Realization and comparison of various mesh refinement strategies near edges

Thomas Apel* Frank Milde†

August 1, 1994

Abstract

This paper is concerned with mesh refinement techniques for treating elliptic boundary value problems in domains with re-entrant edges and corners, and focuses on numerical experiments. After a section about the model problem and discretization strategies, their realization in the experimental code FEMPS3D is described. For two representative examples the numerically determined error norms are recorded, and various mesh refinement strategies are compared.

Contents

*Technische Universität Chemnitz-Zwickau, Fakultät für Mathematik, D-09107 Chemnitz
†Technische Universität Chemnitz-Zwickau, Fakultät für Physik, D-09107 Chemnitz

1 Introduction

This paper is concerned with mesh refinement techniques for treating elliptic boundary value problems in domains with re-entrant edges and corners. It is well known that a non-smooth boundary of the domain causes the regularity of the solutions of the problems to be low in comparison with that of the solution of smooth problems. As a result the approximation of the standard finite element solution of the problem deteriorates, and many specially adapted numerical methods have been developed in recent years, see for example [1, 2, 3, 5, 6, 10, 17, 19, 22, 23, 24]. In this paper we shall focus on mesh refinement strategies.

In Section 2 we give a short introduction into the field. For simplicity we study the Poisson equation with Dirichlet boundary conditions. In Subsection 2.1 the analytical behaviour of the exact solution is characterized. Then we give some basic information on the finite element discretization. In the last two subsections we review a-priori grading and an adaptive algorithm together with some results from the numerical analysis of these strategies. We remark that the analytical behaviour of the solution is similar for a large class of problems including the Lamé equation systems, the biharmonic equation and general boundary conditions. Mesh refinement strategies were also studied for more general problems, see for example [5].

Some variants of the algorithms were realized by the authors in the experimental code FEMPS3D at the Technische Universität Chemnitz-Zwickau. The aim of this paper is to describe the computational realization in more detail than in the more theoretical papers of the first author. This is done in Section 3 where we also discuss some difficulties connected with our realization. But we remark that most of the programming was done in order to investigate whether and how different mesh refinement stategies work. In order to derive approximation orders we are interested in the calculation of problems with as many degrees of freedom as possible, so we often compromised on memory and computation time in the sense of saving memory. We do not claim that our realization is optimal in any sense.

Another aim of this paper is to compare various strategies and realizations for the treatment of elliptic boundary value problems with boundary singularities. For this purpose we computed the corresponding approximation errors in the energy norm for two typical examples. The results are given in Sections 4 and 5. We can conclude that any of the proposed mesh refinement algorithms is better than computing without paying attention to the large error near concave edges and corners. It has been underlined that the proposed a-priori mesh grading algorithms have the optimal convergence order already on coarse meshes. For incorporating this knowledge into adaptive procedures, different proposals were investigated. We get the best results when an anisotropic grading is realized in each refinement step using a coordinate transformation. But the problem is to find the optimal transformation for the domain under consideration.

2 Treatment of elliptic problems with boundary singularities via the finite element method with mesh refinement

2.1 The model problem

Consider the Dirichlet problem for the Poisson equation

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega \\ u &= g \qquad \text{on } \partial\Omega \end{aligned} \tag{2.1}$$

in its weak formulation: Find $u \in V_*$ such that

$$a(u, v) = (f, v) \quad \text{for all } v \in V_0. \tag{2.2}$$

Here, we define $V_0 := \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, $V_* := \{v \in H^1(\Omega) : v|_{\partial\Omega} = g\}$, and $a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx$. We assume that $f \in L_2(\Omega)$, that g is trace of a $H^2(\Omega)$-function, and that the boundary $\partial\Omega$ is at least piecewise smooth.

The regularity of the solution u of (2.2) is determined by the properties of the domain $\Omega \in I\!R^d$, $d = 2,3$. If Ω is a smooth or a convex domain then $u \in H^2(\Omega)$. But this is no longer true, when the domain Ω contains re-entrant corners or edges.

Consider first the two-dimensional case and let $x_0 \in \partial\Omega$ be a boundary point and $\omega \in (\pi, 2\pi)$ the internal angle at this point. For simplicity assume that $\partial\Omega$ is polygonal near x_0. Introduce polar coordinates (r,φ) in the neighbourhood $U := \{\underline{x} \in I\!R^2 : |\underline{x} - \underline{x}_0| \le R_0\}$. Then the solution can be represented by

$$u = \xi(r)\,\gamma\, r^\lambda \sin\lambda\varphi \,+\, u_r, \tag{2.3}$$

where $\xi(.)$ is a smooth cut-off function, γ and $\lambda = \frac{\pi}{\omega} \in (0,1)$ are real numbers, and $u_r \in H^2(\Omega)$ is the regular part of the solution.

In three dimensions, the irregular boundary points are classified as conical corners, edges and polyhedral corners. Near edges we have the same representation formula (2.3). However, r is the distance to the edge, and the coefficient γ is no longer constant. In general the function γ is dependent on all three spatial variables. But under the assumption that not only the right hand side f, but also its derivatives $\frac{\partial f}{\partial z}$ and $\frac{\partial^2 f}{\partial z^2}$ are contained in $L_2(\Omega)$, then γ is just a function of z. Here, we denoted by z the coordinate in direction of the edge.

In the case of polyhedral corners we have a superposition of corner and edge singularities. The additional terms arising from the corners have also a representation in analogy to (2.3). The coefficient γ is constant, r is the distance to the corner, but the function of the spherical angles is more complicated; it can even have singularities itself. Most important for our purposes is that the smoothness properties of this term is again characterized by a real number λ, which is here the smallest eigenvalue of the Laplace-Beltrami-Operator on the intersection of Ω and the unit-sphere centered at the corner.

For more general problems and for a more detailed study of the regularity of the solutions we refer the reader to the books of Grisvard [14] and Kufner/Sändig [15] or to the review in [5, Section 2].

2.2 The finite element discretization

Assume for the moment that the domain Ω is a polygon in $I\!R^2$ or a polyhedron in $I\!R^3$. Then we consider a family of partitions $\mathcal{T}_h$ of $\overline{\Omega}$ with the usual regularity properties; see for example [11]:

(a) $\overline{\Omega} = \bigcup_{i=1}^m \overline{\Omega}_i$, where Ω_i are simplices in $I\!R^d$ (triangles or tetrahedra),

(b) $\Omega_i \cap \Omega_j = \emptyset \quad$ for $i \ne j$,

(c) any edge (for $d = 2,3$) or face (for $d = 3$) of Ω_i is either a subset of $\partial\Omega$ or an edge or face of another Ω_j.

In the standard finite element method there are two additional assumptions. First it is acquired that the aspect ratio h_i/ϱ_i, which is defined as the quotient of the diameter h_i of Ω_i and the diameter ϱ_i of the largest inner ball of Ω_i, is bounded: There exists a constant σ independent of $\mathcal{T}_h$ with

(d) $\frac{h_i}{\varrho_i} < \sigma \quad$ for all $\Omega_i \in \mathcal{T}_h$.

The second assumption is that all elements are approximately of the same size: There exist constants $\underline{C}$ and $\overline{C}$ such that

(e) $\underline{C}h \le h_i \le \overline{C}h \quad$ for all $\Omega_i \in \mathcal{T}_h$.

Note that condition (e) yields that the number m of elements and the number n of vertices are of the order h^{-d}.

Using this partition we introduce a finite-dimensional space $V_{0h} \subset V_0$ and a manifold $V_{*h} \subset V_*$ of continuous functions such that the restriction of any function from V_{0h} or V_{*h} to an element Ω_i is a polynomial of first degree. In analogy to (2.2) the finite element solution $u_h \subset V_{*h}$ is now defined by

$$a(u_h, v_h) \;=\; (f, v_h) \quad \text{for all } v_h \in V_{0h}. \tag{2.4}$$

It has been proved that the approximation error $u - u_h$ can be estimated by

$$\|u - u_h; H^1(\Omega)\| \;\le\; Ch^\alpha \|f; L_2(\Omega)\|, \tag{2.5}$$

where $\alpha = 1$ for a convex domain, $\alpha = \lambda$ for a two-dimensional domain with a re-entrant corner and $\alpha = \lambda - \varepsilon$ (ε is an arbitrarily small positive real number) for a three-dimensional domain with corners and/or edges, for λ see Subsection 2.1 [2, 5, 17].

