COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 77 (1989) 79–112
NORTH-HOLLAND

TOWARD A UNIVERSAL *h-p* ADAPTIVE FINITE ELEMENT STRATEGY, PART 1. CONSTRAINED APPROXIMATION AND DATA STRUCTURE

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

Received 27 March 1989

This is the first of a series of papers devoted to the development of an adaptive finite element method to solve various boundary value problems arising in solid and fluid mechanics. The paper deals with the general concepts of *h-p* finite element methods, constrained approximation, and data structure and related issues. Several numerical examples illustrate the main ideas of the method.

1. Introduction

With over three decades now past since the introduction of finite element methods into the engineering literature, the basic features of finite element approximations are now widely known: piecewise polynomial approximation of weak forms of boundary- and initial-value problems over a partition of the domain of the solution being sought. The subdomains making up the partition are the finite elements and the independent polynomials defining the approximation over the elements are called shape functions. At first thought, it may be surprising that only very recently finite element methods have emerged which attempt to distribute these basic approximation parameters in some optimal way—the sizes h of the elements and the orders p of the shape functions. But further consideration soon reveals that such *h-p* versions of the finite element method represent significant departures from conventional finite element techniques and require the resolution of several formidable problems in their effective implementation, including new data structures, equation solvers and criteria for choosing a distribution of mesh sizes and approximation orders.

In this work and in accompanying papers, we present a general approach to the development and implementation of an *h-p* version of the finite element method. In particular, we describe adaptive *h-p* methods in which a strategy is developed for choosing local mesh sizes and polynomial degrees so as to reduce local (elementwise) errors in appropriate norms below some preset tolerance. In this way, an optimal mesh is attained in the sense that an attempt is made to produce an *h-p* distribution with the least number of degrees of freedom required to deliver a solution with a given accuracy. Such a strategy requires that several key issues be dealt with. These include:

(1) first and foremost, a data structure and data management plan that efficiently copes with the significant bookkeeping problem of labeling, tracking, and addressing element numbers, connectivity, shape-function degrees, node locations, element neighbors, etc.,

(2) a posteriori error estimates to guide the analysis toward a systematic reduction in computational error,

(3) an optimization scheme that can choose optimal distributions of mesh sizes and spectral orders to yield desired accuracy with near minimal degrees of freedom, and

(4) new robust equation solvers that function on unstructured, changing meshes and with sometimes poorly conditioned matrices resulting from high-order approximations.

In the present study, a general *h-p* approach for the analysis of a general class of two- and three-dimensional second-order boundary-value problems with vector-valued solutions and arbitrary linear boundary conditions is described. In Part 1 of this investigation, reported here, we take up general formulation issues, development of a data structure, and the *h-p* adaptive strategy. In Part 2, a posteriori error estimation is discussed and in Part 3 we describe *h-p* mesh optimization techniques and the results of numerical experiments and applications to representative physical problems.

One feature of the techniques and results developed here is especially important: by appropriate sequences of distributions of h and p, exponential rates of convergence are observed. This fact confirms estimates developed by Babuška and his students for *h-p* methods for one-dimensional elliptic boundary-value problems (see e.g. [1, 2, 3]). This fact also suggests that if the basic issues listed earlier can be effectively resolved, *h-p* methods may have far-reaching applications to many classes of physical problems of practical importance. Some progress in this direction has already been made in the work of Devloo et al. [4] on adaptive *h-p* methods for viscous compressible flow problems. The work of Delves and Hall [5] on global element methods describes an approach similar to *h-p* methods, but does not discuss a posteriori error estimation or *h-p* adaptivity.

Following this introduction, we describe weak forms of a general class of linear systems of second-order partial differential equations with variable coefficients and general linear boundary conditions in two and three dimensions. This problem class includes problems in plane and three-dimensional elasticity, Stokesian flow, steady-state convective heat conduction, potential flow, and many more. By special techniques for introducing a time variable, discussed in Appendix A, transient linear advection, advection diffusion, and other time-dependent problems can be brought into this class.

In Section 3, the basic element strategy is discussed. Here 1-irregular meshes (a term defined in Section 2) and hierarchical shape functions are introduced and special element constraint methods are used to invoke continuity across interelement boundaries of elements of different size and with shape functions of differing polynomial degree. Our adaptive *h-p* methodology is discussed in Section 4 and some features of the special data structure are outlined in Section 5. Some numerical results are presented in Section 6.

2. *h*- and *p*-adaptivity. Regular and irregular meshes

A variational formulation

Let Ω be an open bounded domain in $\mathbb{R}^n$, $n = 2, 3$ with a sufficiently regular boundary $\partial\Omega$. In what follows, we shall restrict ourselves to a class of problems that can be formulated in the

following abstract form:

$$\text{Find } u \in X, \quad \text{such that} \quad B(u, v) = L(v) \quad \forall v \in X. \tag{2.1}$$

Here

$$X = X \times X \cdots \times X \ (m \text{ times}), \tag{2.2}$$

where X is a subspace of $H^1(\Omega)$, the Sobolev space of first order, $B(\cdot, \cdot)$ is a bilinear form on $X \times X$ of the following form:

$$B(u, v) = \sum_{i,j=1}^{m} B_{ij}(u_i, v_j), \tag{2.3}$$

where $B_{ij}(\cdot, \cdot)$ are bilinear forms of scalar-valued arguments of the type

$$B_{ij}(u, v) = \int_\Omega \left\{ \sum_{k,l=1}^{n} a_{ij}^{kl} \frac{\partial u}{\partial x_k} \frac{\partial v}{\partial x_l} + \sum_{k=1}^{n} b_{ij}^{k} \frac{\partial u}{\partial x_k} v + c_{ij} uv \right\} \mathrm{d}x + \int_{\partial\Omega} d_{ij} uv \, \mathrm{d}s \tag{2.4}$$

and

$$L(\cdot) \text{ is a linear form on } X \tag{2.5}$$

of the form

$$L(v) = \sum_{j=1}^{m} L_j(v_j) \tag{2.6}$$

with the linear forms $L_j(\cdot)$ acting on the scalar-valued functions

$$L_j(v) = \int_\Omega \left\{ f_j v + \sum_{k=1}^{n} g_j^k \frac{\partial v}{\partial x_k} \right\} \mathrm{d}x + \int_{\partial\Omega} h_j v \, \mathrm{d}s. \tag{2.7}$$

In the above formulas, a_{ij}^{kl}, b_{ij}^{k}, c_{ij}, f_i, g_j^k, d_{ij}, h_j are sufficiently regular functions defined on Ω and on $\partial\Omega$, respectively.

Numerous examples fall into the category of problems described by the abstract formulation (2.1). To mention a few: linear elliptic problems (both single equations and systems), linear problems resulting from a one-step approximation in time for evolution problems, linear steps of a nonlinear problem solution, etc. Some specific examples are given in Appendix A.

In this formulation, both the solution u and the test functions v are members of the same space X. Nonhomogeneous essential boundary conditions are handled by means of a standard penalty approach.

Finite element approximation

We assume that the domain Ω can be represented as a union of finite elements K_e, $e = 1, \ldots, M$. More precisely

$$\bar{\Omega} = \bigcup_{e=1}^{M} \bar{K}_e \tag{2.8}$$

and

$$\text{int } K_e \cap \text{int } K_f = \emptyset \quad \text{for } e \neq f \,. \tag{2.9}$$

Each of the elements K has a corresponding finite dimensional space of shape functions, denoted $X_h(K)$; for instance, the space of polynomials of order p. The global finite element space X_h consists of functions which, when restricted to element K, belong to the local space of shape functions $X_h(K)$. Thus the global approximation is constructed by patching together the local shape functions in the usual way.

We shall adopt the fundamental requirement that the global approximation must be continuous. As we will see, this requirement leads to the notion of constrained approximations. Formally, the continuity assumption guarantees that the finite element space X_h is a subspace of $H^1(\Omega)$ and, with some additional assumptions if necessary, also a subspace of X. The approximate problem is easily obtained from (2.1) by substituting for $\boldsymbol{u}$ and $\boldsymbol{v}$ their approximations $\boldsymbol{u}_h$ and $\boldsymbol{v}_h$:

$$\text{Find } \boldsymbol{u}_h \in \boldsymbol{X}_h \,, \quad \text{such that} \quad B_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = L_h(\boldsymbol{v}_h) \quad \forall \boldsymbol{v}_h \in \boldsymbol{X}_h \,. \tag{2.10}$$

Here

$$\boldsymbol{X}_h = X_h \times \cdots \times X_h \ (m \text{ times}) \,, \tag{2.11}$$

which indicates that the same approximation has been applied to every component of $\boldsymbol{u}$. $B_h(\cdot,\cdot)$ and $L_h(\cdot)$ denote approximations to the original bilinear and linear forms resulting from numerical integration.

Adaptivity

A flow chart of a typical Adaptive Finite Element Method (AFEM) is shown in Fig. 1. The method consists of first generating an initial mesh and solving for the corresponding FEM approximate solution. Next, the error is estimated in some way and based on this (usually crude) approximation, one adapts the mesh, i.e., adds new degrees of freedom. The approximate problem for the new mesh is solved again and the whole procedure continues until certain error tolerances are met. Obviously, such a procedure requires an estimate of the error over each element and a strategy to reduce the error by proper changes in the mesh parameters, h and p.

In the FEM the new degrees of freedom can be added in two different ways: elements may be locally refined or their spaces of shape functions may be enriched by incorporating new shape functions. As noted earlier, in the case of polynomials, this is done by increasing locally the degree of polynomials used to construct the shape functions, the first case being an h-refinement, and the second case a p-refinement. A combination of both is an (adaptive) h-p FEM. We remark that the process of increasing the local polynomial degrees for a fixed mesh size is mathematically akin to increasing the spectral order of the approximation and that, therefore, we also refer to h-p methods as 'adaptive spectral-element' or 'h-spectral' methods.

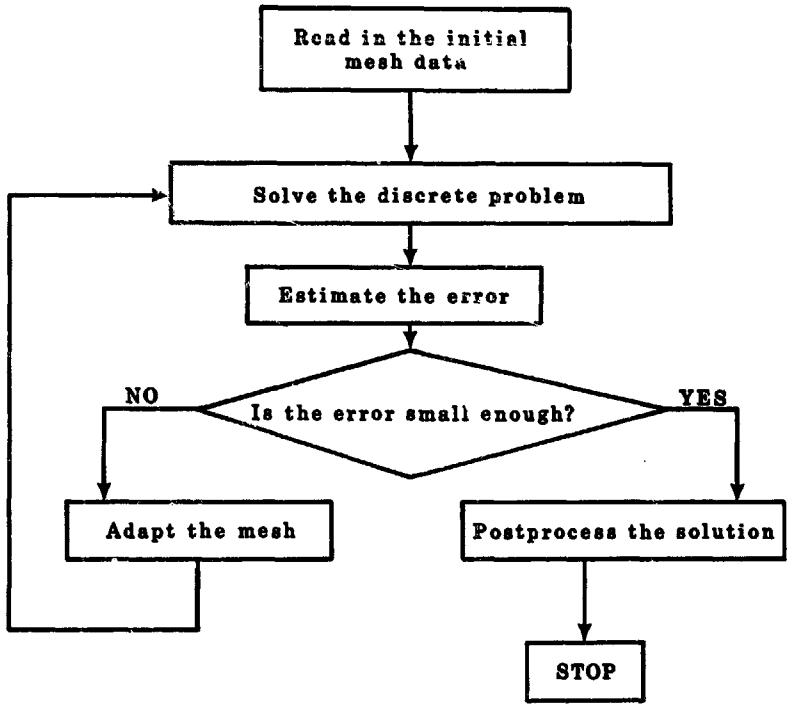

Fig. 1. Typical flow chart of an adaptive method.

