ANISOTROPIC MESH ADAPTATION METHOD FOR COMPRESSIBLE FLOW

Vít DOLEJŠÍ

Charles Univeristy Prague
Faculty of Mathematics and Physics
Department of Numerical Mathematics
Sokolovska 83
186 75 Prague
Czech Republic
e-mail: dolejsi@karlin.mff.cuni.cz

Prague 2003

Contents

Chapter 1

Anisotropic mesh adaptation method

1 Introduction

Numerical solution of complicated problems of computational physics and mathematical engineering gives rise to the necessity of application of adaptive methods, which allow computation of a sufficiently precise solution without enormous requirements for memory and CPU time. The aim of the adaptive methods is to find a suitable discretization of computational domain where sufficiently accurate numerical solution can be computed and moreover the number of unknowns is reasonable. There exists a very large range of different adaptive methods developed during several last decades. Many adaptive methods are based on a priori or a posteriori error analysis. Taking into account the error estimations we define a suitable error indicator. Using the values of numerical solution calculated on the given mesh we compute the values of error indicator for all elements. Then elements, where the value of the indicator is higher than a given tolerance, are refined and we obtain a new grid. This approach is usual for finite element methods. The list of authors dealing with such type of adaptive technique is very large, we cite only the review [Ver96]. Alternative approach is the use of the dual problem, see the review article [Ran98].

Rather different approach is the *anisotropic mesh adaptation* (AMA). It is based on the control of the interpolation error of a piecewise polynomial interpolation of the exact solution on the triangular grid. The mesh adaptation criterion checks the behaviour of the second order derivatives of the solution of the considered problem.

The advantage of AMA is that it can be used without any modification for arbitrary boundary value problem and arbitrary numerical method (finite element method, finite volume method, discontinuous Galerkin method, etc.). The disadvantage of this approach is that we have no control of the discretization error of the numerical solution in fact.

In [DS89], [DS91], [Sim94], it is discussed how these mechanisms arise in the theory of optimal error control, using simple model mesh generation problems. A general recovery technique was developed for determining the derivatives of the finite element solutions in [ZZ92a] and a posteriori error estimation was discussed in [ZZ92b]. The application of AMA for computational fluid dynamics has been performed in INRIA group (see, e.g., [CDBG$^+$96], [DHM95], [BHF97]) and Habashi's group (see, e.g., [FVD$^+$96], [HFaY$^+$96]). In monograph [Ape99], there were presented the anisotropic local interpolation error estimates for several types of two- and three-dimensional finite elements and a priori estimates of the discretization error for model problems with edge singularities or boundary layers.

Our presented mesh adaptation technique for unstructured grids is based on such type of adaptive methods. The original approach of our algorithm is the use of the *edge-based* optimization criterion. In this work we summarize and develop the first formulation of AMA and its use for the numerical solution of compressible flow and heat conduction problem, see [Dol98a], [DF], [Dol01], [BDM] and [DF02].

Although we try to give a mathematically rigorous explanation, we are not able to avoid several rather heuristic steps. On the other hand the presented method is a very efficient tool for the numerical solution of various practical problems. The presented mathematical treatment of the anisotropic mesh adaptation is almost identical for two and three dimensional problems. Therefore it is carried out for dimension d, $d = 2, 3$. The space discretization is called *mesh*, which is a triangulation ($d = 2$) or a tetrahedrization ($d = 3$), and *element* is a triangle ($d = 2$) or tetrahedron ($d = 3$).

The summary of this chapter is the following. At the beginning we consider a *scalar boundary value problem* (BVP) and suppose that its exact solution is known. In Section 2 we derive a necessary condition which guarantees that the discretization error is under a given tolerance. As the necessary condition is not suitable for practical computation, we introduce its stronger (but simpler) version. Moreover, using several heuristic considerations, we define the concept of an *optimal mesh.* Finally, we introduce the *quality parameter* which measures the quality of a mesh.

In Section 3 we describe the AMA algorithm which creates the optimal mesh for a known exact solution. Section 4 shows how to use the AMA algorithm for practical computation, where the exact solution is not known. Several test examples are given in Section 5, where the efficiency of AMA is demonstrated. Section 6 and 7 show the practical application of AMA to the numerical simulation of inviscid as well as viscous flow and heat conduction problems, respectively. For the clarity of the treatment, some (simple but long) proofs of Lemmas are given in Appendix.

2 Interpolation error control

2.1 Necessary condition

Let us consider a scalar boundary value problem (BVP) in the set Ω, where $\Omega \subset I\!R^d$ is a bounded domain. By $\overline{\Omega}$ and $\partial\Omega$ we denote the closure and boundary of Ω, respectively. We introduce the weak formulation of (BVP): Find the function $u : \Omega \to I\!R$ satisfying

$$u \in U, \tag{2.1}$$
$$\mathcal{L}(u, v) = f(v) \quad \forall v \in V, \tag{2.2}$$
$$\mathcal{B}(u) \in V. \tag{2.3}$$

Here U and V are the trial and test function spaces, respectively, (2.2) represents the weak form of the partial differential equation under consideration and (2.3) guarantees that the prescribed boundary conditions are fulfilled in the sense of traces. The function u satisfying (2.1) – (2.3) is called the *exact solution.* Let us suppose that the problem (2.1) – (2.3) has a unique exact solution.

Very often the numerical solution of (2.1) – (2.3) is sought in the space of piecewise polynomial functions and computed by a suitable numerical method (e.g. finite element method, finite volume method, discontinuous Galerkin method). To this end the domain Ω is replaced by its polygonal approximation Ω_h and the mesh $\mathcal{T}_h$ is constructed. The properties of $\mathcal{T}_h$ are presented in Definition 1. The *approximate solution* of problem (2.1) – (2.3) is defined as a function $u_h : \overline{\Omega}_h \to I\!R$ satisfying

$$u_h \in U_h, \tag{2.4}$$

$$\mathcal{L}_h(u_h, v_h) = f_h(v_h) \quad \forall v_h \in V_h, \tag{2.5}$$

$$\mathcal{B}_h(u_h) \in V_h. \tag{2.6}$$

Here

$$U_h = \left\{ u_h \in L^1(\Omega_h); u_h|_T \in P_k(T) \ \forall T \in \mathcal{T}_h \right\}, \tag{2.7}$$

and $P_k(T)$ denotes the set of all polynomials of degree $\le k$ ($k \ge 0$) on the element T and $\mathcal{L}_h$, f_h, $\mathcal{B}_h$, V_h are the discrete analogue of $\mathcal{L}$, f, $\mathcal{B}$, V, respectively. Both possibilities $U_h \subset U$ and $U_h \not\subset U$ are admissible. Let us suppose, that the *discrete problem* (2.4) – (2.6) has a unique solution u_h.

In order to measure how close the approximate solution u_h is to the exact solution u, we define the *discretization error* as

$$e_h \equiv \|u - u_h\|_X, \tag{2.8}$$

where $\|\cdot\|_X$ is a suitable norm such that the expression $\|v\|_X$ has a sense $\forall v \in U \cup U_h$. (If $U_h \subset U$ we can put $\|\cdot\|_X \equiv \|\cdot\|_U$, where $\|\cdot\|_U$ is the norm in the space U.)

Our aim is, for a given tolerance $\omega > 0$, to compute the approximate solution satisfying

$$e_h \le \omega. \tag{2.9}$$

In what follows, we present the necessary condition for the relation (2.9) and show with numerical examples how its satisfaction can be used for the construction of adaptive meshes.

We denote by $\mathcal{O}(U, U_h)$ the set of all operators from U to U_h, i.e.

$$\mathcal{O}(U, U_h) = \{\Pi, \Pi : w \in U \mapsto \Pi w \in U_h\}. \tag{2.10}$$

Let us consider the minimization problem: Find $\Pi_h \in \mathcal{O}(U, U_h)$ such that

$$\|w - \Pi_h w\|_X = \min_{w_h \in U_h} \|w - w_h\|_X \quad \forall w \in U. \tag{2.11}$$

The existence of Π_h follows from the fact that U_h is a finite dimensional space. The operator Π_h may not be uniquely determined, its uniqueness depends on the choice of $\|\cdot\|_X$.

It is evident that for the approximate solution u_h of (2.4) – (2.6)

$$\|u - \Pi_h u\|_X \le e_h. \tag{2.12}$$

This is the crucial point of the proposed adaptation strategy: Any numerical method for the solution of (2.1) – (2.3) computes the approximate solution u_h with the discretization error which is bounded from below by $\|u - \Pi_h u\|_X$, where u is the exact solution of (2.1) – (2.3). Then the *necessary condition* to fulfill (2.9) is

$$\|u - \Pi_h u\|_X \le \omega. \tag{2.13}$$

For a given ω, the condition (2.13) is satisfied if the mesh is sufficiently fine. This can be achieved

(i) by a global mesh refinement, which is rather inefficient from the point of view of computer memory and CPU time in further computations,

(ii) by a local grid refinement/coarsening/alignment strategy, which is the subject–matter of Chapter 1.

It is evident that the magnitude of $\|u - \Pi_h u\|_X$ strongly depends on the choice of U_h and therefore on the choice of $\mathcal{T}_h$. In the first part of this chapter, for a given $\omega > 0$ and given function $u \in C^2(\Omega)$, we describe the construction of the mesh $\mathcal{T}_h$ for which the function $\Pi_h u$ satisfies (2.13) and the mesh has as small a number of elements as possible. In the second part, we apply these results to the numerical solution of physically relevant problems.

The basic principle of AMA can be illustrated on the following 1D example. Let $\langle A, B\rangle \in I\!R^1$ be an interval and $u \in C^2(\langle A, B\rangle)$ the given function, see Figure 1.1. We introduce two partitions $\mathcal{T}_{h_1}$ and $\mathcal{T}_{h_2}$ of the interval $\langle A, B\rangle$. The former partition (having 5 intervals) is uniform whereas the latter (having 3 intervals) is anisotropic. Let us consider a discontinuous piecewice linear approximation. Let $\Pi_{h_1}u$ and $\Pi_{h_2}u$ be the functions given by (2.11) on the partitions $\mathcal{T}_{h_1}$ and $\mathcal{T}_{h_2}$, respectively, where U_h is given by (2.15) and we set $\|\cdot\|_X = \|\cdot\|_{L^\infty(\langle A,B\rangle)}$. Moreover we put

$$\|u - \Pi_{h_1}u\|_X = \omega_1, \qquad \|u - \Pi_{h_2}u\|_X = \omega_2, \tag{2.14}$$

where the values $\max_{x\in\langle A,B\rangle} |u(x) - \Pi_{h_i}u(x)|,\ i = 1, 2$ are plotted on the Figure 1.1.

We observe that $\omega_2 < \omega_1$ and moreover $\#\mathcal{T}_{h_2} < \#\mathcal{T}_{h_1}$, where $\#\mathcal{T}_h$ denotes the number of subintervals of partition $\mathcal{T}_h$. Therefore, the anisotropic partition $\mathcal{T}_{h_2}$ is more suitable for the piecewise linear capturing of u.

In practical problems, of course, the exact solution is not a priori known. In that case we compute the approximate solution u_h on the starting mesh $\mathcal{T}_h$, we smooth u_h and denote this smoothing as $\tilde{u}$ and apply the AMA algorithm for $\tilde{u}$, see Section 4.

2.2 ω-minimal mesh

From the practical point of view, we shall use in (2.7) the discontinuos piecewise linear polynomials, i.e.

$$U_h = \left\{u_h \in L^1(\Omega_h); u_h|_T \in P_1(T)\ \forall T \in \mathcal{T}_h\right\}, \tag{2.15}$$

and we set

$$\|\cdot\|_X = \|\cdot\|_{L^\infty(\Omega_h)}. \tag{2.16}$$

The example of a function from U_h is viewed in Figure 1.2. This choice of U_h covers wide range of numerical methods, piecewise constant approximation used in finite volume method (e.g. [Fei93], [Krö97], [KR94]) and conforming (e.g. [Cia79]), nonconforming (e.g. [ADFF98], [DA96], [Tem77]) and discontinuous Galerkin finite element method (e.g. [Coc]). On the other hand, the use of AMA does not require the piecewise linear approximation and a higher order approximation can be easily adopted for the efficient use of AMA.

