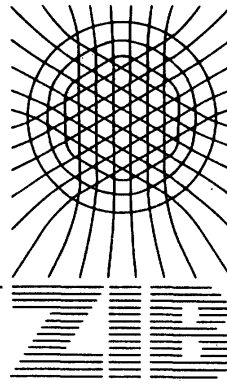

Konrad-Zuse-Zentrum für Informationstechnik Berlin



P. Deuffhard P. Leinen H. Yserentant

Concepts of an Adaptive Hierarchical Finite Element Code

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Strasse 10
1000 Berlin 31
Verantwortlich: Dr. Klaus André
Umschlagsatz und Druck: Verwaltungsdruckerei Berlin

ISSN 0933-7911

P. Deuffhard, P. Leinen, H. Yserentant

Concepts of an Adaptive Hierarchical Finite Element Code

Abstract

The paper presents the mathematical concepts underlying the new adaptive finite element code KASKADE, which, in its present form, applies to linear scalar second-order 2-D elliptic problems on general domains. Starting point for the new development is the recent work on hierarchical finite element bases due to Yserentant (1986). It is shown that this approach permits a flexible balance between iterative solver, local error estimator, and local mesh refinement device - which are the main components of an adaptive PDE code. Without use of standard multigrid techniques, the same kind of computational complexity is achieved - independent of any uniformity restrictions on the applied meshes. In addition, the method is extremely simple and all computations are purely local - making the method particularly attractive in view of parallel computing. The algorithmic approach is illustrated by a well-known critical test problem.

Keywords: finite elements, hierarchical basis, adaptive mesh refinement, preconditioned conjugate gradient methods.

Subject Classification: AMS(MOS): 65N30, 65N50, 65N20, 65F10, 65V05

Contents

0	Introduction	1
1	Basic Idea	2
2	Error Estimation	8
3	The Adaptive Algorithm	13
4	A Numerical Experiment	21
	References	27
	Appendix	29

The authors want to thank S. Wacker for her efficient and excellent $\text{T}_{\text{E}}\text{X}$ -typing of this manuscript.

0. Introduction

In the field of ordinary differential equations, *adaptive techniques* have been standard for many years. In boundary value problems involving partial differential equations, however, most of the work still concentrates on uniform or quasi-uniform grids. Because of the additional complications arising from the *boundary geometry*, this class of problems should even more be attacked by adaptive techniques – in view of real life applications in science and engineering.

The present paper focusses on the mathematical concepts underlying our new adaptive finite element code KASKADE. In its present version, this code applies to scalar self-adjoint second-order plane elliptic problems on general domains. Among the most popular comparable codes in the field are PLTMG due to Bank [3] and NFEARS due to Mesztenyi/Rheinboldt [10]. Both of these codes apply to nonlinear problems and offer path continuation facilities, whereas KASKADE at present just applies to linear problems.

Despite of this present restriction and a number of features that KASKADE shares with PLTMG, the new code opens an independent line of development – both in its concept and its implementation. Starting point is the rather recent work of Yserentant [12] on hierarchical finite element bases. Without use of standard multigrid techniques [8,11,4], the new code nevertheless achieves the same kind of computational complexity – independent of any uniformity restriction on the applied meshes. An important feature of the new method is its enticing *simplicity*. All computations are local, which implies small start-up times and no assembling of the global matrix. This feature makes the method particularly attractive in view of parallel computing. Conceptually, this simplicity permits a rather flexible balance between the three main components of an adaptive PDE code, which are: the iterative solver, the local error estimator, and the local mesh refinement.

The basic idea of the iterative linear equation solver implemented in KASKADE is outlined in Section 1 – there in the simplified context of uniform grid refinement. In Section 2, an edge-oriented estimator for the local discretization error is derived. It is the basic tool for the local mesh refinement strategy described in detail in Section 3. Moreover, this section deals with the hierarchical basis construction for nonuniformly refined meshes, the procedure nesting the termination criteria of the iteration with the discretization error estimator, and a complexity estimate for the general case. Finally, in Section 4, a critical numerical example illustrates the efficiency of the new approach.

1. Basic Idea

Let $\Omega \subset \mathbb{R}^2$ denote a bounded polygonal domain with boundary $\partial\Omega$ composed of two disjoint parts $\partial\Omega_1, \partial\Omega_2$. Consider the following elliptic boundary value problem:

$$\begin{aligned} -\nabla(a(x)\nabla u) + q(x)u &= f(x), \quad x \in \Omega \\ u|_{\partial\Omega_1} &= 0 \\ n \cdot a(x)\nabla u|_{\partial\Omega_2} &= 0, \end{aligned} \tag{1.1}$$

where a is a symmetric positive definite (2,2)-matrix, and a, q, f are assumed to be piecewise continuous. For simplicity, the assumption $q(x) \geq 0$ is made throughout this paper. However, an extension of the method to the indefinite case ($q(x)$ either sign) is possible along the lines worked out in Yserentant [13]. The weak formulation associated with (1.1) is: find a function $U \in H(\Omega)$ satisfying

$$B(U, v) = \int_{\Omega} f(x)v dx, \quad v \in H(\Omega) \tag{1.2}$$

where

$$B(u, v) := \int_{\Omega} [a(x)\nabla u \cdot \nabla v + q(x)u \cdot v] dx. \tag{1.3}$$

Herein the solution space $H(\Omega)$ is defined as

$$H(\Omega) := \left\{ u \in H^1(\Omega) \mid u|_{\partial\Omega_1} = 0 \right\} \tag{1.4}$$

where the Dirichlet boundary condition is understood to hold in the sense of the trace operator. Assume that (1.2) has a unique solution (which excludes, for instance, the case $q(x) \equiv 0, \partial\Omega_1 = \emptyset$).

For a finite element treatment of (1.2), $\bar{\Omega}$ is covered by a coarse triangulation \mathcal{T}_0 . With \mathcal{T}_0 as initial grid, subsequent mesh refinements lead to triangulations $\mathcal{T}_1, \mathcal{T}_2 \dots$. For ease of presentation, *uniform* mesh refinements are assumed (throughout this section only), compare Figure 1.

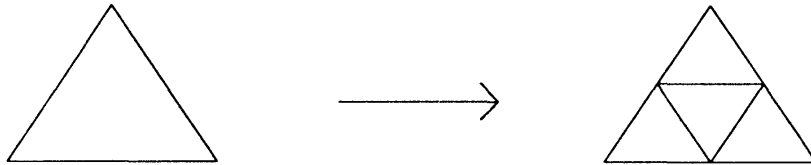


Figure 1 Uniform mesh refinement.

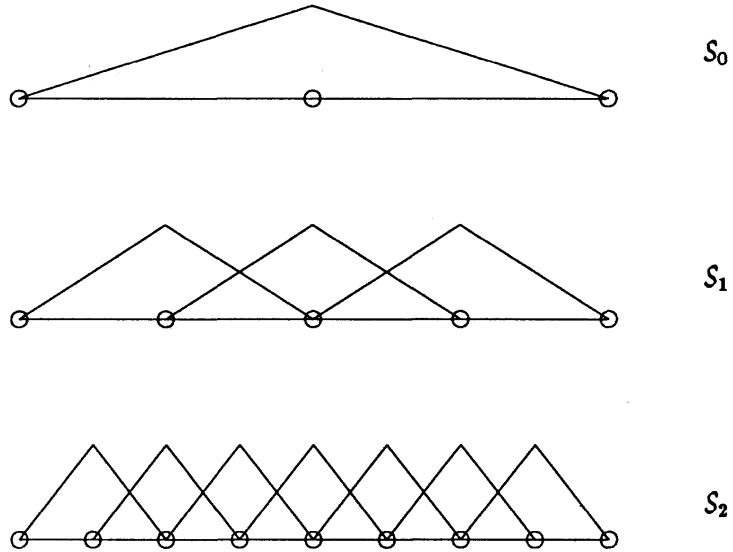


Figure 2 Nodal bases for S_i in the 1-D case (homogeneous Dirichlet boundary conditions).

Each triangulation \mathcal{T}_i is associated with the piecewise linear element space S_i of dimension n_i . Usually, S_i is represented by its *nodal basis* – see Figure 2 for the 1-D case. The method to be described herein uses the alternative representation in terms of *hierarchical bases* (compare Yserentant [12]). In Figure 3, the corresponding 1-D case is illustrated. The essential feature of this representation is that S_{i+1} is partitioned according to

$$S_{i+1} = S_i \oplus \mathcal{V}_{i+1} \quad (1.5)$$

where the functions in \mathcal{V}_{i+1} vanish at the nodes associated with S_i . The hierarchical basis of S_{i+1} is obtained by keeping the hierarchical basis of S_i and representing \mathcal{V}_{i+1} by its nodal basis.

The finite element solution of (1.2) requires the determination of a function $U_i \in S_i$ satisfying

$$B(U_i, v) = \int_{\Omega} f(x)v dx, \quad v \in S_i. \quad (1.6)$$

In *nodal basis representation*, this is equivalent to the linear system

$$A_i u_i = b_i. \quad (1.7)$$

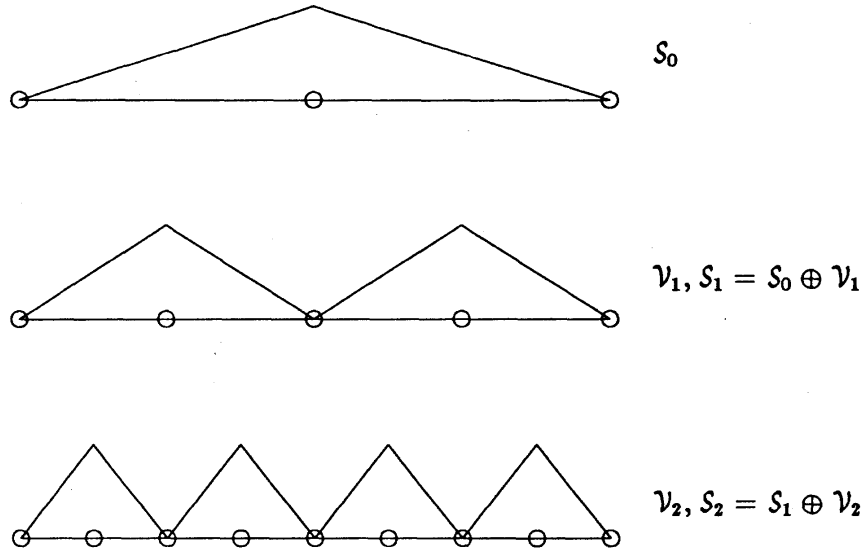


Figure 3 Hierarchical bases for S_i – to be compared with Figure 2.