If the boundary is not polygonal/polyhedral, the domain Ω is in general approximated by a polyhedral domain Ω_h. This can be done in different ways; the crucial point is that the domain error $(\Omega \setminus \Omega_h) \cup (\Omega_h \setminus \Omega)$ has to be contained in a boundary strip with a diameter of order h^2. The consequences are that additional terms have to be estimated in order to get the approximation result (2.5), see [2, 17, 28] for details from different points of view.

Note further that in general the assembly of the system of equations for determining the finite element solution requires numerical integration, at least for the right hand side. The solution of this system introduces another error. In order to get the error estimate (2.5) it is necessary that these error contributions are of lower order, see for example [11] for the error analysis.

2.3 A-priori mesh grading

Because of the practical importance of problems with boundary singularities it has been necessary to develop adapted numerical methods which yield error estimates of the same quality as for problems with a regular solution. Here, we shall focus on finite element methods involving a-priori local mesh grading. The idea is to use the knowledge about the singular solution to determine a relation between the size of the elements and their distance to the corner or edge.

This approach was first investigated in the two-dimensional case [6, 17, 19] and it turned out that the condition (e) from Subsection 2.2 should be replaced in a refinement neighbourhood $U := \{\underline{x} \in I\!R^2 : |\underline{x} - \underline{x}_0| \le R_0\}$ around the corner $\underline{x}_0 = (x_0, y_0)$ by the condition

$$\text{(e')} \quad \begin{array}{rcccll} \underline{C}_1 h^{1/\mu} & \le & h_i & \le & \overline{C}_1 h^{1/\mu} & \text{if } \ \underline{x}_0 \in \overline{\Omega}_i, \\ \underline{C}_2 h r_i^{1-\mu} & \le & h_i & \le & \overline{C}_2 h r_i^{1-\mu} & \text{if } \ \underline{x}_0 \notin \overline{\Omega}_i. \end{array}$$

By $r_i := \operatorname{dist}(\Omega_i, x_0) := \sup_{\underline{x} \in \Omega_i} |\underline{x} - \underline{x}_0|$ we denoted the distance of the element Ω_i to the point $\underline{x}_0$.

For $\mu < \lambda$ it has been proved that the error estimate (2.5) holds with $\alpha = 1$. The easiest way to construct such a mesh is to generate a standard (ungraded) mesh and to move the nodes from U via the coordinate transformation

$$\begin{aligned} r &:= \sqrt{(x - x_0)^2 + (y - y_0)^2}, \\ x &:= x_0 + (\tfrac{r}{R_0})^{-1+1/\mu}, \\ y &:= y_0 + (\tfrac{r}{R_0})^{-1+1/\mu}. \end{aligned} \tag{2.6}$$

Meshes of this type are used in [17] and with a slight modification in [19]. Note that the number of elements and nodes remains unchanged and that condition (d) is still fulfilled after the transformation.

The extension of this approach to three-dimensional domains with edges is natural. However we have to distinguish between two types of meshes which can be generated.

When we consider a neighbourhood of an edge and employ the transformation (2.6) to the nodes of a certain quasiuniform mesh, we get an anisotropic mesh. According to [1], an element is called anisotropic if its diameters in different directions have different asymptotics. Though it turns out that such meshes do not fulfil condition (d), they can be applied successfully. Under some smoothness assumptions on the data, estimate (2.5) has been proved with $\alpha = 1$ and $\mu < \lambda$ [1, 4].

On the other hand, by describing the mesh via condition (e') it is possible to investigate meshes which fulfil condition (d) and to prove the same error estimate (2.5) (with $\alpha = 1$ for $\mu < \lambda$, $f \in L_2(\Omega)$). The disadvantage of such meshes is that for $\mu \leq \frac{1}{3}$ the asymptotic number of elements as well as the asymptotic condition number increase [2, 5]. We suggest to construct these isotropic meshes with the method of dyadic partition [13]: Starting with a coarse mesh the elements are divided until condition (e') is fulfilled with suitable constants $\underline{C}_1, \overline{C}_1, \underline{C}_2$ and $\overline{C}_2$, see also Subsection 3.3.

2.4 Adaptive algorithms

For a detailed knowledge of the errors in a particular finite element approximation and for assessing its acceptability, an a-posteriori error estimator has to be provided. Usually the a-posteriori error estimate is calculated locally and can thus serve as an indicator for regions with large or small errors, respectively, as the quality of a finite element approximation in general varies over the computational domain. Thus it is natural to use so-called automatic mesh adapting finite element strategies for problems with boundary singularities. The process consists in repeating the three steps

- calculating an approximate solution,
- estimating the error locally,
- generating an improved mesh,

until the error is within a desired tolerance ε. For a review of error estimators and refinement strategies see for example [16] or [25, 26].

We use a residual type error estimator based on the one introduced in [7]. Consider a tetrahedron Ω_i with the faces Γ_{ij} and the outer normal vectors $\underline{n}_{ij}$, $(j = 1, \ldots, 4)$. Then the local error contributions η_i are calculated by

$$\begin{aligned} \eta_i^2 &= C \sum_{j=1}^{4} (\text{meas}(\Gamma_{ij}))^{3/2} \, \alpha_{ij}^2, \\ \alpha_{ij} &= \begin{cases} \frac{1}{2}(\nabla u^h|_{\Omega_i} - \nabla u^h|_{\Omega_{ij}}) \cdot \underline{n}_{ij} & \text{if} \quad \Gamma_{ij} = \Omega_i \cap \Omega_{ij}, \\ 0 & \text{if} \quad \Gamma_{ij} \subset \partial\Omega_i, \end{cases} \end{aligned} \tag{2.7}$$

and added to the global estimate

$$\eta^2 = \sum_{i=1}^{m} \eta_i^2. \tag{2.8}$$

The constant C must be taken from experience. For problems with homogeneous Dirichlet boundary conditions (Neumann or Newton boundary conditions can be inhomogeneous) the constant can be extrapolated from two calculations with different meshes: From

$$\|u\|_E^2 - \|u_{h_1}\|_E^2 = C^2 \eta_{C=1,h=h_1}^2$$

$$\|u\|_E^2 - \|u_{h_2}\|_E^2 = C^2 \eta_{C=1,h=h_2}^2$$

we can eliminate the unknown energy $\|u\|_E^2$:

$$C^2 = \frac{\|u_{h_1}\|_E^2 - \|u_{h_2}\|_E^2}{\eta^2_{C=1,h=h_2} - \eta^2_{C=1,h=h_1}} \tag{2.9}$$

When the estimated error η is not within a given tolerance ε, all elements Ω_i with

$$\eta_i^2 \geq \frac{\varepsilon^2}{m}$$

are marked for refinement (m is the number of elements). Our implementation of the refinement procedure is described in Subsection 3.2.

