Hamburger Beiträge
zur Angewandten Mathematik

An adaptive concept for the finite element method for boundary eigenvalue problems

Kai Rothe

Reihe A
Preprint 181
March 2004

Hamburger Beiträge zur Angewandten Mathematik

Reihe A	Preprints
Reihe B	Berichte
Reihe C	Mathematische Modelle und Simulation
Reihe D	Elektrische Netzwerke und Bauelemente
Reihe E	Scientific Computing
Reihe F	Computational Fluid Dynamics and Data Analysis

An adaptive concept for the finite element method for boundary eigenvalue problems

Kai Rothe

Fachbereich Mathematik, Universität Hamburg, Bundesstrasse 55, 20146 Hamburg
Germany

Abstract

Error indicators for static problems can be applied to eigenvalue problems too. If these error indicators are weighted, it is possible to approximate a predetermined part of the spectrum very well. This concept leads to an adaptive algorithm which is tested at two examples with non smooth eigenfunctions, where the non smoothness is caused by re-entrant corners of the domain.

Keywords: boundary eigenvalue problems, finite element method, error indicator, adaptive algorithm

1 Introduction

The approximation quality of the finite element solution for boundary eigenvalue problems depends on the triangulation of the domain. If the solution of the problem is uniformly smooth on the whole domain, an uniform triangulation will be normally a good choice for a useful approximation. If the smoothness of the solution differs from one part of the domain to an other, coarse and fine triangulation has to change on these parts to get a satisfying result. In this case adaptive algorithms which fit the triangulation locally dependend on an elementwise error estimator can be used with advantage. For boundary eigenvalue problems, however, adaptive algorithms are not standard in usual program packages.

Following an idea from Babuška and Rheinboldt [2], it turns out that error estimators for static problems are applicable to eigenvalue problems too. In this paper an elementwise error indicator is developed which by adjusting parameters results in a very high approximation quality for a by the user predetermined part of the spectrum. The efficiency of the resulting adaptive algorithm (cf [7]) is demonstrated at two examples. These examples belong to a class of problems described in the following in which the smoothness of some eigenfunctions is disturbed by re-entrant corners.

We consider the boundary eigenvalue problem in its variational form:

Find $u \in H(\Omega)$ with $u \not\equiv 0$ and $\lambda \in \mathbb{R}$ such that

$$a(u, v) \quad = \quad \lambda b(u, v) \quad \forall\, v \in H(\Omega), \tag{1}$$

with the following symmetric bilinear forms on $H(\Omega) \times H(\Omega)$

$$\begin{aligned} a(u,v) &:= \int_\Omega (\nabla u)^T \boldsymbol{k} \, \nabla v \, + \, q\, u\, v \; d(x,y)\,, \\ b(u,v) &:= \int_\Omega \rho\, u\, v \; d(x,y) \end{aligned}$$

and the Sobolev space $H(\Omega) = H_0^1(\Omega)$ for the Dirichlet problem respectively $H(\Omega) = H^1(\Omega)$ for the Neumann problem.

For the bounded domain $\Omega \subset \mathbb{R}^2$ the boundary is denoted by $\Gamma := \partial\Omega$. We suppose that Γ is a polygonal or Lipschitz boundary for instance. Let the matrix $\boldsymbol{k} = (k_{ij})_{i,j=1,2}$ be symmetric and positive definite on Ω with $k_{ij} \in C^1(\Omega) \cap C(\overline{\Omega})$. For the functions $q, \rho \in C(\overline{\Omega})$ the properties $\rho(x,y) > 0$ and $q(x,y) \geq 0$ are required additionally.

For the coefficient-functions $\boldsymbol{k}, q$ and ρ the requirements of continuity and differentiability are often too strong. They can be replaced by weaker ones based on integrability. In any case for the bilinear forms a and b the following is demanded

$$\begin{aligned} |a(u,v)| &\leq C_1 ||u||_{1,\Omega}\, ||v||_{1,\Omega}\,, & & \\ |b(u,v)| &\leq C_2 ||u||_{0,\Omega}\, ||v||_{0,\Omega} & \forall\, u,v \in H(\Omega) \quad \text{and} \\ a(u,u) &\geq C_3 ||u||_{1,\Omega}^2 & \forall\, u \in H(\Omega), \end{aligned} \tag{2}$$

with positive constants C_1, C_2 and C_3. Thus by $a(u,v)$ and $b(u,v)$ inner products are defined with corresponding norms

$$||u||_a := \sqrt{a(u,u)} \quad \text{and} \quad ||u||_b := \sqrt{b(u,u)}\,.$$

Under these assumptions (1) possesses (cf Babuška und Osborn [1]) the following increasing sequence of eigenvalues

$$0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \ldots$$

with corresponding $||\cdot||_b$ orthonormalized eigenfunctions

$$u_1, u_2, u_3, \ldots\,,$$

which are dense in $L_2(\Omega)$.

2 Error estimation

As finite element space V_h we choose the space of all continuous functions that are linear on each triangle which in the case of the Dirichlet problem is defined as follows

$$V_h := \{ v \in C(\Omega) \mid v|_\Gamma = 0 \,\text{and}\, v|_\tau \,\text{linear}\, \forall\, \tau \in T \}\;.$$

Thereby the set $T = \{\tau_1, \tau_2, \cdots, \tau_m\}$ of triangles τ_i partitions the domain

$$\Omega = \tau_1 \cup \tau_2 \cdots \cup \tau_m$$

with boundary Γ. We notice that $V_h \subset H(\Omega)$. In the case of the Neumann problem the boundary condition $v|_\Gamma = 0$ vanishes.

