ are as defined previously and Ω is the L-shaped domain shown in Fig. 20. We use the version of the algorithm with modifications for singularities (see Example 4).

The mesh resulting from the optimization scheme is shown in Fig. 27. As could be expected, it exhibits features of the optimal meshes determined earlier for both problems with singularities and with shock-like zones. Near the corner, we observe strong domination of h-refinements, while close to the shock, combined h- and p-refinements are chosen, with a limited level of h-refinement, and a high order of p-enrichments. Figure 28 shows the

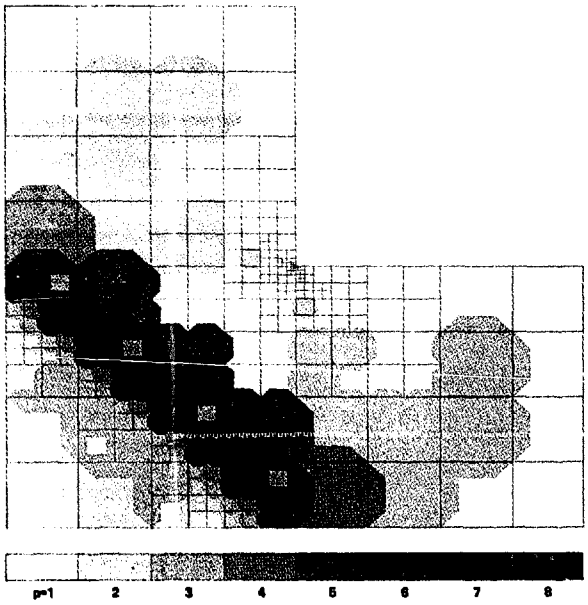


Fig. 27. An optimal *h-p* mesh for a function $u = \frac{3}{2}r^{2/3}\sin\frac{2\theta}{3} + \tan^{-1}\alpha(\xi - \xi_0)$, $\xi = (x+y)/\sqrt{2}$, $\alpha = 30$, $\xi_0 = -0.75$, for error estimates obtained by exact interpolation.

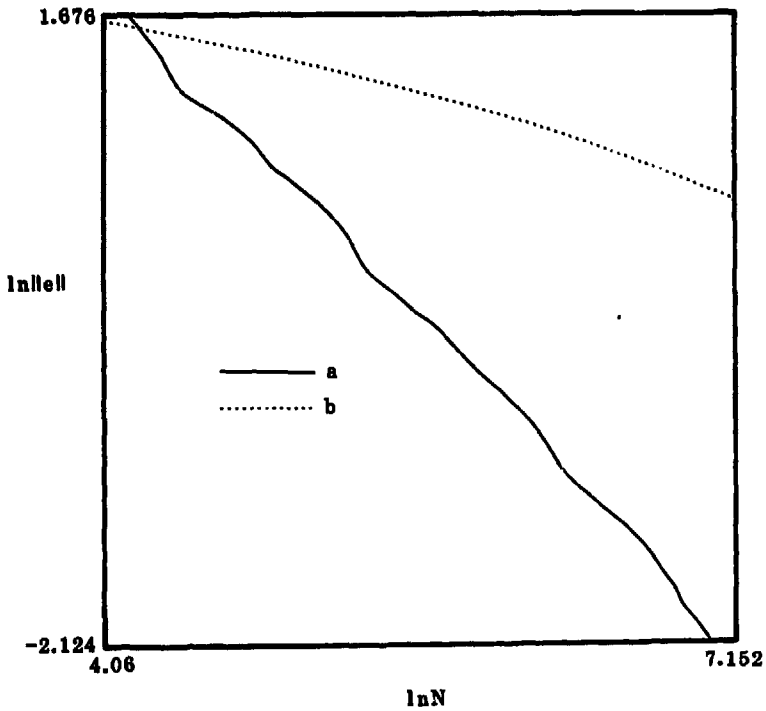


Fig. 28. Rates-of-convergence for $u = \frac{3}{2}r^{2/3}\sin\frac{2\theta}{3} + \tan^{-1}\alpha(\xi - \xi_0)$, $\xi = (x+y)/\sqrt{2}$, $\xi_0 = -0.75$, $\alpha = 30$, for two kinds of meshes: (a) *h-p* adaptive, (b) geometrically graded with increasing p.