3 Realization of the algorithms in the experimental code FEMPS3D

3.1 Basic properties of FEMPS3D

FEMPS3D is a finite element code for solving Poisson's equation with (in general inhomogeneous, mixed) boundary conditions of Dirichlet, Neumann or Newton type. The first version was developed in 1987-1989 at a VAX workstation, and in 1993 it was ported to the UNIX operating system. The main features are the following:

- The mesh can consist of tetrahedra, hexahedra (cubes) and pentahedra (triangular prisms). Linear and quadratic shape functions can be used.
- The code does not contain a general mesh generator. It is possible to read mesh data from a file generated by any code, eventually after adapting the data structure. Recently we developed some special routines to triangulate our test domains.
- The problem data are given in general by function subroutines. For Dirichlet data we developed the additional feature to interpolate some pointwise values along the surface.
- For the assembly of the equation system many different integration rules are programmed. Only the non zero elements of the upper right triangle of the matrix are stored. The system is solved with a conjugate gradient method, preconditioned with different types of incomplete Cholesky factorization (IC(0), IC(1), MIC), see [21].
- The resulting solution can be interpreted with tables of values in subdomains and with a representation of isolines. When the exact solution is known in academic examples, the table of values and the isolines can be given for the error as well. Additionally the error norms in $H^1(\Omega)$, $L_2(\Omega)$ and in a discrete maximum norm are calculated.
- In general real values are stored in double precision, the exception is the array of the coordinates of the nodes, which is single precision for memory reasons. The versions 1 and 2 of FEMPS3D are restricted to a maximum of 32767 nodes. This restriction is removed in version 3 by using 4-byte-integers (optionally) for the storage of the topology.

In 1993/94 the code was extended, but only for linear tetrahedral elements:

- In version 2 we included an error estimator of residual type and an adaptive mesh refinement procedure, see details in [3] and Subsection 3.2.
- In version 3 the restriction to 32767 nodes was removed and some subroutines were reprogrammed with the aim of saving memory. The isotropic a-priori mesh grading by dyadic partition (see Subsection 2.3) was included.

- In the expectation of an optimization of the meshes two nodal relaxation procedures were included: the standard Laplace smoothing and the improved version introduced in [18] for graded meshes, see Subsections 3.4 and 4.4.

- An interface to the visualization package GRAPE [27] was developed.

3.2 The mesh refinement algorithm

Consider the following situation: Given a finite element mesh with some (eventually all) elements marked for refinement due to the result of an error estimation or within the process of dyadic partitioning (see Subsections 2.3 and 3.3). The task is to construct a refined mesh.

One method is the bisection of the elements as described in [8, 20]. But we follow the stronger strategy and divide all marked tetrahedra into 8 smaller ones of equal volume and programmed the stable version of Bey [9]: A tetrahedron with the nodes $1, 2, 3, 4$ and the midpoints $12, 13, 14, 23, 24, 34$ of the edges is split into the following ones:

$$
\begin{array}{lllll}
t_1: & P_1 & P_{12} & P_{13} & P_{14} \\
t_2: & P_{12} & P_2 & P_{23} & P_{24} \\
t_3: & P_{13} & P_{23} & P_3 & P_{34} \\
t_4: & P_{14} & P_{24} & P_{34} & P_4 \\
t_5: & P_{12} & P_{13} & P_{14} & P_{24} \\
t_6: & P_{12} & P_{13} & P_{23} & P_{24} \\
t_7: & P_{13} & P_{14} & P_{24} & P_{34} \\
t_8: & P_{13} & P_{23} & P_{24} & P_{34}
\end{array}
$$

Note that the enumeration is important to avoid degenerating angles. We will call this procedure *red refinement* of a tetrahedron.

The consequence of a local red refinement is the existence of tetrahedra with irregular nodes, that are tetrahedra which were not refined theirselves but at least one of their neighbours with at least one common edge. For these elements we follow with one exception the *green refinement* strategy in [12]; see Figures 3.1 – 3.3 for the three main cases. In the cases of more than three irregular nodes or three irregular nodes which do not belong to one single face, the element is red refined. In opposite to [12] we did not treat the remaining case of two irregular nodes at adjacent edges in a separate way, compare Figure 3.4, but we introduced an additional node and attributed this case to the one in Figure 3.3.

In order to avoid distorted angles, in each refinement step the green refinement of the previous level is removed. Consequently, at a deeper refinement level it is possible that after the red refinement some tetrahedra have edges with more than one irregular node. These elements are treated with red refinement.

During the production of this so-called *regular closure* of the mesh it can happen that new nodes are introduced, eventually even at edges of green tetrahedra. Thus it must be considered as an iterative process. The entire mesh refinement algorithm can shortly be described in the following way:

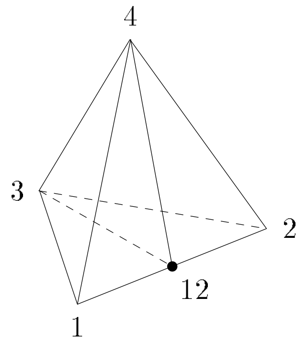
1. Remove all green elements of the given mesh.

2. Divide all marked elements by red refinement.

3. Go through the list of elements and treat all elements with irregular points corresponding to the cases discussed above.

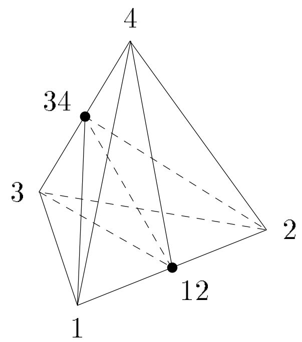
4. Go through the list of elements and remove green tetrahedra with irregular points.

5. If there were any actions in step 3 or 4 then go to 3 else stop.



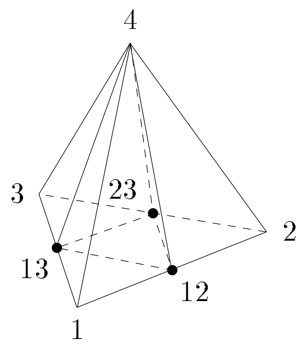
t_1 :	1	3	4	12
t_2 :	2	4	3	12

Figure 3.1: Division of a tetrahedron with one irregular node into two subtetrahedra.



t_1 :	1	3	12	34
t_2 :	2	4	12	34
t_3 :	3	2	12	34
t_4 :	4	1	12	34

Figure 3.2: Division of a tetrahedron with two irregular nodes at opposite edges into four subtetrahedra.



t_1 :	3	23	13	4
t_2 :	2	12	23	4
t_3 :	1	13	12	4
t_4 :	4	23	12	13

Figure 3.3: Division of a tetrahedron with three irregular nodes at one face into four subtetrahedra.

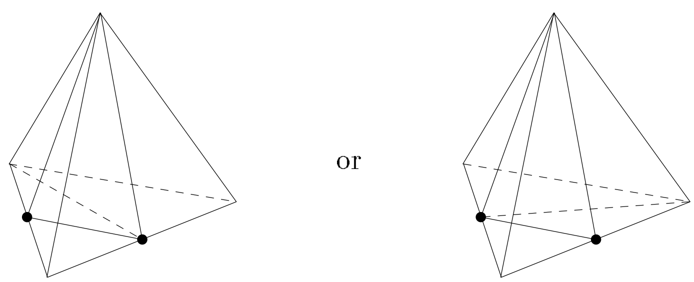


Figure 3.4: Treatment of two irregular nodes at adjacent edges without introducing an additional node.

3.3 The a-priori grading algorithm

As discussed in Subsection 2.3, a mesh can be graded using a coordinate transformation. This type of grading leads near edges to anisotropic meshes. An alternative is the method of *dyadic partitioning* [13]: Given a start mesh proceed as follows:

1. Mark all elements which do not fulfil condition (e').

2. If no element is marked then stop.

3. Run the mesh refinement algorithm from Subsection 3.2 an go to 1.

The main difficulty of this algorithm is the appropriate definition of the constants in condition (e'). We took the following points into consideration.

- Condition (e') can be reformulated in the following way:

$$
\begin{aligned}
\text{(e'')}\quad \underline{C}H(r_i) \quad &\leq \quad h_i \;\leq\; \overline{C}H(r_i) \\
H(r) \quad &:= \quad \begin{cases} R_0^{1-1/\mu}h^{1/\mu} & \text{for } r \leq r_0 \\ \frac{1}{\mu}R_0^{\mu-1}hr^{1-\mu} & \text{for } r > r_0 \end{cases} \\
r_0 \quad &:= \quad \mu^{1/(1-\mu)}R_0^{1-1/\mu}h^{1/\mu} \quad (\mu < 1)
\end{aligned}
$$

  As before R_0 is the radius of the refinement region, in our tests $R_0 = 1$. — Thus we have a continuous function $H(r)$ which is also useful for the node relaxation procedure described in Subsection 3.4.