In *hierarchical basis representation*, one obtains

$$\bar{A}_i \bar{u}_i = \bar{b}_i \quad (1.8)$$

where

$$u_i = S_i \bar{u}_i, \quad \bar{b}_i = S_i^T b_i, \quad \bar{A}_i = S_i^T A_i S_i. \quad (1.9)$$

Assuming natural ordering of hierarchical basis functions, one may define the symmetric positive definite (n_i, n_i) -matrix

$$\bar{D}_i := \begin{pmatrix} \bar{A}_0 & 0 \\ 0 & D_i \end{pmatrix} =: L_i L_i^T, \quad (1.10)$$

where the diagonal matrix D_i contains those diagonal entries of \bar{A}_i that are not contained in \bar{A}_0 . It has been shown in [12] that for

$$\hat{A}_i := L_i^{-1} \bar{A}_i L_i^{-T} \equiv L_i^{-1} S_i^T A_i S_i L_i^{-T} \quad (1.11)$$

the spectral condition number κ satisfies

$$\kappa(\hat{A}_i) \leq C_1 \cdot (i + 1)^2 \quad (1.12)$$

where C_1 is bounded independent of i and of the regularity of the boundary value problem – see also [5]. Recall that in nodal basis representation one has

$$\kappa(A_i) \doteq C_2 \cdot 4^i . \quad (1.13)$$

In view of result (1.12), the matrix

$$S_i^{-T} \bar{D}_i S_i^{-1} \quad (1.14)$$

may serve as an efficient *preconditioner* for the linear system (1.7). This means that the linear system

$$\hat{A}_i \hat{u}_i = \hat{b}_i , \quad \hat{b}_i := L_i^{-1} S_i^T b_i \quad (1.15)$$

can be solved in a fast way by means of *conjugate gradient iteration*. The solution u_i in nodal basis representation is

$$u_i = S_i L_i^{-T} \hat{u}_i : \quad (1.16)$$

Let

$$\|u\| := B(u, u)^{1/2} , \quad u \in H(\Omega) , \quad (1.17)$$

denote the energy norm. Let ε be the user prescribed reduction factor from the initial to the final error - measured in the energy norm. Then the number $m_i(\varepsilon)$ of necessary cg-iterations is well-known [1] to be bounded by

$$m_i(\varepsilon) \leq \frac{1}{2} \sqrt{\kappa} \cdot \left| \log \frac{\varepsilon}{2} \right| . \quad (1.18)$$

Upon inserting (1.12), one thus obtains

$$m_i(\varepsilon) \leq C_3 \cdot (i + 1) \cdot \left| \log \frac{\varepsilon}{2} \right| . \quad (1.19)$$

As shown in [12], neither \hat{A}_i nor \hat{b}_i need to be computed explicitly. Rather routines for the fast evaluation of *matrix times vector* with the matrices

$$L_i^{-1} , \quad L_i^{-T} , \quad S_i , \quad S_i^T , \quad A_i \quad (1.20)$$

are sufficient. Surprisingly, the evaluation of $S_i w$, $S_i^T w$ can be performed recursively and requires only $2n_i$ additions and n_i divisions by 2 each. This means that the amount of work per cg-iteration is proportional to n_i . As a consequence, the total amount of work sums up to

$$\alpha_i(\varepsilon) \leq C_4 \cdot n_i \cdot (i + 1) \cdot \left| \log \frac{\varepsilon}{2} \right| . \quad (1.21)$$

Of course, in order to avoid unnecessary computational work, the value ε should reflect the *discretization error* on level i . In the best case, the discretization error for linear elements behaves like $\mathcal{O}(h_i)$ with

$$h_i = h_0 \cdot \left(\frac{1}{2}\right)^i \quad (1.22)$$

for uniform refinements. This means that a prescribed reduction factor ε less than

$$\varepsilon_i := \left(\frac{1}{2}\right)^i \quad (1.23)$$

is unreasonable. To achieve the reduction ε_i one needs at most

$$m_i(\varepsilon_i) \leq \bar{C}_3 \cdot (i+1)^2 \quad (1.24)$$

preconditioned cg-iterations and, at most,

$$\alpha_i(\varepsilon_i) \leq \bar{C}_4 \cdot n_i \cdot (i+1)^2 \quad (1.25)$$

operations.

A rough interpretation of (1.23) is that level i supplies, in the best case, i binary digits of accuracy. However, $i-1$ bits can be obtained on coarser grids at lower cost. This naturally inspires the following implementation, reminding of the nested multigrid iteration process [8]: rather than solving (1.7) for $i=j$, a *cascade* of linear systems (1.7) for $i=0, 1, \dots, j$ is solved successively. For $i=0$, i.e. on the coarsest grid, Cholesky decomposition may be applied, yielding $U_0 \in \mathcal{S}_0$, thus initializing the cascade. On finer levels, certain approximations of U_i , say $\tilde{U}_i \in \mathcal{S}_i$, are determined by preconditioned cg-iteration. The iteration on level i is started with $U_i^{(0)} := \tilde{U}_{i-1}$ and terminated when

$$\|\tilde{U}_i - U_i\| \leq \bar{\varepsilon} \|\tilde{U}_{i-1} - U\| \quad (1.26)$$

for some reduction factor $\bar{\varepsilon}$ to be discussed below. Recall that the norm on the right-hand side of (1.26) measures the distance between the solution U of the continuous problem (1.2) and the last iterate on level $i-1$.

Repeated induction leads to

$$\|\tilde{U}_j - U\|^2 \leq \sum_{i=0}^j (\bar{\varepsilon}^2)^{j-i} \|U_i - U\|^2. \quad (1.27)$$

This means that for the choice

$$\bar{\varepsilon} = \rho \cdot \frac{1}{2}, \quad 0 < \rho < 1, \quad (1.28)$$

the errors introduced on coarser levels are damped sufficiently fast so that the iterates \tilde{U}_j achieve discretization error accuracy. To see this, assume that, for $0 < \alpha \leq 1$,

$$\|U_i - U\| \leq C_5 \cdot h_i^\alpha = C_5 \cdot h_0^\alpha \cdot \left(\frac{1}{2}\right)^{i\alpha}. \quad (1.29)$$

Then insertion of (1.29) into (1.27) yields

$$\|\tilde{U}_j - U\| \leq C_5 \cdot h_j^\alpha \cdot \left(1 - \left(\frac{1}{4}\right)^{1-\alpha} \rho^2\right)^{-1/2} \quad (1.30)$$

Note that in this estimate the usually critical case of small α is uncritical, whereas the best case $\alpha = 1$ appears to be the critical one.

As shown above, the crucial condition nesting the different levels is (1.26). Because of $\tilde{U}_{i-1} = U_i^{(0)}$ and

$$\|U_i^{(0)} - U\|^2 = \|U_i^{(0)} - U_i\|^2 + \|U_i - U\|^2 \geq \|U_i^{(0)} - U_i\|^2, \quad (1.31)$$

the condition

$$\|\tilde{U}_i - U_i\| \leq \bar{\varepsilon} \|U_i^{(0)} - U_i\| \quad (1.32)$$

is sufficient for (1.26). Therefore, the number

$$m_i(\bar{\varepsilon}) \leq C_3 \cdot (i+1) \cdot \left| \log \frac{\bar{\varepsilon}}{2} \right| \quad (1.33)$$

of cg-iterations needed to satisfy (1.32) is an upper bound for the number of cg-iterations needed to satisfy (1.26). The total amount of work to compute \tilde{U}_j from U_0 can be estimated by

$$\tilde{\alpha}_j \leq \sum_{i=1}^j \alpha_i(\bar{\varepsilon}). \quad (1.34)$$

For further estimation, assume *geometric progression* for the number of nodes, i.e.

$$n_{i+1} \geq s \cdot n_i, \quad s > 1. \quad (1.35)$$

Note that $s \doteq 4$ for uniform refinement of grids (this section only). By (1.21), one obtains

$$\begin{aligned} \frac{\tilde{\alpha}_j}{n_j} &\leq \sum_{i=1}^j C_4 \cdot \frac{n_i}{n_j} \cdot (i+1) \cdot \left| \log \frac{\bar{\varepsilon}}{2} \right| \leq \\ &\leq \sum_{i=1}^j C_4 \cdot \left(\frac{1}{s} \right)^{j-i} \cdot (j+1) \cdot \left| \log \frac{\bar{\varepsilon}}{2} \right|. \end{aligned}$$

Hence,

$$\tilde{\alpha}_j \leq C_4 \cdot n_j \cdot (j+1) \cdot \frac{s}{s-1} \cdot \left| \log \frac{\bar{\varepsilon}}{2} \right|. \quad (1.36)$$

With $n = n_j \approx 4^j$, one obtains a computational complexity of

$$O(n \text{ ld } n).$$

2. Error Estimation

An actual realization of the cascade iteration described in the preceding section requires an estimator for both the discretization error and the iteration error – compare the crucial criterion (1.26). At the same time, a discretization error estimator is anyway required for the construction of nonuniform meshes. In principle, the highly efficient and reliable *triangle oriented* local error estimator of Bank and Weiser [7], which is realized in PLTMG [3], could also be applied in the context here. However, in the hierarchical basis framework, a much simpler *edge oriented* local error estimator seems to be worth discussing. This alternative is the subject of this section. The method to be developed has certain similarities with ideas of Zienkiewicz and Craig; see [14], for example.

Let \mathcal{T} be a fixed triangulation, which is either the initial triangulation or a triangulation obtained in the course of the adaptive process described in Section 3. Let S_L and S_Q denote the spaces of piecewise linear and piecewise quadratic finite element functions associated with \mathcal{T} . In what follows, the distance between the approximate finite element solution S_L and the exact finite element solution in S_Q will be estimated. S_Q can be decomposed into a direct sum

$$S_Q = S_L \oplus \mathcal{V}_Q . \quad (2.1)$$

The functions of \mathcal{V}_Q vanish at the vertices of the triangles in \mathcal{T} and can be characterized by their midpoint values on the edges of these triangles. One immediately observes that this construction corresponds to the splitting (1.5). Now, assume that a hierarchical basis of S_L is given (for details of its construction compare Section 3). Then the hierarchical basis of S_Q is naturally defined by adding the usual nodal basis functions of S_Q spanning \mathcal{V}_Q . For an illustration in the 1-D case see Figure 4. With respect to this hierarchical basis of S_Q every function has a coefficient vector

$$v = \begin{pmatrix} v_L \\ v_Q \end{pmatrix} \quad (2.2)$$

with v_L representing the linear part in S_L and v_Q representing the quadratic part in \mathcal{V}_Q .