Definition 1 *Let $\Omega \subset I\!R^d$. We say that $\mathcal{T}_h = \{T_i\}_{i\in I}$ (I = an index set) is a* mesh *of Ω if:*

1. *$T_i \in \mathcal{T}_h$ is a closed simplex (triangle ($d = 2$) or tetrahedron ($d = 3$)).*

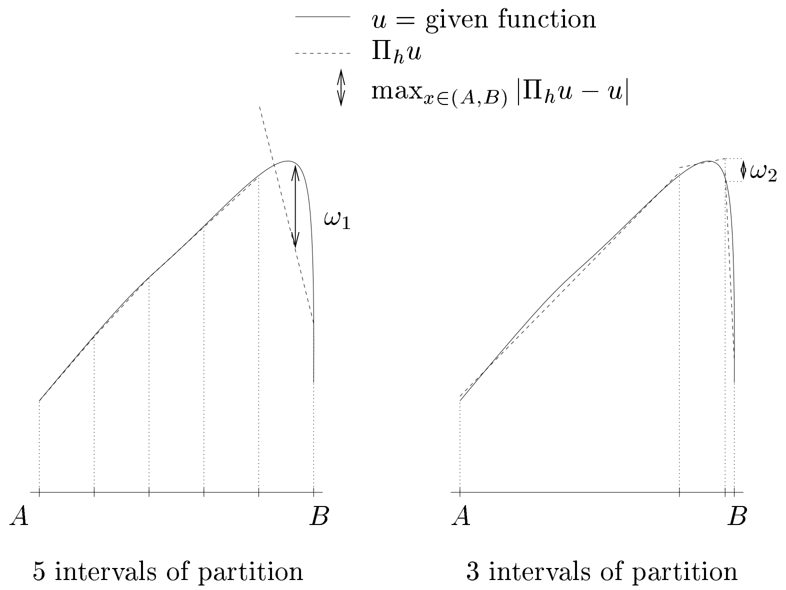

Figure 1.1: Motivation of AMA: partition of the interval AB, u is the given function, $\Pi_{h_1}u$ (left) and $\Pi_{h_2}u$ (right) are discontinuous piecewise linear functions given by (2.11), $\max_{x\in(A,B)} |u - \Pi_h u|$ is denoted by $\updownarrow$.

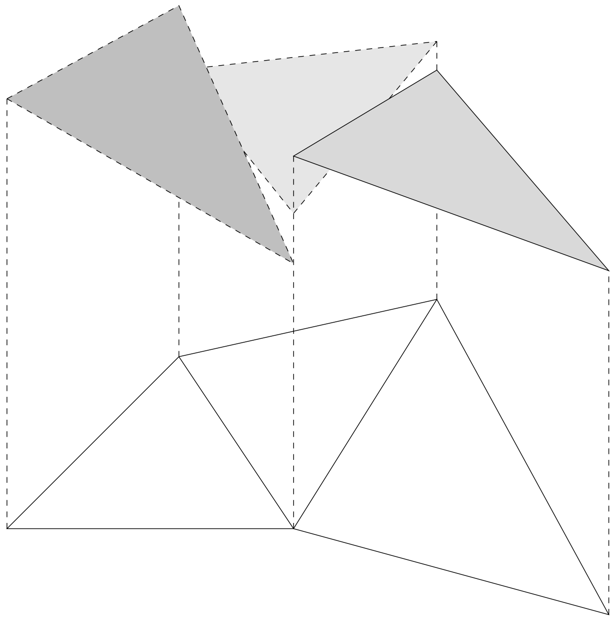

Figure 1.2: Example of a discontinuous piecewise linear function.

2. *$\bigcup_{i\in I} T_i = \overline{\Omega_h}$, where Ω_h is a polygonal approximation of Ω.*

3. *Let x_i be the barycentre of T_i, then $x_i \in \Omega\ \forall T_i \in \mathcal{T}_h$.*

4. *We denote by $\sigma_h = \{P_1, \ldots, P_N\}$ the set of all vertices of all $T \in \mathcal{T}_h$. We suppose that $\sigma_h \cap \partial\Omega_h \subset \partial\Omega$, $\sigma_h \subset \overline{\Omega}$.*

5. *For any $T_i, T_j \in \mathcal{T}_h$, $T_i \neq T_j$, $T_i \cap T_j$ are either disjoint or their intersection $T_i \cap T_j$ is a common vertex or $T_i \cap T_j$ is a common edge or $T_i \cap T_j$ is a common face (only for $d = 3$).*

The step size h *of the mesh is given by*

$$h = \max_{i\in I} \operatorname{diam}(T_i), \tag{2.17}$$

where $\operatorname{diam}(T_i)$ *denotes the diameter of* T_i.

Without proof we present the following Lemma:

Lemma 1 *Let $\Omega \in \mathbb{R}^d$ be a computational domain and $\bar{h} > 0$. Then there exists a mesh $\mathcal{T}_h$ of Ω with the step size h, $h \leq \bar{h}$.*

Definition 2 *Let $w \in C^2(\overline{\Omega} \cup \Omega_h)$ and $\mathcal{T}_h = \{T_i\}_{i\in I}$ be a mesh of Ω. Let $x_i = (x_i^{(1)}, \ldots, x_i^{(d)})$ be the centre of gravity of $T_i \in \mathcal{T}_h$. We say that the operator $r_h \in \mathcal{O}(C^2(\overline{\Omega} \cup \Omega_h), U_h)$ is the* ∇–interpolation *if*

$$\begin{aligned} r_h w(x_i) &= w(x_i) \qquad \forall T_i \in \mathcal{T}_h, \\ \nabla r_h w(x_i) &= \nabla w(x_i) \qquad \forall T_i \in \mathcal{T}_h. \end{aligned} \tag{2.18}$$

Lemma 2 *Let $w \in C^2(\overline{\Omega} \cup \Omega_h)$ and $\mathcal{T}_h = \{T_i\}_{i\in I}$ be a mesh of Ω. Let U_h, $\|\cdot\|_X$, Π_h and r_h are given by (2.15), (2.16), (2.11) and (2.18), respectively. Then*

$$\|w - \Pi_h w\|_X \leq \|w - r_h w\|_X. \tag{2.19}$$

Proof. As $r_h w \in U_h$, the assertion follows immediately from (2.11). □

Remark 1 *Lemma 2 is valid also for more general $\|\cdot\|_X$.*

Definition 3 *Let u be the exact solution of the considered problem (2.1) – (2.3) in the computational domain Ω and r_h be the operator (2.18), where U_h is given by (2.15). The* interpolation error function $E_\mathrm{I} : \Omega \cap \Omega_h \to \mathbb{R}^1$ *is defined by*

$$E_\mathrm{I}(x) = |u(x) - r_h u(x)|, \text{ for almost every } x = (x^{(1)}, \ldots, x^{(d)}) \in \Omega \cap \Omega_h. \tag{2.20}$$

Lemma 2 allows us to control the necessary condition (2.13) by the interpolation error function E_I. The condition

$$\|u - r_h u\|_X \leq \omega \tag{2.21}$$

is stronger than the necessary condition (2.13) but as it is easily computable, we use it for further consideration. Therefore our aim is to construct a mesh $\mathcal{T}_h$ of Ω so that the interpolation error function $E_\mathrm{I}(x)$ is bounded from above by a given tolerance. On the other hand, we require that the number of elements of the mesh is as small as possible (in order to save a memory and CPU time). The demonstration of the efficiency of (2.21) is given in Section 5.1

Definition 4 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ and $\mathcal{T}_h = \{T_i\}_{i\in I}$ be a mesh of Ω. Let E_I be given by (2.20). We say that $\mathcal{T}_h$ is a ω–*mesh

$$E_\mathrm{I}(x) \le \omega \qquad \forall x \in \Omega \cap \Omega_h. \tag{2.22}$$

Theorem 1 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ and $\omega > 0$. Then there exists an ω-mesh.*

Proof. For $u \in C^2(\overline{\Omega} \cup \Omega_h)$ there exists a constant $\hat{c} > 0$ such that

$$\left| \frac{\partial^2 u(x)}{\partial x^{(i)} \partial x^{(j)}} \right| \le \hat{c}, \quad \forall i, j = 1, \dots, d \;\forall x \in \overline{\Omega}. \tag{2.23}$$

Lemma 1 gives that there exists a mesh $\mathcal{T}_h$ with step size

$$h \le \sqrt{\frac{2\omega}{\hat{c}d^2}}. \tag{2.24}$$

We prove that $\mathcal{T}_h$ is a ω–mesh. Let x_i be the centre of gravity of $T_i \in \mathcal{T}_h$. There exists the uniquely defined function $r_h u$ given by (2.18). Then

$$r_h u(x) = u(x_i) + \sum_{j=1}^{d} \frac{\partial u(x_i)}{\partial x^{(j)}} (x^{(j)} - x_i^{(j)}), \; x \in T_i. \tag{2.25}$$

Let $x \in \Omega \cap \Omega_h$, then there exists element $T_i \in \mathcal{T}_h$ such that $x \in T_i$. Then the Taylor series expansion gives

$$u(x) = r_h u(x) + \frac{1}{2} \sum_{j,k=1}^{d} \frac{\partial^2 u(\xi)}{\partial x^{(j)} \partial x^{(k)}} (x^{(j)} - x_i^{(j)})(x^{(k)} - x_i^{(k)}), \tag{2.26}$$

where $\xi \in \Omega$ is a point lying between x and x_i. As $|x^{(j)} - x_i^{(j)}| \le h, \; j = 1, \dots, d$ then using (2.23) and (2.25), we have from (2.26)

$$|u(x) - r_h u(x)| \le \frac{1}{2} d^2 \hat{c} h^2, \qquad x \in \Omega \cap \Omega_h, \tag{2.27}$$

which together with (2.24) proves the theorem. □

Definition 5 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ and $\omega > 0$. We say that $\mathcal{T}_h^{\min}$ is the ω-*minimal mesh *if $\mathcal{T}_h^{\min}$ is an ω-mesh and*

$$\#\mathcal{T}_h^{\min} = \min \#\mathcal{T}_h, \tag{2.28}$$

where $\#\mathcal{T}_h$ denotes the number of elements of $\mathcal{T}_h$ and the minimum is taken over all ω-meshes $\mathcal{T}_h$ of Ω.