- It is not possible with the algorithm of dyadic partitioning to produce elements of an exactly given size. The elements will have a diameter which is the start mesh size divided by 2^k with k being the refinement level.

Thus we used condition (e'') with $\underline{C} = 0.7$ and $\overline{C} = 1.4$ but tested only the upper bound of (e'') to mark elements. Here, we want to remark that there are always elements which do not fulfill the lower bound for several reasons:

- The initial mesh size is to small. This can be observed for elements Ω_i with r_i close to R_0; note that $H(R_0) = \frac{1}{\mu}h$. The factor $\frac{1}{\mu}$ was introduced for compatibility reasons with the construction of the mesh via a coordinate transformation.

- After the subdivision of an element some of the *child elements* have a larger distance to the edge/corner than their *father* had. This affects particularly the elements close to the edge/corner.

- The regularization of the mesh splits elements which were not marked. Here, especially the case of red refinement introduces elements with smaller mesh size.

In our tests we observed another effect, which should be remarked here. In condition (e') there appears also the parameter h, which characterizes the mesh size outside the refinement region. In our realization, h is also the mesh size of the start mesh of the dyadic partitioning algorithm and chosen to be $\frac{1}{k}$, $k = 1, 2, 3, \ldots$ The smaller h is, the more refinement steps are necessary to generate the smallest elements (those at the edge). The resulting number of nodes $n(k+1)$ changes slightly in comparison to $n(k)$ when for $h = \frac{1}{k}$ and $h = \frac{1}{k+1}$ the same number of refinement steps are necessary and it changes very rapidly when one more step must be executed. This effect leads to the exceptional points in the error diagram in Figure 5.2.

Because of these observations for the standard algorithm we modified our algorithm by including two node relaxation procedures.

3.4 Node relaxation

The idea of the standard Laplace smoothing is that each node should be located in the center of gravity of its neighbours. Let $\underline{x}^{(i)}$ $(i = 1, \ldots, n)$ be the coordinates of the nodes, and denote by $I(i)$ the set of the numbers of those nodes which have a common edge with $\underline{x}^{(i)}$. Then the equations

$$\underline{x}^{(i)} = \frac{1}{\operatorname{card} I(i)} \sum_{j \in I(i)} \underline{x}^{(j)} \tag{3.1}$$

should be fulfilled for all inner nodes. Boundary nodes must remain at the boundary; for simplicity we let them fixed.

This algorithm does not include information about the desired mesh sizes. Thus it was modified in [18] in the following way:

$$\underline{x}^{(i)} = \left(\sum_{j \in I(i)} \frac{e_{ij}}{h_{ij}} \right)^{-1} \left(\sum_{j \in I(i)} \frac{e_{ij}}{h_{ij}} \, \underline{x}^{(j)} \right) \tag{3.2}$$

with $e_{ij} := |\underline{x}^{(i)} - \underline{x}^{(j)}|$, $h_{ij} := \frac{1}{2}\left(h(\underline{x}^{(i)}) + h(\underline{x}^{(j)}) \right)$. The mesh density function $h(\underline{x})$ has already been described in (e”), $r(\underline{x}) := \operatorname{dist}(\underline{x}, M) := \min_{\underline{y} \in M} |\underline{x} - \underline{y}|$; M is the set of edges with interior angle $\omega > \pi$.

The equation systems (3.1) and (3.2) are already in iterative form and approximately solved by maximal 20 iterations with an Underrelaxation Method (relaxation parameter 0.5).

4 Comparison of isotropic and anisotropic mesh refinement near an edge

4.1 Description of the test example

In [3] we treated the example given below by different mesh refinement strategies:

- *anisotropic* a-priori grading (with different parameters μ),
- adaptive mesh refinement,
- adaptive mesh refinement starting with an *anisotropic*, graded initial mesh,
- adaptive mesh refinement with grading for all meshes.

In this section we want to compare these strategies with the following additional strategies:

- *isotropic* a-priori grading,
- adaptive mesh refinement starting with an *isotropic*, graded initial mesh,
- adaptive mesh refinement starting with an *isotropic*, graded initial mesh and using a coordinate transformation in each refinement step.

Additionally, we present some experiences with node relaxation and error estimation. As the example we considered Laplace's equation with essential boundary conditions

$$\left. \begin{aligned} -\Delta u &= 0 \quad &&\text{in } \Omega \\ u &= g \quad &&\text{on } \partial\Omega \end{aligned} \right\} \tag{4.1}$$

in the three-dimensional domain $\Omega = \{(x_1, x_2, x_3) = (r\cos\varphi, r\sin\varphi, z) \in I\!R^3 : r < 1, 0 < \varphi < \frac{3}{2}\pi, 0 < z < 1\}$. The right hand side g is taken such that

$$u = (10 + z)\, r^{2/3} \sin \tfrac{2}{3}\varphi$$

is the exact solution of the problem. It has the typical singular behaviour at the edge, compare (2.3).

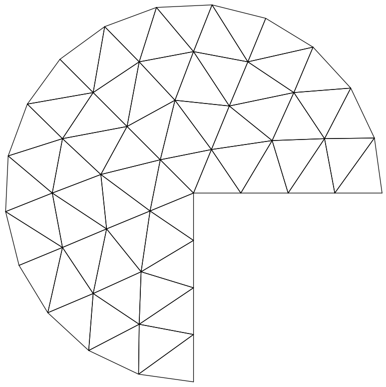


Figure 4.1: Triangulation of the basis ($M = 4$).

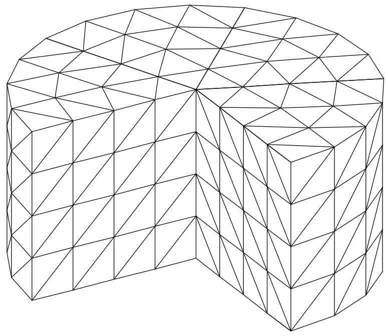


Figure 4.2: Ungraded initial mesh for the sector of a cylinder ($M = 4$).

The initially ungraded meshes are characterized by an integer parameter M and constructed in the following way:

1. Triangulation of the two-dimensional basis of the domain: in each circular arc with radius $r_i = \frac{i}{M}$ $(i = 0, \ldots, M)$ we place $4i + 1$ equidistant nodes. Then the topology is created in a regular way such that each layer between r_i and r_{i+1} $(i = 0, \ldots, M-1)$ contains $4(2i + 1)$ triangles, that means the sector of the disc is subdivided into $4M^2$ elements, see Figure 4.1 for $M = 4$.

2. The two-dimensional triangulation is then used at each plane $z = \frac{i}{M}$ $(i = 0, \ldots, M)$ to form in each slice $\frac{i-1}{M} < z < \frac{i}{M}$ $(i = 1, \ldots, M)$ $4M^2$ triangular prisms which are divided into 3 tetrahedra each, see Figure 4.2 for $M = 4$. That means the triangulation consists of $m = 12M^3$ tetrahedra, $n = (2M + 1)(M + 1)^2$ nodes, and $N = (2M - 1)(M - 1)^2$ unknowns.

4.2 A-priori mesh grading

In this subsection we want to investigate the influence of the *isotropic* a-priori mesh grading on the behaviour of the finite element error, especially on the convergence order of the error. Then we compare the results with those for the anisotropic mesh grading in [3]. We constructed the mesh by dyadic partitioning (see Subsection 3.3) and varied the parameters M $(3, 6, 9, \ldots)$ and μ $(1.0, 0.9, \ldots)$, for M see Subsection 4.1 and for μ see condition (e') in 2.3. From the numerical solution and the known exact solution, the energy norm $\|e\|_E$ of the finite element error $e = u - u_h$ was computed by numerical integration with a 14-point-formula. The relative norms $\|e\|_\% := \|e\|_E / \|u_h\|_E$ are arranged for some values of μ in a double logarithmic scale in Figure 4.3.