Regular and irregular meshes

As the result of local h-refinements, irregular meshes are introduced. Recall (see [6]) that a node is called regular if it constitutes a vertex for each of the neighboring elements; otherwise it is irregular. If all nodes in a mesh are regular, then the mesh itself is said to be regular. In the context of two-dimensional meshes, the maximum number of irregular nodes on an element side is referred to as the index of irregularity. Meshes with an index of irregularity equal to one are called 1-irregular meshes. The notion can be easily generalized to the three-dimensional case. (See [7] and literature cited therein for additional references.)

In the present work, we accept only 1-irregular meshes. In the two-dimensional context, this translates into the requirement that a 'large' neighbor of an element may have no more than two 'small' neighbors on a side; in the three-dimensional case, the number of neighbors sharing a side may go up to four, while the number of neighbors sharing an edge can be no more than two. This is frequently called the '*two-to-one*' rule (cf. [7]). Examples of regular and irregular meshes are shown in Fig. 2. There are several practical and theoretical reasons to accept only 1-irregular meshes, especially in the context of *h-p* methods. For a detailed argument, we refer to [8].

Our restriction to 1-irregular meshes imposes a simple restriction on the way any h-refinement can proceed: before an element is refined, a check for 'larger' neighbors must be made. If any such neighbors exist, they must be refined first and only then can the element in question be refined.

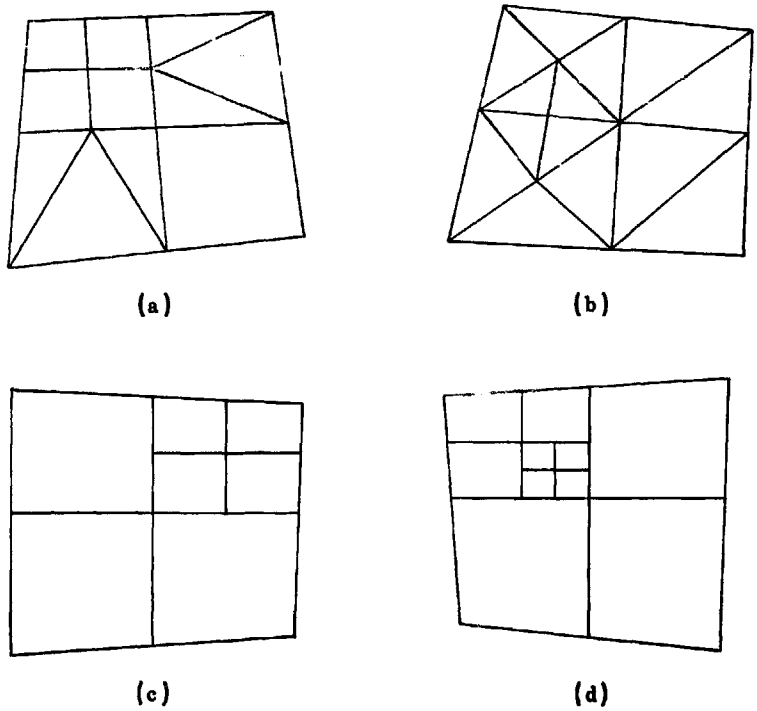

Fig. 2. Examples of regular and irregular meshes: (a) and (b)–regular mesh; (c) 1-regular mesh (index of irregularity = 1); (d) 2-irregular mesh.

Continuity and constrained approximation

The presence of irregular nodes makes the handling of the continuity assumption non-standard and leads to the notion of a constrained approximation. As an example, consider a mesh of three rectangular Q^2 elements (quadrilateral elements with biquadratic shape functions) with standard Lagrange degrees of freedom, as shown in Fig. 3. Clearly, a function

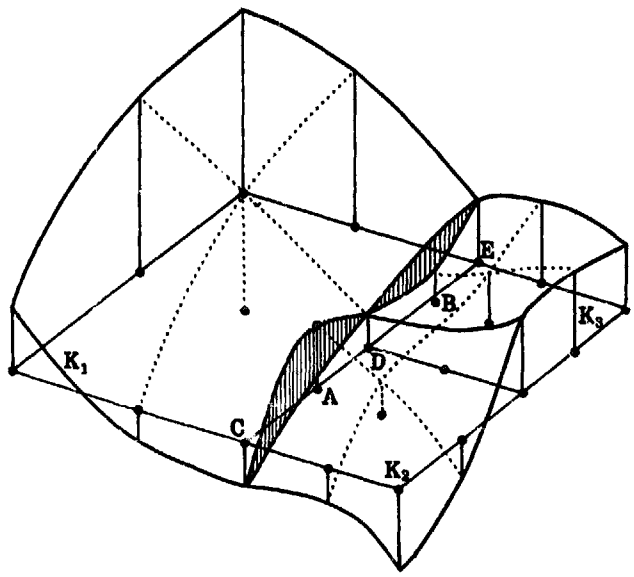

Fig. 3. Example of an unconstrained, discontinuous approximation.

u_h defined over these elements need not be continuous across element interfaces due to the presence of two irregular nodes, A and B. For instance, the values of u_h on side CDE of element K_1 are determined uniquely by (say) its values at points C, D and E, while those on side CD of element K_2 are defined by specifying degrees of freedom in K_1. In order to make u_h continuous, the value of u_h at A from the side of K_2 must be forced to be equal to the value of u_h at A from the side of K_1, which is equivalent to the elimination of the degree of freedom associated with point A by enforcing the constraint

$$u_h(A) = \alpha u_h(C) + \beta u_h(D) + \gamma u_h(E) \tag{2.12}$$

with proper coefficients α, β and γ. The situation is much more complicated in the case of different orders of approximation within each element, and we discuss it in detail in Section 3.

3. Constrained approximation

In this section, we develop the general concept of constrained approximations. We shall see in particular the impact of the constraints on such basic ingredients to the FEM as element stiffness matrix and load vector calculations.

Approximation on the element level

Let K be a finite element with the corresponding space of shape functions $X_h(K)$. The set Q_K of element degrees of freedom N_K, as usual, is viewed as a set of linear functionals defined on $X_h(K)$. We assume that the set Q_K of degrees of freedom

$$\{\varphi_{i,K} : X_h(K) \to \mathbb{R} \mid i \in N_K\} \tag{3.1}$$

is $X_h(K)$-unisolvent (see [9, 10]), i.e.,

$\varphi_{i,K}$, $i \in N_K$ are linearly independent, and

$\oplus_i \mathbb{R}\varphi_{i,K}$, $i \in N_K$ is a dual space of $X_h(K)$.

As indicated, we assign to each linear functional $\varphi_K \in Q_K$ an integer label i, $\varphi \to \varphi_{i,K}$ and denote by N_K the set of such labels for element K. The element shape functions $\chi_{j,K}$ are defined as a dual basis to $\varphi_{i,K}$, i.e.,

$$\langle \varphi_{i,K}, \chi_{j,K} \rangle = \delta_{ij}, \quad i, j \in N_K, \tag{3.2}$$

and the finite element approximation u_h within the element K is of the form

$$u_h = \sum_{i \in N_K} u_h^i \chi_{i,K},$$

where $u_h^i = \varphi_{i,K}(u_h)$.

In what follows, we restrict ourselves to Lagrange- and Hermite-types of degrees of freedom. In other words, we assume that each of the functionals φ_i is of the form

$$\varphi : u \to D_x^k u(\xi_1, \ldots, \xi_k), \quad k = 0, 1, \ldots \tag{3.3}$$

where $D_x^k u$ denotes the kth order differential of u evaluated at point x (as usual, $D_x^0 u = u(x)$). Vectors $\xi_1, \ldots, \xi_k$ at this point denote arbitrary vectors in $\mathbb{R}^n$, $n = 2, 3$. Thus, every degree of freedom can be identified with a point x and k vectors $\xi_1, \ldots, \xi_k$. (Later on, in the context of subparametric finite elements, we shall accept functionals which are linear combinations of functionals of the form (3.3).)

Construction of the unconstrained finite element space $\tilde{X}_h$

We introduce the following formal definition of the unconstrained finite element space $\tilde{X}_h$. A function $u_h : \Omega \to \mathbb{R}$ belongs to $\tilde{X}_h$ if and only if the following two conditions are satisfied:

(1) The restriction of u_h to an element K belongs to the local space of shape functions, i.e.,

$$u_h|_K \in X_h(K) \quad \forall K . \tag{3.4}$$

(2) For every pair of elements K_e and K_f and corresponding pair of degrees of freedom

$$\varphi_{K_e} : X_h(K_e) \to \mathbb{R}, \qquad \varphi_{K_f} : X_h(K_f) \to \mathbb{R},$$

such that φ_{K_e} and φ_{K_f} are defined by (3.3) through the same, common point x and vectors $\xi_1, \ldots, \xi_k$, the two degrees of freedom take on the same value on u_h when restricted to K_e or K_f respectively, i.e.,

$$\varphi_{K_e}(u_h|_{K_e}) = \varphi_{K_f}(u_h|_{K_f}) . \tag{3.5}$$

Note that the elements of $\tilde{X}_h$ need not be continuous (see Fig. 3).

Global degrees of freedom

Due to the construction of the space $\tilde{X}_h$, we can introduce the global degrees of freedom identified by points (nodes) x and vectors $\xi_1, \ldots, \xi_k$. We define the linear functional Φ on $\tilde{X}_h$,

$$\Phi : \tilde{X}_h \to \mathbb{R}, \quad \Phi(u_h) = \varphi_K(u_h|_K), \tag{3.6}$$

where K is an element with the corresponding degree of freedom identified with point x and vectors $\xi_1, \ldots, \xi_k$. Note that due to the definition of $\tilde{X}_h$, the global degrees of freedom are well-defined.

The unconstrained base functions

The unconstrained base functions $\tilde{e}_i$ are introduced as a dual basis to the space of the global degrees of freedom, i.e.,

$$\langle \Phi_j, \tilde{e}_i \rangle = \delta_{ij} . \tag{3.7}$$

Note that $\tilde{e}_i$ may be discontinuous.

Construction of the constrained finite element space X_h

At this point, somewhat arbitrarily, we divide all global degrees of freedom into two subsets: active and constrained. We use the following notation:

—N^a is the set of indices assigned to active degrees of freedom and
—N^c is the set of indices assigned to constrained degrees of freedom.

By an 'active' degree of freedom, we mean one of a set of linearly independent functionals that define the parameters associated with a global stiffness matrix for the problem at hand; 'constrained' degrees of freedom are linear combinations of active degrees of freedom defined by constraints associated with element connectivity.

We assume that for each constrained degree of freedom Φ_i, $i \in N^c$, there exists a set $I(i)$ of corresponding active degrees of freedom, $I(i) \subset N^a$, and a vector R_{ij}, $j \in I(i)$, such that the following equality holds:

$$\Phi_i(u_h) = \sum_{j \in I(i)} R_{ij} \Phi_j(u_h) . \tag{3.8}$$

We now introduce the constrained finite element space X_h as

$$X_h = \left\{ u_h \in \tilde{X}_h \,|\, \Phi_i(u_h) = \sum_{j \in I(i)} R_{ij} \Phi_j(u_h) \;\; \forall i \in N^c \right\} . \tag{3.9}$$

Assuming that the constraints are linearly independent, we see that X_h is dual to the space spanned by only active degrees of freedom. As usual, we define the base functions e_j, $j \in N^a$, as a dual basis to the set of active degrees of freedom:

$$e_j \in X_h , \quad \langle \Phi_i, e_j \rangle = \delta_{ij} , \quad i, j \in N^a . \tag{3.10}$$

Though, at this point, the choice of constrained degrees of freedom is arbitrary, we implicitly assume that with the proper choice of constraints the resulting finite element space X_h consists of only continuous functions. This will involve a proper choice of the array R_{ij}.