Theorem 2 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ and $\omega > 0$. Then there exists an ω-minimal mesh.*

Proof. It follows from Theorem 1, that the set of ω-meshes is not empty. As the operator $\# : \mathcal{T}_h \to \mathbb{N}$ ($\mathbb{N}$ = natural numbers) is bounded from below, it attains its minimum. □

2.3 Optimal element

In order to approach $\mathcal{T}_h^{\min}$, we introduce the concept of an *optimal element.* Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ and let its Hessian matrix $\boldsymbol{H}$

$$\boldsymbol{H}(x) = \begin{pmatrix} \frac{\partial^2 u(x)}{\partial x^{(1)} \partial x^{(1)}} & \cdots & \frac{\partial^2 u(x)}{\partial x^{(1)} \partial x^{(d)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 u(x)}{\partial x^{(d)} \partial x^{(1)}} & \cdots & \frac{\partial^2 u(x)}{\partial x^{(d)} \partial x^{(d)}} \end{pmatrix}, \qquad x \in \Omega \cup \Omega_h \tag{2.29}$$

be symmetric positive definite. Let $\mathcal{T}_h$ be an ω–mesh and $r_h u$ be the ∇–interpolation given by (2.18). Let $x_i = (x_i^{(1)}, \ldots, x_i^{(d)})$ be the centre of gravity of $T_i \in \mathcal{T}_h$ and $x = (x^{(1)}, \ldots, x^{(d)}) \in T_i$. Then, using the Taylor series expansion at x_i, we have

$$u(x) = r_h u(x) + \frac{1}{2} \sum_{j,k=1}^{d} \frac{\partial^2 u(x_i)}{\partial x^{(j)} \partial x^{(k)}} (x^{(j)} - x_i^{(j)})(x^{(k)} - x_i^{(k)}) + o(|x - x_i|^2). \tag{2.30}$$

In what follows we shall neglect the terms of order $o(|x - x_i|^2)$ in equalities and inequalities. Up to terms of this order we can write

$$\begin{aligned} E_{\mathrm{I}}(x) &= |u(x) - r_h u(x)| = \qquad\qquad (2.31) \\ &= \frac{1}{2} \left| \sum_{j,k=1}^{d} \frac{\partial^2 u(x_i)}{\partial x^{(j)} \partial x^{(k)}} (x^{(j)} - x_i^{(j)})(x^{(k)} - x_i^{(k)}) \right|, \quad x \in T_i. \end{aligned}$$

As $\mathcal{T}_h$ is the ω–mesh we have

$$\left| \sum_{j,k=1}^{d} \frac{\partial^2 u(x_i)}{\partial x^{(j)} \partial x^{(k)}} (x^{(j)} - x_i^{(j)})(x^{(k)} - x_i^{(k)}) \right| \le 2\omega, \quad x \in T_i, \tag{2.32}$$

which can be rewritten as

$$\left(x^{(1)} - x_i^{(1)}, \ldots, x^{(d)} - x_i^{(d)}\right) \begin{pmatrix} \frac{\partial^2 u(x_i)}{\partial x^{(1)} \partial x^{(1)}} & \cdots & \frac{\partial^2 u(x_i)}{\partial x^{(1)} \partial x^{(d)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 u(x_i)}{\partial x^{(d)} \partial x^{(1)}} & \cdots & \frac{\partial^2 u(x_i)}{\partial x^{(d)} \partial x^{(d)}} \end{pmatrix} \begin{pmatrix} x^{(1)} - x_i^{(1)} \\ \vdots \\ x^{(d)} - x_i^{(d)} \end{pmatrix} \le 2\omega, \; x \in T_i. \tag{2.33}$$

The points $x \in I\!R^d$ satisfying the relation (2.33) form an ellipse (for $d = 2$) and an ellipsoid (for $d = 3$) $\epsilon_i(\omega) \subset I\!R^d$, see Appendix.

From this it follows that the interpolation error function $E_{\mathrm{I}}(x)$ is bounded from above by ω for all $x \in \epsilon_i(\omega)$. Consequently, if T is such a element that $T \subset \epsilon_i(\omega)$, then $E_{\mathrm{I}}(x) \le \omega$ for all $x \in T$.

Definition 6 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ be a function with symmetric positive definite Hessian matrix, $\omega > 0$, $\mathcal{T}_h$ an ω–mesh, $T_i \in \mathcal{T}_h$ and $\epsilon_i(\omega)$ the ellipse ($d = 2$) or the ellipsoid ($d = 3$) given by (2.33). We say that T_i is an* optimal element *if*

$$T_i \subset \epsilon_i(\omega) \tag{2.34}$$

and

$$\mathrm{meas}(T_i) \ge \mathrm{meas}(T) \text{ for all elements } T \subset \epsilon_i(\omega). \tag{2.35}$$

Lemma 3 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ be a function with symmetric positive definite Hessian matrix, $\omega > 0$ be given, $x_i \in \Omega$ and $\epsilon_i(\omega)$ the ellipse defined by (2.33). Then there exist elements T_i^j, $j = 1, \dots, N(d)$ with the centre of gravity x_i satisfying (2.34) and (2.35), where $N(2) = 4$ and $N(3) = 24$.*

Proof. See Appendix. □

If T_i is the optimal element for every $T_i \in \mathcal{T}_h$ then $E_{\mathrm{I}}(x) \leq \omega$ for every $x \in \Omega \cap \Omega_h$. Therefore $\mathcal{T}_h$ is the ω–mesh and due to (2.35) $\mathcal{T}_h$ has the reasonable small number of elements and approximates well $\mathcal{T}_h^{\min}$.

In order to describe geometrically the optimal element, we introduce the following

Definition 7 *Let $\boldsymbol{M}$ be a $d \times d$ symmetric positive definite matrix and $\boldsymbol{v} \in I\!\!R^d$. We define* the norm *of the vector $\boldsymbol{v}$* corresponding *to the matrix $\boldsymbol{M}$ as*

$$\|\boldsymbol{v}\|_{\boldsymbol{M}} \equiv \left(\boldsymbol{v}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{v}\right)^{\frac{1}{2}}. \tag{2.36}$$

Remark 2 *If $\boldsymbol{M} = \boldsymbol{I}$ (=identity matrix) then $\|\boldsymbol{v}\|_{\boldsymbol{M}} = |\boldsymbol{v}|$, where $|\boldsymbol{v}|$ is the Euclidean norm of $\boldsymbol{v}$.*

Lemma 4 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ be a function with symmetric positive definite Hessian matrix $\boldsymbol{H}(u(x))$, $x \in \Omega$, $\omega > 0$, $\mathcal{T}_h$ a mesh of Ω, $T_i \in \mathcal{T}_h$ with centre of gravity x_i and $\epsilon_i(\omega)$ the ellipse given by (2.33). Let V_i^j, $j = 1, \dots, d+1$ be vertices of T_i. Then T_i is the optimal element if and only if*

$$\|V_i^k - V_i^l\|^2_{\boldsymbol{M}} = 2C(d)\omega, \quad k, l = 1, \dots, d+1, \ k \neq l, \tag{2.37}$$

where we put $\boldsymbol{M} = \boldsymbol{H}(u(x_i))$ and $C(2) = 3$ and $C(3) = \frac{8}{3}$.

Proof. See Appendix □

2.4 Optimal mesh

Definition 7 and Lemma 4 give us a suitable tool for the practical construction of the mesh $\mathcal{T}_h^{\min}$. On the other hand the condition (2.37) for $i = i_1, i_2$ can not be generally satisfied for two elements $T_{i_1} \neq T_{i_2}$ with the common edge $\partial T_{i_1} \cap \partial T_{i_2} \neq \emptyset$. In order to avoid these difficulties, we pass from the centre of gravities associated Hessian matrixes to the edge associated Hessian matrixes. This edge associated approach is simpler for the numerical implementation.

Definition 8 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ be a function with symmetric positive definite Hessian matrix and $\omega > 0$ be given. Let $\mathcal{T}_h$ be a mesh of the computational domain Ω. Let $\boldsymbol{H}_k$ be the Hessian matrix of u evaluated in the centre of the edge $\boldsymbol{e}_k$, $k \in K$(= an index set) of the mesh $\mathcal{T}_h$. We say that the mesh $\mathcal{T}_h$ is* edge-optimal, *if*

$$\|\boldsymbol{e}_k\|_{\boldsymbol{H}_k} = \sqrt{2C(d)\omega} \qquad \forall k \in K. \tag{2.38}$$

Remark 3 *The matrix $\boldsymbol{H}_k$ can be interpreted as an* interpolation error matrix *and the norm $\|\boldsymbol{e}_k\|_M$ as an* interpolation error over the edge *$\boldsymbol{e}_k$ (see [CDBG+96], [FVD+96]). It means that the interpolation error (considered in terms of $\|\cdot\|_{\boldsymbol{H}_k}$) is uniformly distributed over the edges of the mesh.*

Remark 4 *It is easy to see that the optimal mesh exists only for special types of domains Ω and the Hessian matrixes. Let $\boldsymbol{H}_k = c\boldsymbol{I}$ $\forall k \in K$ where $\boldsymbol{I}$ is the unit matrix and $c > 0$. Then the edge-optimal mesh is uniform, i.e., the (Euclidean) lengths of $\boldsymbol{e}_k$, $k \in K$ are the same therefore all elements are equilateral. Let $\Omega = [0,1]^2$ then the optimal triangulation does not exist because the angle of triangles at the corners of the domain cannot be equal to $\pi/3$.*

It lead us to a weaker form of the concept of edge-optimal mesh. We introduce the quality parameter $Q_{\mathcal{T}_h}$, which measures the edge-optimality in the least squares sense.

Definition 9 *Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$ be a function with symmetric positive definite Hessian matrix and $\omega > 0$ be given. Let $\mathcal{T}_h$ be a mesh of the computational domain Ω. Let $\boldsymbol{H}_k$ be the Hessian matrix of u evaluated in the centre of the edge $\boldsymbol{e}_k$, $k \in K$(= an index set) of the mesh $\mathcal{T}_h$. We say that the mesh $\mathcal{T}_h^{\rm opt}$ is* **optimal** *if*

$$Q_{\mathcal{T}_h^{\rm opt}} = \min Q_{\mathcal{T}_h}, \tag{2.39}$$

where the minimum is taken over all meshes of Ω and

$$Q_{\mathcal{T}_h} = \frac{1}{\#\mathcal{T}_h} \sum_{k\in K} \left(\|\boldsymbol{e}_k\|_{\boldsymbol{H}_k} - \sqrt{2C(d)\omega} \right)^2, \tag{2.40}$$

where $\#\mathcal{T}_h$ is the number of edges of $\mathcal{T}_h$. We call the quantity $Q_{\mathcal{T}_h}$ the quality parameter *of $\mathcal{T}_h$.*

The quality parameter $Q_{\mathcal{T}_h}$ is a computable quantity for the given function u and the mesh $\mathcal{T}_h$. Definition 9 gives us an idea how to adapt the mesh. The mesh adaptation algorithm is described in Section 3. In Section 2.5, we present another determination of AMA which leads to the same definition of the quality of a mesh.

2.5 Alternative determination of AMA

It is possible to introduce the term edge-optimal mesh from Definition 8 without the term optimal element from Section 2.3. Our aim is to adapt a given mesh $\mathcal{T}_h$ so that the new one satisfies the stronger condition (2.21) and $\#\mathcal{T}_h$ is minimal. Let us suppose that the dependence of $\|u - r_h u\|_X$ on $\#\mathcal{T}_h$ is monotone. Then the inverse problem can be formulated: Adapt a given mesh $\mathcal{T}_h$ so that the interpolation error $\|u - r_h u\|_X$ is minimal for a fixed $\#\mathcal{T}_h$.

The basic idea is the following: In order to minimize the interpolation error for a given $\#\mathcal{T}_h$, the interpolation error function $E_{\rm I}$ defined by (2.20) should be equidistantly distributed over the whole computational domain Ω, i.e.

$$E_I(x) \approx C \qquad \forall x \in \Omega, \tag{2.41}$$

where $C > 0$ is a constant.