computed accuracy of the approximation on optimal meshes compared with that obtained for geometrically graded meshes with p increasing with the number of layers of elements. This result again determines the superior performance of the h-p strategy even though we are able to detect the exponential ($\exp(\beta\sqrt[3]{N})$) behavior of the error only for the geometrically graded meshes.

EXAMPLE 8. As a final numerical example, we consider a model two-dimensional boundary value problem

$$-\Delta u = f \quad \text{in } \Omega = (0,1)\times(0,1), \qquad u = 0 \quad \text{on } \partial\Omega\,, \tag{6.5}$$

where $f = f(x, y)$ is chosen to correspond to the exact solution.

$$u(x, y) = x(1-x)y(1-y)\tan^{-1}\left[20\left(\frac{x+y}{\sqrt{2}} - 0.80\right)\right].$$

The problem is solved on an initial mesh and the optimization algorithm is applied using estimates based on post-processed values of u and ∇u in computing the interpolation error. The finite element solution is then updated after a sequence of refinements.

Figure 29 presents the computed optimal mesh. Comparing it with the mesh corresponding to the same global accuracy but obtained using exact values of u and ∇u in the optimization process (Fig. 30), we find that the results in terms of number of degrees-of-freedom are essentially the same.

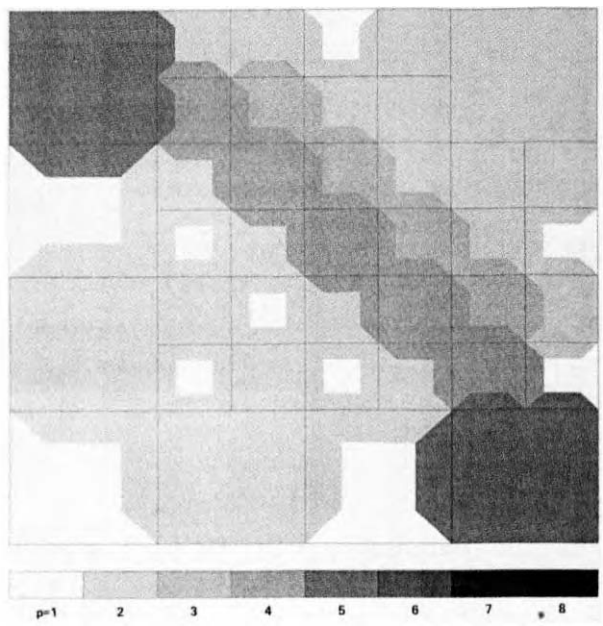


Fig. 29. An optimal h-p mesh for a function $u = x(1-x)y(1-y)\tan^{-1}\alpha(\xi - \xi_0)$, $\xi = (x+y)/\sqrt{2}$, $\xi_0 = 0.8$, $\alpha = 20$, for error estimates obtained using extraction procedures.