By λ_h and $u_h \in V_h$ we denote the finite element approximations of the eigenvalue λ and the corresponding eigenfunction u, thus

$$a(u_h, v) \;=\; \lambda_h b(u_h, v) \quad \forall\, v \in V_h \subset H(\Omega)$$

is valid. Now we interpret u_h as finite element solution of a static problem with right hand side

$$f(v) := \lambda_h b(u_h, v)$$

whose exact solution w solves the following equation

$$a(w, v) \;=\; f(v) \quad \forall\, v \in H(\Omega)\,. \tag{3}$$

Theorem
For the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ we assume that (2) is valid. Then it holds

$$\left|\frac{\lambda_h - \lambda}{\lambda}\right| \;\leq\; C\,||u_h - w||_a\,. \tag{4}$$

Proof

From equation (3) we get for all $v \in H(\Omega)$

$$a(u_h, v) \;=\; a(u_h, v) - a(w, v) + \lambda_h b(u_h, v)\,. \tag{5}$$

Subtracting (1) from (5), we have for all $v \in H(\Omega)$

$$\begin{aligned} a(u_h - u, v) \;&=\; a(u_h - w, v) + \lambda_h b(u_h, v) - \lambda b(u, v) \\ &=\; a(u_h - w, v) + (\lambda_h - \lambda) b(u_h, v) + \lambda b(u_h - u, v)\,. \end{aligned} \tag{6}$$

If we choose $v = u$ in (6), we obtain by remembering $u_h - u \in H(\Omega)$ and (1)

$$a(u_h - w, u) + (\lambda_h - \lambda) b(u_h, u) \;=\; 0\,.$$

Using the inequality of Cauchy-Schwarz, we get the following estimation

$$|\lambda_h - \lambda|\,|b(u_h, u)| \;=\; |a(u_h - w, u)| \;\leq\; ||u_h - w||_a ||u||_a$$

and with that by reason of $||u||_a = \sqrt{a(u,u)} = \lambda^{1/2}\sqrt{b(u,u)}$

$$\left|\frac{\lambda_h - \lambda}{\lambda}\right| \;\leq\; \frac{\sqrt{b(u,u)}}{\lambda^{1/2}|b(u_h, u)|}||u_h - w||_a\,. \tag{7}$$

Since the eigenfunction u has been supposed to be $||\cdot||_b$ orthonormalized and because of the convergence of the finite element solution $u_h \rightarrow u$, the quotient on the right hand side of (7) is bounded by a positive constant C which proves (4). □

3 Error indicator

Important for the practicability of an adaptive algorithm is the time spent on the computation of the local error estimator. In this chapter we are going to develope a suitable error indicator for one fixed eigenvalue with corresponding eigenfunction.

As it is necessary for inequality (4), the error will be determined elementwise in the energy norm

$$||u_h - w||_a^2 = a(e,e) = \sum_{\tau \in T} a_\tau(e,e)\,,$$

where a_τ is the restriction of the bilinearform a to triangle $\tau \in T$.

From (3) we obtain the *error equation*

$$a(e,v) \;=\; a(u_h,v) - \lambda_h b(u_h,v) \quad \forall\, v \in H(\Omega). \tag{8}$$

In this article we only treat the case of bilinearforms a with diagonal shaped coefficient matrix $\boldsymbol{k}$ (cf (1)), i e, $k_1 := k_{11}, k_2 := k_{22}$ and $k_{12} = 0 = k_{21}$, and refer to an idea of Bank and Weiser [4] for the calculation of error estimators for statical problems which is also discussed by Goering, Roos and Tobiska in [5].

For further treatment the right hand side of (8) is rewritten with respect to the triangulation

$$\begin{aligned} a(u_h,v) &= \int_\Omega k_1\, u_{h,x}v_x \;+\; k_2\, u_{h,y}v_y + qu_h v \; d(x,y) \\ &= \sum_{\tau\in T}\int_\tau k_1\, u_{h,x}v_x \;+\; k_2\, u_{h,y}v_y + qu_h v\; d(x,y) \\ &= -\sum_{\tau\in T}\int_\tau ((k_1u_{h,x})_x + (k_2u_{h,y})_y - qu_h)\, v \;\, d(x,y) \\ &\quad + \sum_{\tau\in T}\int_{\partial\tau} (k_1u_{h,x}n_x + k_2u_{h,y}n_y)\, v\;\, ds\,. \end{aligned} \tag{9}$$

The second inequality of (9) is only valid if the triangulation represents the boundary exactly as it is the case for e.g. polygonal boundaries. Otherwise we get an additional error here and in the following arguementation. In the case of constant coefficients and linear shape functions at triangle τ the terms $(k_1u_{h,x})_x$ and $(k_2u_{h,y})_y$ in (9) are equal to zero.

Now we will rewrite the boundary integrals of (9). With E_0 we denote the set of all edges and with $E_1 \subset E_0$ the set of all inner edges. For each edge $\epsilon \in E_0$ we define a normal direction n^ϵ. In the case of an edge which lies on the boundary, this will be the usual outward normal. Otherwise the choice is arbitrary.

Let ϵ be an inner edge of two adjacent triangles (see Figure 3.1) and let τ^{in} denote the triangle for which n^ϵ is the outward normal. The other triangle is named τ^{out}.

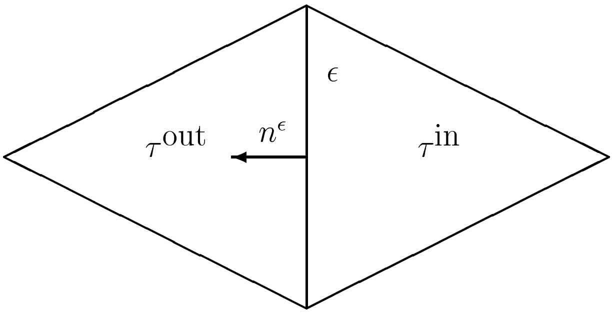

Figure 3.1 triangles $\tau^{\rm in}$ and $\tau^{\rm out}$ with edge ϵ

We define the jumps of the partial derivatives of u_h across ϵ

$$\begin{aligned} (u_{h,x})^\epsilon &:= u_{h,x}^{\rm out} - u_{h,x}^{\rm in} \\ (u_{h,y})^\epsilon &:= u_{h,y}^{\rm out} - u_{h,y}^{\rm in}. \end{aligned}$$