The finite element equations corresponding to the boundary value problem (1.2) and to this hierarchical basis of S_Q are

$$\begin{pmatrix} A_{LL} & A_{LQ} \\ A_{QL} & A_{QQ} \end{pmatrix} \begin{pmatrix} u_L^* \\ u_Q^* \end{pmatrix} = \begin{pmatrix} b_L \\ b_Q \end{pmatrix} \quad (2.3)$$

or, in short form,

$$Au^* = b . \quad (2.4)$$

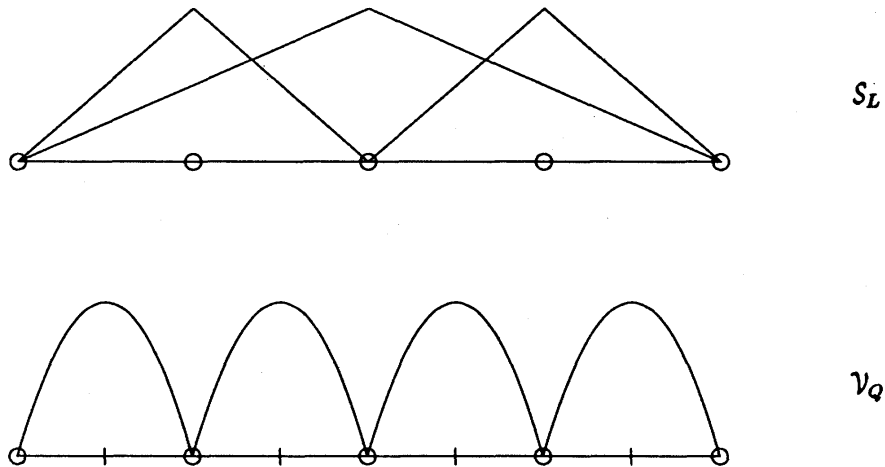


Figure 4 Hierarchical basis representation of $S_Q = S_L \oplus V_Q$

The finite element equations with respect to S_L only are

$$A_{LL}u_L = b_L . \quad (2.5)$$

Assume that an approximate solution \tilde{u}_L of (2.5) is at hand. In the hierarchical basis of S_Q the coefficient vector $(\tilde{u}_L, 0)$ represents the same function. The defect

$$\begin{pmatrix} d_L \\ d_Q \end{pmatrix} := \begin{pmatrix} u_L^* \\ u_Q^* \end{pmatrix} - \begin{pmatrix} \tilde{u}_L \\ 0 \end{pmatrix} \quad (2.6)$$

represents the difference between the exact piecewise quadratic finite element solution in S_Q and the given approximate piecewise linear solution in S_L . It satisfies

$$\begin{pmatrix} A_{LL} & A_{LQ} \\ A_{QL} & A_{QQ} \end{pmatrix} \begin{pmatrix} d_L \\ d_Q \end{pmatrix} = \begin{pmatrix} r_L \\ r_Q \end{pmatrix} \quad (2.7)$$

with

$$r_L := b_L - A_{LL}\tilde{u}_L , \quad (2.8)$$

$$r_Q := b_Q - A_{QL}\tilde{u}_L . \quad (2.9)$$

Of course, an exact solution of (2.7) would be far too expensive. To estimate d_L and d_Q , (2.7) is substituted by a much simpler system

$$\begin{pmatrix} B_{LL} & B_{LQ} \\ B_{QL} & B_{QQ} \end{pmatrix} \begin{pmatrix} \tilde{d}_L \\ \tilde{d}_Q \end{pmatrix} = \begin{pmatrix} r_L \\ r_Q \end{pmatrix} \quad (2.10)$$

or, in short notation,

$$B\tilde{d} = r . \quad (2.11)$$

In the context of hierarchical bases

$$\begin{pmatrix} B_{LL} & B_{LQ} \\ B_{QL} & B_{QQ} \end{pmatrix} := \begin{pmatrix} D_{LL} & 0 \\ 0 & D_{QQ} \end{pmatrix} \quad (2.12)$$

is a natural choice. The matrix D_{LL} corresponds to the already defined nearly diagonal matrix (1.10), and D_{QQ} is the diagonal part of A_{QQ} . Then \tilde{d}_L and \tilde{d}_Q can be computed from

$$D_{LL}\tilde{d}_L = r_L , \quad D_{QQ}\tilde{d}_Q = r_Q . \quad (2.13)$$

The replacement of d by \tilde{d} essentially means that instead of the optimal global correction in S_Q a set of optimal one-dimensional corrections in each of the directions of the hierarchical basis functions spanning S_Q is determined. In the interior of the domain and on the free part of the boundary these corrections are computed utilizing the underlying variational principle. On the Dirichlet part of the boundary the prescribed boundary values are used. If \tilde{u}_L is the exact solution of (2.5), one may interpret this procedure as solving Dirichlet problems with respect to one-dimensional spaces associated with the edges.

An appropriate measure for the size of the error is its energy norm

$$|A^{1/2}d|^2 = (d, Ad) \quad (2.14)$$

where (\cdot, \cdot) denotes the Euclidean inner product and $|\cdot|$ the induced norm. This value is approximated by

$$|B^{1/2}\tilde{d}|^2 = |D_{LL}^{1/2}\tilde{d}_L|^2 + |D_{QQ}^{1/2}\tilde{d}_Q|^2 \quad (2.15)$$

where

$$|D_{LL}^{1/2}\tilde{d}_L|^2 = (\tilde{d}_L, D_{LL}\tilde{d}_L) = (D_{LL}^{-1}r_L, r_L) = |D_{LL}^{-1/2}r_L|^2 , \quad (2.16)$$

$$|D_{QQ}^{1/2}\tilde{d}_Q|^2 = (\tilde{d}_Q, D_{QQ}\tilde{d}_Q) = (D_{QQ}^{-1}r_Q, r_Q) = |D_{QQ}^{-1/2}r_Q|^2 . \quad (2.17)$$

In order to justify this approach, the question of whether the value (2.15) is a reasonable estimate for (2.14) must be studied.

Lemma:

Let B be an arbitrary symmetric and positive definite preconditioner for A and assume that μ_1 and μ_2 are positive constants with

$$\frac{1}{\mu_1}(v, Bv) \leq (v, Av) \leq \mu_2(v, Bv) \quad (2.18)$$

for all coefficient vectors v . Let r be a given residual vector and let d and \tilde{d} be the solutions of

$$Ad = r, B\tilde{d} = r, \quad (2.19)$$

respectively. Then

$$\frac{1}{\sqrt{\mu_1}} |A^{1/2}d| \leq |B^{1/2}\tilde{d}| \leq \sqrt{\mu_2} |A^{1/2}d|. \quad (2.20)$$

Proof: Using (2.19) one gets

$$\begin{aligned} |A^{1/2}d| &= |A^{1/2}A^{-1}B\tilde{d}| \leq |A^{-1/2}B^{1/2}| |B^{1/2}\tilde{d}|, \\ |B^{1/2}\tilde{d}| &= |B^{1/2}B^{-1}Ad| \leq |B^{-1/2}A^{1/2}| |A^{1/2}d|, \end{aligned}$$

and because of

$$|C|^2 = |C^T|^2 = |CC^T| = |C^TC|$$

one has

$$\begin{aligned} |A^{-1/2}B^{1/2}|^2 &= |(B^{-1/2}AB^{-1/2})^{-1}| \leq \mu_1, \\ |B^{-1/2}A^{1/2}|^2 &= |B^{-1/2}AB^{-1/2}| \leq \mu_2. \end{aligned}$$

Inserting above confirms the proposition. \blacksquare

In the following the constants μ_1 and μ_2 are estimated for the matrix (2.12). First, there exist positive constants γ_1 and γ_2 with

$$\begin{aligned} &\frac{1}{\gamma_1} \{(v_L, A_{LL}v_L) + (v_Q, D_{QQ}v_Q)\} \\ &\leq \left(\begin{pmatrix} v_L \\ v_Q \end{pmatrix}, \begin{pmatrix} A_{LL} & A_{LQ} \\ A_{QL} & A_{QQ} \end{pmatrix} \begin{pmatrix} v_L \\ v_Q \end{pmatrix} \right) \\ &\leq \gamma_2 \{(v_L, A_{LL}v_L) + (v_Q, D_{QQ}v_Q)\} \end{aligned} \quad (2.21)$$

for all coefficient vectors v partitioned as usual. γ_1 and γ_2 depend only on the degeneration of the triangles and on the local ellipticity of the boundary value problem, not on the meshsize or any quasi-uniformity assumption for the triangulation, and not on global properties of the boundary value problem. The estimate (2.21) can be proved by considering each triangle separately and by summing up the local estimates. The upper bound in (2.21) is an easy consequence of the triangle and the Cauchy-Schwarz inequality. Splitting a quadratic function into its linear part and the remaining three parts associated with the edges of the given triangle one gets

$$\gamma_2 \leq 4. \quad (2.22)$$

For the proof of the lower bound in (2.21), one utilizes the locally uniform ellipticity of the bilinear form (1.3) and transforms the individual triangles to a fixed reference triangle.

In a second step one utilizes that D_{LL} is spectrally nearly equivalent to A_{LL} ; it has been shown in [12] that there exist positive constants K_1 and K_2 with

$$\frac{1}{K_1 j^2} (v_L, D_{LL} v_L) \leq (v_L, A_{LL} v_L) \leq K_2 (v_L, D_{LL} v_L) \quad (2.23)$$

for all coefficient vectors v_L . Here j denotes the *depth* of the triangulation \mathcal{T} ; for its definition see Section 3 below. Like the constants γ_1 and γ_2 in (2.21), the constants K_1 and K_2 only depend on the degeneration of the triangles and on the local ellipticity of the boundary value problem; see also [5]. Inserting (2.23) into (2.21), the estimate (2.18) follows with

$$\mu_1 \leq K_1 \gamma_1 j^2, \quad \mu_2 \leq K_2 \gamma_2 \leq 4K_2. \quad (2.24)$$

Summarizing one obtains

$$\frac{1}{\sqrt{K_1 \gamma_1} j} |A^{1/2} d| \leq |B^{1/2} \tilde{d}| \leq 2\sqrt{K_2} |A^{1/2} d|. \quad (2.25)$$

This result confirms that $|B^{1/2} \tilde{d}|$ has the usually required properties of an error estimator [7].

The slightly unsatisfactory appearance of j in (2.25) disappears as \tilde{u}_L approaches the exact solution of the linear system (2.5). To see this, just observe that for

$$r_L = b_L - A_{LL} \tilde{u}_L = 0 \quad (2.26)$$

the linear systems

$$\begin{pmatrix} D_{LL} & 0 \\ 0 & D_{QQ} \end{pmatrix} \begin{pmatrix} \tilde{d}_L \\ \tilde{d}_Q \end{pmatrix} = \begin{pmatrix} r_L \\ r_Q \end{pmatrix} \quad (2.27)$$

and

$$\begin{pmatrix} A_{LL} & 0 \\ 0 & D_{QQ} \end{pmatrix} \begin{pmatrix} \tilde{d}_L \\ \tilde{d}_Q \end{pmatrix} = \begin{pmatrix} r_L \\ r_Q \end{pmatrix} \quad (2.28)$$

have the same solution

$$\tilde{d}_L = 0, \quad \tilde{d}_Q = D_{QQ}^{-1} r_Q. \quad (2.29)$$

For this case the estimate (2.25) can be improved to

$$\frac{1}{\sqrt{\gamma_1}} |A^{1/2} d| \leq |B^{1/2} \tilde{d}| \leq 2 |A^{1/2} d|. \quad (2.30)$$

3. The Adaptive Algorithm

In the preceding sections, the main ingredients necessary for the construction of our adaptive finite element code KASKADE have been presented. In Section 1, the iterative solver has been outlined – however, in the restricted situation of uniform grid refinement, which is relieved in the present section. In Section 2, an estimator for the local discretization error has been worked out – which will be used to derive the strategy for nonuniform mesh refinement.

The process for solving the boundary value problem (1.2) starts at some initial triangulation \mathcal{T}^0 – usually given by the user or generated by some preprocessor. It is assumed that \mathcal{T}^0 reflects the geometry of the domain Ω sufficiently well. Typically, \mathcal{T}^0 will consist of only few triangles. Let n_0 be the number of vertices of \mathcal{T}^0 and let S^0 denote the space of piecewise linear finite element functions with respect to \mathcal{T}^0 . The finite element solution $U^0 \in S^0$ of problem (1.2) is obtained by sparse elimination techniques. This establishes cascade level $i = 0$.