The calculations here and in [3] demonstrate that isotropic as well as anisotropic, graded meshes are useful for treating edge singularities, for diminishing the error and achieving the

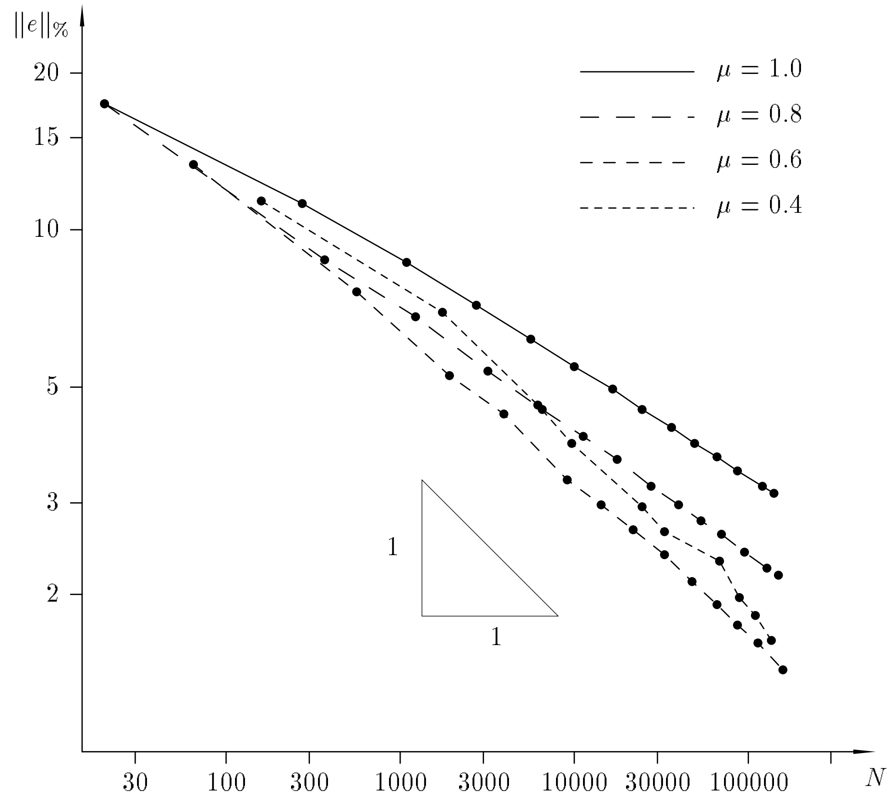


Figure 4.3: Behaviour of the error for different grading parameters μ, isotropic case.

optimal approximation order. In order to compare both strategies we arranged the values of both strategies in Figure 4.4.

The curves for $\mu = 0.5$ are nearly parallel, but the anisotropic strategy gives a slightly smaller error. This can be taken as an indication that the large amount of nodes near the edge in the neighbourhood of the edge is not necessary, and that anisotropic meshes are the more appropriate way for treating edge singularities. On the other hand, the difference between both strategies is very small (factor 0.9), thus the result from one test should not be overrated.

4.3 Adaptive mesh refinement with grading

In extension of the computations in [3] we carried out tests with two variants of the adaptive mesh refinement strategy starting with an isotropic, graded initial mesh. The variants differ in the way new nodes are introduced. While they are located in the middle of an edge in the first variant, we carried out a coordinate transformation in each refinement step of the second variant, thus the new nodes were introduced not necessarily in the center of the edge.

The parameters M $(3, 4, 5)$ and μ $(0.4, 0.5, 0.6)$ for the initial mesh, as well as the tolerance ε $(0.05\|u_h\|_E, 0.03\|u_h\|_E)$ for the relative error were varied and the results were compared with the other adaptive strategies (without mesh grading; with anisotropic, graded initial mesh; with anisotropic, graded mesh in each refinement step).

The qualitative behaviour of the strategies is similar for each choice of the parameters. We present the results for $M = 3$, $\mu = 0.6$, $\varepsilon = 0.03$ in Figure 4.5. It can be observed that the error in the first variant is larger than in the anisotropic strategies, again an indication for the superiority of anisotropic mesh grading. In comparison with the standard strategy without mesh grading we see, that the final error level is received with a about the same number of unknowns, but with one refinement step minus. — The second strategy can be considered as an amalgamation of isotropic and anisotropic strategies, and it behaves like this.

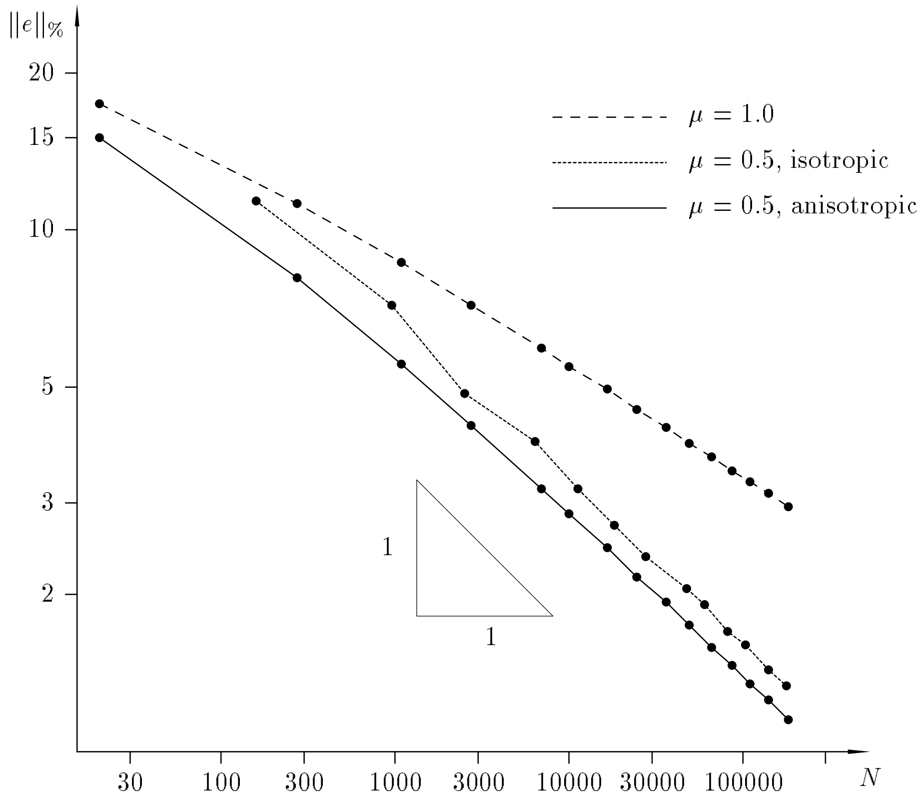


Figure 4.4: Comparison of isotropic and anisotropic mesh grading.

4.4 Node relaxation

The construction of isotropic meshes via the method of dyadic partitioning has the disadvantage that the ratio of the diameters of adjacent elements is more or less exactly 1 or 2. Moreover, there is the problem of a suitable choice of the constants in (e') from Subsection 2.3, see also the explanation in Subsection 3.3. Thus one could suppose that these meshes cannot fit the singular functions as good as meshes which are constructed via a coordinate transformation. We hoped that a node relaxation procedure would improve the meshes.

Our tests with the standard Laplace smoothing and with the smoothing introduced in [18] (see also Subsection 3.4) produced only minimal differences to the meshes used before, see Figure 4.6. It seems that the meshes are without smoothing good enough, or that the smoothing procedure has to be tuned for our purposes. We did not make further investigations.

Note that anisotropic meshes are not compatible with these smoothing procedures; an algorithm that takes into account the different mesh sizes in the different directions, has not been programmed.