Relation between unconstrained and constrained base functions

Let u_h be an arbitrary function belonging to X_h. Then u_h must be of the following form:

$$u_h = \sum_{i \in N^a} u_i \tilde{e}_i + \sum_{j \in N^c} u_j \tilde{e}_j = \sum_{i \in N^a} u_i \tilde{e}_i + \sum_{j \in N^c} \sum_{k \in I(j)} R_{jk} u_k \tilde{e}_j . \tag{3.11}$$

Introducing for every $i \in N^a$ the set

$$S(i) = \{ j \in N^c \,|\, i \in I(j) \} , \tag{3.12}$$

we rewrite u_h in the form

$$u_h = \sum_{i \in N^a} u_i \tilde{e}_i + \sum_{k \in N^a} \sum_{j \in S(k)} u_h R_{jk} \tilde{e}_j = \sum_{i \in N^a} u_i \Big(\tilde{e}_i + \sum_{j \in S(i)} R_{ji} \tilde{e}_j \Big) . \tag{3.13}$$

We claim that functions

$$e_i = \tilde{e}_i + \sum_{j \in S(i)} R_{ji} \tilde{e}_j , \quad i \in N^a , \tag{3.14}$$

form the dual basis to functionals Φ_i, $i \in N^a$. Indeed

$$\langle \Phi_j, e_i \rangle = \langle \Phi_j, \tilde{e}_i \rangle + \sum_{k \in S(i)} R_{ki} \langle \Phi_j, \tilde{e}_k \rangle = \delta_{ij} , \tag{3.15}$$

since $S(i) \subset N^c$.

Calculation of the global load vector and stiffness matrix

For simplicity, we restrict ourselves to the case of a single equation. In the case of systems, the same procedure is applied for every linear form (2.7) and bilinear form (2.4).

Substituting (3.14) into both sides of (2.10), we obtain formulas for the load vector and stiffness matrix:

$$L_h(e_i) = L_h(\tilde{e}_i) + \sum_{k \in S(i)} R_{ki} L_h(\tilde{e}_k) , \tag{3.16}$$

$$B_h(e_i, e_j) = B_h(\tilde{e}_i, \tilde{e}_j) + \sum_{k \in S(i)} R_{ki} B_h(\tilde{e}_k, \tilde{e}_j) + \sum_{l \in S(j)} R_{li} B_h(\tilde{e}_i, \tilde{e}_l) + \sum_{k \in S(i)} \sum_{l \in S(j)} R_{ki} R_{lj} B_h(\tilde{e}_k, \tilde{e}_l) . \tag{3.17}$$

Element level revisited—modified element stiffness matrix and load vector

Consider an element K. Let $N^a(K)$ and $N^c(K)$ denote sets of indices corresponding to active and constrained degrees of freedom for the element. Assuming that, as usual, the load vector and stiffness matrix are calculated by summing up the contributions of all elements, i.e.,

$$L_h(u_h) = \sum_K L_{h,K}(u_h|_K) \tag{3.18}$$

and

$$B_h(u_h, v_h) = \sum_K B_{h,K}(u_h|_K, v_h|_K) , \tag{3.19}$$

we arrive at the practical issue of how to calculate the contributions of element K to the global load vector and stiffness matrix. Toward resolving this question, we introduce

—the usual element load vector

$$\tilde{b}_{i,K} = L_{h,K}(\chi_{i,K}) , \quad i \in N^a(K) \cup N^c(K) , \tag{3.20}$$

—the element stiffness matrix

$$\tilde{B}_{ij,K} = B_{h,K}(\chi_{i,K}, \chi_{j,K}), \quad i, j \in N^a(K) \cup N^c(K), \tag{3.21}$$

—the set of associated active degrees of freedom for element K

$$N(K) = N^a(K) \cup \bigcup_{j \in N^c(K)} I(j), \tag{3.22}$$

(notice that the two sets on the right-hand side of (3.22) need not be disjoint!)

—the element contribution to the global load vector (modified element load vector)

$$b_{i,K} = L_{h,K}(e_i|_K), \quad i \in N(K), \tag{3.23}$$

—the element contribution to the global stiffness matrix (modified element stiffness matrix)

$$B_{ij,K} = B_{h,K}(e_i|_K, e_j|_K), \quad i, j \in N(K). \tag{3.24}$$

Algorithms for the calculation of $b_{i,K}$ and $B_{ij,K}$

The load vector and stiffness matrix for the constrained degrees of freedom are now computed by transformations embodied in the following sample algorithms:

```
FOR i ∈ N(K)
  b_{i,K} = 0
ENDFOR
FOR i ∈ N^a(K) ∪ N^c(K)
  IF i ∈ N^a(K) THEN
  b_{i,K} = b_{i,K} + b̃_{i,K}
  ENDIF
  IF i ∈ N^c(K) THEN
    FOR k ∈ I(i)
    b_{k,K} = b_{k,K} + R_{ik} b̃_{i,K}
    ENDFOR
  ENDIF
ENDFOR
```

The $B_{ij,K}$ algorithm:

```
FOR i, j ∈ N(K)
  B_{ij,K} = 0
ENDFOR
FOR i, j ∈ N^a(K) ∪ N^c(K)
  IF i ∈ N^a(K) AND j ∈ N^a(K) THEN
    B_{ij,K} = B_{ij,K} + B̃_{ij,K}
  ENDIF
```

```
  IF i ∈ N^c(K) AND j ∈ N^a(K) THEN
    FOR k ∈ I(i)
    B_kj,K = B_kj,K + R_ik B̃_kj,K
    ENDFOR
  ENDIF
  IF i ∈ N^a(K) AND j ∈ N^c(K) THEN
    FOR l ∈ I(j)
    B_il,K = B_il,K + R_jl B̃_il,K
    ENDFOR
  ENDIF
  IF i ∈ N^c(K) AND j ∈ N^c(K) THEN
    FOR k ∈ I(i), l ∈ I(j)
    B_kl,K = B_kl,K + R_ik R_jl B̃_kl,K
    ENDFOR
  ENDIF
ENDFOR
```

4. An *h-p* adaptive finite element method

This section is devoted to a presentation of an *h-p* adaptive finite element method which provides for instantaneous h- and p-refinements and unrefinements. For simplicity, the discussion is restricted to the two-dimensional case.

We shall adopt the following assumptions:

—the initial mesh is (topologically) a portion of a regular, rectangular grid in $\mathbb{R}^2$,
—only 1-irregular meshes are accepted for all h-refinements,
—the local orders of approximation may differ in each element,
—the approximation must be continuous.

One-dimensional hierarchical element of an arbitrary order p

We begin our discussion with the definition of the standard one-dimensional hierarchical master element defined on the interval $[-1, 1]$.

The element degrees of freedom are identified with three nodes: two endpoints and the midpoint $x = 0$. If the element is of the first order, only two degrees of freedom are present—these are the nodal values at $x = -1$ and $x = 1$. Starting with $p = 2$, new degrees of freedom are added which are of the form

$$\varphi_p(u) = \frac{1}{\lambda_p} \frac{\mathrm{d}^p u}{\mathrm{d}x^p}(0), \tag{4.1}$$

where λ_p is a scaling factor.

The corresponding shape functions are illustrated in Fig. 4.

Definition of the hierarchical square Q^p master element

Setting $\hat{K} = [-1, 1] \times [-1, 1]$, we define the space of shape functions as

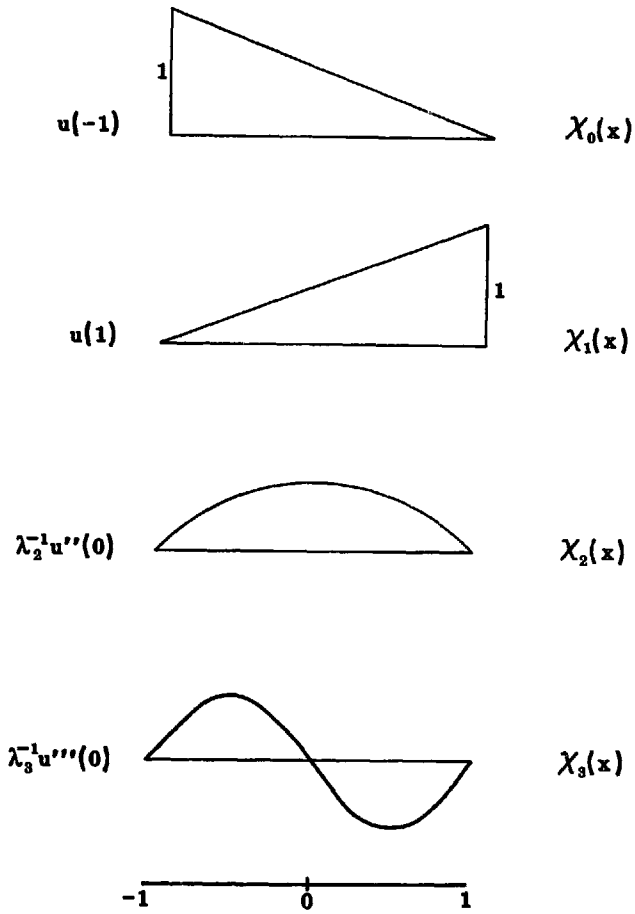

Fig. 4. One-dimensional hierarchical master element. Degrees of freedom and corresponding shape functions.

$$X_h(\hat{K}) = Q^p(\hat{K}), \tag{4.2}$$

where Q^p denotes the space of polynomials up to pth order with respect to each of the variables separately.

The degrees of freedom are defined as follows:

—function values at four vertices:

$$u(-1,-1), \quad u(1,-1), \quad u(1,1), \quad u(-1,1), \tag{4.3}$$

—tangential derivatives (up to a multiplicative constant) up to pth order associated with the midpoints of the four edges:

$$\begin{aligned} &\lambda_k^{-1}\frac{\partial^k u}{\partial x^k}(0,-1), \quad k=2,\ldots,p, \qquad \lambda_k^{-1}\frac{\partial^k u}{\partial y^k}(1,0), \quad k=2,\ldots,p, \\ &\lambda_k^{-1}\frac{\partial^k u}{\partial x^k}(0,1), \quad k=2,\ldots,p, \qquad \lambda_k^{-1}\frac{\partial^k u}{\partial y^k}(-1,0), \quad k=2,\ldots,p, \end{aligned} \tag{4.4}$$

—mixed order derivatives associated with the central node

$$\lambda_k^{-1}\lambda_l^{-1}\frac{\partial^{k+1}u}{\partial x^k\,\partial y^l}(0,0)\,,\quad k,l=2,\ldots,p\,. \tag{4.5}$$

One can easily see that the space of the shape functions $Q^p(\hat{K})$ is a tensor product of $P^p(-1,1)$ with itself and the degrees of freedom just introduced are simply the tensor products of the degrees of freedom for the one-dimensional element. More precisely, if $u\in Q^p$, then u is of the form

$$u(x,y)=\sum_k u^k v_k(x)w_k(y)\,, \tag{4.6}$$

where $v_k, w_k\in P^p(-1,1)$, and each of the degrees of freedom can be represented in the form

$$(\varphi_i\otimes\varphi_j)(u)=\sum_k u^k(\varphi_i\otimes\varphi_j)(v_k\otimes w_k)\,. \tag{4.7}$$

This in particular implies that the corresponding shape functions can be identified with the tensor products of one-dimensional shape functions, which are of the form

$$\chi_i(x)\chi_j(y)\,,\quad i,j=0,1,\ldots,p\,. \tag{4.8}$$

For $i,j=0,1$, we have the usual bilinear element with four nodal degrees of freedom.