Mesh refinement strategy. For the time being, the transition from cascade level $i = 0$ to cascade level $i = 1$ is described. First, the local error estimator of Section 2 is activated, which supplies the components of the vector

$$D_{QQ}^{1/2} \tilde{d}_Q = D_{QQ}^{-1/2} r_Q , \quad (3.1)$$

one component for each edge of the triangulation. Then a threshold value Θ is computed, and those edges are marked for local refinement that satisfy

$$\left(D_{QQ}^{1/2} \tilde{d}_Q|_{\text{edge}} \right)^2 \geq \Theta . \quad (3.2)$$

Our present choice is

$$\Theta := \sigma \cdot \bar{m} , \quad \sigma := 0.95 , \quad (3.3)$$

with \bar{m} denoting the mean value of the squares of the components of the weighted residual (3.1). Next, as a standard procedure, each triangle having at least one edge marked for refinement, is refined regularly (“red” refinement in the notation of [6]) – see Figure 5. An exception is made for obtuse triangles, which are refined to reduce the largest angle – see Figure 6.

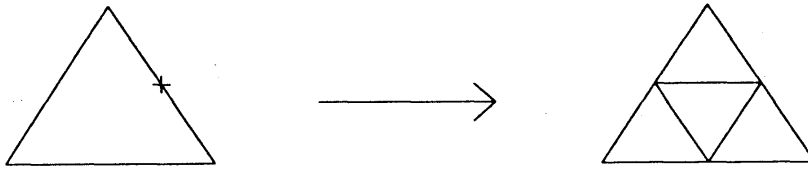


Figure 5 Standard regular or “red” refinement [6].

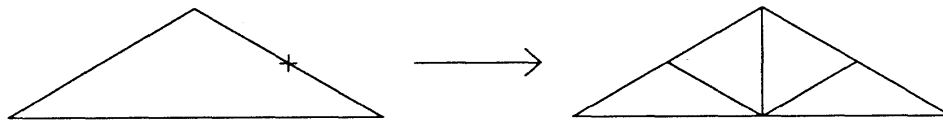


Figure 6 Regular or “red” refinement for obtuse triangles. (See package PLTMG, edition 5.0 [3]).

The triangulation is then completed by possibly using additional “red” refinements and finally “green” refinements (in the notation of [6]). The typical situations occurring are depicted in Figure 7 and Figure 8.

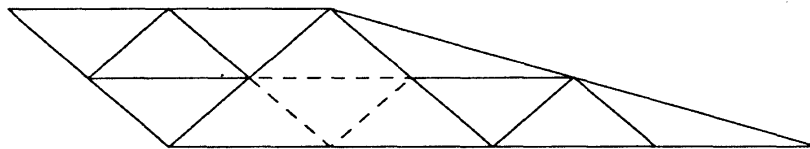


Figure 7 Completion of triangulation by regular or “red” refinement.

Let n be the number of vertices of the thus defined triangulation \mathcal{T} . In view of (1.35) one now checks the condition

$$n \geq s \cdot n_0 \tag{3.4}$$

with $s := 2$ in the present version of KASKADE. If (3.4) holds, then cascade level $i = 1$ is established and $\mathcal{T}^1 := \mathcal{T}$ is the associated triangulation.

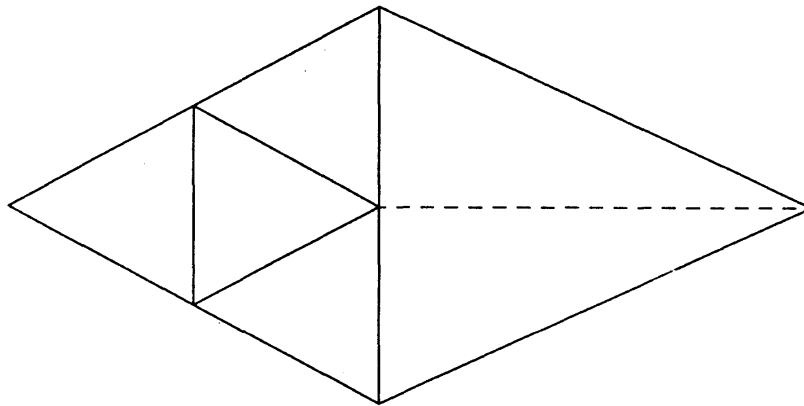


Figure 8 Completion of triangulation by irregular or “green” refinement.

Otherwise, the local error estimator is once more activated leading to a new set of marked edges. Next, green edges are removed, replacing two green triangles by one red triangle. Then those triangles are refined regularly, which have at least one marked red edge. To close the triangulation, first red refinements as shown in Figure 7 and Figure 9 are performed. Finally, green refinements complete the triangulation. The whole refinement process just described must be repeated, until the test (3.4) is passed. After a finite number of steps the cascade level $i = 1$ is established and the triangulation \mathcal{T}^1 is defined.

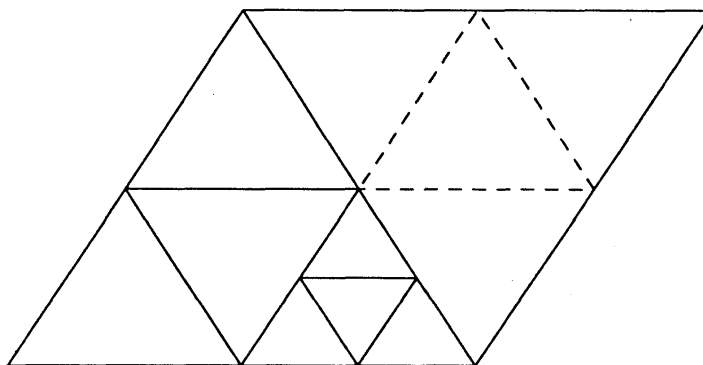


Figure 9 Additional red refinement.

The above techniques describe in sufficient detail the general transition from triangulation \mathcal{T}^i on cascade level i to triangulation \mathcal{T}^{i+1} on cascade level $i + 1$.

Hierarchical basis. Given a triangulation \mathcal{T} obtained by the above mesh refinement process (intermediate level or cascade level), now the hierarchical basis of the associated finite element space S is constructed. The hierarchical basis is needed for both the evaluation of the error estimator

(see Section 2) and the realization of the preconditioned cg-iteration (see Section 1 for uniformly refined meshes). The problem is to find an appropriate hierarchy of subspaces of S , defined by the corresponding hierarchy of triangulations.

Independent of any history of the adaptive process, the triangulation \mathcal{T} is understood as the last member \mathcal{T}_j of a hierarchy of nested triangulations $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \dots$ which is uniquely determined by \mathcal{T} and the triangulation \mathcal{T}^0 establishing the initial cascade level. \mathcal{T}_0 is identical to this triangulation \mathcal{T}^0 . In the transition from \mathcal{T}_k to \mathcal{T}_{k+1} triangles are either kept or are refined as shown in Figure 5 and Figure 6, and in Figure 8, respectively. The triangles of \mathcal{T}_0 are the *triangles of level 0*. The *triangles of level $k+1$* are created by a red or a green refinement of a triangle of level k . To get a unique decomposition of \mathcal{T} , only triangles of level k in \mathcal{T}_k are allowed to be refined in the transition to \mathcal{T}_{k+1} . As a result of this construction, the finite element space S_k associated with \mathcal{T}_k is a subspace of S_{k+1} . With $\mathcal{T} = \mathcal{T}_j$, j is the *depth of the triangulation \mathcal{T}* .

It should be remarked that in the case of uniform refinement (as treated in Section 1) the cascade triangulations $\mathcal{T}^0, \mathcal{T}^1, \dots, \mathcal{T}^i$ are identical to the internal triangulations $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_i$ associated with \mathcal{T}^i . For the general case of nonuniform refinement, however, the depth j_i of triangulation \mathcal{T}^i of cascade level i may be greater than i . (A more precise notation would require double indices!).

The vertices of the triangles in \mathcal{T}_0 are the vertices or *nodes of level 0*. The vertices created by the refinement of a level k triangle are the vertices or *nodes of level $k+1$* ; see Figure 10. Note that the definition of the level of a node is unique because green edges are never refined during the construction of the internal triangulations.

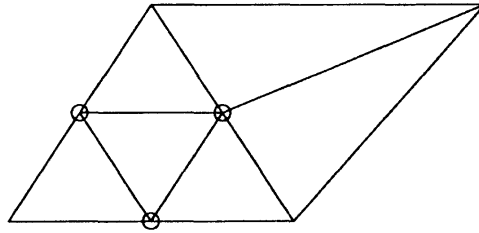


Figure 10 The level $k+1$ nodes created by the refinement of two level k triangles.

The subspace \mathcal{V}_{k+1} of S_{k+1} is spanned by the nodal basis functions of S_{k+1} associated with level $k+1$ nodes. Corresponding to (1.5)

$$S_{k+1} = S_k \oplus \mathcal{V}_{k+1} \quad (3.5)$$

holds. The hierarchical basis of S_0 is the nodal basis of this space. The hierarchical basis of S_{k+1} consists of the hierarchical basis functions of S_k and the nodal basis functions of S_{k+1} spanning \mathcal{V}_{k+1} .

With the definition of the hierarchical basis of $S = S_j$ corresponding to the triangulation $\mathcal{T} = \mathcal{T}_j$ the definition of the preconditioner for the case of nonuniform mesh refinements is also established. As in the case of uniform refinement a condition number estimate like (1.12) holds, now with i replaced by the depth of \mathcal{T} . For details compare [12] and [5].

Termination Criteria for the Iterative Solver. Assume that an approximate solution $\tilde{U}_{i-1} \in S^{i-1}$ of problem (1.2) is given, which has discretization error accuracy, i.e.

$$\|\tilde{U}_{i-1} - U_{i-1}\| \lesssim \|U_{i-1} - U\| \quad (3.6)$$

in the notation of Section 1. This relation certainly holds for $i = 1$ (direct solution on cascade level 0). The aim is now to compute the approximate solution $\tilde{U}_i \in S^i$ on the already given triangulation \mathcal{T}^i . First, an initial approximation $U_i^{(0)} \in S^i$ is determined. For this purpose, one merely keeps all hierarchical basis coefficients associated with \tilde{U}_{i-1} and sets the new coefficients to zero. Then the conjugate gradient method is applied to the preconditioned form corresponding to (1.15) of the arising linear system. From (1.26), the ideal termination criterion for the iteration would be

$$\|\tilde{U}_i - U_i\| \leq \bar{\epsilon}_i \|\tilde{U}_{i-1} - U\| . \quad (3.7)$$

As in Section 1, an efficient choice of the parameter $\bar{\epsilon}_i$ deserves special consideration. The principle transferred from Section 1 is that the $\bar{\epsilon}_i$ should reflect the best possible behavior of the discretization error. For an H^2 -regular problem and quasi-uniform meshes the discretization error is well-known to behave like $\mathcal{O}(h)$, which means

$$\|U_i - U\| \sim 1/n_i^{1/2} . \quad (3.8)$$

Using conformally mapped uniform meshes one can show that this approximation order can be reached for a fairly general class of problems. This leads to the choice

$$\bar{\epsilon}_i := \rho \cdot (n_{i-1}/n_i)^{1/2} , \quad 0 < \rho < 1 , \quad (3.9)$$

wherein the safety factor ρ takes care of the considerations of Section 1 and of the estimation of the norms arising in (3.7) as described in Section 2. In the present version of KASKADE the value $\rho := 1/2$ is selected (compare

also (2.30)). With the choice (3.9) of $\bar{\varepsilon}_i$ (3.7) leads to

$$\begin{aligned} \|\tilde{U}_i - U\|^2 \leq & \|U_i - U\|^2 + \rho^2 \frac{n_{i-1}}{n_i} \|U_{i-1} - U\|^2 + \\ & + \dots + \rho^{2i} \frac{n_0}{n_i} \|U_0 - U\|^2 \end{aligned} \quad (3.10)$$

which corresponds to (1.27). Therefore efficiency would only be lost for finite element approximations behaving better than (3.8).