4.5 Reliability of the error estimator

All the previous results in this section were achieved with an elementwise integration of the exact error $u - u_h$; it was possible because the solution u of (4.1) is known. For studying the reliability of the error estimator programmed, we always estimated the error (using $C = 1$, confer Subsection 2.4) and computed the efficiency index

$$\theta := \frac{\eta}{\|u - u_h\|_E},$$

for η see (2.8). If this value is nearly independent of M and μ, then we can try to estimate the constant C and get $\theta \approx 1$. (Note that (2.9) is not applicable for inhomogeneous Dirichlet boundary conditions.)

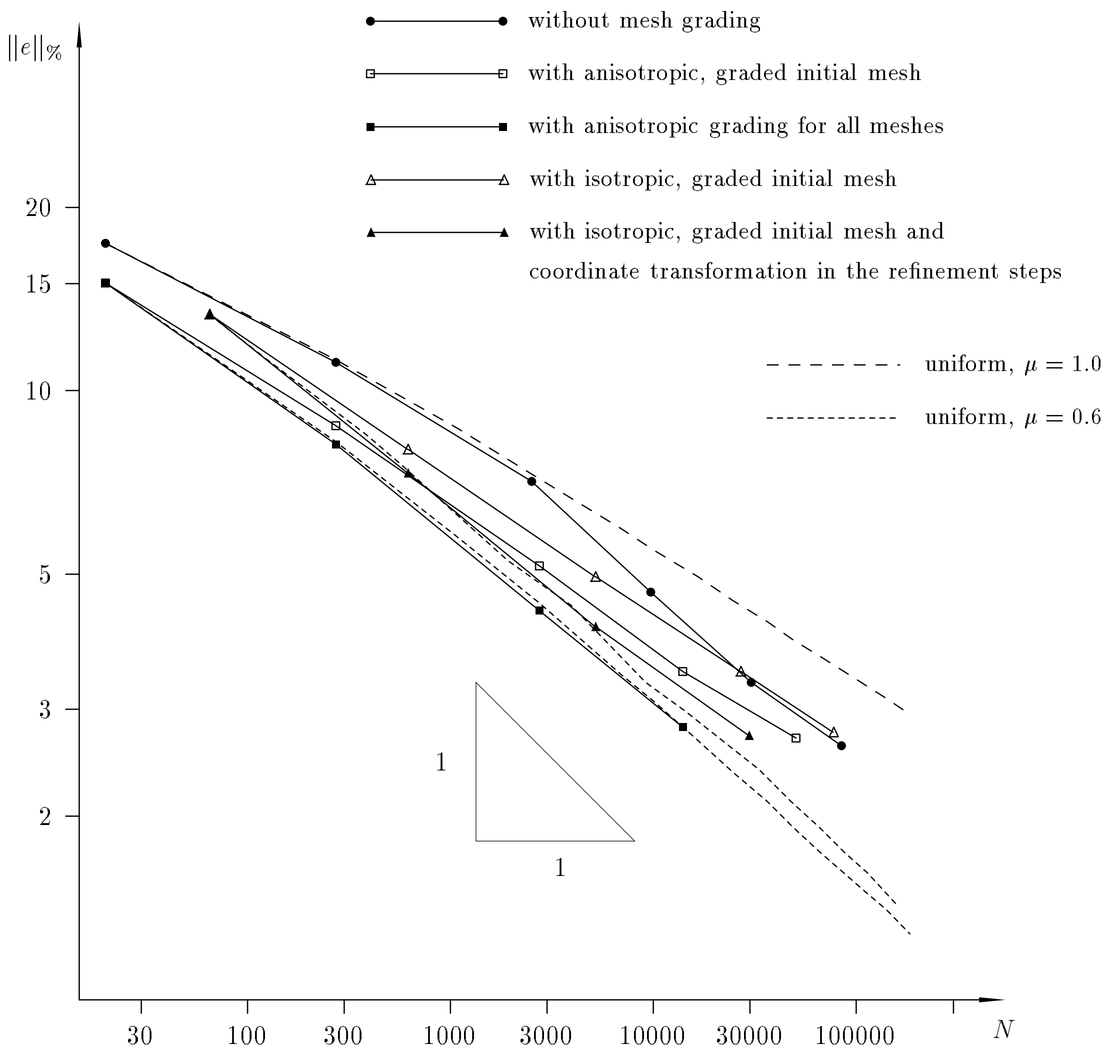


Figure 4.5: Error in the energy norm for various adaptive mesh refinement strategies, $M = 3$, $\mu = 0.6$, $\varepsilon = 0.03\|u_h\|_E$.

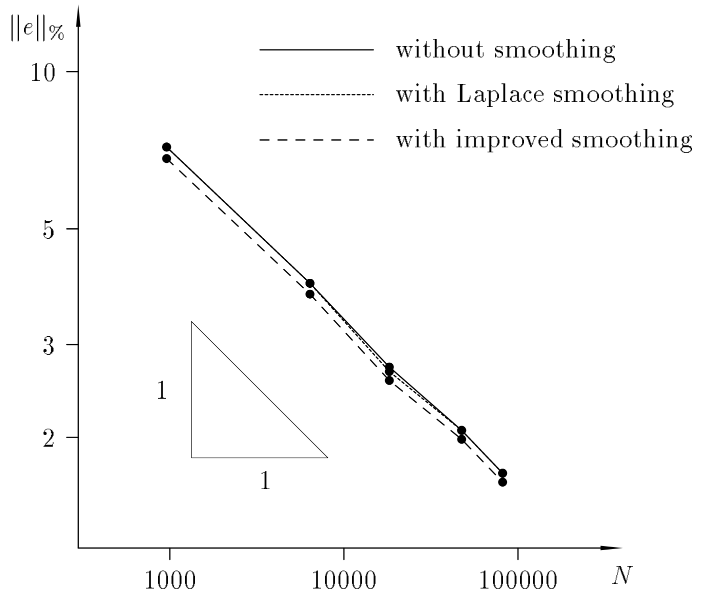


Figure 4.6: Error in the energy norm for isotropic, graded meshes ($\mu = 0.6$): influence of node relaxation.

μ	M=3	M=12	M=24	M=36
1.0	4.530	4.523	4.512	4.507
0.9	4.645	4.800	4.875	4.922
0.8	4.781	5.125	5.306	5.419
0.7	4.939	5.488	5.782	5.964
0.6	5.118	5.854	6.233	6.459
0.5	5.327	6.178	6.571	6.785
0.4	5.612	6.452	6.779	6.931
0.3	6.125	6.730	6.956	7.043

Table 4.1: Dependence of the effectivity index θ on the mesh parameters M and μ: anisotropic grading.

μ	M=3	M=6	M=12	M=18	M=24	M=30	M=36
1.0	4.530	4.538	4.523	4.516	4.512	4.509	4.507
0.8	4.530	4.611	4.656	4.641	4.642	4.635	4.631
0.7	4.525	4.611	4.780	4.790	4.789	4.795	
0.6	4.525	4.724	4.780	4.902	4.909	4.892	
0.5	4.598	4.795	4.928	4.943	5.010	5.021	
0.4	4.598	4.830	4.950	4.982	5.031		
0.3	4.685	4.852	4.965				

Table 4.2: Dependence of the effectivity index θ on the mesh parameters M and μ: isotropic grading. (There are some values missing for memory reasons.)

In Tables 4.1 and 4.2 we give the values for θ, and we can underline the observation in [3] that the error estimator under consideration is sensitive with respect to M and μ if the mesh is anisotropically graded. A better estimator is desired.

5 The Fichera corner: a test example with both a corner and several edges

As a second test example we consider the Poisson equation with a specific right hand side $f \in L_2(\Omega)$ ($f \notin L_p(\Omega)$ for $p > 2$), together with homogeneous Dirichlet boundary conditions:

$$\left.\begin{array}{rcll} -\Delta u & = & r^{-3/2}\,(\ln \frac{r}{1000})^{-1} & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega. \end{array}\right\} \tag{5.1}$$

The domain $\Omega := (-1,1)^3 \setminus [0,1]^3$ (see Figure 5.1) has three edges with interior angle $\omega = \frac{3}{2}\pi$ which meet in the center of coordinates; we denote by r the distance to this point. Sometimes such a corner is called Fichera corner. The edge singularities are described by $\lambda = \frac{2}{3}$ (for λ see (2.3)), the corner singularity is weaker, see [5, Subsection 4.2].