Note the products $(u_{h,x})^\epsilon n_x^\epsilon$ and $(u_{h,y})^\epsilon n_y^\epsilon$ are independent of the direction $n^\epsilon = (n_x^\epsilon, n_y^\epsilon)$. With this notation the boundary integrals on the right hand side of (9) can be formulated in the following way

$$\begin{aligned} a(u_h, v) &= -\sum_{\tau \in T} \int_\tau ((k_1 u_{h,x})_x + (k_2 u_{h,y})_y - q u_h)\, v \, d(x,y) \qquad (10) \\ &+ \int_\Gamma (k_1 u_{h,x} n_x + k_2 u_{h,y} n_y)\, v \, ds - \sum_{\epsilon \in E_1} \int_\epsilon (k_1 (u_{h,x})^\epsilon n_x^\epsilon + k_2 (u_{h,y})^\epsilon n_y^\epsilon)\, v \, ds. \end{aligned}$$

3.1 Neumann boundary condition

In this subsection we choose $H(\Omega) := H^1(\Omega)$ and specialize the preceding results. To guarantee the properties (2), we consider the bilinear form a in its shifted form with $c > 0$:

$$a(u_h, v) = \int_\Omega k_1\, u_{h,x} v_x + k_2\, u_{h,y} v_y + q u_h v + c \, \rho \, u_h \, v \;\; d(x,y). \qquad (11)$$

From (8) in connection with (10) we obtain for all $v \in H(\Omega)$

$$\begin{aligned} a(e, v) &= (c - \lambda_h) b(u_h, v) - \sum_{\tau \in T} \int_\tau ((k_1 u_{h,x})_x + (k_2 u_{h,y})_y - q u_h)\, v \, d(x,y) \qquad (12) \\ &+ \int_\Gamma (k_1 u_{h,x} n_x + k_2 u_{h,y} n_y)\, v \, ds - \sum_{\epsilon \in E_1} \int_\epsilon (k_1 (u_{h,x})^\epsilon n_x^\epsilon + k_2 (u_{h,y})^\epsilon n_y^\epsilon)\, v \, ds. \end{aligned}$$

We notice that the shift c in the error equation (12) vanishes by subtraction, i e, the factor $c - \lambda_h$ coincides with the negative eigenvalue approximation of the unshifted problem since λ_h is the eigenvalue approximation of the shifted problem.

If we restrict equation (12) to each triangle, we obtain the error equation for each element. Instead of solving this equation in $H^1(\tau)$, we solve it only approximately in the vector space $P_2^0(\tau)$ of quadratic polynomials which vanish on the three nodes of τ.

To be more precisely, we seek the error approximation $\tilde{e}$ which satisfies the following equation for all $v \in P_2^0(\tau)$

$$\begin{aligned} a_\tau(\tilde{e}, v) &= \int_\tau ((q + (c - \lambda_h)\rho)u_h - ((k_1 u_{h,x})_x + (k_2 u_{h,y})_y))\, v\, d(x,y) \\ &\quad + \int\limits_{\partial\tau\cap\Gamma} (k_1 u_{h,x} n_x^\epsilon + k_2 u_{h,y} n_y^\epsilon)\, v\, ds - \frac{1}{2} \int\limits_{\partial\tau\setminus\partial\tau\cap\Gamma} (k_1 (u_{h,x})^\epsilon n_x^\epsilon + k_2 (u_{h,y})^\epsilon n_y^\epsilon)\, v\, ds \\ &=: f_\tau(v)\,. \end{aligned} \tag{13}$$

The jump of the partial derivatives across an inner edge is divided by two and each half is appointed to the corresponding triangles. Thus we hope to approximate the actual error e of each element τ very well. Since the error is not calculated exactly, this method leads us only to an error indicator but not to an error estimator.

3.2 Dirichlet boundary condition

For the Dirichlet problem we chosse $H(\Omega) := H_0^1(\Omega)$.

With the same notation as before, we get (12) with two differences, first no shift is performed, i e, $c = 0$, and second the boundary integral across Γ vanishes. This is due to the fact that v satisfies the Dirichlet condition.

This leads us to the error equation

$$\begin{aligned} a(e, v) &= -\lambda_h b(u_h, v) - \sum_{\tau\in T} \int_\tau ((k_1 u_{h,x})_x + (k_2 u_{h,y})_y - q\, u_h)\, v\, d(x,y) \\ &\quad - \sum_{\epsilon\in E_1} \int_\epsilon (k_1 (u_{h,x})^\epsilon n_x^\epsilon + k_2 (u_{h,y})^\epsilon n_y^\epsilon)\, v\, ds. \end{aligned} \tag{14}$$

Restricting equation (14) to each triangle τ yields the error equation for each element again which is solved for all $v \in P_2^0(\tau)$

$$\begin{aligned} a_\tau(\tilde{e}\,,\, v) &= \int_\tau ((q - \lambda_h\, \rho)u_h - ((k_1 u_{h,x})_x + (k_2 u_{h,y})_y))\, v\ d(x,y) \\ &\quad -\frac{1}{2} \int\limits_{\partial\tau\setminus\partial\tau\cap\Gamma} (k_1 (u_{h,x})^\epsilon n_x^\epsilon + k_2 (u_{h,y})^\epsilon n_y^\epsilon)\, v\, ds \\ &=: f_\tau(v)\,. \end{aligned} \tag{15}$$

An extension to a boundary condition of mixed Dirichlet and Neumann typ can be performed accordingly.

3.3 Computation of the error indicator

For triangle τ the nodes are denoted by P_j, $j = 1, 2, 3$. Since the basis functions $v_i \in P_2^0(\tau)$ satisfy the conditions $v_i(P_j) = 0$, the linear space $P_2^0(\tau)$ is only of dimension three, i e, $i = 1, 2, 3$.

We suppose that the error approximation $\tilde{e}$ possesses the following representation on τ

$$\tilde{e} = \sum_{j=1}^{3} \alpha_j v_j \; .$$

Then $\tilde{e}$ can be calculated by error equation (13) respectively (15) as the solution of the following positive definite 3×3 system of linear equations

$$\sum_{j=1}^{3} a_\tau(v_j, v_i)\alpha_j \quad = \quad f_\tau(v_i) \; , \quad i = 1, 2, 3 \, .$$

If we choose v_i as the usual quadratic basis functions with value one at a midpoint of an edge and zero at all other nodes, the coefficients $a_\tau(v_i, v_j)$ are those of one block of the usual element stiffness matrix.