Complexity Estimate. In order to estimate the complexity of this procedure, first one needs some knowledge about the growth of the number of unknowns from one cascade level to the next. Let n_i be the number of vertices of the triangulation \mathcal{T}^i . Since \mathcal{T}^i , $i \geq 1$, has to pass the test (3.4), one has

$$s \leq \frac{n_i}{n_{i-1}}, \quad i \geq 1. \quad (3.11)$$

To get an upper bound for this quotient assume that \mathcal{T} is the last intermediate triangulation before \mathcal{T}^i . The number n of vertices of \mathcal{T} does not pass the test (3.4). Thus

$$n < s \cdot n_{i-1}.$$

After the green edges of \mathcal{T} have been removed, every remaining triangle is subdivided at most once into four red triangles. Therefore in the transition from \mathcal{T} to \mathcal{T}^i the number of newly created vertices is bounded by the number of edges of \mathcal{T} . Let γ_0 denote the maximal number of edges meeting at a vertex of \mathcal{T}^0 . Then at most $\gamma = \max(\gamma_0, 6)$ edges meet at a vertex of \mathcal{T} . Therefore

$$n_i - n \leq \frac{\gamma}{2} \cdot n,$$

and finally

$$\frac{n_i}{n_{i-1}} \leq \left(1 + \frac{\gamma}{2}\right) \cdot s \quad (3.12)$$

follows.

Corresponding to (1.33), the number of preconditioned cg-iterations on cascade level i necessary to achieve

$$\|\tilde{U}_i - U_i\| \leq \bar{\varepsilon}_i \|U_i^{(0)} - U\| \quad (3.13)$$

can be estimated by

$$m_i(\bar{\varepsilon}_i) \leq C_3 \cdot (j_i + 1) \cdot \left| \log \frac{\bar{\varepsilon}_i}{2} \right|. \quad (3.14)$$

Herein $j_i \geq i$ denotes the depth of the triangulation \mathcal{T}^i . Recalling the construction of $U_i^{(0)} \in \mathcal{S}^i$ from $\tilde{U}_{i-1} \in \mathcal{S}^{i-1}$, one can assume that

$$\|U_i^{(0)} - U\| \doteq \|\tilde{U}_{i-1} - U\|. \quad (3.15)$$

Thereby (3.14) becomes an estimate for the number of iterations needed to satisfy (3.7). Combining the definition (3.9) of $\bar{\varepsilon}_i$ and (3.12) one has

$$\rho \left(\frac{1}{1 + \gamma/2} \cdot \frac{1}{s} \right)^{1/2} \leq \bar{\varepsilon}_i < 1, \quad (3.16)$$

which leads to

$$m_i(\bar{\varepsilon}_i) \leq C_3^* \cdot (j_i + 1) \quad (3.17)$$

for properly defined C_3^* . With n_i the number of vertices associated with \mathcal{T}^i , the work estimate corresponding to (1.21) comes out to be

$$\alpha_i(\bar{\varepsilon}_i) \leq C_4^* \cdot n_i \cdot (j_i + 1). \quad (3.18)$$

Let \bar{i} denote the final cascade level, $j = j_{\bar{i}}$ the associated depth of the final triangulation $\mathcal{T}^{\bar{i}}$ and $n = n_{\bar{i}}$ the number of vertices of this triangulation. Then the total amount of *iteration work* $\tilde{\alpha}$ can be estimated by

$$\tilde{\alpha} \leq \sum_{i=0}^{\bar{i}} \alpha_i(\bar{\varepsilon}_i). \quad (3.19)$$

Insertion of (3.16) and condition (3.11) lead to

$$\begin{aligned} \frac{\tilde{\alpha}}{n} &\leq \sum_{i=0}^{\bar{i}} C_4^* \cdot \frac{n_i}{n} \cdot (j_i + 1) \leq \\ &\leq \sum_{i=0}^{\bar{i}} C_4^* \cdot \left(\frac{1}{s} \right)^{\bar{i}-i} \cdot (j + 1) \end{aligned}$$

Summing up, one obtains

$$\tilde{\alpha} \leq C_4^* \cdot \frac{s}{s-1} \cdot n \cdot (j + 1). \quad (3.20)$$

For the choice $s := 2$ made in the code KASKADE, the result (3.20) estimates the total amount of iteration work by roughly the double amount of iteration work on the finest cascade level.

In addition to the iteration work, the work for building the local element matrices and right-hand sides (associated with the individual triangles), for the computation of the local error estimates and for the administration of the triangulations needs to be considered. For a clever arrangement of these computations the involved amount of additional work can be bounded proportional to the number of triangles on the final cascade level. The amount of work for selecting the edges according to (3.2) can be neglected from a practical point of view.

In actual computation, the unavailable terms on both sides of (3.7) are replaced according to

$$\|\tilde{U}_i - U_i\|^2 \rightarrow |\bar{D}_i^{-1/2} r_i|^2 \quad (3.21)$$

$$\|\tilde{U}_{i-1} - U\|^2 \rightarrow |D_{LL,i-1}^{-1/2} r_{L,i-1}|^2 + |D_{QQ,i-1}^{-1/2} r_{Q,i-1}|^2 . \quad (3.22)$$

Herein, the notation in (3.21) agrees with the notation of Section 1, the notation in (3.22) is an adaptation of the notation of Section 2. Both estimates (3.21) and (3.22) represent the preconditioned residual in the associated scaled hierarchical basis.

Assume that for some iterate on cascade level i , the condition (3.7) with the replacements (3.21), (3.22) and the choice (3.9) for $\bar{\epsilon}_i$ holds. Then the discretization error estimator described in Section 2 is called on level i . In order to mimic the computationally unavailable condition (3.6), one checks for the substitute condition

$$|D_{LL,i}^{-1/2} r_{L,i}| \leq \bar{\rho} \cdot \tau, \quad 0 < \bar{\rho} < 1, \quad (3.23)$$

where

$$\tau = \left(|D_{LL,i}^{-1/2} r_{L,i}|^2 + |D_{QQ,i}^{-1/2} r_{Q,i}|^2 \right)^{1/2} \quad (3.24)$$

is the current estimate for the discretization error. If this condition holds, the refinement procedure is called. If (3.23) does not hold, one keeps τ fixed and continues the iteration on level i until

$$|D_{LL,i} r_{L,i}| \leq \tilde{\rho} \cdot \tau, \quad 0 < \tilde{\rho} < \bar{\rho}, \quad (3.25)$$

is satisfied. After a new call of the error estimator refinement begins. In the present version of KASKADE the values $\bar{\rho} = 1/2$ and $\tilde{\rho} = 1/4$ are used. It may be worth mentioning that in the examples tested so far a restart of the cg-iteration never occurred.

The presentation up to now ignored the question which kind of data structure is needed to handle the algorithm in an efficient way. Note that a careless treatment of this question might easily lead to a computational amount behaving worse than the estimate (3.20). A detailed presentation of this important part of the development would be far beyond the scope of the present paper and is left to [9].

4. A Numerical Experiment

The above described code KASKADE has been applied to a number of examples including those with re-entrant corners (such as L-shaped regions), discontinuous Dirichlet boundary conditions or discontinuous coefficients (material constants). As it turns out, the basic performance of the code can be sufficiently illustrated by a documentation of a well-known nasty test problem [2,5]. Consider Laplace's equation

$$-\Delta u = 0 \quad (4.1)$$

on a circle Ω of radius 1 (centered at the origin) having a crack along the positive x -axis. Homogeneous Dirichlet (Neumann) boundary conditions are imposed on the top (bottom) of the crack, giving a singular solution with leading term

$$u = r^{1/4} \sin(\varphi/4) . \quad (4.2)$$

This function is imposed as Dirichlet boundary condition on the remaining part of the boundary, yielding (4.2) as exact solution. The associated contour plot of (4.2) is given in Figure 11.

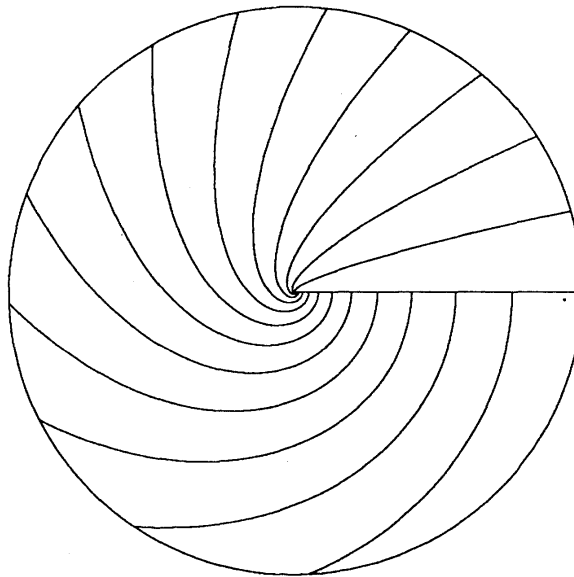


Figure 11 Contour map of the solution $u = r^{1/4} \sin(\varphi/4)$ as obtained by KASKADE.

The initial triangulation \mathcal{T}^0 used to start KASKADE is presented in Figure 12. Further triangulations as obtained by the code are documented in Figures 13-15.

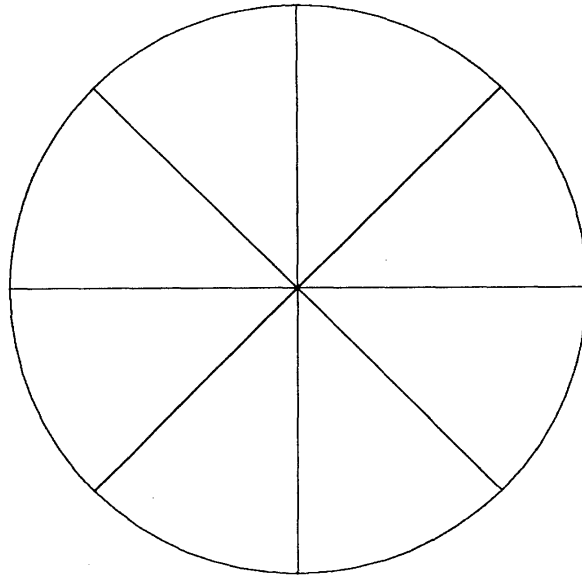


Figure 12 The initial triangulation \mathcal{T}^0 (10 nodes).