In order to minimize the interpolation error function E_I, we introduce the discrete version of the condition (2.41). We consider the interpolation error function E_I over edges for practical reasons. Let $\boldsymbol{e}$ be an edge of $\mathcal{T}_h$ (connecting two nodes of $\mathcal{T}_h$), $|\boldsymbol{e}|$ notes the Euclidean length of $\boldsymbol{e}$ and let $x_{\boldsymbol{e}}$ be the centre of $\boldsymbol{e}$. We approximate $E_I|_{\boldsymbol{e}}$ by the mean value of E_I over $\boldsymbol{e}$. Then the omitting the terms of higher order yields

$$\begin{aligned} E_I|_{\boldsymbol{e}} &\approx \frac{1}{|\boldsymbol{e}|} \int_{\boldsymbol{e}} |u(x) - r_h u(x)| \, \mathrm{d}S \approx \\ &\approx \frac{1}{2} \frac{1}{|\boldsymbol{e}|} \int_{\boldsymbol{e}} |(x - x_{\boldsymbol{e}}) \boldsymbol{H}(x_{\boldsymbol{e}})(x - x_{\boldsymbol{e}})| \, \mathrm{d}S = \frac{1}{24} \left| \vec{e}^{\rm T} \boldsymbol{H}(x_{\boldsymbol{e}}) \vec{e} \right|, \end{aligned} \tag{2.42}$$

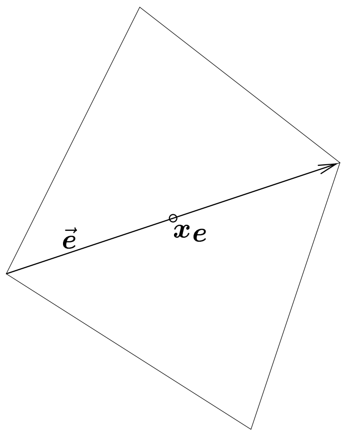

Figure 1.3: The vector $\vec{\boldsymbol{e}}$ corresponding to the edge $\boldsymbol{e} \in \mathcal{T}_h$ and its centre $x_{\boldsymbol{e}}$.

where $\vec{\boldsymbol{e}} \in I\!R^d$ is the vector parallel with the edge $\boldsymbol{e}$ and with the Euclidean norm equal to $|\boldsymbol{e}|$, see Figure 1.3.

Using (2.36), the norm of the edge $\boldsymbol{e}$ corresponding to the matrix $\boldsymbol{H}$ is

$$\|\boldsymbol{e}\|_{\boldsymbol{H}} \equiv \left(\left|\vec{\boldsymbol{e}}^{\mathrm{T}} \boldsymbol{H}(x_{\boldsymbol{e}})\vec{\boldsymbol{e}}\right|\right)^{1/2}. \tag{2.43}$$

We have from (2.42) and (2.43), that the interpolation error function is uniformly distributed over the mesh $\mathcal{T}_h$ if

$$E_I|_{\boldsymbol{e}} \approx \frac{1}{24}\|\boldsymbol{e}\|^2_{\boldsymbol{H}} \approx \omega \qquad \text{for any edge } \boldsymbol{e} \text{ of } \mathcal{T}_h, \tag{2.44}$$

where $\omega > 0$ is the given accuracy. With the aid of (2.44) we define the following:

Definition 10 *The mesh $\mathcal{T}_h$ is* optimal *if and only if*

$$\|\boldsymbol{e}_k\|_{\boldsymbol{H}_k} = \sqrt{24\,\omega} \quad \forall \boldsymbol{e}_k \in \mathcal{T}_h, \qquad \boldsymbol{e}_k \textit{ edge of } \mathcal{T}_h, \quad \boldsymbol{H}_k \equiv \boldsymbol{H}(x_{\boldsymbol{e}_k}), \tag{2.45}$$

where ω is a given constant which plays a role of accuracy.

Remark 5 *Definition 10 is identical (modulo a scaling constant) to Definition 8. Therefore it leads also to Definition 9 introducing the term optimal mesh. The first concept (Sections 2.2 – 2.4) has better geometrical interpretation. However, the second approach (Section 2.5) should be used in order to derive some a priori error estimations.*

3 Mesh optimization

The quality parameter $Q_{\mathcal{T}_h}$ given by (2.40) is always nonnegative and it is equal to zero if the mesh is edge-optimal in the sense of the Definition 8. Therefore we adapt the mesh $\mathcal{T}_h$ in order to decrease $Q_{\mathcal{T}_h}$ and to approach $\mathcal{T}_h^{\mathrm{opt}}$ and so $\mathcal{T}_h^{\mathrm{min}}$. Let $u \in C^2(\overline{\Omega} \cup \Omega_h)$, $\omega > 0$, $\mathcal{T}_{\mathrm{old}}$ be a mesh of Ω and $\boldsymbol{H}_k,\ k \in K$ be the Hessian matrices of u evaluated in the centre of the edge $\boldsymbol{e}_k$. The mesh contains two type of nodes: I (for internal) and B (for boundary). An II edge is an edge that joins two I-nodes, and the other combinations are defined analogously. We want to find a new mesh $\mathcal{T}_{\text{new}}$ such that $Q_{\mathcal{T}_{\text{new}}}$ is minimal from all possible meshes. To obtain $\mathcal{T}_{\text{new}}$, an iterative process is used. Each iteration consists of a certain number of the local mesh operations described in the next paragraphs.

3.1 Mesh adaptation – d=2

The local mesh operations for $d = 2$ (viewed in Figure 1.4) are the following

- **Adding a node** ($\mathcal{A}$) - If the norm of some edge of a triangle is greater than $\frac{4}{3}\sqrt{2C(d)\omega}$ then we put a new node.

  - *II or IB-edge*: we put a new node at the centre of this edge.
  - *BB-edge*: a new node is the orthogonal projection of the centre of this edge on $\partial\Omega$.

  The constant $\frac{4}{3}\sqrt{2C(d)\omega}$ is the smallest positive solution of the inequality

$$\left| \|\boldsymbol{e}\|_{\boldsymbol{H}} - \sqrt{2C(d)\omega} \right| \geq \left| \frac{1}{2}\|\boldsymbol{e}\|_{\boldsymbol{H}} - \sqrt{2C(d)\omega} \right|, \tag{3.1}$$

  which means that after adding a node, two edges with the norms $\frac{1}{2}\|\boldsymbol{e}\|_{\boldsymbol{H}}$ arise and the difference $\frac{1}{2}\|\boldsymbol{e}\|_H$ and $\sqrt{2C(d)\omega}$ is smaller than the difference of $\|\boldsymbol{e}\|_H$ and $\sqrt{2C(d)\omega}$.

- **Removing an edge** ($\mathcal{R}$) - If the norm of some edge of a triangle is smaller than $\frac{2}{3}\sqrt{2C(d)\omega}$ we remove this edge. The initial and final nodes of this side pass into a new node, whose position depends on the type of edge.

  - *II-edge*: the new position is at the centre of removed edge.
  - *IB-edge*: the new position is at the position of the B-node
  - *BB-edge*: the new position is at the orthogonal projection of the centre of removed edge on $\partial\Omega$.

  The constant $\frac{2}{3}\sqrt{2C(d)\omega}$ is the highest solution of the inequality

$$\left| \|\boldsymbol{e}\|_{\boldsymbol{H}} - \sqrt{2C(d)\omega} \right| \geq \left| 2\|\boldsymbol{e}\|_{\boldsymbol{H}} - \sqrt{2C(d)\omega} \right|, \tag{3.2}$$

  which introduce a heuristical consideration opposite to (3.1).

- **Swapping**($\mathcal{S}$) the diagonal of the quadrilateral formed by any pair of adjacent elements. The edge is swapped if the quality parameter $Q_{\mathcal{T}}$ after swapping is smaller than before.

- **Moving a node** ($\mathcal{M}$) is the only non discrete local operation. We seek a new position of a node (positions of other nodes are fixed), which minimize the quality parameter $Q_{\mathcal{T}}$.

  - *B-node*: we move the node along $\partial\Omega$ and the new position minimizes the quality parameter $Q_{\mathcal{T}}$.
  - *I-node*: the moving of node P_i consists of the following steps:

1. Let $\mathcal{P}(i)$ be the set of nodes $P_j \in \sigma_h$ such that $\boldsymbol{e}_{ij} \equiv P_iP_j$ is an edge of the mesh. Then we define the nodes
$$P^*_{i,j} \equiv P_i + \alpha_j(P_j - P_i) \quad \forall j \in \mathcal{P}(i)$$
where $\alpha_j > 0$ is chosen in such a way that $\|P^*_{i,j} - P_i\|_{\boldsymbol{H}_{\boldsymbol{e}_{ij}}} = \sqrt{2C(d)\omega}\ \forall j \in \mathcal{P}(i)$.
2. We define the node P^*_i by
$$P^*_i \equiv \frac{1}{\#\mathcal{P}(i)} \sum_{j\in\mathcal{P}(i)} P^*_{i,j},$$
where $\#\mathcal{P}(i)$ denotes the number of nodes having a common edge with P_i.
3. Let $P'_i(s)$ be a point such that
$$P'_i(s) = P_i + s(P^*_i - P_i), \quad s \in [0,1]$$
If we move P_i into $P'_i(s)$, we obtain a triangulation $\mathcal{T}(s)$ with the quality parameter $Q_{\mathcal{T}}(s)$. We find $s^* \in [0,1]$ such that $Q_{\mathcal{T}}(s^*) \le Q_{\mathcal{T}}(s)$ for all $s \in [0,1]$ and the new position of P is $P_i := P'_i(s^*)$.

Remark 6 *The local operation moving a node should seem to be not very effective. There exists another approach ([FOG97]), where the optimal position is sought by a gradient method. On the other hand, the advantage of our proposed algorithm is its simplicity and therefore quickness. As the iterative process consists of many repetitions of all local operations, it is sufficient to find the new position of the node approximately. Numerical experiments show that our algorithm is more effective from CPU-time point of view.*

3.2 Mesh adaptation – d=3

The two dimensional operations **adding a node** ($\mathcal{A}$), **removing an edge** ($\mathcal{R}$) and **moving a node** ($\mathcal{M}$) should be easily extended also for the case $d = 3$. Only the operation **swapping** is more complicated and we distinguish the following cases similarly as in [FOG97].

- **Face swapping** ($\mathcal{S}_F$) reconnects the tetrahedra separated by a single interior face. The operation is viewed in Figure 1.5 and Figure 1.6 where two tetrahedra yields three and two new ones for interior and boundary face, respectively. The face is swapped if the quality parameter $Q_{\mathcal{T}}$ after swapping is smaller than before.

- **Edge swapping** ($\mathcal{S}_E$) is performed for the set of M tetrahedra ($M \ge 3$) having a common edge AB. Moreover, the tetrahedra have the nodes $P_1, \dots, P_M$, see Figure 1.7. So that we have tetrahedra $ABP_jP_{j+1},\ j = 1, \dots, M$ where $P_{M+1} \equiv P_1$. Performing the edge swapping we obtain $2M - 4$ new tetrahedra having the nodes $AP_iP_jP_k$ and $BP_iP_jP_k$, see Figure 1.7. It means that the polygon $P_1 \dots P_M$ is divided into $M - 2$ triangles $P_iP_jP_k$. Figure 1.8 shows how we can divide polygon $P_1 \dots P_M$ into $M - 2$ triangles for $M = 3, \dots, 7$. The upper indexes of the pictures denote the number of subcases arising by the index shifting. The edge is swapped if the quality parameter $Q_{\mathcal{T}}$ after swapping is smaller than before.