Master element of an enriched order

Let $\hat{K}$ be an element of pth order. By adding additional shape functions corresponding to the $(p+1)$th order, together with the corresponding degrees of freedom, we obtain a well-defined finite element whose space of shape functions includes Q^p. In particular, by adding all the additional degrees of freedom corresponding to the $(p+1)$th order, we pass to the element Q^{p+1} without modifying the existing shape functions and degrees of freedom of Q^p. Equivalently, we can begin an adaptive process with element Q^{p+1} and eliminate some degrees of freedom to pass to an incomplete element of a lower order.

Subparametric hierarchical elements

Consider the master element of (possibly) an incomplete order p. Even though p can be arbitrarily large, the element may be only Q^1-complete, which means that some of the nodes may be missing. An example of such an element is presented in Fig. 5. An arbitrary location of the seven nodes in the plane (x,y) determines uniquely a map T from the master element into $\mathbb{R}^2$, the components of T belonging to the incomplete Q^2 space. More precisely, if ψ_i, $i=1,\ldots,9$, are the regular shape functions for the nine-node biquadratic element, then

$$T(\hat{x},\hat{y})=\sum_{i=1}^{9} a_i\psi_i(\hat{x},\hat{y})\,, \tag{4.9}$$

with the assumption that $a_6=\frac{1}{2}(a_2+a_3)$ and $a_7=\frac{1}{2}(a_3+a_4)$.

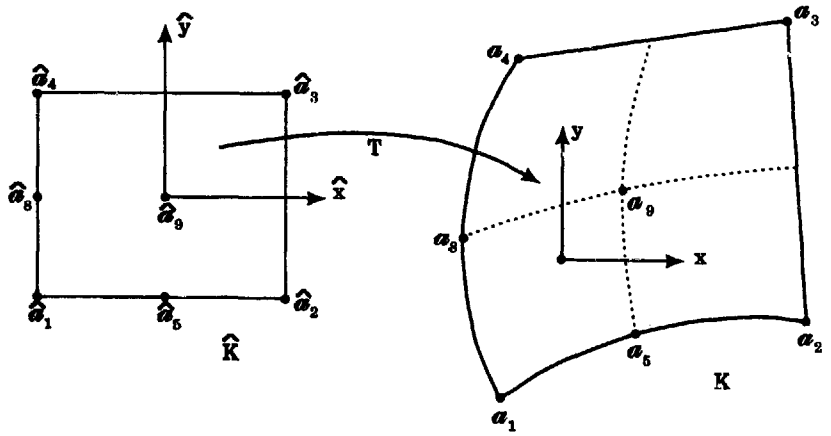

Fig. 5. Concept of the subparametric element.

We have the classical definition of the subparametric element,

$$K = T(\hat{K}), \tag{4.10}$$

with the space of test functions defined as

$$X_h(K) = \{u = \hat{u} \circ T^{-1} \mid \hat{u} \in X_h(\hat{K})\} \tag{4.11}$$

and the degrees of freedom

$$\langle \varphi, u \rangle = \langle \hat{\varphi}, \hat{u} \rangle, \text{ where } u = \hat{u} \circ T^{-1}. \tag{4.12}$$

Interpretation of the degrees of freedom

The degrees of freedom associated with vertices are simply the function values evaluated at these points. The degrees of freedom associated with the midpoints of element edges and central nodes are more complicated. It follows from (4.12) that they may be interpreted as certain linear combinations of directional derivatives. The form of directional or mixed derivatives appropriate for the master element is preserved only if the map T is linear. Thus, it is understood that the degrees of freedom discussed in the previous section should be interpreted broadly enough to include linear combinations of these Hermite-type degrees of freedom.

Continuity for regular meshes

One of the fundamental advantages of using the hierarchical shape functions is the ease with which they allow one to construct a continuous approximation with locally variable order of approximation. A typical situation is illustrated in Fig. 6. If elements K_1 and K_2 are to support polynomials of degree, say, one and three, respectively, then there are at least two ways to enforce continuity across the interelement boundary. One way is to add two extra

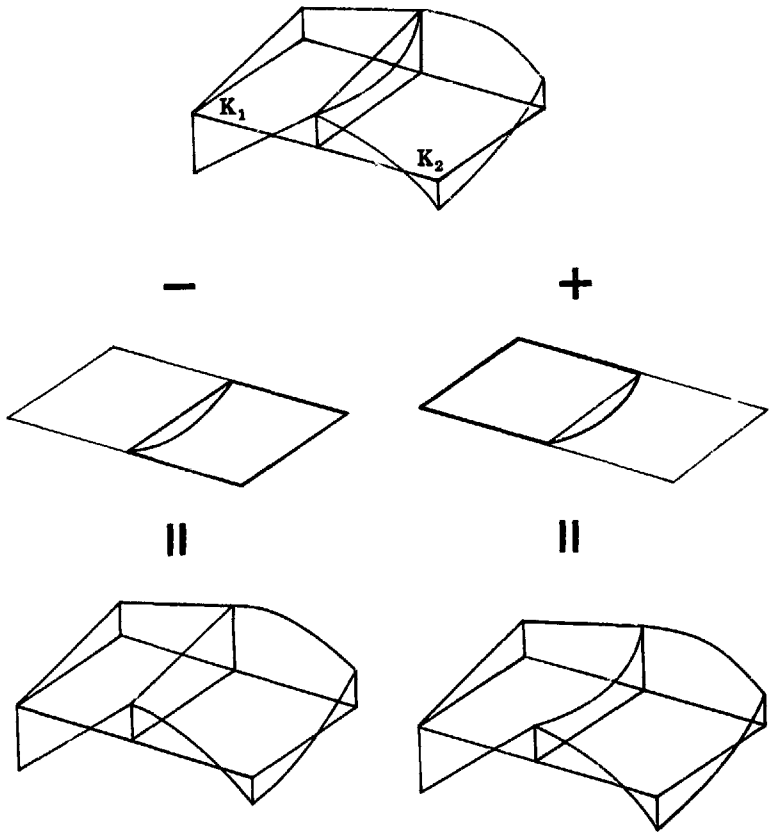

Fig. 6. Continuity by hierarchical shape functions.

shape functions of second and third order corresponding to the middle node A of element K_1. Alternatively, the same two shape functions may be deleted from element K_2. One can, of course, establish a 'golden rule' method by adding one extra shape function of second order to K_1, and deleting the third order shape function from K_2. In all these cases, a common order of approximation along the interelement boundary can be enforced by simply adding or deleting the respective shape functions from the neighboring elements. While any of these choices can be made, the results described here employ the 'maximum rule' in which the higher-order approximation dominates lower orders. Thus, if an element is p-refined, i.e., a higher order approximation of degree $\bar{p}$ is introduced, the neighbors of lower order are enriched by the addition of extra shape functions of degree $\bar{p}$ necessary to guarantee continuity of the approximation.

Constraints in the one-dimensional case

In the case of irregular meshes, continuity has to be enforced by means of the constrained approximation. To fix ideas, consider the generic, one-dimensional case shown in Fig. 7. The approximation on the small elements $[-1, 0]$ and $[0, 1]$ must match the approximation on the large element $[-1, 1]$.

We first choose the scaling factors λ_p in (4.1) in such a way that the corresponding shape functions for the one-dimensional master element have the following form

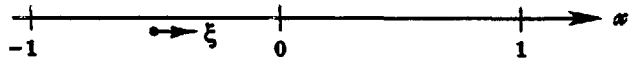

Fig. 7. Derivation of the constraint coefficients.

$$\chi_0 = \tfrac{1}{2}(1-\xi)\,, \qquad \chi_1 = \tfrac{1}{2}(1+\xi)\,, \qquad \chi_p(\xi) = \begin{cases} \xi^p - 1\,, & p = 2,4,6,\ldots \\ \xi^p - \xi\,, & p = 3,5,7,\ldots \end{cases} \tag{4.13}$$

Assume next that all degrees of freedom for the large element are active. The question is: what degrees of freedom must the small elements take on in order that the functions supported on the two small elements coincide exactly with shape functions of the large element?

From the fact that (4.1) is a dual basis to (4.13), we get

$$\varphi_p(\chi_p) = \frac{1}{\lambda_p}\, p! = 1\,, \quad p = 2,3,\ldots \tag{4.14}$$

and therefore $\lambda_p = p!$.

The transformation map from $[-1,1]$ onto $[-1,0]$ is of the form

$$x = -\tfrac{1}{2} + \tfrac{1}{2}\xi\,, \tag{4.15}$$

with inverse $\xi = 2x+1$. This yields the following formulas for the shape functions ${}^1\chi_p$, $p = 0,1,2,\ldots$ For the (left-hand side) element $[-1,0]$ (recall (4.11)):

$$\begin{aligned} &{}^1\chi_0(x) = -x\,, \\ &{}^1\chi_1(x) = x+1\,, \\ &{}^1\chi_p(x) = (2x+1)^p - 1\,, \quad p = 2,4,6,\ldots\,, \\ &{}^1\chi_p(x) = (2x+1)^p - (2x+1)\,, \quad p = 3,5,7,\ldots\,, \end{aligned} \tag{4.16}$$

and the corresponding formulas for the degrees of freedom are (recall (4.12))

$$\begin{aligned} &\langle {}^1\varphi_0, u\rangle = u(-1)\,, \qquad \langle {}^1\varphi_1, u\rangle = u(0)\,, \\ &\langle {}^1\varphi_p, u\rangle = \frac{1}{2^p p!}\,\frac{\mathrm{d}^p u}{\mathrm{d}x^p}\left(-\frac{1}{2}\right), \quad p = 2,3,\ldots \end{aligned} \tag{4.17}$$

Now let $u(x)$ ($x = \xi$ for the master element) be any function defined by the shape functions on $[-1,1]$, i.e.,

$$u(x) = \sum_{q=0}^{k} \varphi_q(u)\chi_q(x)\,. \tag{4.18}$$

In order to represent $u(x)$ for $x \in [-1,0]$ in terms of the shape functions on $[-1,0]$, we have to calculate the values of the degrees of freedom (4.17). We get

$$\langle {}^1\varphi_0, u\rangle = \varphi_0(u)\langle {}^1\varphi_0, \chi_0\rangle + \sum_{q=1}^{k} \varphi_q(u)\langle {}^1\varphi_0, \chi_q\rangle = \langle \varphi_0, u\rangle ,$$

$$\langle {}^1\varphi_1, u\rangle = \varphi_0(u)\langle {}^1\varphi_1, \chi_0\rangle + \varphi_1(u)\langle {}^1\varphi_1, \chi_1\rangle + \sum_{q=2}^{k} \varphi_q(u)\langle {}^1\varphi_1, \chi_q\rangle \tag{4.19}$$

$$= \tfrac{1}{2}\langle \varphi_1, u\rangle + \sum_{q=2}^{k} {}^1R_{q1}\langle \varphi_q, u\rangle ,$$

where

$${}^1R_{q1} = \langle {}^1\varphi_1, \chi_q\rangle = \begin{cases} -1 & \text{if } q \text{ is even}, \\ 0 & \text{otherwise}. \end{cases} \tag{4.20}$$

For $p \geq 2$,

$$\langle {}^1\varphi_p, u\rangle = \varphi_0(u)\langle {}^1\varphi_p, \chi_0\rangle + \sum_{q=1}^{k} \varphi_q(u)\langle {}^1\varphi_p, \chi_q\rangle = 0 + \sum_{q=1}^{k} {}^1R_{qp}\langle \varphi_q, u\rangle ,$$

where

$${}^1R_{qp} = \langle {}^1\varphi_p, \chi_q\rangle = \begin{cases} 0, & \text{for } q < p, \\ \dfrac{1}{2^q}(-1)^{p+q}\binom{q}{p} = \dfrac{(-1)^{p+q}}{2^q}\dfrac{q!}{p!(q-p)!} & \text{for } q \geq p. \end{cases} \tag{4.21}$$

The same procedure applied to the right-hand side element $[0,1]$ yields the following.