After computing the error approximation $\tilde{e}$, the error indicator η_τ for triangle τ is calculated by

$$\eta_\tau^2 \; := \; a_\tau(\tilde{e}, \tilde{e}) \; = \; \sum_{i=1}^{3} \alpha_i f_\tau(v_i) \; .$$

4 Adaptive concept

With that we have worked out a base for an adaptive algorithm. If for all triangles τ the error indicators are computed, we are able to determine by the size of each indicator in which part of the triangulation the approximation is bad. That triangles with the largest indicators will be refined in the next mesh.

With the construction up to now we are able to improve one single eigenvalue approximation with corresponding eigenfunction adaptively. If we want to approximate a given number of eigenvalues, numbered by $j = 1, \cdots, n$, in a predetermined part of the spectrum especially well, we have to use the information of all eigenvalue approximations from the actual mesh. For this reason we try to improve the by $\tilde{\omega}_j$ weighted sum of relative errors with respect to the eigenvalue approximation:

$$\sum_{j=1}^{n} \tilde{\omega}_j \left| \frac{\lambda_{h,j} - \lambda_j}{\lambda_j} \right|^2 \qquad \text{with} \qquad \tilde{\omega}_j \geq 0 \, . \tag{16}$$

If we add j as index to the developed formulas, we get from (16) in connection with (4)

$$\begin{aligned}\sum_{j=1}^{n} \tilde{\omega}_j \left|\frac{\lambda_{h,j}-\lambda_j}{\lambda_j}\right|^2 &\le \sum_{j=1}^{n} C_j^2 \tilde{\omega}_j \,||u_{h,j}-w_j||_a^2 = \sum_{j=1}^{n} C_j^2 \tilde{\omega}_j \left(\sum_{\tau\in T} a_\tau(e_j,e_j)\right) \\ &\approx \sum_{j=1}^{n} \underbrace{C_j^2 \tilde{\omega}_j}_{=:\omega_j} \left(\sum_{\tau\in T} \eta_{j,\tau}^2\right) = \sum_{\tau\in T}\left(\sum_{j=1}^{n} \omega_j\, \eta_{j,\tau}^2\right) \end{aligned} \tag{17}$$

a weighted error indicator

$$\tilde{\eta}_\tau^2 \;:=\; \sum_{j=1}^{n} \omega_j\, \eta_{j,\tau}^2\,. \tag{18}$$

So the user is able to improve the error term (16) adaptively by a special choice of the weights from (18). By that choice we can give the wanted eigenvalues priority . In particular the weight ω_j is chosen larger than zero only if there exists a prior interest in the improvement of eigenvalue λ_j. If there are no special knowledges about the eigenvalue approximation properties, we recommend to fix the weights in question uniformly equal to one. The adaptive improvement of one single eigenvalue is realized in this concept by choosing all weights equal to zero except the corresponding weight.

We propose the following algorithm 4.1 to mark out triangles for refinement by the weighted error indicator (18).

```
1.  choose $\gamma = 0.95$ and $\delta$ and compute $\tilde{\eta}_{\max} := \max \tilde{\eta}_\tau$
2.  let $i = 0$ (refinement counter for triangles)
3.  loop about all triangles $\tau$
    a) if triangle $\tau$ is unmarked yet and $\tilde{\eta}_\tau \ge \gamma\,\tilde{\eta}_{\max}$,
       then mark out triangle $\tau$ and let $i := i + 1$
    b) if $i \ge \delta$ or $i \ge$ number of triangles go to 5.
    end of loop
4.  let $\gamma := \gamma - 0.05$ and go to 3.
5.  generate a new mesh
```

Figure 4.1

This algorithm marks out all triangles for refinement by the control parameter γ step by step, starting with that which possess the largest error indicator $\tilde{\eta}_\tau$. By the maximum number of triangles δ to be refined per step, it is possible to control the growth of the mesh size.

5 Adaptive algorithm

The developed adaptive concept in Figure 4.1 describes step 3 of the adaptive algoritm (see Figure 5.1).

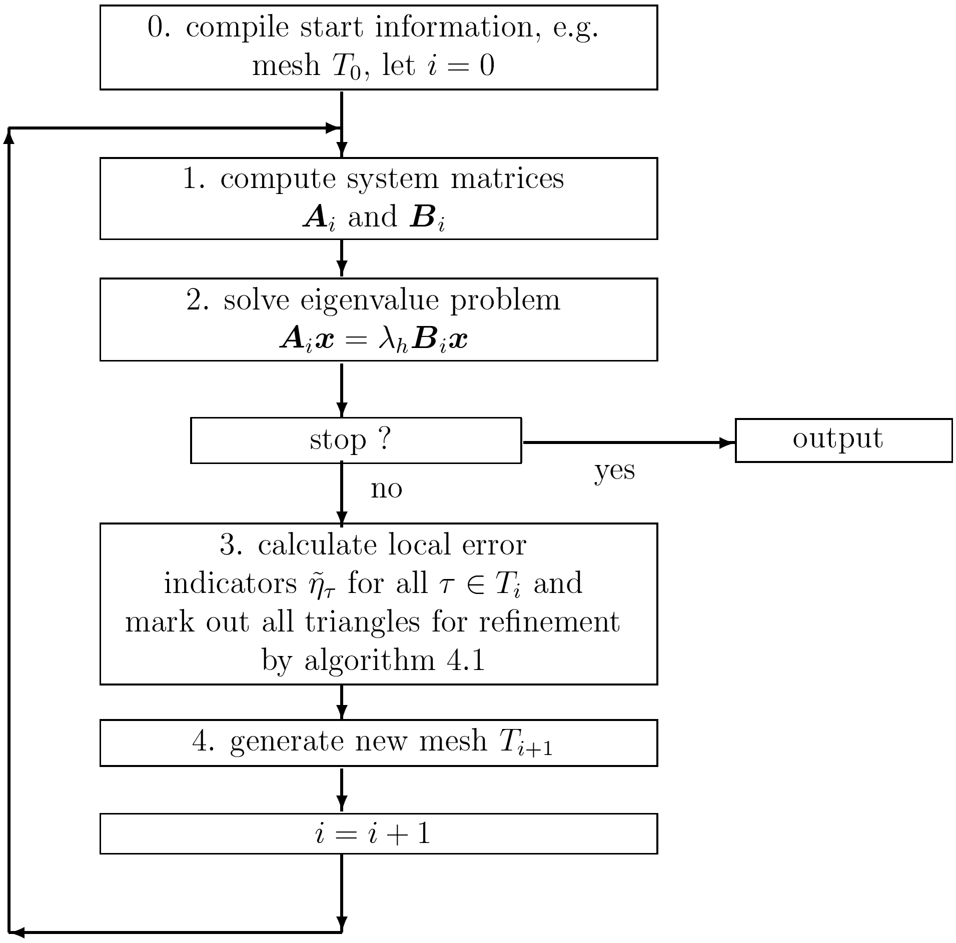
0. compile start information, e.g.
mesh T_0, let $i = 0$
1. compute system matrices
$\boldsymbol{A}_i$ and $\boldsymbol{B}_i$
2. solve eigenvalue problem
$\boldsymbol{A}_i\boldsymbol{x} = \lambda_h \boldsymbol{B}_i\boldsymbol{x}$
stop ?
yes
output
no
3. calculate local error
indicators $\tilde{\eta}_\tau$ for all $\tau \in T_i$ and
mark out all triangles for refinement
by algorithm 4.1
4. generate new mesh T_{i+1}
$i = i + 1$