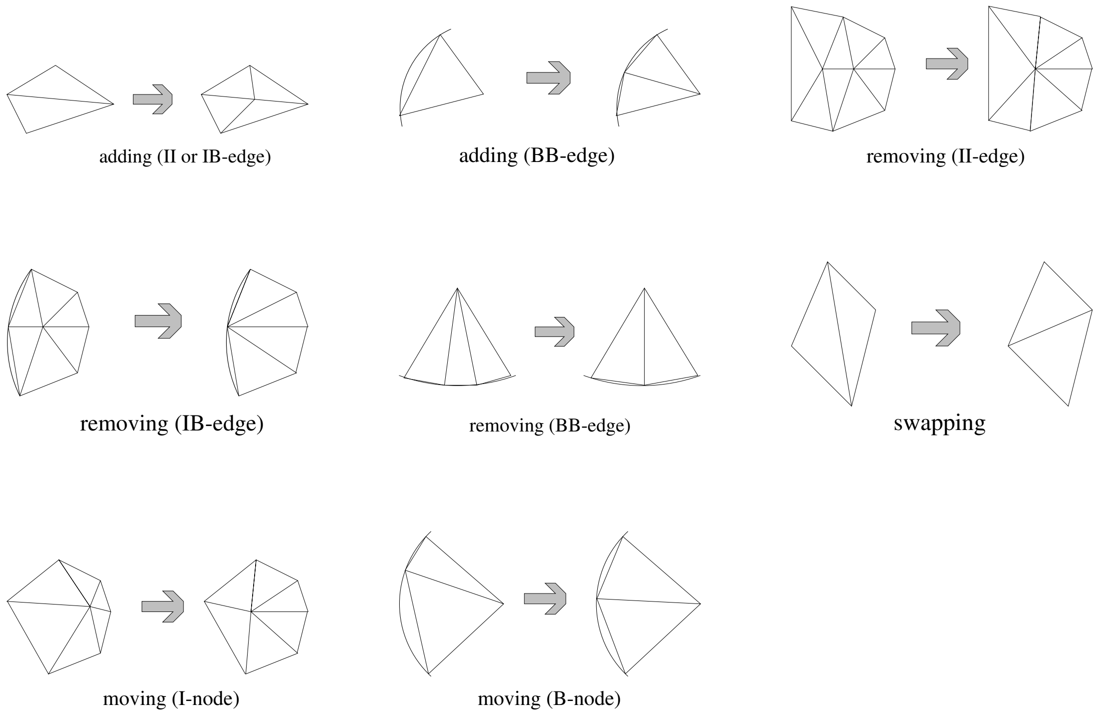

Figure 1.4: Two dimensional local operations.

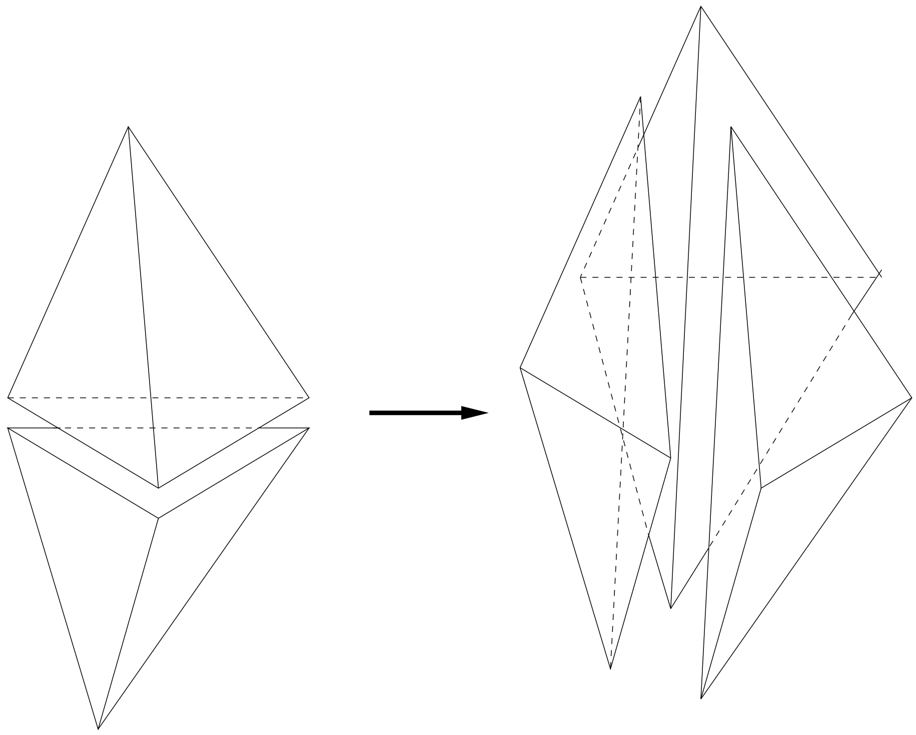

Figure 1.5: Face swapping for an interior face.

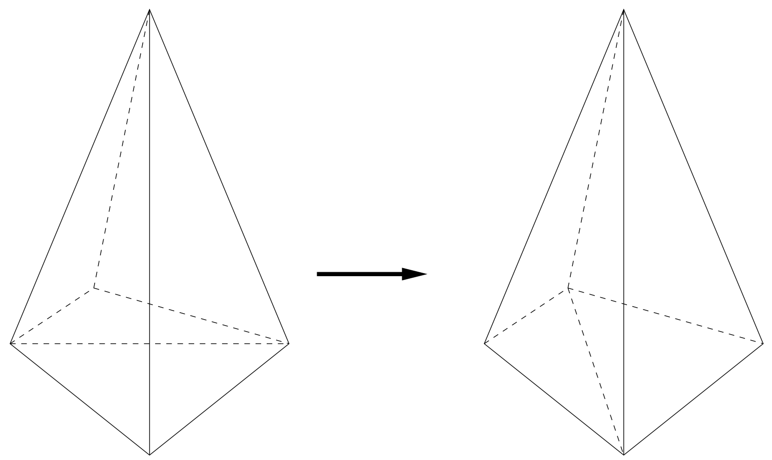

Figure 1.6: Face swapping for a boundary face, the bottom faces belongs to $\partial\Omega_h$.

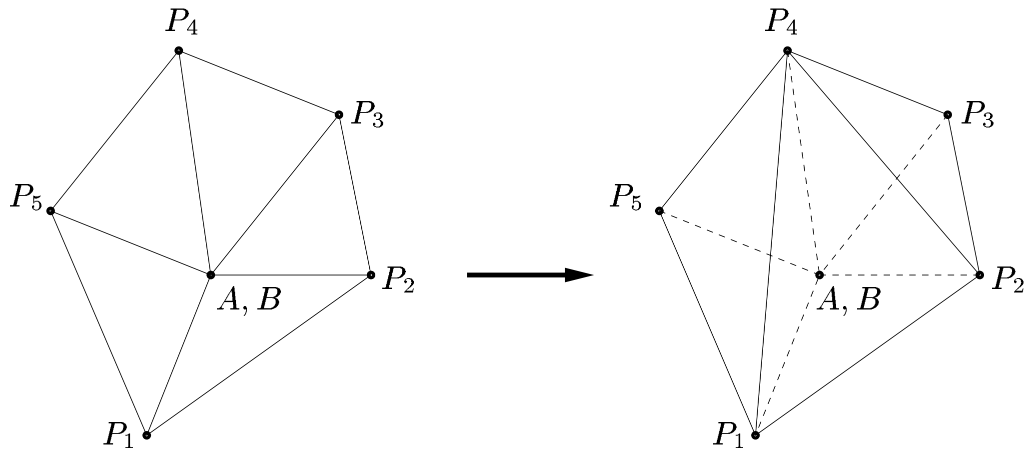

Figure 1.7: Example of edge swapping from 5 to 6 tetrahedra. The view is parallel with the edge AB, therefore the edge AB pass into one point.

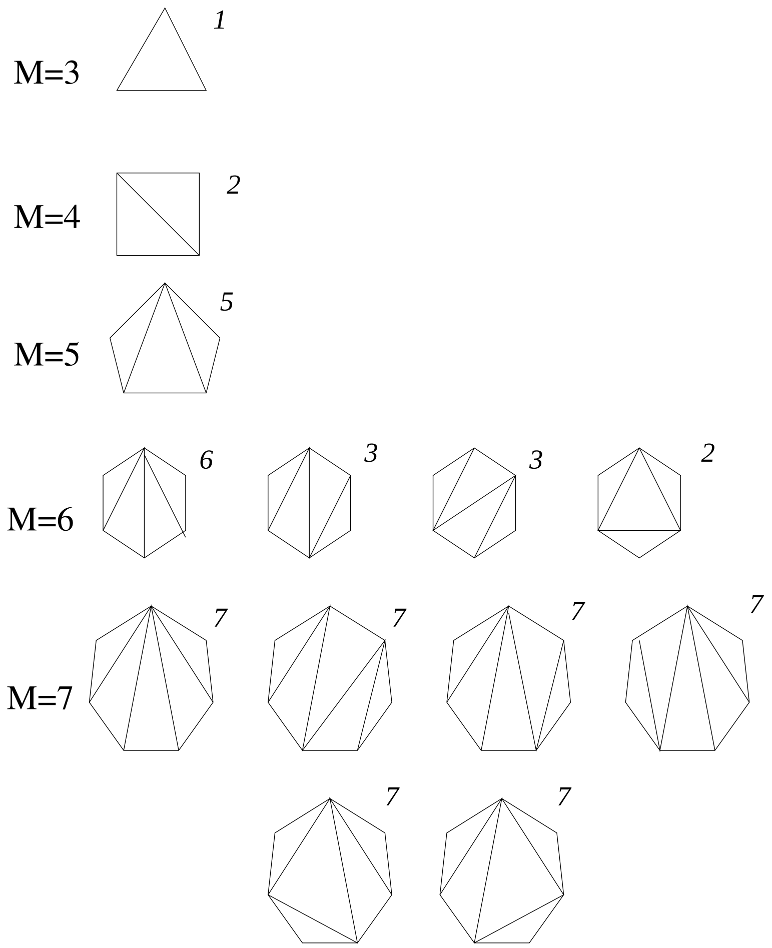

Figure 1.8: Triangles after edge swapping for $3 \leq M \leq 7$, the upper index denotes the number of unique rotations for each case.

3.3 Minimal angle condition

From theoretical as well as computational reasons, it is suitable to ensure the shape regularity of the mesh. The mesh is shape regular if there exists a constant β such that

$$\frac{\text{diam }(T)}{\rho_T} \leq \beta \qquad \forall T \in \mathcal{T}_h, \tag{3.3}$$

where ρ_T denotes the radius of a maximal $d-$dimensional ball inscribed into $T \in \mathcal{T}_h$. To guarantee that the condition (3.3) is valid, we define the condition

$$\frac{\text{meas}_d(T)}{\sum_{\boldsymbol{e}_k^T \subset T} |\boldsymbol{e}_k^T|^d} \geq \gamma \qquad \forall T \in \mathcal{T}_h, \tag{3.4}$$

where $\text{meas}_d(T)$ is the $d-$dimensional measure of T, $|\boldsymbol{e}_k^T|$ is the (Euclidean) length of the edges $\boldsymbol{e}_k^T$ forming the element $T \in \mathcal{T}_h$. Then any local operation is performed only if the condition (3.4) is satisfied for a prescribed number $\gamma > 0$. In examples presented in this work we put $\gamma = 0.02$.

3.4 Iterative process

Based on a very large range of numerical experiments we present the AMA algorithm using the combination of the above operations $\mathcal{A}, \mathcal{R}, \mathcal{S}$, and $\mathcal{M}$ as (for $d = 3$ we put $\mathcal{S} = \mathcal{S}_F + \mathcal{S}_E$)

$$m \times \{\mathcal{M} + n_1 \times [\mathcal{R} + \mathcal{S}] + \mathcal{M} + n_2 \times [\mathcal{A} + \mathcal{S}] + M + \mathcal{S}\} + \mathcal{M}, \tag{3.5}$$

(the sequence of operations goes from the left to the right), where m, n_1 and n_2 are integers denoting the number of repetitions of the sequence of operations in the square brackets. The sequence in the square brackets is applied until no local discrete operation $\mathcal{S}$, $\mathcal{R}$ and $\mathcal{A}$ are performed. Numerical examples show that the algorithm (3.5) converges, i.e. it finish after a finite number of steps.

4 Application of AMA in numerical solution process

In the previous section we have shown how to construct the mesh $\mathcal{T}_h^{\text{opt}}$ for the smooth known exact solution u. In practical examples, of course, we do not know the exact solution u and therefore we use a numerical approximation u_h on a mesh $\mathcal{T}_h$, on which we apply a smoothing procedure and use the technique from previous section. What we need is not the smoothed function itself but its Hessian matrices $\boldsymbol{H}$ evaluated in the centres of edges of $\mathcal{T}_h$.