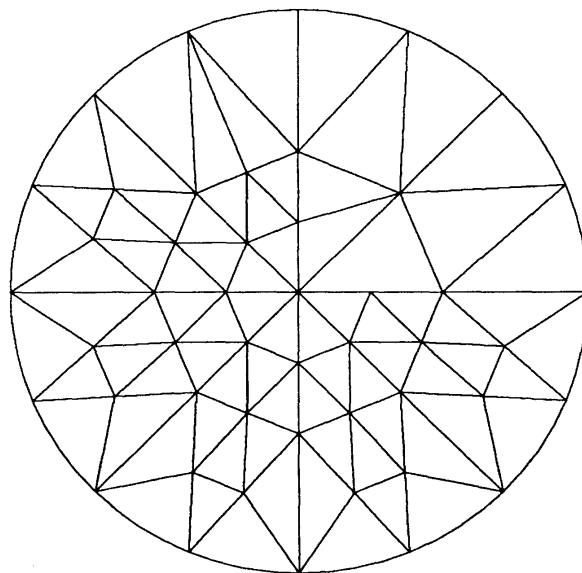


Figure 13 Triangulation \mathcal{T}^2 (50 nodes, depth 2)

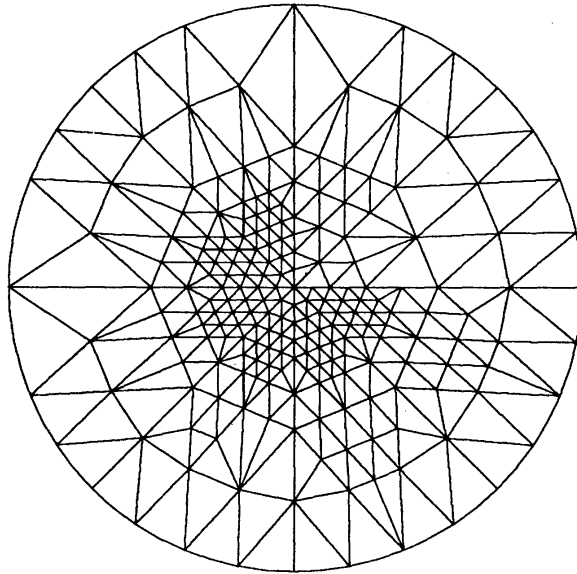


Figure 14 Triangulation 7^3 (218 nodes, depth 4)

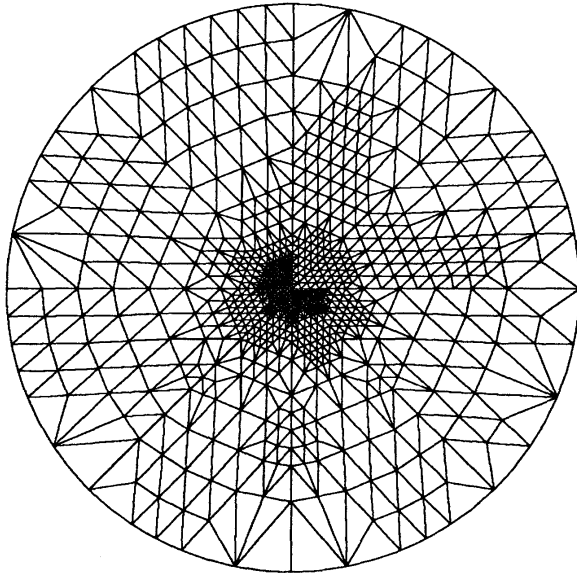


Figure 15 Triangulation 7^4 (769 nodes, depth 6)

This documentation illustrates the efficiency of the *mesh refinement strategy* of KASKADE. The actual performance of the suggested *error estimator* is illustrated by the convergence history in Figure 16. For reasons of a clear presentation, each approximate solution on the grids $\mathcal{T}^0, \dots, \mathcal{T}^5$ is embedded into the piecewise linear finite element space associated with the final triangulation \mathcal{T}^6 by setting appropriate hierarchical basis coefficients to zero. The quantity DISTL is defined as the distance between the embedded approximate solution on \mathcal{T}^i , $i = 0, \dots, 6$, and the exact piecewise *linear* finite element solution corresponding to \mathcal{T}^6 . The quantity DISTQ is defined as the equivalent distance to the exact piecewise *quadratic* solution corresponding to \mathcal{T}^6 . Both distances are measured in the energy norm. As the space of piecewise linear functions is a subspace of the space of piecewise quadratic functions, DISTL is always less than DISTQ. The *estimated error* ERR-EST is defined by formula (2.15). It should roughly behave like DISTQ. Each dotted vertical line indicates the appearance of a new cascade level (here up to $j = 6$).

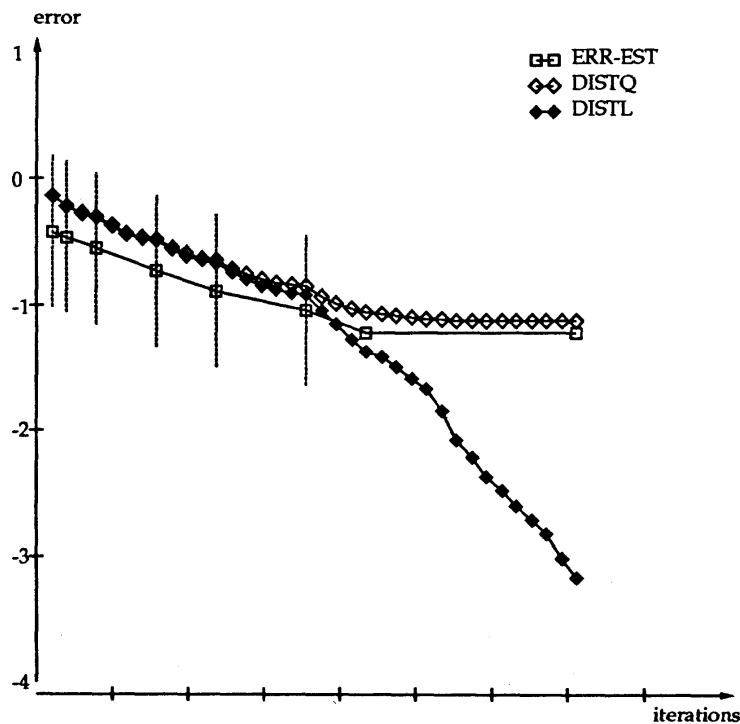


Figure 16 The convergence history of the cascade (logarithmic scale).

From Figure 16, the following observations can be made:

- The quantity ERR-EST is only a slight underestimate of the quantity DISTQ - thus nicely confirming our error estimation device (Section 2).
- On the final cascade level ($j = 6$), the iteration would normally terminate after 4 steps. Further iterations, which have deliberately been added for illustration purposes, do not "improve" the solution - a fact, which nicely demonstrates the efficiency of our termination criterion (3.7)/(3.9).
- The number of iterations necessary on each cascade level remains small (not greater than 6).

The total computing time for this problem on a Macintosh II work station using the MPW PASCAL compiler is given in Table 1.

Cascade level i	Depth j_i	Nodes n_i	CPU-Time t [sec]	$(t/n_i) \cdot 100$
0	0	10	0.02	0.20
1	1	23	0.20	0.86
2	2	50	0.52	1.04
3	4	218	2.88	1.32
4	6	769	10.23	1.33
5	8	2774	42.00	1.51
6	10	6156	98.20	1.60

Table 1

Summary. This test problem and a series of further examples illustrate the efficiency (robustness, reliability, speed) of our adaptive hierarchical finite element code KASKADE. Independent of the special implementation, the authors feel that with this approach a new door has been opened towards a more efficient treatment of elliptic PDE's on arbitrary 2-D domains arising in science and engineering applications.

Acknowledgments. The authors want to thank R. Roitzsch for his invaluable computational assistance and many helpful discussions. The third author was supported by a Konrad Zuse Fellowship.

References

- [1] O. Axelsson, V.A. Barker: *Finite Element Solution of Boundary Value Problems: Theory and Computation*. New York: Academic Press (1984).
- [2] R.E. Bank: *A-posteriori error estimates, adaptive local mesh refinement and multigrid iteration*. In: W. Hackbusch, U. Trottenberg (eds.): *Multigrid Methods II*. Springer Verlag, Berlin, Heidelberg, New York: Lect. Notes Math. Vol 1228, p. 7-23 (1986).
- [3] R.E. Bank: *PLTMG User's Guide, Edition 5.0*. Technical Report, Department of Mathematics, University of California at San Diego (1988).
- [4] R.E. Bank, T. Dupont: *An Optimal Order Process for Solving Finite Element Equations*. Math. Comp. 36, p. 35-51 (1981).
- [5] R.E. Bank, T. Dupont, H. Yserentant: *The Hierarchical Basis Multigrid Method*. Numer. Math. 52, p. 427-458 (1988).
- [6] R.E. Bank, A.H. Sherman, A. Weiser: *Refinement Algorithms and Data Structures for Regular Local Mesh Refinement*. Scientific Computing, R. Stepleman et al (eds.), Amsterdam: IMACS/North Holland p. 3-17 (1983).
- [7] R.E. Bank, A. Weiser: *Some a-posteriori Error Estimators for Elliptic Partial Differential Equations*. Math. Comp. 44, p. 283-301 (1985).
- [8] W. Hackbusch: *Multigrid Methods and Applications*. Berlin, Heidelberg, New York: Springer (1985).
- [9] P. Leinen: *Work done in preparation of a dissertation*.
- [10] Ch. K. Mesztenyi, W.C. Rheinboldt: *NFEARS: A Nonlinear Adaptive Finite Element Solver*. Report ICMA-87-113 Department of Mathematics and Statistics, University of Pittsburgh (1987).
- [11] K. Stüben, U. Trottenberg: *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis, and Applications* In: *Multigrid Methods* (W. Hackbusch, U. Trottenberg, eds.), Lect. Notes Math. 960, Berlin, Heidelberg, New York: Springer Verlag p. 1-176 (1982).
- [12] H. Yserentant: *On The Multi-Level Splitting of Finite Element Spaces*. Numer. Math. 49, p. 379-412 (1986).

-
- [13] H. Yserentant: *Preconditioning Indefinite Discretization Matrices*. To appear in Numer. Math.
- [14] O.C. Zienkiewicz, A. Craig: *Adaptive Refinement, Error Estimates, Multigrid Solution, and Hierarchic Finite Element Method Concepts*. In: Accuracy Estimates and Adaptive Refinements in Finite Element Computations, Chapter 2 (edited by I. Babuska, O. C. Zienkiewicz, J. Gago, and E.R. de A. Oliveira) J. Wiley and Sons Ltd. (1986).

Appendix

This section contains a documentation of further test problems. For each example, the initial triangulation \mathcal{T}^0 and the final triangulation \mathcal{T}^j , a contour plot of the solution on \mathcal{T}^j and the convergence history (as in Figure 13) are presented.

The following examples are arranged:

- A.1 The Notorious Model Problem
- A.2 L-shaped Region
- A.3 Discontinuous Coefficients
- A.4 Discontinuous Dirichlet Boundary Conditions

The computing times given in the subsequent tables refer to a Macintosh II work station using the MPW PASCAL compiler.

A.1 The Notorious Model Problem

For the sake of completeness, the standard model problem

$$\begin{aligned} -\Delta u &= 1 \\ u|_{\partial\Omega} &= 0 \end{aligned}$$

for $\Omega = [0, 1]^2$ is included. In this example, the discretization error estimate is nearly exact – see Figure 4. Moreover, note that KASKADE performs just as in the case of the L-shaped region (Example A.2).