Figure 5.1 adaptive algorithm

Triangles are refined in step 4 in accordance with the rules of the software package PLTMG from Bank [3].

6 Examples

Both examples belong to the following class of eigenvalue problems

$$\int_\Omega u_x v_x + u_y v_y \, d(x,y) \quad = \quad \lambda \int_\Omega uv \, d(x,y) \quad \forall\, v \in H(\Omega)\,. \tag{19}$$

6.1 Vibration of a membrane

For the vibration of a plane membrane we consider the section of a circle Ω (see Figure 6.1) with inner angle $\omega = 5\pi/3$ and radius $R = 1$. The eigenfunctions u of the Dirichlet problem satisfy equation (19) with $H(\Omega) = H_0^1(\Omega)$.

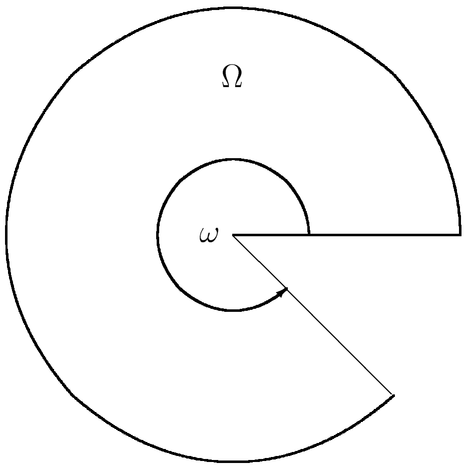

Figure 6.1 section of a circle

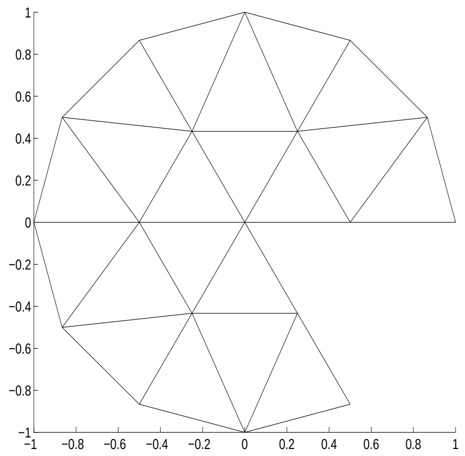

Figure 6.2
starting triangulation T_0

In [6] all eigenvalues with corresponding eigenfunctions are calculated exactly. It turns out that some eigenfunctions possess a singularity at the re-entrant corner, in particular we have $u_1, u_6 \notin C^1(\Omega) \cap C_0(\overline{\Omega})$ (see Figure 6.3 and 6.5). In this case we can use the adaptive algorithm with advantage.

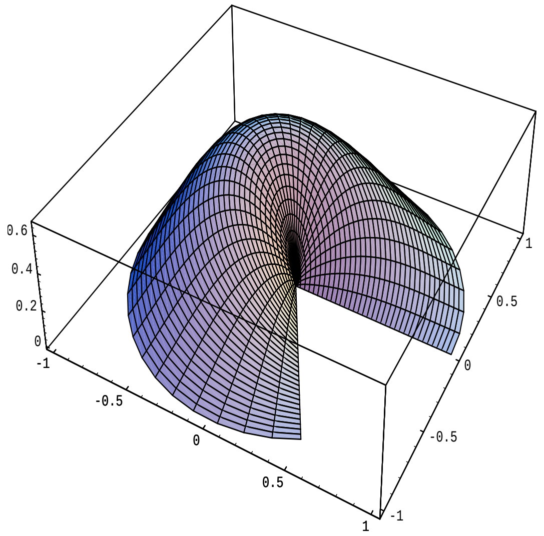

Figure 6.3 u_1 for $\lambda_1 = 10.775105$

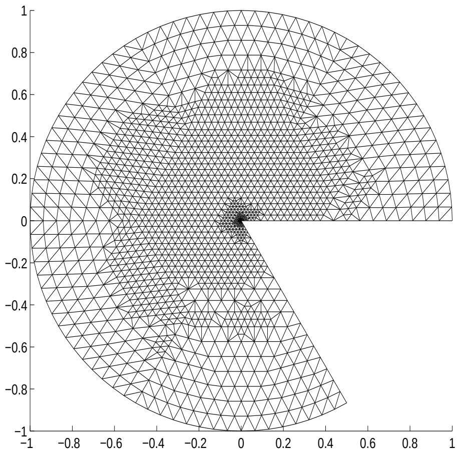

Figure 6.4
mesh T_8 for u_1,
(dimension = 1645) with $\delta = 100$,
weights $\omega_1 = 1$, $\omega_{i \neq 1} = 0$