The domain Ω can be split naturally into 7 cubes. The triangulation of Ω is generated by dividing each cube into M^3 ($M = 2, 3, \ldots$) congruent smaller cubes, which are then divided into 6 tetrahedra of the same volume each ($m = 42M^3$ tetrahedra, $n = (2M+1)^3 - M^3$ nodes, $N = (2M-1)^3 - M^3$ unknowns). Refined meshes with various grading parameters $\mu < 1$ were constructed using the method of dyadic partitioning, see Subsection 3.3. An example with $M = 3$ and $\mu = 0.5$ is shown in Figure 5.1. We were not able to find a suitable

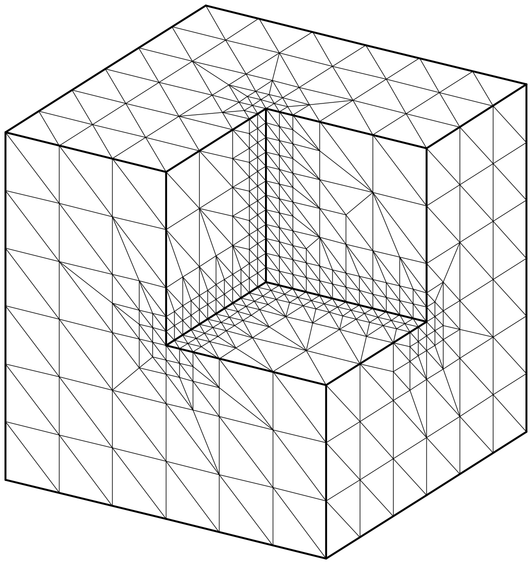


Figure 5.1: Fichera corner with refined mesh ($M = 3$, $\mu = 0.5$)

coordinate transformation to construct an anisotropic refinement near the three concave edges, which becomes isotropic near the concave vertex.

The finite element error in the energy norm is measured with the error estimator described in Subsection 2.4. Note that, in contrast to the example in Section 4, we do not know the exact solution of Problem (5.1). — For uniform mesh grading the results are already published in [5] and shall not be repeated here. But we will compare these results with

- adaptive mesh refinement and
- adaptive mesh refinement starting with an isotropically graded initial mesh.

The memory resources of the computer we used allowed only to work with an error bound of at least $\varepsilon = 0.15\|u_h\|_E$. We varied the initial mesh size by starting with $M = 3, 4, 5, 6$, and the grading parameter $\mu = 0.4, 0.5, 0.6$. In Figure 5.2 we present the results for $M = 4$ and $\mu = 0.6$ for the two stategies mentioned above, for comparison together with the curves for uniform refinement ($\mu = 1.0$ and $\mu = 0.6$). The results in the other cases are similar, for $M = 3, 4$ sometimes the stopping criterion $\eta < \varepsilon$ (for η see (2.8)) has not been achieved for memory reasons.

The behaviour of the error is similar to the tests in Subsection 4.3, that means, a second test has shown that starting with a graded mesh is favourable in the sense that a smaller error is achieved with less refinement steps.

The following observations shall be remarked:

- We got a slightly different error in the two cases
  - $N = 6$, $\mu = 1$,
  - $N = 3$, $\mu = 1$ plus one global refinement step.

  The reason is the different topology of the meshes because of the different algorithms for their construction.

- For the choice of an appropriate constant C in (2.7) we need two calculations with two different meshes, see (2.9). In the uniform mesh grading we used the results from the previous mesh, and omitted the first error norm ($M = 1$, which was calculated with $C = 1$) in the presentation.

  But for adaptive algorithms we need reliable error indicators η_i already for the first mesh. Here, we used the constants from the calculations with uniform mesh grading.

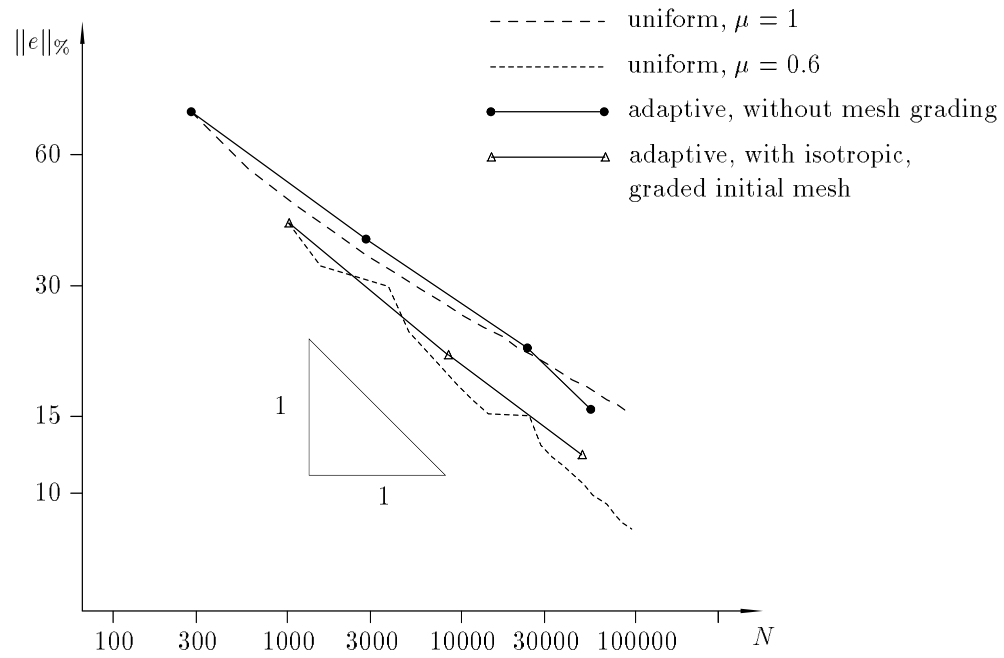


Figure 5.2: Error in the energy norm for adaptive mesh refinement strategies, $M = 4$, $\varepsilon = 0.15\|u_h\|_E$. (For comparison the curves for uniform mesh refinement are included.)

It is clear that this information is not available in practical calculations; then one has to use a start value C from experience or $C = 1$.

- As in the tests in Section 4, node relaxation either did not work or did not influence the results markedly. In some cases we had even problems, that near the concave corner some inner points moved out of the domain.

Acknowledgement. Both authors were supported by DFG (German Research Foundation), No. La 767-3/1.

References

[1] Th. Apel and M. Dobrowolski. Anisotropic interpolation with applications to the finite element method. *Computing*, 47:277–293, 1992.

[2] Th. Apel and B. Heinrich. Mesh refinement and windowing near edges for some elliptic problem. *SIAM J. Numer. Anal.*, 31, 1994. To appear.

[3] Th. Apel, R. Mücke, and J. R. Whiteman. An adaptive finite element technique with a-priori mesh grading. Technical Report 9, BICOM Institute of Computational Mathematics, 1993.

[4] Th. Apel and S. Nicaise. Elliptic problems in domains with edges: anisotropic regularity and anisotropic finite element meshes. Preprint 16, TU Chemnitz-Zwickau, 1994.

[5] Th. Apel, A.-M. Sändig, and J. R. Whiteman. Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains. Technical Report 12, BICOM Institute of Computational Mathematics, 1993. To appear in Math. Meth. Appl. Sci.

[6] I. Babuška, R. B. Kellogg, and J. Pitkäranta. Direct and inverse error estimates for finite elements with mesh refinements. *Numer. Math.*, 33:447–471, 1979.

[7] I. Babuška and W. C. Rheinboldt. On the reliability and optimality of the finite element method. *Comput. Struct.*, 10:87–94, 1979.