The transformation from $[-1,1]$ *onto* $[0,1]$:

$$x = \tfrac{1}{2} + \tfrac{1}{2}\xi , \tag{4.22}$$

with inverse $\xi = 2x - 1$.

The shape functions ${}^r\chi_p$, $p = 0, 1, 2, \ldots$:

$${}^r\chi_0(x) = 1 - x , \tag{4.23}$$

$${}^r\chi_1(x) = x , \tag{4.24}$$

$${}^r\chi_p(x) = \begin{cases} (2x-1)^p - 1 & p \text{ even}, \\ (2x-1)^p - (2x-1) & p \text{ odd}. \end{cases} \tag{4.25}$$

The degrees of freedom ${}^r\varphi_p$:

$$\langle {}^{r}\varphi_0, u\rangle = u(0)\,,$$
$$\langle {}^{r}\varphi_1, u\rangle = u(1)\,, \tag{4.26}$$
$$\langle {}^{r}\varphi_p, u\rangle = \frac{1}{2^p p!}\frac{\mathrm{d}^p u}{\mathrm{d}x^p}\left(\frac{1}{2}\right).$$

The constraints

$$\langle {}^{r}\varphi_0, u\rangle = \tfrac{1}{2}\langle \varphi_0, u\rangle + \tfrac{1}{2}\langle \varphi_1, u\rangle + \sum_{q=2}^{k} {}^{r}R_{q0}\langle E\varphi_q, u\rangle\,, \tag{4.27}$$

where

$${}^{r}R_{q0} = \begin{cases} -1 & \text{if } q \text{ is even}\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{4.28}$$

$$\langle {}^{r}\varphi_1, u\rangle = \langle \varphi_1, u\rangle\,, \tag{4.29}$$

and for $p \geq 2$

$$\langle {}^{r}\varphi_p, u\rangle = \sum_{q=2}^{k_r} R_{qp}\langle \varphi_q, u\rangle\,, \tag{4.30}$$

$${}^{l}R_{qp} = \begin{bmatrix} 1 & 1/2 & & & & & \\ & 1/2 & & & & & \\ & -1 & 1/4 & & & & \\ & 0 & -3/8 & 1/8 & & & \\ & -1 & 6/16 & -4/16 & 1/16 & & \\ & 0 & -10/32 & 10/32 & -5/32 & 1/32 & \\ & -1 & 15/64 & -20/64 & 15/64 & -6/64 & 1/64 \end{bmatrix}$$

$${}^{r}R_{qp} = \begin{bmatrix} 1/2 & & & & & \\ 1/2 & 1 & & & & \\ -1 & 1/4 & & & & \\ 0 & 3/8 & 1/8 & & & \\ -1 & 6/16 & 4/16 & 1/16 & & \\ 0 & 10/32 & 10/32 & 5/32 & 1/32 & \\ -1 & 15/64 & 20/64 & 15/64 & 6/64 & 1/64 \end{bmatrix}$$

Fig. 8. The constraint coefficients for a sixth order approximation. The unfilled coefficients are zero.

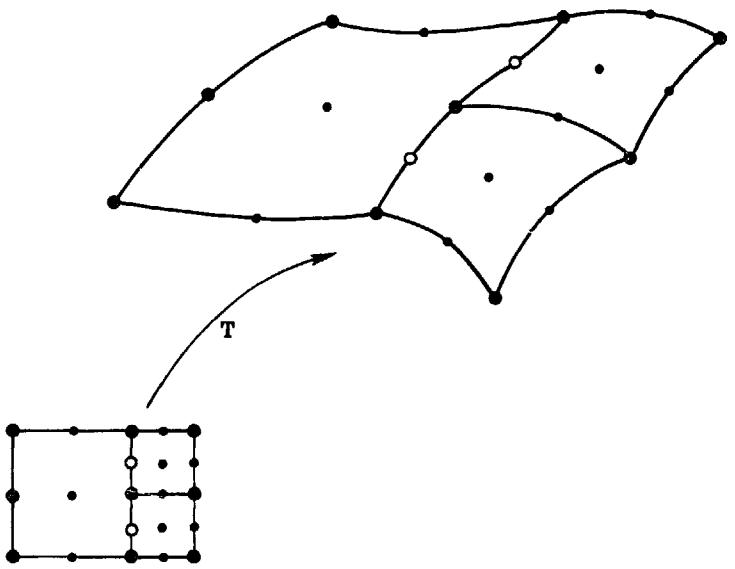

Fig. 9. Illustration of the constraints for the subparametric elements.

where

$$ {}^{r}R_{qp} = \langle {}^{r}\varphi_p, \chi_q \rangle \tag{4.31} $$

$$ = \begin{cases} 0 & \text{for } q < p\,, \\ \dfrac{1}{2^q}\binom{q}{p} & \text{for } q \geqslant p\,. \end{cases} \tag{4.32} $$

Arrays ${}^{l}R_{qp}$ and ${}^{r}R_{qp}$, $q, p = 0, \ldots, 5$ are presented in Fig. 8.

Constraints for 2-D subparametric elements

Since the shape functions for the 2-D master element are defined as tensor products of the 1-D functions, the results for the 1-D case hold exactly in the same form in the 2-D situation, the only difference being that the calculated constraint equations have to be applied to the proper degrees of freedom (see Fig. 9). It follows from the definition of the subparametric elements that the constraints coefficients are exactly the same, even when the elements have curved boundaries. This follows from the fact that the shape functions' behavior in a subparametric element on a part of its boundary depends exclusively upon the deformation of the part of the boundary, and therefore, any relation defined for the shape functions in the generic situation carries over immediately to the case of two small elements sharing an edge with a large element, so long as the deformation of the edge is identical in all three elements. The situation is illustrated in Fig. 9.

5. Some details concerning the data structure

In the classical FEM, elements as well as nodes are usually numbered consecutively in an attempt to produce a minimal band within the global stiffness matrix. When the program

identifies an element to process its contribution to the global matrices, the minimal information needed is the node numbers associated with the element. Adaptive refinement and unrefinement algorithms require much more information on the mesh structure than the classical assembly process.

First of all, we introduce the notion of a family. Whenever an element is refined a new family is created. The original element is called the father of the family and the four new elements are called its sons. Graphically, the geneology on families can be presented in a family tree structure as illustrated in Fig. 10.

An examination of refinement and unrefinement algorithms (see [7] for details) reveals that for a given element NEL, one must have access to the following information:

—the element node numbers,
—the element neighbors,
—the tree structure information, including
  —the number of the element family,
  —the number of the father,
  —the numbers of sons,
  —the refinement level (number of generation).

For a given NODE we also require,

—the node coordinates,
—the values of the degrees of freedom associated with the node.

In general, some information is stored explicitly in a database consisting of a number of arrays, some other information is recovered from the database by means of simple algorithms. A careful balance should be maintained between the amount of information stored (storage requirements) and recovered (time).

The following is a short list of arrays used in the database:

(1) The tree structure is stored in the condensed, family-like fashion [7], [8] in two arrays

NSON(NRELEI), NTREE(5,MAXNRFAM)

where NRELEI is the number of elements in the initial mesh and MAXNRFAM is the anticipated maximum number of families. For an element NEL of the initial mesh, NSON-

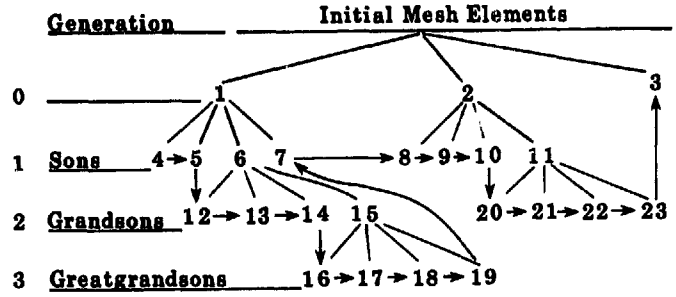

Fig. 10. A tree structure and the natural order of elements: 4, 5, 12, 13, 14, 16, 17, 18, 19, 7, 8, 9, 10, 20, 21, 22, 23, 3.

(NEL) contains its first son number (if there is any). For a family NFAM, NTREE(1,NFAM) contains the number of the father of the family while the other four entries NTREE(2:5,NFAM) are reserved for the 'first-born' sons of the sons of the family (the first-born 'grandsons' of the father).

(2) The initial mesh neighbor information is stored explicitly in array

NEIG(4,NRELEI),

containing up to four neighbors for each element of the initial mesh (elements adjacent to the boundary may have less neighbors).

(3) For every active element, up to nine nicknames are stored in array

NODES(9,MAXNRELEM),

where MAXNRELEM is the anticipated maximum number of elements. For a regular node, the nickname is defined as

NODE * 100 + NORDER,

where NODE is the node number and NORDER the order of approximation associated with the node. For an irregular node, the nickname is defined as

−NORDER,

where NORDER is again the order of approximation corresponding to the node.

(4) For a particular component IEL of a vector-valued solution, the corresponding degrees of freedom are stored sequentially in array

U(MAXNRDOF,IEL)

where MAXNRDOF is the anticipated maximal number of degrees of freedom. Two extra integer arrays are introduced to handle the information stored in array U. Array

NADRES(MAXNRNODE)

contains for every node, NODE, the address of the first from the degrees of freedom corresponding to NODE in array *U*. If K = NADRES(NODE) is such an address, the address for the next degree of freedom can be found in

NU(K)

and so on, until NU(K) = 0, which means that the last degree of freedom for a node has been found (see [7] for a detailed discussion). The parameter MAXNRNODE above is the anticipated maximal number of nodes.

(5) The node coordinates are stored in array

XNODE(2,MAXNRNODE).

The rest of the necessary information is reconstructed from the data structure by means of simple algorithms. These include

—calculation of up to eight neighbors for an element,
—calculation of local coordinates of nine nodes for an element determining its geometry (the irregular nodes coordinates have to be reconstructed by interpolating regular nodes coordinates),
—recovery of the tree-structure related information, e.g., level of refinement, the sons numbers, etc.,
—an algorithm establishing the natural order of elements.