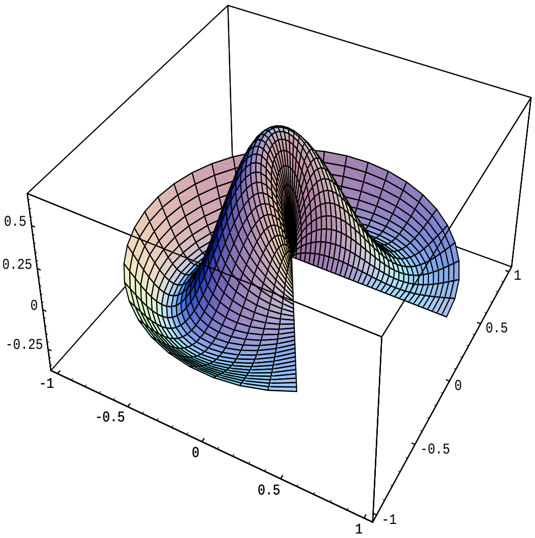

Figure 6.5 u_6 for $\lambda_6 = 41.368167$

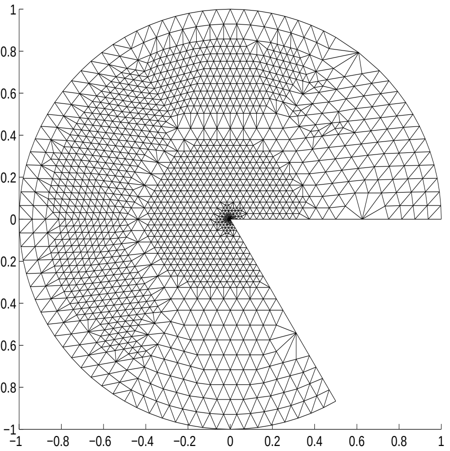

Figure 6.6
mesh T_8 for u_6,
(dimension $= 1649$) with $\delta = 100$,
weights $\omega_6 = 1$, $\omega_{i\neq 1} = 0$

The Figures 6.4 and 6.6 illustrate that the adaptive algorithm refines the mesh strongly at a re-entrant corner, i e, the part of the domain with absence of smoothness. For smooth eigenfunctions as u_7 the algorithm generates a mesh which is refined uniformly (see Figure 6.8), where the increase is controled by the parameter δ.

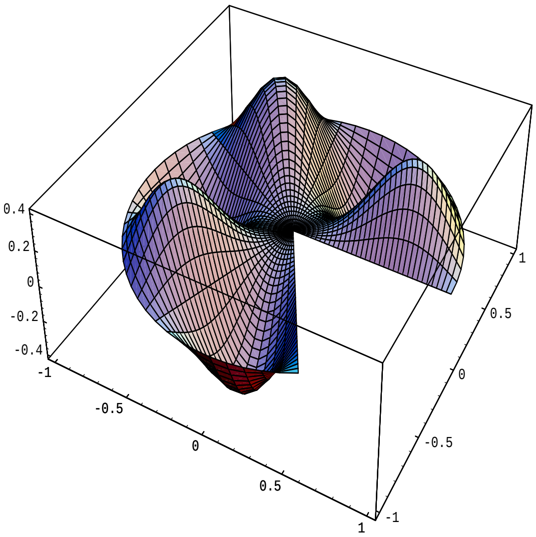

Figure 6.7 u_7 for $\lambda_7 = 50.531635$

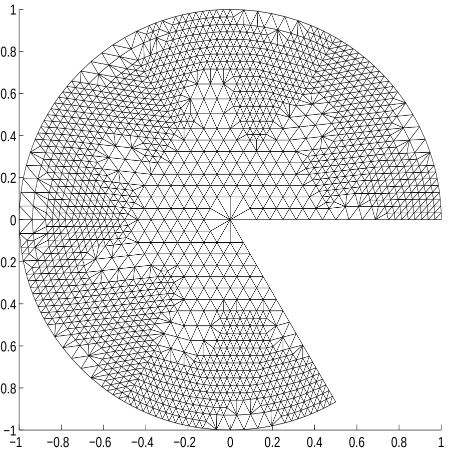

Figure 6.8
mesh T_8 for u_7,
(dimension $= 1818$) with $\delta = 100$,
weights $\omega_7 = 1$, $\omega_{i\neq 1} = 0$

6.2 Longitudinal section of a car

The computation of acoustical eigenfrequencies with corresponding waves for the interior of a car (see Figure 6.9) with acoustical stiff walls leads us to a Neumann problem (cf Schwarz [8]). This is described by (19) with $H(\Omega) = H^1(\Omega)$. Since we have the Neumann condition on the complete boundary, the smallest eigenvalue with corresponding eigenfunction is well known, i e, $\lambda_1 = 0$ and $u_1 \equiv const$. The resulting theoretical and numerical problems can be removed by the discussed spectral shift (see (11)). We do not pay attention to this eigenpair in the following because it is well known and not interesting for the adaptive algorithm.

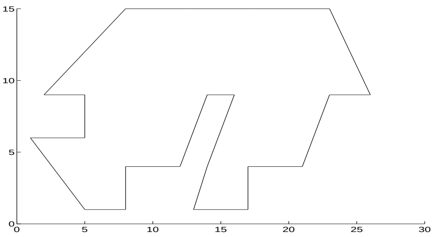

Figure 6.9 region Ω for the longitudinal section of a car

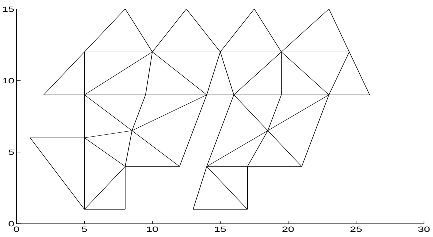

Figure 6.10 starting triangulation T_0

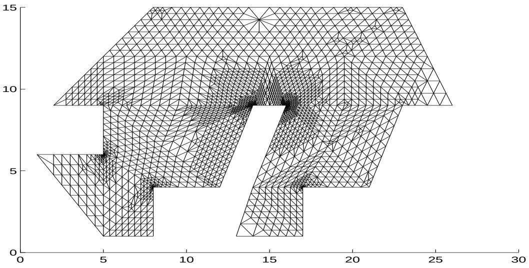

Figure 6.11
mesh T_7 for u_2,
(dimension = 2203) with $\delta = 150$,
weights $\omega_2 = 1, \omega_{i \neq 2} = 0$