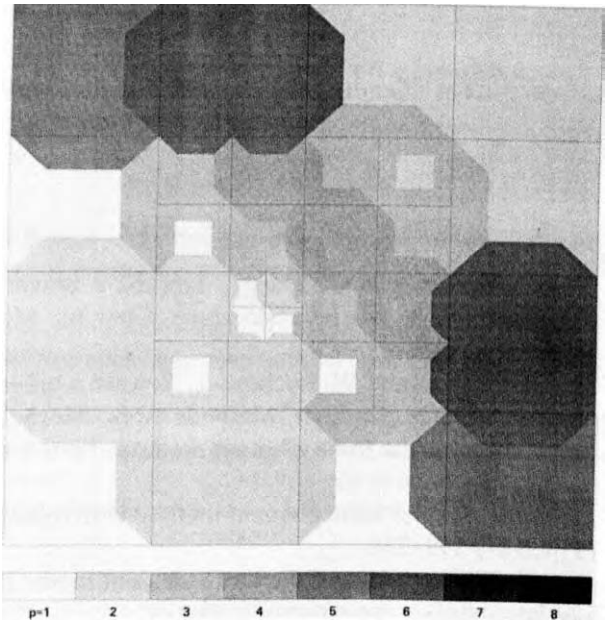


Fig. 30. An optimal *h-p* mesh for a function $u = x(1-x)y(1-y)\tan^{-1}\alpha(\xi - \xi_0)$, $\xi = (x+y)/\sqrt{2}$, $\xi_0 = 0.8$, $\alpha = 20$, for error estimates obtained by exact interpolation.

Appendix A. Numerical integration

Calculation of interpolation errors (defined in [2]) involves the crucial design of an appropriate numerical integration scheme. To compare the accuracies of interpolants for different spaces of interpolating functions (on an enriched or subdivided element), we must ensure that the discrete representation of an interpolated function—its values at integration points—is the same for all cases considered. For this reason, integration over any element of order p is performed by subdividing it into subdomains—sons resulting from anticipated divisions, and using for each of them a $p+2$-point Gaussian quadrature. In practice, in the one-dimensional case such an approach guarantees that the interpolation error decreases when the space of shape functions is extended; in the two-dimensional case, this property is not guaranteed, but increases of the error seem to occur only in rare circumstances.

The problem of computing integrals of singular functions is resolved herein as follows:

(1) In the one-dimensional case, special Gaussian quadrature schemes are used that integrate exactly products $x^\alpha f(x)$, $f(x)$ being a polynomial, $\alpha > -1$.

(2) In the two-dimensional case, functions $f(x, y)$ with an r^α-type singularity at a corner of a rectangular element are considered. The integration procedure consists of subdividing the element into two triangles and introducing new coordinates, $r = x$, $\theta = y/x$, with Jacobian $J = x$. This transformation leads to the integration of the nonsingular function $f \cdot J$ over a rectangular domain and can be carried out by standard Gaussian quadrature procedures.

Acknowledgement

The support of the Office of Naval Research is gratefully acknowledged.

References

1. L. Demkowicz, J.T. Oden, W. Rachowicz and O. Hardy, Toward a univeral *h-p* adaptive finite element strategy, Part 1. Constrained approximation and data structure, Comput. Methods Appl. Mech. Engrg. 77 (1989) 79–112.
2. J.T. Oden, L. Demkowicz, T.A. Westerman and W. Rachowicz, Toward a universal *h-p* adaptive finite element strategy, Part 2. A posteriori error estimates, Comput. Methods Appl. Mech. Engr. 77 (1989) 113–180.
3. B. Guo and I. Babuška, The *h-p* version of the finite element method, Parts 1 and 2, Comput. Mech. 1 (1986) 21–41, 203–220.
4. I. Babuška and M. Suri, The *h-p* version of the finite element method with quasiuniform meshes, RAIRO Math. Mod. and Numer. Anal. 21 (2) (1987) 199–238.
5. W. Gui and I. Babuška, The *h*, *p* and *h-p* versions of the finite element method in one dimension, Parts 1, 2, 3, Numer. Math. 49 (1986) 577–683.
6. I. Babuška and B. Guo, The *h-p* version of the finite element method for domains with the curved boundaries, Tech. Note BN-1057, Institute for Physical Science and Technology, Univ. of Maryland, 1986.
7. L. Demkowicz, Ph. Devloo and J.T. Oden, On an *h*-type mesh refinement strategy based on minimization of interpolation errors, Comput. Methods Appl. Mech. Engrg. 53 (1985) 67–89.