Cascade level i	Depth j_i	Nodes n_i	CPU-Time t [sec]	$(t/n_i) \cdot 100$
0	0	13	0.03	0.23
1	1	41	0.27	0.66
2	2	137	1.12	0.82
3	3	485	4.63	0.95
4	4	1685	18.38	1.09
5	5	5377	62.38	1.16

Table 2

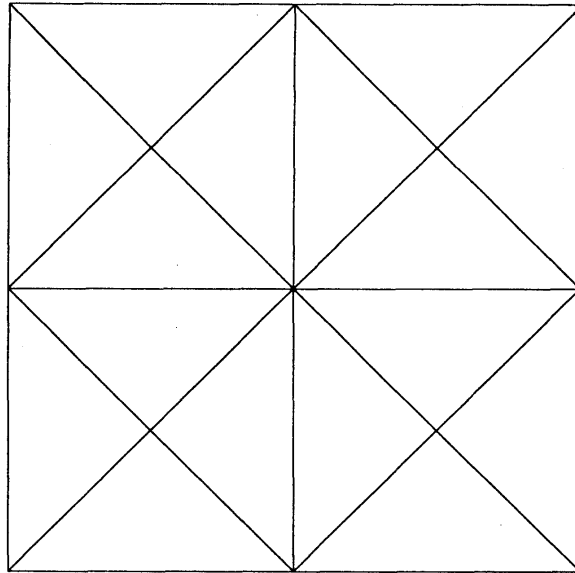


Figure 1 Initial triangulation for Example A.1.

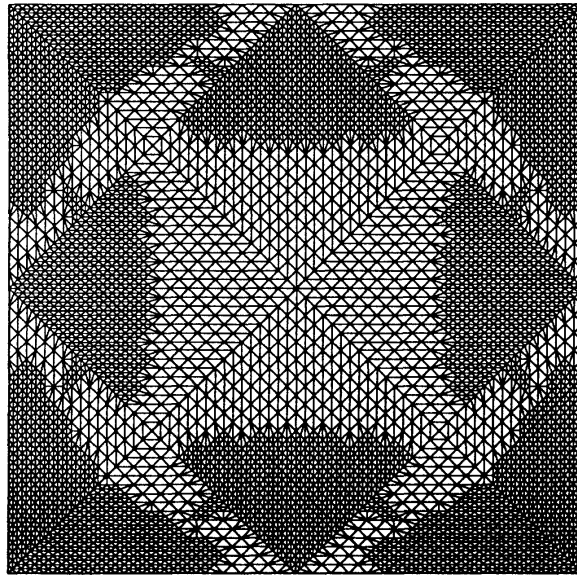


Figure 2 Final triangulation (level 5, depth 5) for Example A.1.

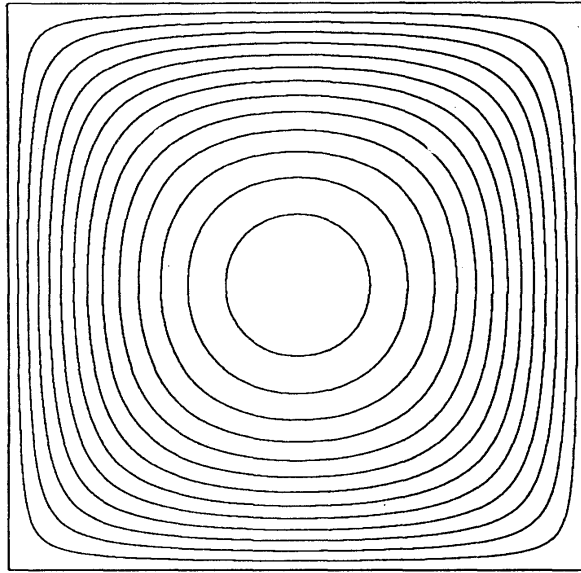


Figure 3 Contour plot of solution of Example A.1 on final triangulation as obtained from KASKADE.

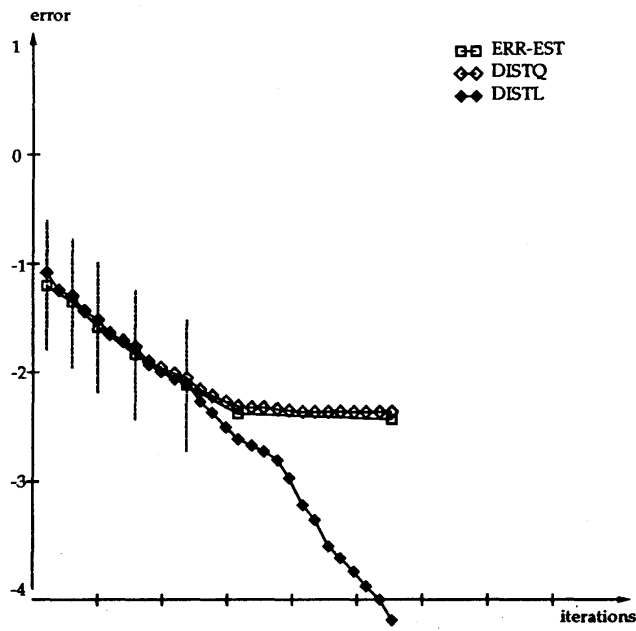


Figure 4 Convergence history for Example A.1.

A.2 L-shaped Region

In many engineering applications, re-entrant corners arise. A typical example is

$$\begin{aligned} -\Delta u &= 1 \\ u|_{\partial\Omega} &= 0 \end{aligned}$$

with $\Omega = [0, 1]^2 \setminus [\frac{1}{2}, 1]^2$.

Once more, one observes that the error estimate is nearly exact! Moreover, compare the convergence behavior with the behavior in the standard problem A.1.

Cascade level i	Depth j_i	Nodes n_i	CPU-Time t [sec]	$(t/n_i) \cdot 100$
0	0	8	0.02	0.25
1	1	21	0.10	0.48
2	2	65	0.45	0.69
3	3	208	1.68	0.80
4	4	645	5.75	0.89
5	5	1876	19.28	1.03
6	6	4993	51.07	1.02

Table 3

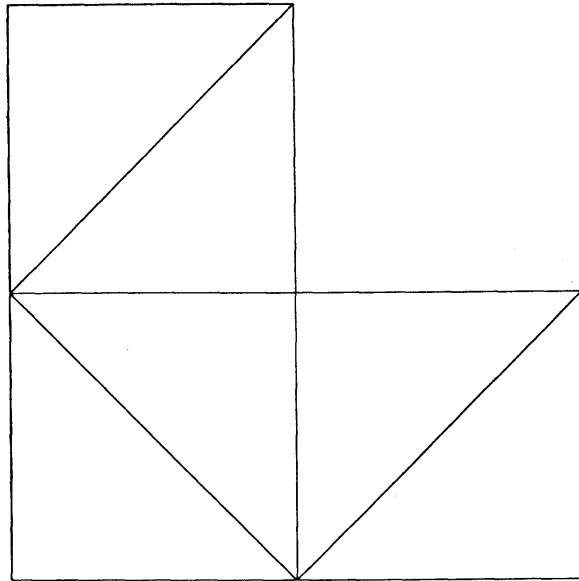


Figure 5 Initial triangulation for Example A.2.

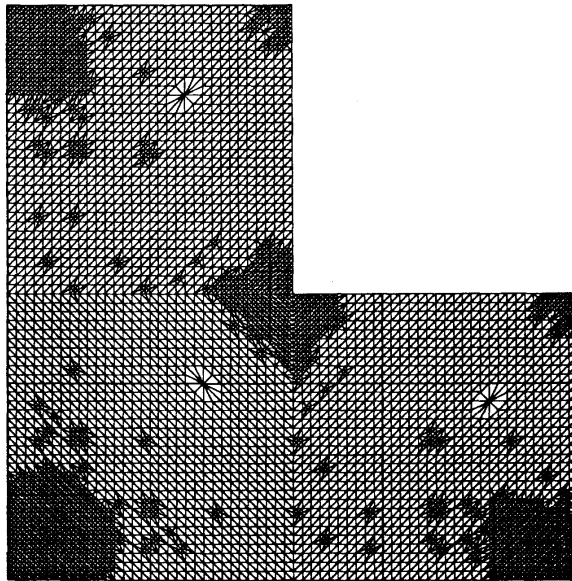


Figure 6 Final triangulation (level 6, depth 6) for Example A.2.

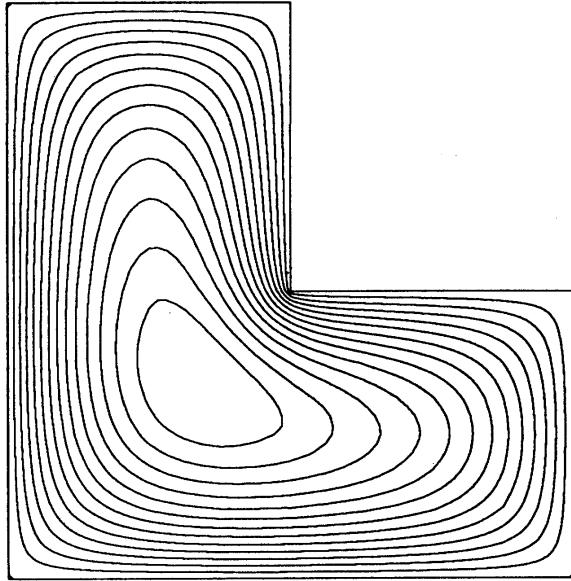


Figure 7 Contour plot of solution of Example A.2 on final triangulation as obtained from KASKADE.

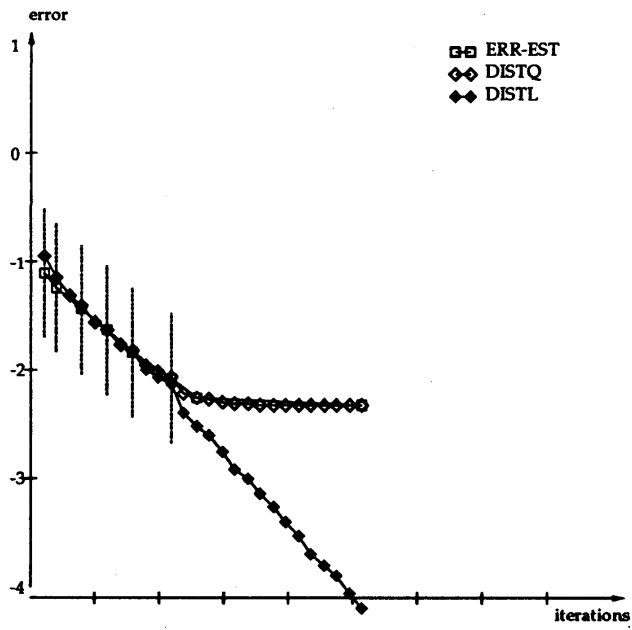


Figure 8 Convergence history for Example A.2.