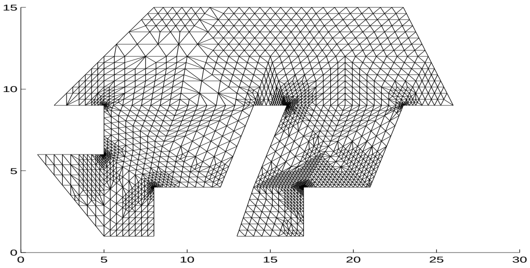

Figure 6.12
mesh T_7 for u_3,
(dimension = 2162) with $\delta = 150$,
weights $\omega_3 = 1, \omega_{i \neq 3} = 0$

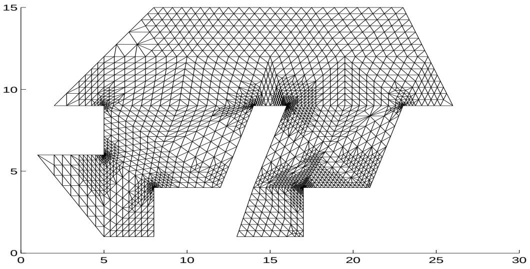

Figure 6.13
mesh T_7 (dimension = 2155) with $\delta = 150$,
weights $\omega_2 = \omega_3 = 1, \omega_{i \neq 2,3} = 0$

There are re-entrant corners in this example too which are responsible for lacking smoothness of some eigenfunctions. Thus we can use the adaptive algorithm here with advantage.

We have given preference to only one eigenvalue at the adapative algorithm in Figures 6.11 and 6.12. This leads to strong refinements at the re-entrant corners with distinct accentuation. Figure 6.13 shows the comparable mesh, where both eigenvalues have the same weights. With regard to the number of triangles we see a superposition of the characterics from both preceding Figures.

Because the exact eigenvalues are unknown, we have calculated eigenvalue approximations $\tilde{\lambda}$ with the PDE Toolbox from MATLAB. These reference values are shown in Table 6.14. Thereby the starting mesh has been refined uniformly by 32768 linear triangles with a resulting dimension of 16769 for the system of linear equations.

$\tilde{\lambda}_1 = 0$	$\tilde{\lambda}_6 = 0.1367$	$\tilde{\lambda}_{11} = 0.3454$	$\tilde{\lambda}_{16} = 0.5583$
$\tilde{\lambda}_2 = 0.01263$	$\tilde{\lambda}_7 = 0.1423$	$\tilde{\lambda}_{12} = 0.3735$	$\tilde{\lambda}_{17} = 0.615$
$\tilde{\lambda}_3 = 0.04449$	$\tilde{\lambda}_8 = 0.1998$	$\tilde{\lambda}_{13} = 0.3848$	$\tilde{\lambda}_{18} = 0.6834$
$\tilde{\lambda}_4 = 0.05627$	$\tilde{\lambda}_9 = 0.2703$	$\tilde{\lambda}_{14} = 0.3953$	$\tilde{\lambda}_{19} = 0.731$
$\tilde{\lambda}_5 = 0.1159$	$\tilde{\lambda}_{10} = 0.2902$	$\tilde{\lambda}_{15} = 0.4656$	$\tilde{\lambda}_{20} = 0.7658$

Table 6.14 eigenvalue approximations for the longitudinal section of a car

The main idea of the developed adaptive algorithm is to use a given number of triangles most efficiently. For this reason the size and the place of the used triangles should be determined suitably so that the wanted eigenvalues are approximated very well. Both of the following Tables 6.15 and 6.16 confirm this effect for the used algorithm.

	$\omega_2 = 1$	$\omega_3 = 1$	$\omega_4 = 1$	$\omega_5 = 1$	$\omega_6 = 1$	$\omega_7 = 1$	$\omega_8 = 1$
r_2	$3.2e-6$	$2.5e-5$	$3.0e-5$	$2.2e-5$	$2.0e-5$	$1.5e-5$	$5.0e-5$
r_3	$1.8e-5$	$4.5e-6$	$1.8e-5$	$4.5e-5$	$1.7e-5$	$1.1e-5$	$7.6e-5$
r_4	$2.2e-5$	$1.3e-5$	$5.0e-6$	$3.0e-5$	$1.0e-5$	$2.2e-5$	$3.9e-5$
r_5	$1.8e-5$	$1.8e-5$	$1.7e-5$	$5.2e-6$	$1.6e-5$	$2.1e-5$	$1.6e-5$
r_6	$1.8e-5$	$2.1e-5$	$1.2e-5$	$3.4e-5$	$6.6e-6$	$1.8e-5$	$4.2e-5$
r_7	$4.4e-5$	$2.0e-5$	$3.9e-5$	$8.3e-5$	$2.8e-5$	$7.0e-6$	$1.2e-5$
r_8	$2.6e-5$	$3.0e-5$	$3.1e-5$	$1.7e-5$	$2.6e-5$	$2.8e-5$	$6.5e-6$

Table 6.15 relative error r_j for λ_j at mesh T_8 for $\delta = 150$

The strategy of choosing only one weight different from zero results in a very good approximation of the corresponding eigenvalue as we can pick off Table 6.15. The other eigenvalues are approximated evidently worser.