During the *h*- and *p*-refinements and unrefinements, both elements and nodes are created and deleted in a rather random way. This makes it impossible to denumerate them in a consecutive way, according to their numbers (for instance, as a result of unrefinements some numbers may be simply missing!). Thus a new ordering of elements has to be introduced which is based on some scheme other than an element number criterion. In the code discussed here, we use 'the natural order of elements' based on the initial mesh elements ordering and the tree structure. The concept is illustrated in Fig. 10. One has to basically follow the tree of elements obeying the order of elements in the initial mesh and the order of sons in a family.

The natural order of elements may serve as a basis for defining an order for nodes and, consequently, for degrees of freedom, when necessary.

For a detailed discussion of the data structure as well as a critical review of different data structures in context of different *h*-refinement techniques, we refer again to [7].

6. Numerical examples

We conclude Part 1 of this work with a presentation of results obtained for selected representative examples. Details on forms of the corresponding variational formulations can be found in Appendix A.

Example 1. Poisson's equation on an L*-shaped domain*

The following boundary value problem was solved in the L-shaped domain shown in Fig. 11.

$$\begin{aligned} &-\Delta u = f \quad \text{in } \Omega, \\ &u = 0 \quad \text{on } \Gamma_u, \\ &\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_t. \end{aligned} \tag{6.1}$$

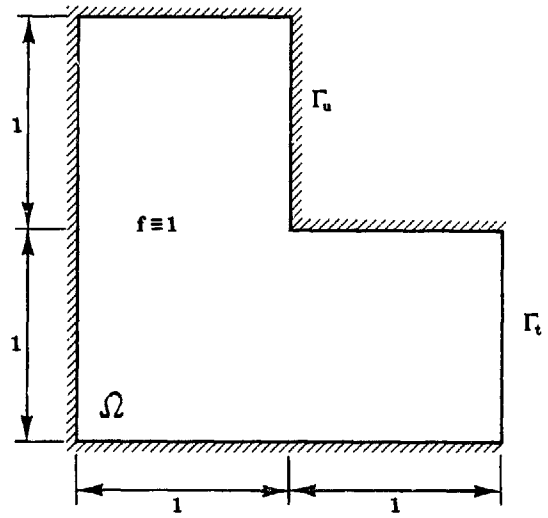

Fig. 11. Poisson equation in L-shaped domain.

Figure 12 presents an optimal *h-p* finite element mesh for the problem (see Part 3 in this series) and the corresponding finite element solution is shown in Fig. 13 (contour map) and Fig. 14 (a three-dimensional perspective). The scale below the mesh in Fig. 13 corresponds to different orders of approximation (different polynomial degrees p).

Example 2. Crack problem in linear elasticity

The linear elasticity equations

$$-\mu\,\Delta \boldsymbol{u} - (\mu + \lambda)\,\mathbf{grad}\,\mathrm{div}\,\boldsymbol{u} = \boldsymbol{f}\,, \tag{6.2}$$

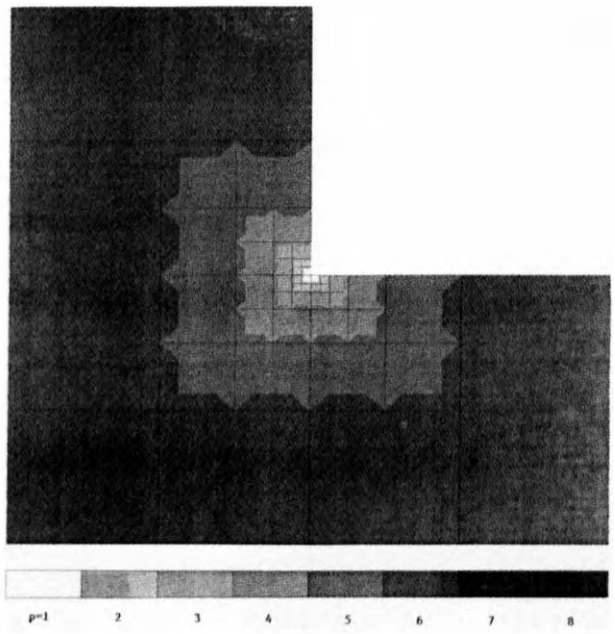

Fig. 12. Poisson equation in L-shaped domain. An optimal *h-p* FEM mesh.

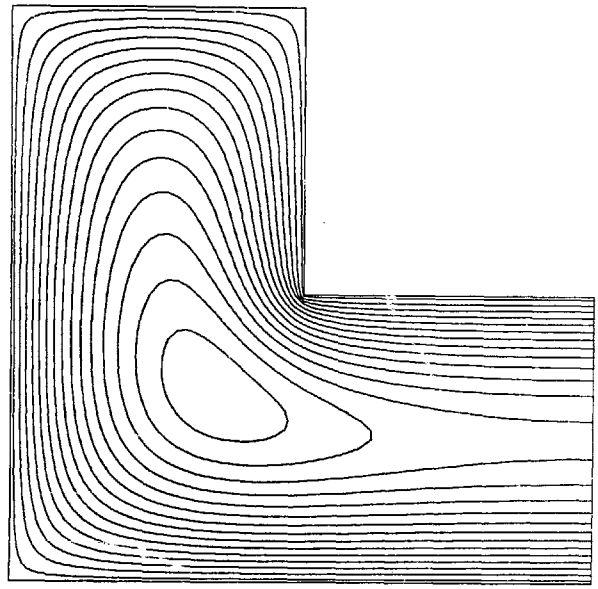

Fig. 13. Poisson equation in L-shaped domain. Contour map of the solution.

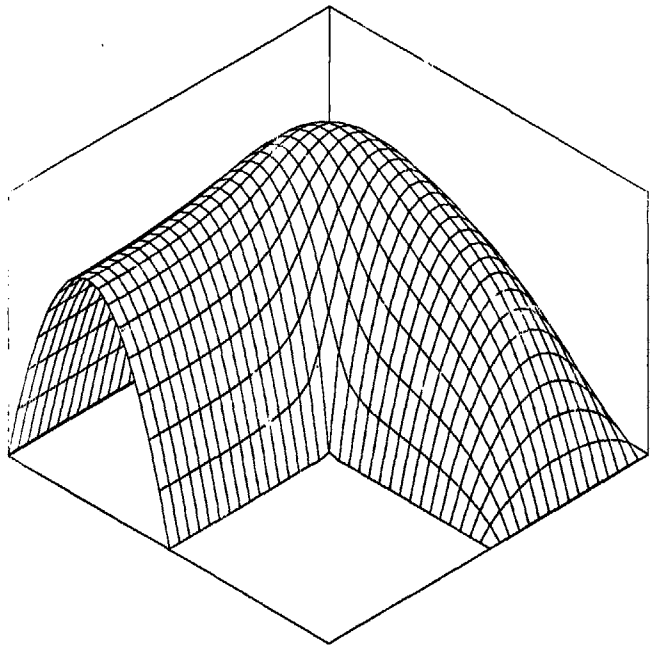

Fig. 14. Poisson equation in L-shaped domain. A three-dimensional plot of the solution.

with homogeneous kinematic boundary conditions on Γ_u and traction boundary conditions on Γ_t (Fig. 15) was solved using the data:

$$\text{body force } \boldsymbol{f} = \boldsymbol{0}\,, \qquad \mu = \lambda = 1.0\,. \tag{6.3}$$

An optimal *h-p* finite element mesh is shown in Fig. 16. Three-dimensional plots of the solution components are shown in Figs. 17a and 17b.

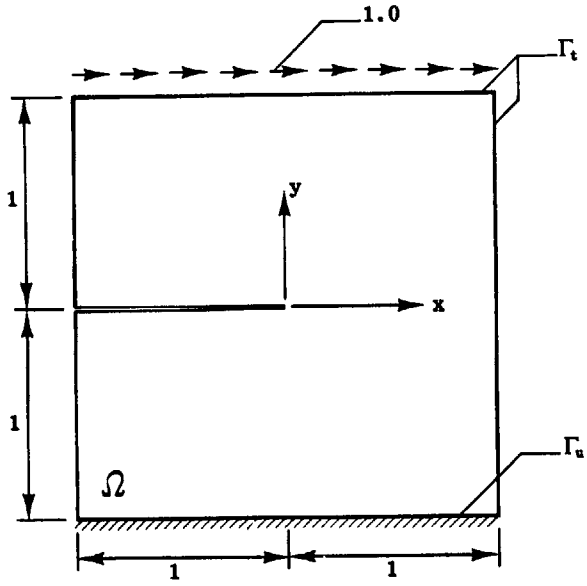

Fig. 15. Crack problem in linear elasticity.

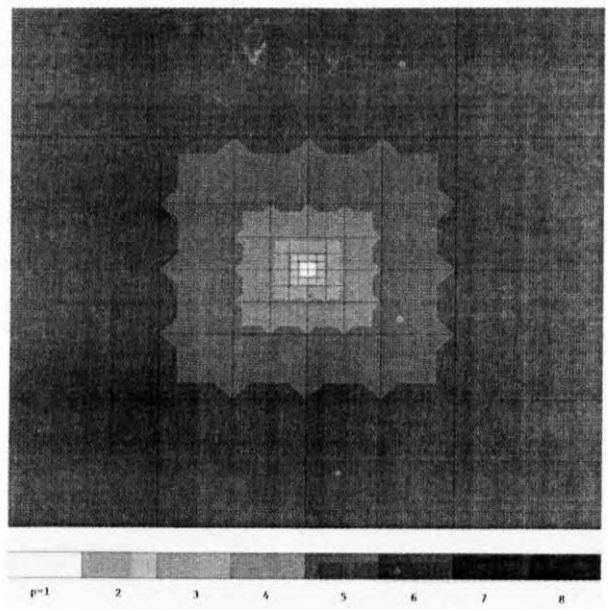

Fig. 16. Crack problem in linear elasticity. An optimal *h-p* FE mesh.

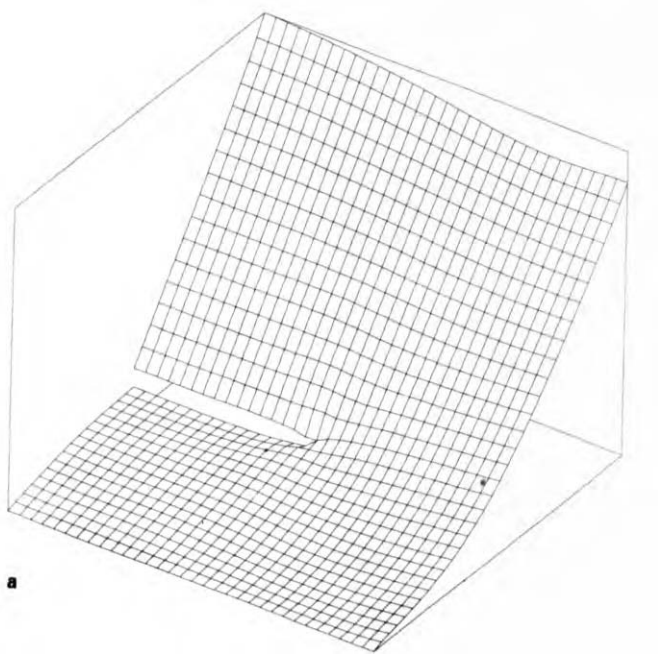

Fig. 17a. Crack problem in linear elasticity. The x component of the displacement vector.