[8] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *IMPACT of Computing in Science and Engineering*, 3:181–191, 1991.

[9] J. Bey. Der BPX-Vorkonditionierer in 3 Dimensionen: Gitter-Verfeinerung, Parallelisierung und Simulation. Preprint 3, Universität Heidelberg, IWR, 1992.

[10] H. Blum and M. Dobrowolski. On finite element methods for elliptic equations on domains with corners. *Computing*, 28:53–63, 1982.

[11] P. Ciarlet. *The finite element method for elliptic problems.* North-Holland Publishing Company, Amsterdam, 1978.

[12] B. Erdmann and R. Roitzsch. 3d-ELLKASK: Programmer's Manual; Version 1.0. Technical report, ZIB Berlin, 1993.

[13] R. Fritzsch. *Optimale Finite–Elemente–Approximationen für Funktionen mit Singularitäten.* PhD thesis, Technische Universität Dresden, 1990.

[14] P. Grisvard, W. Wendland, and J. R. Whiteman, editors. *Singularities and constructive methods of their treatment.* Lecture Notes in Mathematics, vol. 1121. Springer–Verlag, Berlin–Heidelberg–New York–Tokyo, 1985.

[15] A. Kufner and A.-M. Sändig. *Some Applications of Weighted Sobolev Spaces.* BSB B. G. Teubner Verlagsgesellschaft, Leipzig, 1987.

[16] J. T. Oden, L. Demkowicz, T. A. Westermann, and W. Rachowicz. Toward a universal h-p adaptive finite element strategy. Part 2. A posteriori error estimates. *Comp. Meth. Appl. Mech. Eng.*, 77:113–180, 1989.

[17] L. A. Oganesyan and L. A. Rukhovets. *Variational-difference methods for the solution of elliptic equations.* Izd. Akad. Nauk Armyanskoi SSR, Jerevan, 1979. (Russian).

[18] E. Rank, M. Schweingruber, and M. Sommer. Adaptive mesh generation and transformation of triangular to quadrilateral meshes. *Comm. Num. Meth. Eng*, 9:121–129, 1993.

[19] G. Raugel. Résolution numérique par une méthode d'éléments finis du problème Dirichlet pour le Laplacien dans un polygone. *C. R. Acad. Sci. Paris, Sér. A*, 286(18):A791–A794, 1978.

[20] M. C. Rivara. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Int. J. Num. Meth. Eng.*, 20:745–756, 1984.

[21] S. Rjasanow. Dokumentation und theoretische Grundlagen zum Programm SOLKLZ. Preprint 15, TU Karl-Marx-Stadt (Chemnitz), 1986.

[22] A. H. Schatz and L. B. Wahlbin. Maximum norm estimates in the finite element method on plane polygonal domains. Part 2: Refinements. *Mathematics of Computation*, 33(146):465–492, 1979.

[23] E. Stephan and J. R. Whiteman. Singularities of the Laplacian at corners and edges of three–dimensional domains and their treatment with finite element methods. *Math. Meth. Appl. Sci.*, 10(3):339–350, 1988.

[24] G. Strang and G. Fix. *An analysis of the finite element method.* Prentice–Hall Inc., Englewood Cliffs, 1973.

[25] T. Strouboulis and K. A. Haque. Recent experiences with error estimation and adaptivity. Part 1: Review of error estimators for scalar elliptic problems. *Comp. Meth. Appl. Mech. Eng.*, 97:399–436, 1992.

[26] T. Strouboulis and K. A. Haque. Recent experiences with error estimation and adaptivity. Part 2: Error estimation for h-adaptive approximations on grids of triangles and quadrilaterals. *Comp. Meth. Appl. Mech. Eng.*, 100:359–430, 1992.

[27] A. Wierse and M. Rumpf. GRAPE – Eine objektorientierte Visualisierungs- und Numerikplattform. *Informatik Forsch. Entw.*, 7:145–151, 1992.

[28] A. Ženíšek. *Nonlinear elliptic and evolution problems and their finite element approximations.* Academic Press, London, 1990.

Other titles in the SPC series:

93_1 G. Haase, T. Hommel, A. Meyer, and M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen. Preprint SPC 93_1, TU Chemnitz, DFG-Forschergruppe, Mai 1993.

93_2 M. Pester and S. Rjasanow. A parallel version of the preconditioned conjugate gradient method for boundary element equations. Preprint SPC 93_2, TU Chemnitz, DFG-Forschergruppe, June 1993.

93_3 G. Globisch. PARMESH – a parallel mesh generator. Preprint SPC 93_3, TU Chemnitz, DFG-Forschergruppe, June 1993.

94_1 J. Weickert and T. Steidten. Efficient time step parallelization of full-multigrid techniques. Preprint SPC 94_1, TU Chemnitz-Zwickau, DFG-Forschergruppe, January 1994.

94_2 U. Groh. Lokale Realisierung von Vektoroperationen auf Parallelrechnern. Preprint SPC 94_2, TU Chemnitz-Zwickau, DFG-Forschergruppe, March 1994.

94_3 A. Meyer. Preconditoning the Pseudo-Laplacian for Finite Element simulation of incompressible flow. Preprint SPC 94_3, TU Chemnitz-Zwickau, DFG-Forschergruppe, February 1994.

94_4 M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen. Preprint SPC 94_4, TU Chemnitz-Zwickau, DFG-Forschergruppe, March 1994. (aktualisierte Fassung).

94_5 U. Groh, Chr. Israel, St. Meinel, and A. Meyer. On the numerical simulation of coupled transient problems on MIMD parallel systems. Preprint SPC 94_5, TU Chemnitz-Zwickau, DFG-Forschergruppe, April 1994.

94_6 G. Globisch. On an automatically parallel generation technique for tetrahedral meshes. Preprint SPC 94_6, TU Chemnitz-Zwickau, DFG-Forschergruppe, April 1994.

94_7 K. Bernert. Tauextrapolation – theoretische Grundlagen, numerische Experimente und Anwendungen auf die Navier-Stokes-Gleichungen. Preprint SPC 94_7, TU Chemnitz-Zwickau, DFG-Forschergruppe, Juni 1994.

94_8 G. Haase, U. Langer, A. Meyer, and S. V. Nepomnyaschikh. Hierarchical extension and local multigrid methods in domain decomposition preconditoners. Preprint SPC 94_8, TU Chemnitz-Zwickau, DFG-Forschergruppe, June 1994.

94_9 G. Kunert. On the choice of the basis transformation for the definition of DD Dirichlet preconditioners. Preprint SPC 94_9, TU Chemnitz-Zwickau, DFG-Forschergruppe, June 1994.

94_10 M. Pester and T. Steidten. Parallel implementation of the Fourier Finite Element Method. Preprint SPC 94_10, TU Chemnitz-Zwickau, DFG-Forschergruppe, June 1994.

94_11 M. Jung and U. Rüde. Implicit Extrapolation Methods for Multilevel Finite Element Computations: Theory and Applications. Preprint SPC 94_11, TU Chemnitz-Zwickau, DFG-Forschergruppe, June 1994.

94_12 A. Meyer and M. Pester. Verarbeitung von Sparse-Matrizen in Kompaktspeicherform KLZ/KZU. Preprint SPC 94_12, TU Chemnitz-Zwickau, DFG-Forschergruppe, Juni 1994.

94_13 B. Heinrich and B. Weber. Singularities of the solution of axisymmetric elliptic interface problems. Preprint SPC 94_13, TU Chemnitz-Zwickau, DFG-Forschergruppe, June 1994.

94_14 K. Gürlebeck, A. Hommel, and T. Steidten. The method of lumped masses in cylindrical coordinates. Preprint SPC 94_14, TU Chemnitz-Zwickau, DFG-Forschergruppe, July 1994.

Some papers can be accessed via anonymous ftp from server ftp.tu-chemnitz.de, directory pub/Local/mathematik/SPC. (Note the capital L in Local!)