strategy		μ_4	σ_4	μ_5	σ_5
1.	$\omega_2 = 1$	1.74	0.73	1.36	0.59
2.	$\omega_3 = 1$	1.69	0.50	1.22	0.32
3.	$\omega_4 = 1$	1.80	0.59	1.40	0.50
4.	$\omega_5 = 1$	2.19	1.26	1.86	1.33
5.	$\omega_6 = 1$	1.74	0.65	1.32	0.46
6.	$\omega_7 = 1$	1.69	0.34	1.24	0.27
7.	$\omega_8 = 1$	2.15	0.97	1.90	1.01
8.	$\omega_2 = \omega_3 = 1$	1.51	0.44	1.15	0.38
9.	$\omega_3 = \omega_4 = 1$	1.51	0.42	1.15	0.38
10.	$\omega_4 = \omega_5 = 1$	1.51	0.58	1.22	0.53
11.	$\omega_5 = \omega_6 = 1$	1.58	0.66	1.15	0.47
12.	$\omega_6 = \omega_7 = 1$	1.38	0.20	1.14	0.19
13.	$\omega_7 = \omega_8 = 1$	1.46	0.19	1.16	0.16
14.	$\omega_2 = \omega_3 = \omega_4 = 1$	1.45	0.42	1.12	0.41
15.	$\omega_3 = \omega_4 = \omega_5 = 1$	1.39	0.42	1.13	0.40
16.	$\omega_4 = \omega_5 = \omega_6 = 1$	1.50	0.57	1.11	0.42
17.	$\omega_5 = \omega_6 = \omega_7 = 1$	1.35	0.22	1.07	0.19
18.	$\omega_6 = \omega_7 = \omega_8 = 1$	1.43	0.17	1.08	0.13
19.	$\omega_2 = \cdots = \omega_5 = 1$	1.39	0.43	1.08	0.41
20.	$\omega_3 = \cdots = \omega_6 = 1$	1.39	0.42	1.08	0.35
21.	$\omega_4 = \cdots = \omega_7 = 1$	1.35	0.27	1.05	0.21
22.	$\omega_5 = \cdots = \omega_8 = 1$	1.33	0.25	1.05	0.19
23.	$\omega_2 = \cdots = \omega_6 = 1$	1.38	0.43	1.08	0.39
24.	$\omega_3 = \cdots = \omega_7 = 1$	1.35	0.26	1.04	0.21
25.	$\omega_4 = \cdots = \omega_8 = 1$	1.35	0.30	1.07	0.20
26.	$\omega_2 = \cdots = \omega_7 = 1$	1.34	0.27	1.02	0.24
27.	$\omega_3 = \cdots = \omega_8 = 1$	1.35	0.28	1.04	0.21
28.	$\omega_2 = \cdots = \omega_8 = 1$	1.35	0.32	1.02	0.23

Table 6.16
mean values μ_i of the error percentages for $\lambda_2, \cdots, \lambda_8$
with deviation σ_i for the meshes T_i, $i = 4, 5$ and different weights

If the lower part of the spectrum, e.g. the eigenvalues $\lambda_2, \cdots, \lambda_8$, should be approximated very well, we have to adjust the matching weights of the local error indicator (18). Table 6.16 presents different strategies. Different strategies yielded different dimensions of the eigenvalue problems that had to be solved. At mesh T_4 the mean value of these dimensions was equal to 489 with small deviation due to the parameter $\delta = 50$. At T_5 we got the mean value 632 for the dimension. The values μ_i and σ_i in Table 6.16 have been calculated by linear interpolation of the 4. and 5. mesh with evaluation at the mean values 489 and 632 of the dimension to get a better comparison. Therefore we obtain from these values only a qualitative statement.

The strategies from row one to seven in Table 6.16, where only one weight is different from zero, lead to a comparatively large mean value of error percentages, according to the result of Table 6.15. By taking more weights, μ_i can be decreased. Finally, μ_i has nearly the smallest value if we use for all eigenvalues the same weight (see row 28). The deviations show a similar tendency.

As strategy for a uniform approximation of a fixed by the user predetermined number of eigenvalues from the lower part of the spectrum, we recommend the same weights in the local error indicator (18) for the eigenvalues in question. This strategy should be used if there is no further information about the behaviour of the approximation of the eigenvalues. If it turns out during the calculation that the behaviour of approximation differs from one to another eigenvalue, we are able to influence this behaviour by the weights.

7 Conclusion

We describe in this article how static error indicators can be applied to eigenvalue problems. Error indicators for each triangle and each eigenvalue are the result of this investigation. They can be condensed for a given number of eigenvalues to over-all local error indicators by using weights. The triangulations of the resulting adaptive algorithm are adjusted step by step by a suitable choice of the weights so that we have a strong refinement at the problem zones which are caused by e.g. re-entrant corners. This algorithm fits size and place of the triangles that are at the users disposal so that the predetermined part of the spectrum can be approximated better than other parts. Thus it is also possible to overcome a principle of the finite element method by which the eigenvalues are approximated worse with their increasing size if uniform triangulations are used.

References

[1] I. Babuška and J. Osborn,
Eigenvalue Problems, in: P.G. Ciarlet and J.L. Lions (eds.), Handbook of Numerical Analysis, Vol.II, Finite Element Methods (Part 1)
(North-Holland, Amsterdam, New York, Oxford, Tokio 1991.)

[2] I. Babuška and W.C. Rheinboldt,
A-posterior error estimates for the finite element method.
(Internat. J. Numer. Meth. Engrg. 12 (1978) 1597 — 1615.)

[3] R.E. Bank,
PLTMG: A software package for solving elliptic partial differential equations. Users Guide 7.0
(SIAM,Philadelphia,1994).

[4] R.E. Bank and A. Weiser,
Some a posteriori error estimators for elliptic partial differential equations.
(Math. Comp. Vol.44 No.170 (1985) 283 — 301.)

[5] H. Goering,H.G. Roos and L. Tobiska,
Finite Element Methode.
(Akademie Verlag, Berlin, 3.Aufl., 1993).

[6] K. Rothe,
Randeigenwertprobleme im Kreissektor mit einspringender Ecke.
(Hamburger Beiträge zur Angewandten Mathematik, Bericht 37, Reihe B, Universität Hamburg, 2002).

[7] K. Rothe,
Ein adaptiver finite Elementalgorithmus für Randeigenwertprobleme.
(Hamburger Beiträge zur Angewandten Mathematik, Bericht 38, Reihe B, Universität Hamburg, 2003).

[8] H.R. Schwarz,
Methode der Finiten Elemente.
(Teubner, Stuttgart, 3.Aufl., 1991).