4.1 Computation of the second order derivatives

The Hessian matrix $\boldsymbol{H}$ evaluated in the centre of an edge $\boldsymbol{e}_k$ of $\mathcal{T}_h$ is computed as the average of the Hessian matrixes evaluated in the initial and final point forming $\boldsymbol{e}_k$. Let $u_h \in U_h$ (defined in (2.15)) be an approximate solution of (2.4) – (2.6) and

$$\begin{aligned} W_h &= \{w_h; w_h \in C(\overline{\Omega_h}), w_h \text{ is linear on each } T \in \mathcal{T}_h\} \\ W_h^0 &= \{w_h; w_h \in W_h, w_h = 0 \text{ on } \partial\Omega_h\}. \end{aligned} \tag{4.1}$$

We introduce the operator $u_h \in U_h \mapsto \bar{u}_h \in W_h$, such that

$$\bar{u}_h(P_i) = \frac{\sum_{T \in D_i} \int_T u_h(\xi)\, d\xi}{\sum_{T \in D_i} \operatorname{meas}(T)}, \tag{4.2}$$

where $P_i \in \sigma_h$ (see Definition 1) and D_i is the set of elements from $\mathcal{T}_h$ having P_i in common.

We compute the elements of the Hessian matrix $\boldsymbol{H}$ using the evaluation of the second order derivatives of $\bar{u}_h$ in the following sense. We approximate $\frac{\partial^2 \bar{u}_h}{\partial x^{(i)} \partial x^{(j)}},\ i,j = 1,\dots,d$ by a function $\boldsymbol{d}^{(i,j)} u_h \in W_h$ such that

$$\left(\frac{\partial^2 \bar{u}_h}{\partial x^{(i)} \partial x^{(j)}}, \varphi \right) \approx \left(\boldsymbol{d}^{(i,j)} u_h, \varphi \right) = -\left(\frac{\partial \bar{u}_h}{\partial x^{(i)}}, \frac{\partial \varphi}{\partial x^{(j)}} \right) \quad \forall \varphi \in W_h^0,\ i,j = 1,\dots,d, \tag{4.3}$$

where $(\cdot,\cdot)$ means the scalar product in $L^2(\Omega_h)$. The R.H.S. of (4.3) has meaning because $\bar{u}_h, \varphi \in W_h$ and therefore the first derivatives exist and are piecewise constant. The choice of the standard finite element hat test functions from W_h^0 leads to the explicit formula for $\boldsymbol{d}^{(i,j)} u_h,\ i,j = 1,\dots,d$. The relations (4.3) determine the values of $\boldsymbol{d}^{(i,j)} u_h,\ i,j = 1,\dots,d$ in inner nodes $P \in \sigma_h \cap \Omega$. For nodes lying on $\partial\Omega$ we use the extrapolation from interior part of the computational domain. The values of $\boldsymbol{d}^{(i,j)} u_h,\ i,j = 1,\dots,d$ in boundary nodes are averages of $\boldsymbol{d}^{(i,j)} u_h$ in nodes having a common edge with a boundary nodes. Numerical examples confirm that such simple extrapolation yields satisfactory results. Finally, we put

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{d}^{(1,1)} u_h & \dots & \boldsymbol{d}^{(1,d)} u_h \\ \vdots & \ddots & \vdots \\ \boldsymbol{d}^{(d,1)} u_h & \dots & \boldsymbol{d}^{(d,d)} u_h \end{pmatrix}. \tag{4.4}$$

Lemma 5 *Let $u_h \in W_h^0$. Then the approximations of second order derivatives given by (4.3) are symmetric, i.e.*

$$\boldsymbol{d}^{(i,j)} u_h = \boldsymbol{d}^{(j,i)} u_h \qquad \forall i,j = 1,\dots,d. \tag{4.5}$$

Proof. We perform the proof for the case $d = 3$, the two dimensional case we obtain by an easy simplification. Let $\{\varphi_k,\ \varphi_k(P_l) = \delta_{kl},\ k,l \in J_h^\circ\}$ be a basis of W_h^0, where $\{P_l,\ l \in J_h^\circ\}$ is the set of all inner nodes of $\mathcal{T}_h$. Then using (4.3), it is sufficient to prove that

$$\left(\frac{\partial \varphi_k}{\partial x^{(i)}}, \frac{\partial \varphi_l}{\partial x^{(j)}} \right) = \left(\frac{\partial \varphi_k}{\partial x^{(j)}}, \frac{\partial \varphi_l}{\partial x^{(i)}} \right) \qquad \forall i,j = 1,\dots,d,\ \forall k,l \in J_h^\circ. \tag{4.6}$$

The relations (4.6) are nontrivial if the functions $\partial\varphi_k$ and $\partial\varphi_l$ have a nonempty common support which means that there exists an edge $\boldsymbol{e}_{kl}$ such that P_k and P_l are its final nodes. Then there exists M tetrahedra having the edge $\boldsymbol{e}_{kl}$ in common, we denote them by $T_m \equiv P_k P_l P_m P_{m+1},\ m = 1,\dots,M$ (we put $P_{M+1} := P_1$), see Figure 1.9.

Let $D_m,\ m = 1,\dots,M$ be a 4×4 matrix corresponding to the tetrahedron T_m given by

$$\boldsymbol{D}_m \equiv \left\{ (\boldsymbol{D}_m)_{r,s} \right\}_{r,s=1}^4 = \begin{pmatrix} x_k^{(1)} & x_k^{(2)} & x_k^{(3)} & 1 \\ x_l^{(1)} & x_l^{(2)} & x_l^{(3)} & 1 \\ x_m^{(1)} & x_m^{(2)} & x_m^{(3)} & 1 \\ x_{m+1}^{(1)} & x_{m+1}^{(2)} & x_{m+1}^{(3)} & 1 \end{pmatrix}, \qquad m = 1,\dots,M, \tag{4.7}$$

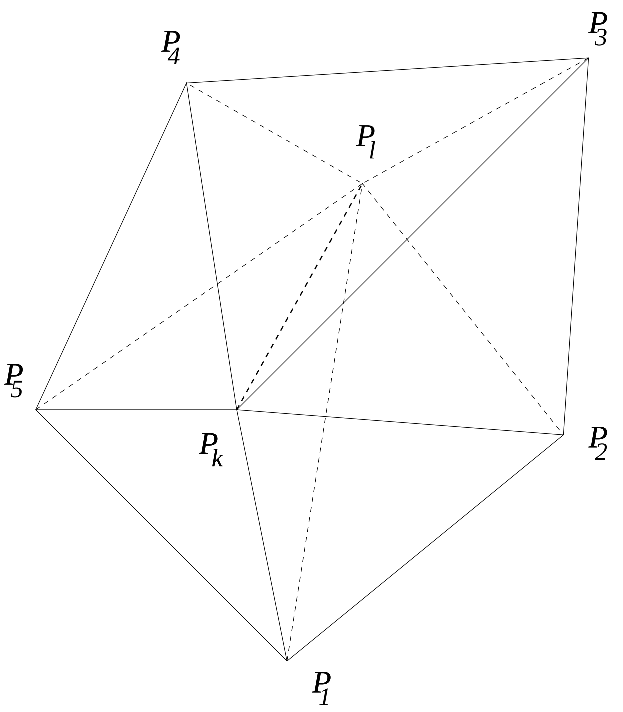

Figure 1.9: Tetrahedra having the edge P_kP_l in common.

where $(x_k^{(1)}, x_k^{(2)}, x_k^{(3)})$ are the coordinates of node P_k and similarly for other nodes. Moreover we define the matrixes $\boldsymbol{D}_m^{p,q},\ p,q = 1,\dots,4,\ m = 1,\dots,M$ which arise replacing the q-th column of $\boldsymbol{D}_m$ by the column $(\delta_{p1}, \delta_{p2}, \delta_{p3}, \delta_{p4})$, i.e.

$$\boldsymbol{D}_m^{p,q} \equiv \left\{ (\boldsymbol{D}_m^{p,q})_{r,s} \right\}_{r,s=1}^{4}, \qquad (\boldsymbol{D}_m^{p,q})_{r,s} = \begin{cases} (\boldsymbol{D}_m)_{r,s} & \text{if } q \neq s \\ \delta_{pr} & \text{if } q = s. \end{cases} \tag{4.8}$$

As $\varphi_k(P_l) = \delta_{kl},\ k,l \in J_h^\circ$ and $\varphi_k,\ k \in J_h^\circ$ is linear over each tetrahedron, it is possible observe that

$$\frac{\partial \varphi_p}{\partial x^{(q)}}|_{T_m} = \frac{\det \boldsymbol{D}_m^{p,q}}{\det \boldsymbol{D}_m}, \qquad p = 1,2,\ q = 1,2,3,\ m = 1,\dots,M, \tag{4.9}$$

where we put $\varphi_1 = \varphi_k$ and $\varphi_2 = \varphi_l$. From (4.6) and (4.9) we have

$$\begin{aligned} \left(\frac{\partial \varphi_k}{\partial x^{(i)}}, \frac{\partial \varphi_l}{\partial x^{(j)}}\right) - \left(\frac{\partial \varphi_k}{\partial x^{(j)}}, \frac{\partial \varphi_l}{\partial x^{(i)}}\right) &= \sum_{m=1}^{M} \int_{T_m} \left(\frac{\partial \varphi_k}{\partial x^{(i)}} \frac{\partial \varphi_l}{\partial x^{(j)}} - \frac{\partial \varphi_k}{\partial x^{(j)}} \frac{\partial \varphi_l}{\partial x^{(i)}} \right) \mathrm{d}x \\ = \sum_{m=1}^{M} |T_m| \left(\frac{\det \boldsymbol{D}_m^{k,i}}{\det \boldsymbol{D}_m} \frac{\det \boldsymbol{D}_m^{l,j}}{\det \boldsymbol{D}_m} - \frac{\det \boldsymbol{D}_m^{k,j}}{\det \boldsymbol{D}_m} \frac{\det \boldsymbol{D}_m^{l,i}}{\det \boldsymbol{D}_m} \right) \\ = \frac{1}{6} \sum_{m=1}^{M} \frac{\det \boldsymbol{D}_m^{k,i} \det \boldsymbol{D}_m^{l,j} - \det \boldsymbol{D}_m^{k,j} \det \boldsymbol{D}_m^{l,i}}{\det \boldsymbol{D}_m}, \end{aligned} \tag{4.10}$$

where $|T_m| = \det \boldsymbol{D}_m / 6$ denotes the volume of T_m. After simple but time consuming computation it is possible to show that

$$\frac{\det \boldsymbol{D}_m^{k,i} \det \boldsymbol{D}_m^{l,j} - \det \boldsymbol{D}_m^{k,j} \det \boldsymbol{D}_m^{l,i}}{\det \boldsymbol{D}_m} = x_m^{(\kappa_{ij})} - x_{m+1}^{(\kappa_{ij})}, \quad m = 1,\dots,M, \tag{4.11}$$

where $\kappa_{ij} \in \{1,2,3\} \setminus \{i,j\}$. Now, the equalities (4.10) and (4.11) with the property

$$\sum_{m=1}^{M} (x_m^{(n)} - x_{m+1}^{(n)}) = 0, \qquad n = 1,2,3 \tag{4.12}$$

yield the relation (4.6). $\square$

Remark 7 *The matrix $\boldsymbol{H}$ given by (4.4) is symmetric but not positive definite as is required in previous sections. Moreover if $\bar{u}_h$ is linear in Ω_h then $\boldsymbol{d}^{(i,j)}u_h = 0$ $\forall i,j = 1\ldots,d$ in Ω. Therefore $\|\boldsymbol{e}_k\|_{\boldsymbol{H}_k} = 0$ $\forall k \in K$ which corresponds with the relation $r_h\bar{u}_h(x) = u_h(x)$ $\forall x \in \Omega \cup \Omega_h$. Consequently $Q_{\mathcal{T}_h} = 2C(d)\omega$ in (2.40) for any mesh $\mathcal{T}_h$ of Ω and the iterative process from the Section 3 can not be applied for minimization of $Q_{\mathcal{T}_h}$. From these reasons we use instead of $\boldsymbol{H}$ its modification $\boldsymbol{M}$.*