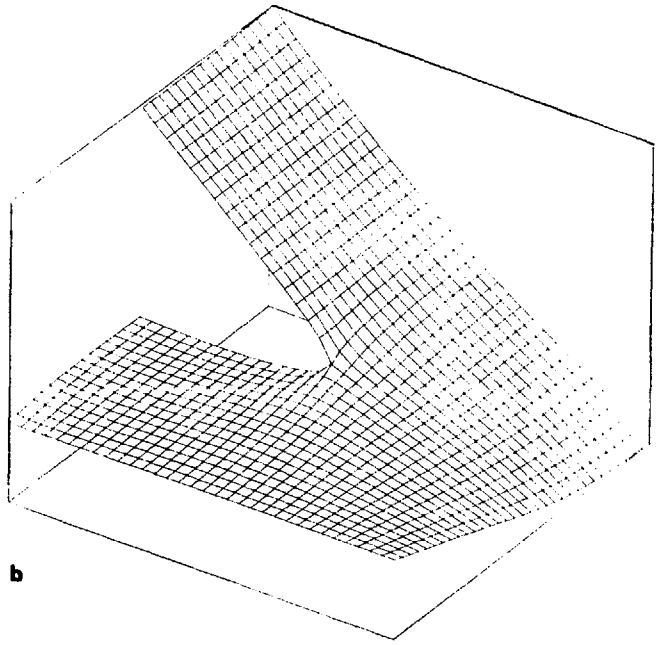

Fig. 17b. Crack problem in linear elasticity. The y component of the displacement vector.

Example 3. Rotation of a cone problem

The linear convection problem,

$$u_t + \boldsymbol{c} \cdot \nabla u = 0 \quad \text{in } \Omega\,, \tag{6.4}$$

was solved in the square domain $[-1, 1]^2$ (Fig. 18) using the second-order Taylor–Galerkin method described in Appendix A. The divergence-free velocity field $\boldsymbol{c}$ was selected as

$$\boldsymbol{c} = (v \cos\theta, v \sin\theta)\,, \quad v = 1\,, \tag{6.5}$$

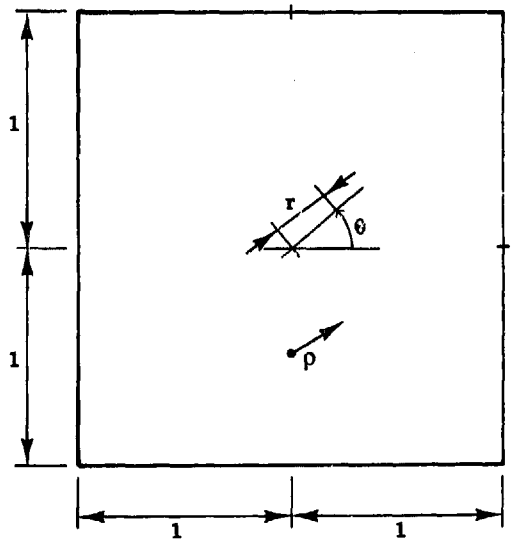

Fig. 18. Rotation of a cone problem.

and the initial data were specified as

$$u_0 = \begin{cases} \frac{1}{2}(1+\cos 4\rho\pi), & \text{if } \rho \leqslant \frac{1}{4}, \\ 0, & \text{otherwise}, \end{cases} \tag{6.6}$$

where

$$\rho^2 = (x-0)^2 + (y+0.5)^2 . \tag{6.7}$$

Homogeneous inflow boundary conditions ($u = 0$) and a constant time step

$$\Delta t = 2\pi/400 \tag{6.8}$$

were employed. Figures 19a and 19b present an adaptively changing finite element mesh and a corresponding finite element solution after one revolution (400 time steps). The top of the cone diminished from 1.0 to 0.984.

In all three problems a penalty method was used to implement the essential boundary conditions with a penalty parameter

$$\varepsilon = 0.001 . \tag{6.9}$$

Details on solution techniques for these classes of problems are to be provided in a later paper.

Example 4. A three-dimensional h-p finite element mesh

Though conceptionally the same, the three-dimensional formulation involves many more

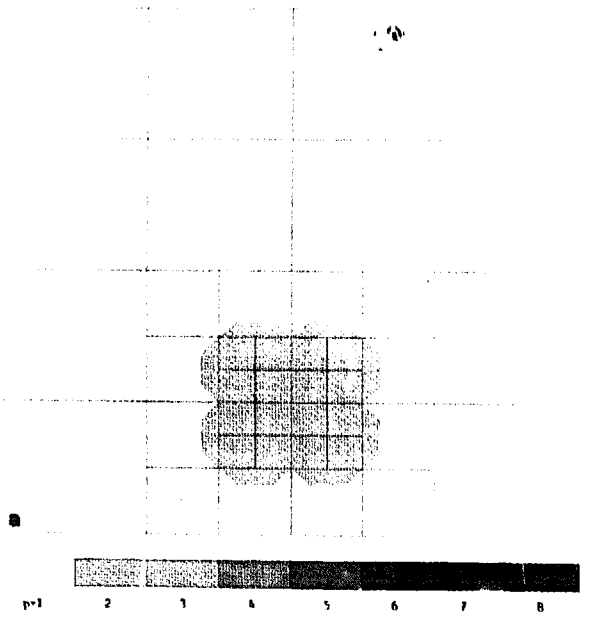

Fig. 19a. Rotation of a cone problem. An optional *h-p* FE mesh after one revolution.

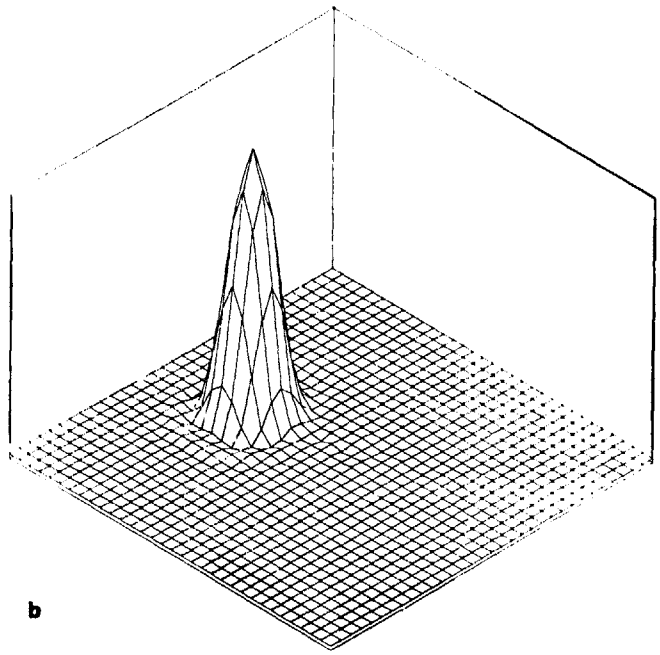

Fig. 19b. Rotation of a cone problem. FE solution after one revolution.

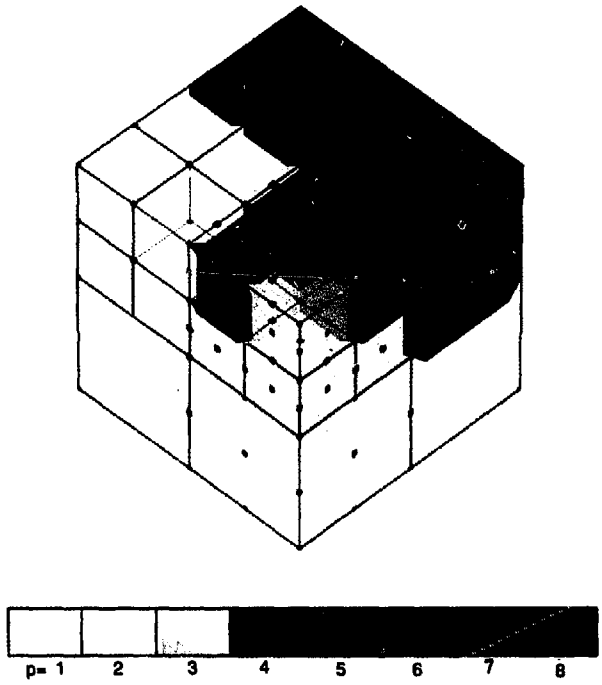

Fig. 20. Example of a three-dimensional *h-p* FE mesh.

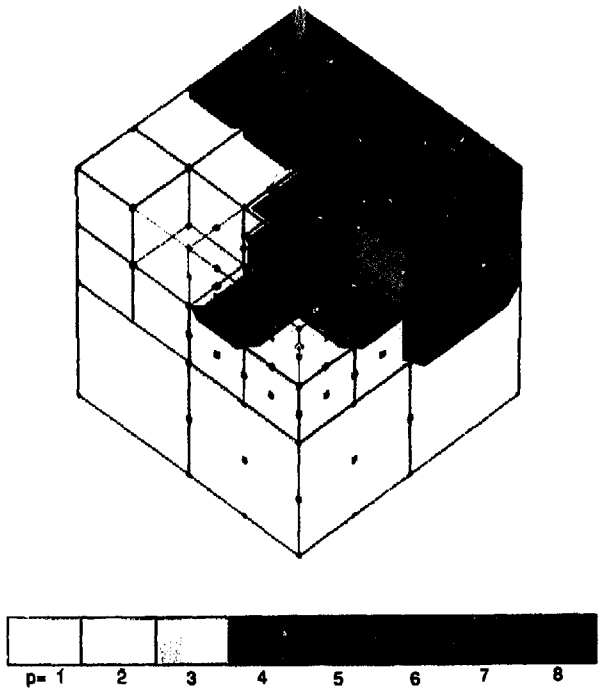

Fig. 21. The three-dimensional *h-p* mesh with elements removed to reveal interior connectivities and spectral orders.

technical details. We conclude the illustrative examples with a sample illustration of a three-dimensional mesh in a cubic domain $(-1,1)^3$ shown in Fig. 20. Figure 21 presents the same mesh with some elements removed to reveal the spectral orders chosen by the code for certain interior elements. The various colors correspond to polynomial degrees p, as indicated. Overlapping colors signify the enrichment of polynomial interfaces to maintain continuity of shape functions across interelement boundaries.

Appendix A. Examples of boundary value problems

In this appendix we present selected examples of problems which fall into the category of equations described by the abstract formulation (2.1).

A single elliptic equation

Consider the following elliptic boundary-value problem:

Find $u(x)$, such that

$$-\sum_{i,j=1}^{2}\frac{\partial}{\partial x_j}\left(a_{ij}\frac{\partial u}{\partial x_i}\right)+\sum_{i=1}^{2}b_i\frac{\partial u}{\partial x_i}+c=f \quad \text{in } \Omega\,, \tag{A.1a}$$

$$u=u_0 \quad \text{on } \Gamma_u\,, \tag{A.1b}$$

$$\sum_{i,j=1}^{2}a_{ij}\frac{\partial u}{\partial x_i}n_j=g \quad \text{on } \Gamma_t\,. \tag{A.1c}$$

Here Γ_u and Γ_t are two disjoint parts of the boundary $\partial\Omega$, $\boldsymbol{n}=(n_j)$ is the outward normal unit to $\partial\Omega$ and a_{ij}, b_i, e, f, u_0, g stand for given functions prescribed in the domain or on the boundary respectively. Customary assumptions on the coefficients are assumed so as to guarantee the ellipticity of the problem. In order to avoid the direct elimination of degrees of freedom due to the essential boundary conditions, the Dirichlet condition (A.1b) is replaced with the general natural boundary condition as follows:

$$u+\varepsilon\sum_{i,j=1}^{2}a_{ij}\frac{\partial u}{\partial x_i}n_j=g \quad \text{on } \Gamma_t\,, \tag{A.2}$$

where $\varepsilon>0$ is a sufficiently small penalty parameter.