A.3 Discontinuous Coefficients

This type of problem arises, whenever different materials are combined. As a model of such an occurrence, consider the example for $\Omega = [0, 1]^2$

$$\begin{aligned} -\nabla \cdot (\sigma \nabla u) &= 1 \\ u|_{\partial\Omega} &= 0 \end{aligned}$$

with

$$\sigma = \begin{cases} 1 & | \quad [0.25, 0.75]^2 \setminus [0.375, 0.625]^2 \\ 10^6 & | \quad \text{else} \end{cases}$$

Cascade level i	Depth j_i	Nodes n_i	CPU-Time t [sec]	$(t/n_i) \cdot 100$
0	0	25	0.10	0.40
1	1	45	0.35	0.78
2	2	145	1.47	1.01
3	3	521	5.42	1.04
4	4	1553	16.88	1.09
5	5	4017	45.72	1.14
6	6	10029	116.05	1.17

Table 4

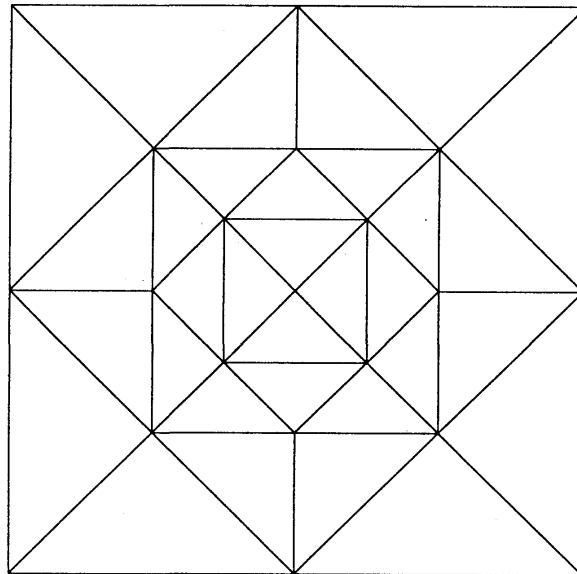


Figure 9 Initial triangulation for Example A.3.

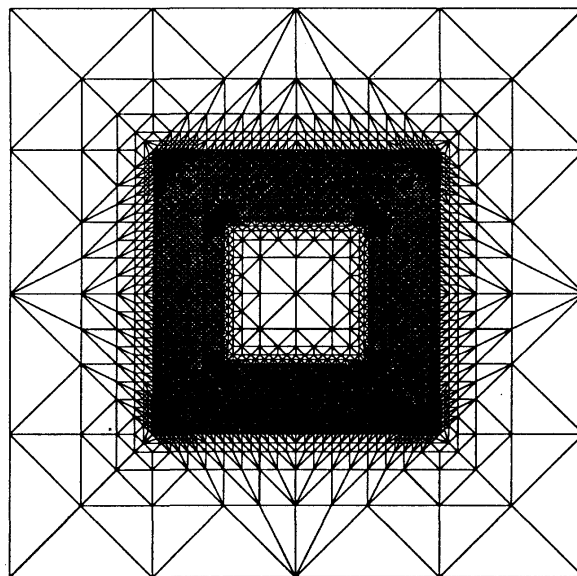


Figure 10 Final triangulation (level 6, depth 6) for Example A.3.

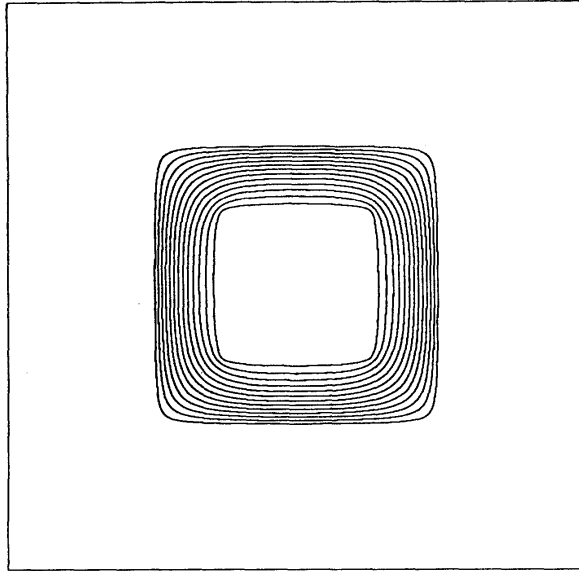


Figure 11 Contour plot of solution of Example A.3 on final triangulation as obtained from KASKADE.

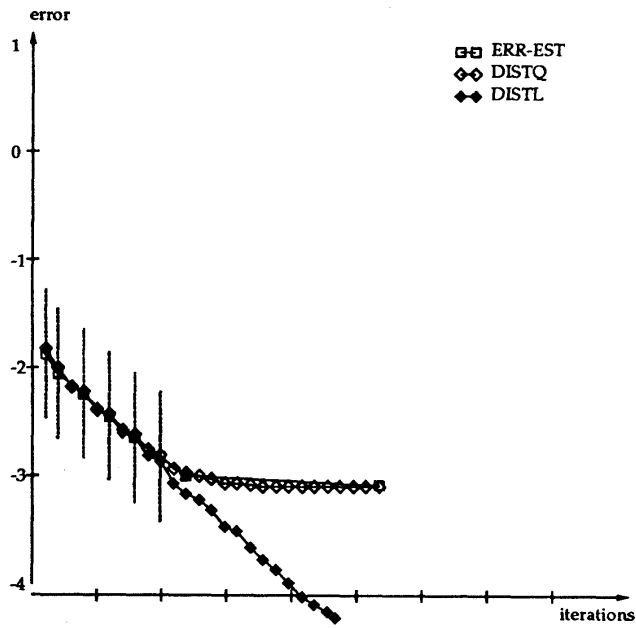


Figure 12 Convergence history for Example A.3.

A.4 Discontinuous Dirichlet Boundary Conditions

For $\Omega = [0, 1]^2$ consider Laplace's equation

$$\Delta u = 0$$

with boundary conditions

$$u = \begin{cases} 1 & x < \frac{1}{2} \wedge y < \frac{1}{2} \\ \frac{1}{2} & (x = 0 \wedge y = \frac{1}{2}) \vee (x = \frac{1}{2} \wedge y = 0) \\ 0 & \text{else} \end{cases}$$

The solution is known to be globally in $H^{1/2}(\Omega)$ – compare Lions/Magenes, Non-Homogeneous Boundary Value Problems and Applications I, p. 188-189, Theorem 7.4. The energy norms DISTQ and ERR-EST are asymptotically constant – because the energy norm is equivalent to the H^1 -norm. Nevertheless, the problem is satisfactorily solved, since only local ellipticity properties are required by KASKADE.

Cascade level i	Depth j_i	Nodes n_i	CPU-Time t [sec]	$(t/n_i) \cdot 100$
0	0	13	0.03	0.23
1	1	41	0.27	0.66
2	3	198	1.97	0.99
3	5	633	7.27	1.14
4	8	1654	22.52	1.36
5	11	3768	59.73	1.59
6	15	8217	143.95	1.75

Table 5

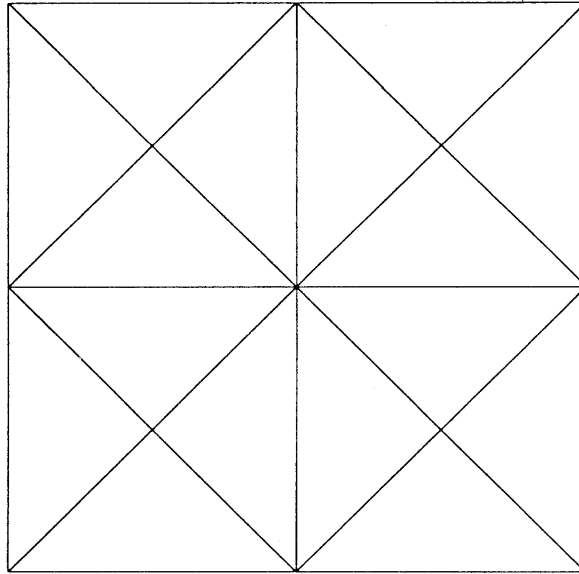


Figure 13 Initial triangulation for Example A.4.

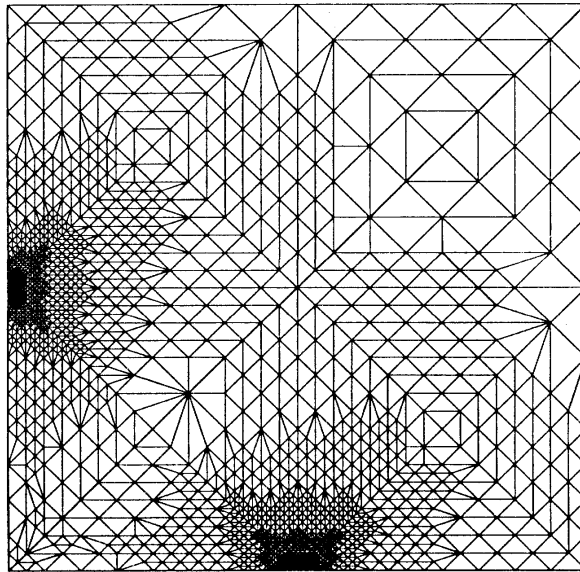


Figure 14 Final triangulation (level 6, depth 15) for Example A.4.

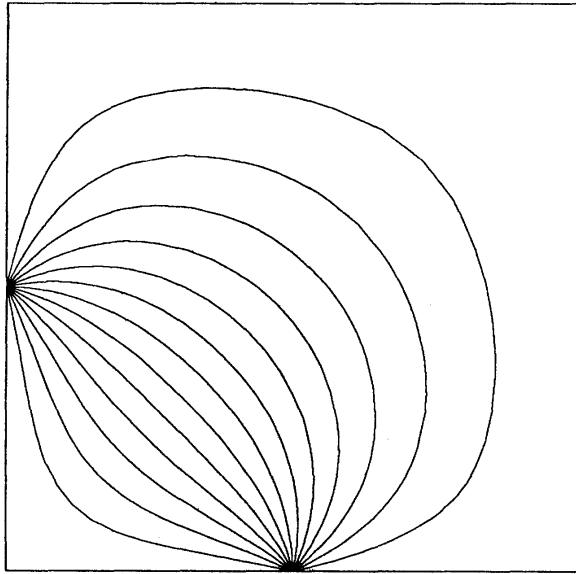


Figure 15 Contour plot of solution of Example A.4 on final triangulation as obtained from KASKADE.

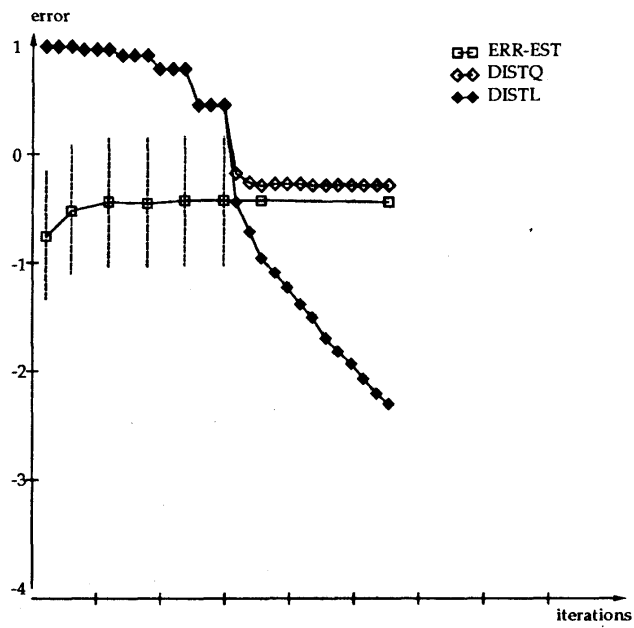


Figure 16 Convergence history for Example A.4.