4.2 Generation of the matrices M

Let $\boldsymbol{H}(P_i)$, $P_i \in \sigma_h$ be a Hessian matrix (4.4) evaluated in the vertex P_i. As $\boldsymbol{H}(P_i)$ is symmetric, we decompose it in the following way:

$$\boldsymbol{H}(P_i) = R_i \begin{pmatrix} \lambda_{i1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_{id} \end{pmatrix} R_i^{-1}, \quad i \in J_h, \tag{4.13}$$

where R_i, $i \in J_h$(= a suitable index set) are orthogonal matrices. As matrices $\boldsymbol{H}(P_i)$ are generally not positive definite, we put

$$\bar{\boldsymbol{H}}(P_i) = R_i \begin{pmatrix} |\lambda_{i1}| & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & |\lambda_{id}| \end{pmatrix} R_i^{-1} \quad i \in J_h. \tag{4.14}$$

The matrices $\bar{\boldsymbol{H}}(P_i)$, $P_i \in \sigma_h$ are symmetric positive definite. To avoid the problems mentioned in Remark 7 we put

$$\boldsymbol{M}(P_i) = c\left[\boldsymbol{I} + \alpha(\|\bar{\boldsymbol{H}}(P_i)\|)\bar{\boldsymbol{H}}(P_i))\right] \quad P_i \in \sigma_h, \tag{4.15}$$

where $\boldsymbol{I}$ is a unit $d \times d$ matrix, $c > 0$ is a constant and $\alpha : \langle 0,\infty) \to \langle 0,\infty)$ is a function. In (4.15) we put $\|\bar{\boldsymbol{H}}\| = \max_{i,j=1,\dots,d} |\bar{h}_{ij}|$, where $\bar{h}_{ij}$, $i,j = 1,\dots,d$ are the elements of $\bar{\boldsymbol{H}}$. The first term in the square bracket of (4.15) guarantees, that the matrices $\boldsymbol{M}(P_i)$ are always regular.

Setting of parameter c

If $\boldsymbol{H}(P_i) = 0$ $\forall P_i \in \sigma_h$ then $\boldsymbol{M}(P_i) = c\boldsymbol{I}$ $\forall P_i \in \sigma_h$. If the edge-optimal mesh from Definition 8 exists, then it consists of equal equilateral elements. As $\boldsymbol{M}(P_i) = c\boldsymbol{I}$ $\forall P_i \in \sigma_h$ then (2.38) yields

$$\|\boldsymbol{e}_k\|^2_{\boldsymbol{M}(P_i)} = ch^2 = 2C(d)\omega \qquad \forall k \in K, \tag{4.16}$$

where h is the length of edges defined in (2.17) and then

$$h = \sqrt{\frac{2C(d)\omega}{c}}. \tag{4.17}$$

The d-dimensional measure of equilateral element T with the length of edge is h is

$$\mathrm{meas}_d(T) = C_1(d)h^d, \tag{4.18}$$

where $C_1(2) = \sqrt{3}/4$ and $C_1(3) = \sqrt{2}/12$. In this case, the edge-optimal mesh $\mathcal{T}_h$ of Ω consists of $N_{\mathcal{T}_h}$ equal equilateral elements, where

$$N_{\mathcal{T}_h} = \frac{\mathrm{meas}_d(\Omega)}{\mathrm{meas}_d(T)}. \tag{4.19}$$

This, (4.17) – (4.19) imply that

$$c = \left(\frac{N_{\mathcal{T}_h} C_1(d)(2C(d)\omega)^{d/2}}{\mathrm{meas}_d(\Omega)} \right)^{2/d} \tag{4.20}$$

and

$$h = \left(\frac{\mathrm{meas}_d(\Omega)}{C_1(d)\, N_{\mathcal{T}_h}} \right)^{1/d}. \tag{4.21}$$

It means that for a given $N_{\mathcal{T}_h}$ the relation (4.15) leads to the same mesh $\mathcal{T}_h$ independently on $\omega > 0$ provided that $\boldsymbol{H}(P_i) = 0\ \forall P_i \in \sigma_h$.

Remark 8 *In order to guarantee the regularity of matrices $\boldsymbol{M}$ we have introduced the term $c\boldsymbol{I}$ in the relation (4.15). Then the parameter ω has lost the meaning of the accuracy. It indicates how fine should be a mesh provided that $\boldsymbol{H}(P_i) = 0\ \forall P_i \in \sigma_h$. This can be order also by the value $N_{\mathcal{T}_h}$ in (4.20) so that the value ω can be fixed. We set $\omega = 1/2$ in (4.20) and then $N_{\mathcal{T}_h}$ is the number of elements prescribed by user for the construction of the initial mesh $\mathcal{T}_{h_0}$.*

Setting of α

The simplest way to define α is to put

$$\alpha(\|\bar{\boldsymbol{H}}(P_i)\|) = \hat{C},\ \ \hat{C} = \mathrm{const},\ \ \hat{C} > 0. \tag{4.22}$$

When solving problems with discontinuities (e.g. transonic flow with strong shocks), the norm $\|\bar{\boldsymbol{H}}(P_i)\|$ can tend to infinity for some $P_i \in \sigma_h$. Then the drawback of the choice (4.22) is that the condition

$$h^d \leq \bar{c}\,\mathrm{meas}(T),\ \ T \in \mathcal{T}_h,\ \ \bar{c} = \text{constant independent of } h, \tag{4.23}$$

is not satisfied as $\|\boldsymbol{H}(P_i)\| \to \infty$ for some $P_i \in \sigma_h$. The condition (4.23) is the standard requirement for the finite volume/element methods, see e.g. [Fei93]. To fulfill (4.23) we propose the relation

$$\alpha(\|\bar{\boldsymbol{H}}(P_i)\|) = \frac{\varepsilon_1}{\varepsilon_2 + \|\bar{\boldsymbol{H}}(P_i)\|}, \quad P_i \in \sigma_h, \tag{4.24}$$

where ε_1 and ε_2 are positive constants. The relation (4.15) and (4.24) give

$$\lim_{\|\bar{\boldsymbol{H}}(P_i)\| \to \infty} \|\boldsymbol{M}(P_i)\| = c(1 + \varepsilon_1), \quad P_i \in \sigma_h, \tag{4.25}$$

from which it follows that the norms of $\boldsymbol{M}(P_i)$, $P_i \in \sigma_h$ are bounded and therefore the lengths of edges of $\mathcal{T}$ are bounded from below. The relation (4.24) is used in further computation.

The parameters $\varepsilon_1, \varepsilon_2$ are determined in the following way. Let $\bar{\boldsymbol{H}}(P_i) = 0$ for some $P_i \in \sigma_h$. Then $\boldsymbol{M}(P_i) = c\boldsymbol{I}$ and the corresponding optimal element has equal edges with length (in Euclidean norm)

$$l_{\max} = \sqrt{\frac{3}{C(d)}}. \tag{4.26}$$

Let $\bar{\boldsymbol{H}}(P_i) = \beta\boldsymbol{I}$ for some $P_i \in \sigma_h$ and let $\beta \gg 1$. Then $\boldsymbol{M}(P_i) \to c(1+\varepsilon_1)\boldsymbol{I}$ for $\beta \to \infty$, as follows from (4.25) and the corresponding optimal element has equal edges with limit length (in Euclidean norm)

$$l_{\min} = \sqrt{\frac{C(d)}{c(1+\varepsilon_1)}}. \tag{4.27}$$

It follows from (4.26) and (4.27) that the length l of any edge from the edge-optimal mesh satisfies (provided $\boldsymbol{M}(P_i)$ is defined by (4.15) and (4.24))

$$l_{\min} = \sqrt{\frac{C(d)}{c(1+\varepsilon_1)}} \leq l \leq \sqrt{\frac{C(d)}{c}} = l_{\max}. \tag{4.28}$$

It follows from (4.26) and (4.27) that

$$\frac{l_{\max}}{l_{\min}} = \sqrt{1+\varepsilon_1} \tag{4.29}$$

and then

$$\varepsilon_1 = \left(\frac{l_{\max}}{l_{\min}}\right)^2 - 1. \tag{4.30}$$

Relation (4.30) explains the role of ε_1. Setting ε_1 we control the ratio between the longest and the shortest edge admissible in the edge-optimal mesh.

Based on series of practical computations we use for ε_2 the following relation

$$\varepsilon_2 = \frac{\varepsilon_1}{p} \tag{4.31}$$

where p is some positive number. The role of p is viewed in Figure 1.10, where the function

$$\|\bar{\boldsymbol{H}}\| \mapsto \|\boldsymbol{M}\| = c\left(1 + \frac{\varepsilon_1}{\frac{\varepsilon_1}{p} + \|\bar{\boldsymbol{H}}\|}\|\bar{\boldsymbol{H}}\|\right) \tag{4.32}$$

is plotted for different p. The number p controls the transition of the coarsest part of the mesh (with edge length l_{max} given by (4.26)) to the finest one (with edge length l_{min} given by (4.27)) as it is demonstrated on Figure 1.11.

4.3 Multilevel computation

The technique described above is applied in the following way, see Figure 1.12: We start from an initial mesh $\mathcal{T}_{h_0}$ of computational domain Ω. We compute the solution u_{h_0} on the mesh $\mathcal{T}_{h_0}$ using a suitable numerical method (finite volume method, finite element method, ...) for approximate solution of the problem (2.1) – (2.3). We compute the Hessian matrices

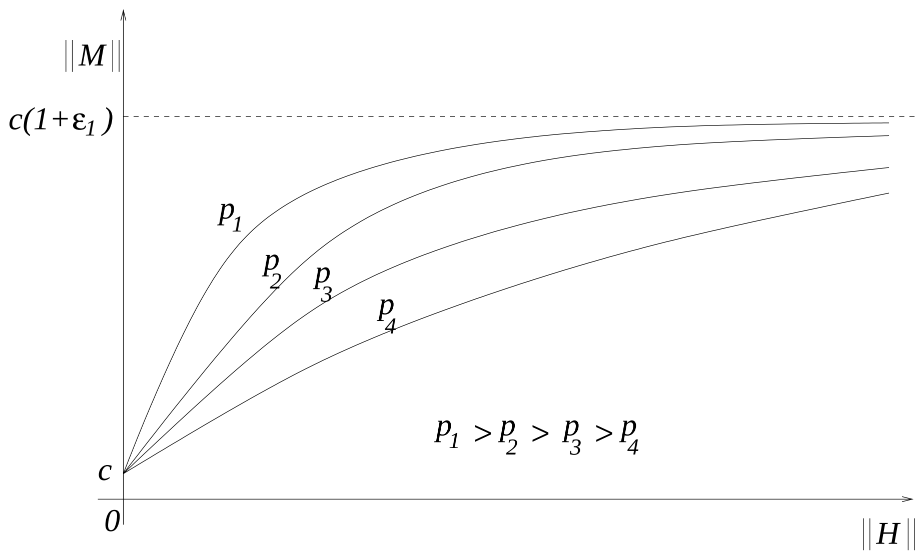

Figure 1.10: Dependence of $\|\boldsymbol{M}\|$ on $\|\bar{\boldsymbol{H}}\|$ for different p.