The variational formulation of such a modified boundary value problem is as follows:

Find $u=u(x)$, such that

$$\begin{aligned}&\int_\Omega\left\{\sum_{i,j=1}^{2}a_{ij}\frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j}+\sum_{i=1}^{2}b_i\frac{\partial u}{\partial x_i}v+cuv\right\}\mathrm{d}x+\int_{\Gamma_u}\frac{1}{\varepsilon}uv\,\mathrm{d}s\\&\quad=\int_\Omega fv\,\mathrm{d}x+\int_{\Gamma_u}\frac{1}{\varepsilon}u_0v\,\mathrm{d}s+\int_{\Gamma_t}gv\,\mathrm{d}s\end{aligned} \tag{A.3}$$

for every $v=v(x)$.

Plane elasticity problem

Let $\Omega \subset \mathbb{R}^2$ be a domain with a boundary consisting of three disjoint parts Γ_u, Γ_m and Γ_t. The displacement-based plane elasticity problem can be formulated as follows:

Find the displacement field $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$, such that

$$\sigma_{ij}(\boldsymbol{u})_{,j} + f_i = 0 \quad \text{in } \Omega\,, \tag{A.4a}$$

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \quad \text{on } \Gamma_u\,, \tag{A.4b}$$

$$u_{\mathrm{N}} = \hat{u}_{\mathrm{N}} \quad \text{on } \Gamma_m\,, \tag{A.4c}$$

$$\sigma_{\mathrm{T}}(\boldsymbol{u}) = \hat{\sigma}_{\mathrm{T}} \quad \text{on } \Gamma_m\,, \tag{A.4d}$$

$$\boldsymbol{t}(\boldsymbol{u}) = \hat{\boldsymbol{t}} \quad \text{on } \Gamma_t\,. \tag{A.4e}$$

The following notation is used:

—The stress tensor corresponding to $\boldsymbol{u}$

$$\sigma_{ij} = E_{ijkl}\varepsilon_{kl}(\boldsymbol{u})\,, \tag{A.5}$$

where the strain tensor is defined as

$$\varepsilon_{kl}(\boldsymbol{u}) = \tfrac{1}{2}(u_{k,l} + u_{l,k}) \tag{A.6}$$

and E_{ijkl} is the tensor of elasticities satisfying the customary symmetry and ellipticity conditions.

—The stress vector on the boundary $\partial\Omega$

$$t_i = \sigma_{ij}n_j\,, \tag{A.7}$$

with $\boldsymbol{n} = (n_j)$ the outward normal unit to $\partial\Omega$.

—The normal and tangential displacement on the boundary

$$u_{\mathrm{N}} = u_i n_i = u_1 n_1 + u_2 n_2\,, \qquad u_{\mathrm{T}} = u_1(-n_2) + u_2 n_1\,. \tag{A.8}$$

—The normal and tangential stress on the boundary

$$\sigma_{\mathrm{N}} = t_i n_i = t_1 n_1 + t_2 n_2\,, \qquad \sigma_{\mathrm{T}} = t_1(-n_2) + t_2 n_1\,. \tag{A.9}$$

As usual, the summation convention is used and commas denote the partial differentiation.

We emphasize that numerous other boundary conditions can be considered including elastic supports on the boundary, prescribed tangential displacements and normal stresses, etc. In order to remain compatible with the abstract variational formulation, the essential (kinematic) boundary conditions are again modified using the penalty method.

In place of (A.4b) we have

$$u_i + \varepsilon t_i = \hat{u}_i \quad \text{on } \Gamma_u\,, \tag{A.10}$$

and (A.4c) is replaced with

$$u_N + \varepsilon\sigma_N(u) = \hat{u}_N \quad \text{on } \Gamma_m \,. \tag{A.11}$$

The variational formulation of such a modified boundary value problem is as follows:

$$\begin{aligned} &\text{Find } u = u(x)\,, \quad \text{such that} \\ &\int_\Omega E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)\,dx + \int_{\Gamma_u} \frac{1}{\varepsilon}\, u_i v_i\,ds + \int_{\Gamma_m} \frac{1}{\varepsilon}\, u_N v_N\,ds \\ &\quad = \int_\Omega f_i v_i\,dx + \int_{\Gamma_u} \frac{1}{\varepsilon}\,\hat{u}_i v_i\,ds + \int_{\Gamma_m} \left(\frac{1}{\varepsilon}\,\hat{u}_N v_N + \hat{\sigma}_T v_T\right) ds + \int_{\Gamma_t} \hat{t}_i v_i\,ds \\ &\text{for every } v = v(x)\,. \end{aligned} \tag{A.12}$$

Linear convection problem

As an example of an evolution problem we consider a linear convection problem of the form

$$\begin{aligned} &\text{Find } u = u(x, t)\,, \quad \text{such that} \\ &u_{,t} + c_i u_{,x_i} = 0 \quad \text{in } \Omega \\ &u = \hat{u} \quad \text{on } \Gamma_{\text{in}}\,, \\ &u = u_0 \quad \text{at } t = 0\,. \end{aligned} \tag{A.13}$$

Here $c = c(x)$ is a time independent divergence free velocity field prescribed in Ω, repeated indices are summed, and the inflow boundary is defined as

$$\Gamma_{\text{in}} = \{x \in \partial\Omega \,|\, c \cdot n < 0\}\,, \tag{A.14}$$

where n is the outward normal unit to $\partial\Omega$.

Due to the assumption that $\operatorname{div} c = 0$, the convection equation can be written in the conservation form

$$u_{,t} + (c_i u)_{,x_i} = 0\,. \tag{A.15}$$

Solving (A.15) for $u_{,t}$, we get

$$u_{,t} = -(c_i u)_{,x_i} = -c_i u_{,x_i}\,, \tag{A.16}$$

and differentiating with respect to time we get

$$u_{,tt} = -(c_i u)_{,tx_i} = -(c_i u_{,t})_{,x_i} = (c_i c_j u_{,x_j})_{,x_i}\,. \tag{A.17}$$

As a basis for an approximation in time we select the following second order formula:

$$u(t+\Delta t) - \tfrac{1}{2}\Delta t^2 u_{tt}(t+\Delta t) = u(t) + \Delta t u_t(t) + O(\Delta t^3)\,. \tag{A.18}$$

Substituting (A.16) and (A.17) into (A.18), we get

$$u(t+\Delta t) - \tfrac{1}{2}\Delta t^2 (c_i c_j u(t+\Delta t)_{,x_j})_{,x_i} = u(t) - \Delta t c_i u(t)_{,x_i}\,. \tag{A.19}$$

In order to obtain a variational formulation for (A.19), we multiply (A.19) by a test function v and integrate the second order term by parts. The resulting boundary term is

$$-\frac{1}{2}\Delta t^2 \int_{\partial\Omega} c_i c_j u(t+\Delta t)_{,x_j} n_i v \, ds\,. \tag{A.20}$$

Assuming that the test function v vanishes on the inflow boundary, we reduce the integration in (A.20) to the outflow boundary only:

$$-\frac{1}{2}\Delta t^2 \int_{\Gamma_{out}} c_i c_j u(t+\Delta t)_{,x_j} n_i v \, ds = \frac{1}{2}\Delta t^2 \int_{\Gamma_{out}} u_t(t+\Delta t) c_i n_i v \, ds\,. \tag{A.21}$$

The boundary integral suggests the natural boundary condition of the form

$$u_t(t+\Delta t) = \frac{1}{\Delta t}\,[u(t+\Delta t) - u(t)] + O(\Delta t)\,, \tag{A.22}$$

which results in the third order error in the boundary integral.

Summing up, we approximate (A.13) with a series of one-time step problems of the form

$$\begin{aligned}
&\text{Find } u^{n+1} = u^{n+1}(x)\,, \quad \text{such that}\\
&u^{n+1} - \tfrac{1}{2}\Delta t^2 (c_i c_j u^{n+1}_{,x_j})_{,x_i} = u^n - \Delta t c_i u^n_{,x_i}\,,\\
&u^{n+1} = \hat{u} \quad \text{on } \Gamma_{in}\,,\\
&-\tfrac{1}{2}\Delta t^2 c_i c_j u^{n+1}_{,x_j} n_i = \tfrac{1}{2}\Delta t c_i n_i (u^{n+1} - u^n) \quad \text{on } \Gamma_{out}\,,
\end{aligned} \tag{A.23}$$

with the initial condition u_0 replacing u^n for $n = 1$. Next, we modify the essential boundary condition on the inflow boundary replacing it with the condition of the form

$$u^{n+1} + \tfrac{1}{2}\varepsilon\Delta t^2 c_i c_j u^{n+1}_{,x_j} n_i = u_0 \quad \text{on } \Gamma_{in}\,. \tag{A.24}$$

Finally, the resulting variational formulation is as follows:

$$\text{Find } u^{n+1} = u^{n+1}(x)\,, \quad \text{such that}$$

$$\int_\Omega u^{n+1} v \, dx + \frac{1}{2}\Delta t^2 \int_\Omega c_i c_j u^{n+1}_{,x_j} v_{,x_i} \, dx + \int_{\Gamma_{in}} \frac{1}{\varepsilon}\, u^{n+1} v \, ds + \frac{1}{2}\,\Delta t \int_{\Gamma_{out}} (c_i n_i) u^{n+1} v \, ds$$

$$= \int_{\Omega} (u^n - \Delta t c_i u^n_{,x_i}) v \, dx + \int_{\Gamma_{in}} \frac{1}{\varepsilon} \hat{u} v \, ds + \frac{1}{2} \Delta t \int_{\Gamma_{out}} (c_i n_i) u^n v \, ds \tag{A.25}$$

for every $v = v(x)$.

Acknowledgment

Our work on theoretical issues concerned with data structures and error estimation for elliptic boundary-value problems began under the support of the Office of Naval Research.

References

[1] B. Guo and I. Babuška, The *h-p* version of the finite element method, Parts 1 and 2, Comput. Mech. (1986) (1) 21–41, 203–220.

[2] 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.

[3] 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.

[4] Ph. Devloo, J.T. Oden and P. Pattani, An *h-p* adaptive finite element method for the numerical simulation of compressible flow, Comput. Methods Appl. Mech. Engrg. 70 (1988) 203–235.

[5] L.M. Delves and C. Hall, An implicit matching principle for global element calculations, J. Inst. of Math. and its Applications 23 (1979) 223–234.

[6] W.C. Rheinboldt and Ch.K. Mesztenyi, On a data structure for adaptive finite element mesh refinements, ACM Trans. Math. Software 6 (2) (1980) 166–187.

[7] L. Demkowicz and J.T. Oden, A review of local mesh refinement techniques and corresponding data structures in *h*-type adaptive finite element methods, TICOM Rept. 88-02, The Texas Institute for Computational Mechanics, The University of Texas at Austin, Texas 78712.

[8] R.E. Bank, A.H. Sherman and A. Weiser, Refinement algorithms and data structures for regular mesh refinement, in: R.S. Stepleman, M. Carver, R. Peskin, W.F. Ames, R. Vichnevetsky, eds., IMACS Transactions on Scientific Computation, Vol. I: Scientific Computing (North-Holland, Amsterdam, 1983) 3–17.

[9] P.G. Ciarlet, The Finite Element Method for Elliptic Problems (North-Holland, Amsterdam, 1978).

[10] J.T. Oden and G.F. Carey, Finite Elements: Mathematical Aspects IV (Prentice-Hall, Englewood Cliffs, 1983).