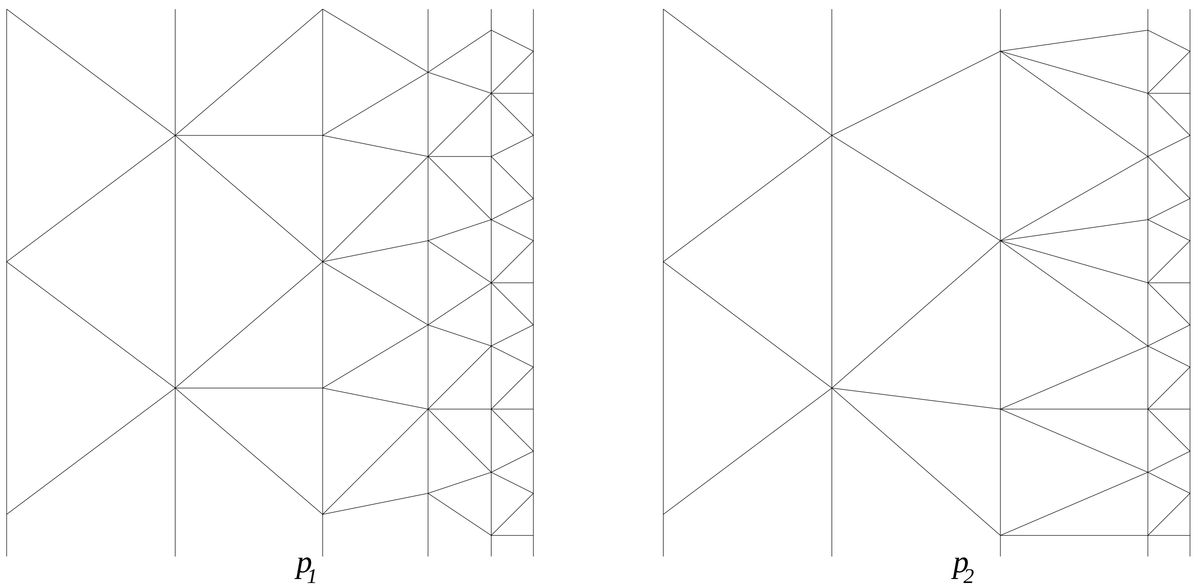

Figure 1.11: Transition of the coarsest part to the finest one for two different $p_1 > p_2$.

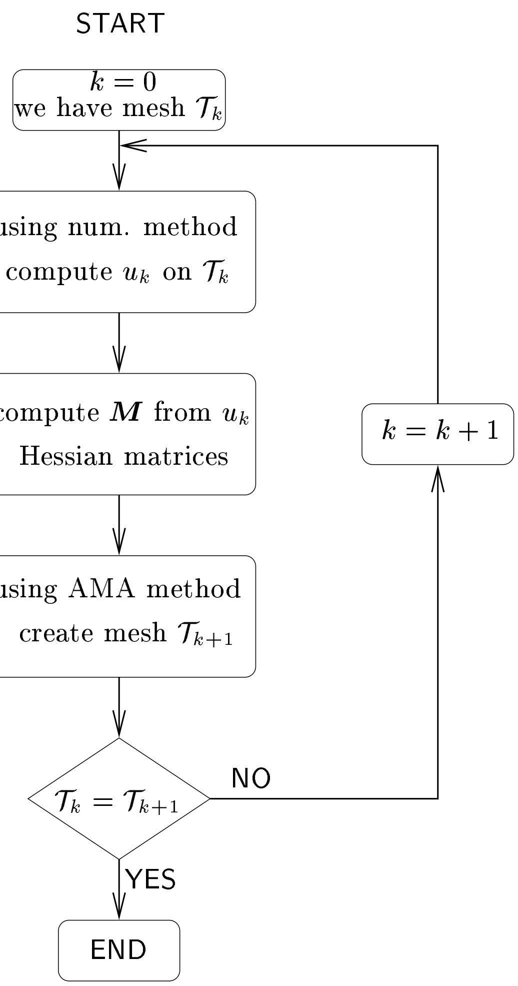

Figure 1.12: Multilevel adaptation AMA algorithm.

$\boldsymbol{M}$ by the technique described above at each vertex of $\mathcal{T}_{h_0}$ and we construct the mesh $\mathcal{T}_{h_1}$ using the iterative process (3.5). The results from $\mathcal{T}_{h_0}$ are interpolated onto $\mathcal{T}_{h_1}$ and the numerical solution procedure is applied on $\mathcal{T}_{h_1}$ producing u_{h_1}. In the similar way we construct $\mathcal{T}_{h_2}, u_{h_2}; \mathcal{T}_{h_3}, u_{h_3}; , \dots$. The multilevel computational process is terminated on the mesh $\mathcal{T}_{h_k}$ if two successive mesh $\mathcal{T}_{h_k}$ and $\mathcal{T}_{h_{k+1}}$ are (approximately) identical. Numerical examples show that for a fixed values of $N_{\mathcal{T}_h}$, ε_1 and p the multilevel computational process terminates, see Section 6.3.

5 Demonstration of efficiency of AMA

In previous sections, we describe the use of the AMA method in combination with a suitable solver for the numerical solution of BVP. Before presenting practical results, we show several examples demonstrating the efficiency of the proposed adaptive algorithm.

5.1 Importance of the stronger necessary condition

At first we show the importance of the stronger necessary condition (2.21). Let u be a solution of the given problem. Then the function $\Pi_h u$ introduced in (2.11) is the best possible solution from the space U_h in the sense of $\|\cdot\|_X$ norm. Moreover, the value

$$\|u - \Pi_h u\|_X \tag{5.1}$$

is the minimal discretization error. Our aim is to show that using anisotropic meshes we can essentially decrease the value of (5.1). As the construction of $\Pi_h u$ is difficult, we estimate (5.1) by the interpolation error

$$\|u - r_h u\|_X, \tag{5.2}$$

where r_h is the ∇-interpolation defined by (2.18).

In two examples we investigate (for a given function u) the interpolation error $\|u - r_h u\|_X$ in the norms

$$\|u - r_h u\|_{L^2(\Omega)}, \qquad |u - r_h u|_{H^1(\Omega,\mathcal{T}_h)}, \qquad \|u - r_h u\|_{L^\infty(\Omega)}, \tag{5.3}$$

where

$$|u|_{H^1(\Omega,\mathcal{T}_h)} \equiv \left(\sum_{T\in\mathcal{T}_h} \int_T |\nabla u|^2 \,\mathrm{d}x \right)^{1/2} \tag{5.4}$$

is the broken H^1 seminorm known from the discontinuous Galerkin method, see e.g. [CKe00]. In (5.3), u is a given function and $r_h u$ is defined by (2.18) where x_i is a barycentre of $T_i, \mathcal{T}_i \in \mathcal{T}_h$. We compare the values of the interpolation errors (5.3) for two types of triangulations: the former uniform triangulation meshes consisting of (almost) equilateral triangles and the latter anisotropic ones obtained by AMA method presented in Section 4.

Example 1: Let $\Omega = [-2;2] \times [-2;2]$ and

$$u(x_1, x_2) = \frac{1}{2}(100 x_1^2 + x_2^2). \tag{5.5}$$

The eigenvalues of the Hessian matrix of u are $\lambda_1 = 100,\ \lambda_2 = 1$. Let as consider two sets of triangulations,

$$\left\{\mathcal{T}^{\mathrm{I}}_{h_1},\ \mathcal{T}^{\mathrm{I}}_{h_2},\ \mathcal{T}^{\mathrm{I}}_{h_3},\ \mathcal{T}^{\mathrm{I}}_{h_4}\right\} \quad \text{and} \quad \left\{\mathcal{T}^{\mathrm{A}}_{h_1},\ \mathcal{T}^{\mathrm{A}}_{h_2},\ \mathcal{T}^{\mathrm{A}}_{h_3},\ \mathcal{T}^{\mathrm{A}}_{h_4}\right\}. \tag{5.6}$$

The former and the latter sets contain uniform (isotropic) and anisotropic meshes, respectively and

$$\#\mathcal{T}^{\mathrm{I}}_{h_j} \approx \#\mathcal{T}^{\mathrm{A}}_{h_j}, \quad j = 1, \dots, 4, \tag{5.7}$$

where $\#\mathcal{T}_h$ denotes the number of triangles of $\mathcal{T}_h$. Figure 1.13 shows meshes $\mathcal{T}^{\mathrm{I}}_{h_j}, \mathcal{T}^{\mathrm{A}}_{h_j},\ j = 1, \dots, 3$.

The computed results are in the following table:

mesh	$\#\mathcal{T}_h$	$\|u - r_h u\|_{L^2(\Omega)}$	$\|u - r_h u\|_{H^1(\Omega,\mathcal{T}_h)}$	$\|u - r_h u\|_{L^\infty(\Omega)}$
$\mathcal{T}_{h_1}^{\mathrm{I}}$	108	4.7131	48.33	8.9073
$\mathcal{T}_{h_1}^{\mathrm{A}}$	110	0.8496	17.37	1.3152
$\mathcal{T}_{h_2}^{\mathrm{I}}$	478	1.1712	24.05	2.6671
$\mathcal{T}_{h_2}^{\mathrm{A}}$	480	0.1828	7.261	0.3496
$\mathcal{T}_{h_3}^{\mathrm{I}}$	1874	0.2770	12.03	0.6582
$\mathcal{T}_{h_3}^{\mathrm{A}}$	1885	0.0462	3.649	0.0930
$\mathcal{T}_{h_4}^{\mathrm{I}}$	7472	0.0713	5.966	0.1697
$\mathcal{T}_{h_4}^{\mathrm{A}}$	7499	0.0116	1.848	0.0259

Example 2: Let $\Omega = [-2;2] \times [-2;2]$ and

$$u(x_1, x_2) = 10(x_1^2 + x_2^2) \exp\left[-10\left((x_1^2 + x_2^2)^{1/2} - 1\right)^2\right]. \tag{5.8}$$

This is an example with steep gradients in the neighborhood of a unit circle. Let as consider set of triangulations from previous example

$$\left\{\mathcal{T}_{h_1}^{\mathrm{I}},\ \mathcal{T}_{h_2}^{\mathrm{I}},\ \mathcal{T}_{h_3}^{\mathrm{I}},\ \mathcal{T}_{h_4}^{\mathrm{I}}\right\} \tag{5.9}$$

and the new set of anisotropic triangulations

$$\left\{\mathcal{T}_{h_1}^{\mathrm{G}},\ \mathcal{T}_{h_2}^{\mathrm{G}},\ \mathcal{T}_{h_3}^{\mathrm{G}},\ \mathcal{T}_{h_4}^{\mathrm{G}}\right\}. \tag{5.10}$$

There is again

$$\#\mathcal{T}_{h_j}^{\mathrm{I}} \approx \#\mathcal{T}_{h_j}^{\mathrm{G}}, \quad j = 1, \dots, 4. \tag{5.11}$$

Figure 1.14 shows meshes $\mathcal{T}_{h_j}^{\mathrm{G}}$, $j = 1, \dots, 4$. We have investigated the same interpolation errors is in previous example. The results are in the following table:

mesh	$\#\mathcal{T}_h$	$\|u - r_h u\|_{L^2(\Omega)}$	$\|u - r_h u\|_{H^1(\Omega,\mathcal{T}_h)}$	$\|u - r_h u\|_{L^\infty(\Omega)}$
$\mathcal{T}_{h_1}^{\mathrm{I}}$	108	4.2131	39.02	17.22
$\mathcal{T}_{h_1}^{\mathrm{G}}$	114	1.6801	24.43	5.433
$\mathcal{T}_{h_2}^{\mathrm{I}}$	478	0.9241	19.54	4.282
$\mathcal{T}_{h_2}^{\mathrm{G}}$	488	0.3791	11.88	1.664
$\mathcal{T}_{h_3}^{\mathrm{I}}$	1874	0.2371	9.961	1.148
$\mathcal{T}_{h_3}^{\mathrm{G}}$	1880	0.1023	5.887	0.567
$\mathcal{T}_{h_4}^{\mathrm{I}}$	7472	0.0581	4.952	0.350
$\mathcal{T}_{h_4}^{\mathrm{G}}$	7467	0.0243	2.862	0.125

The interpolation errors (5.3) are several times smaller for anisotropic meshes than for uniform ones. Consequently, taking into account (2.12) and (2.19), the disretization error (2.8) will be smaller with the use of anisotropic meshes.

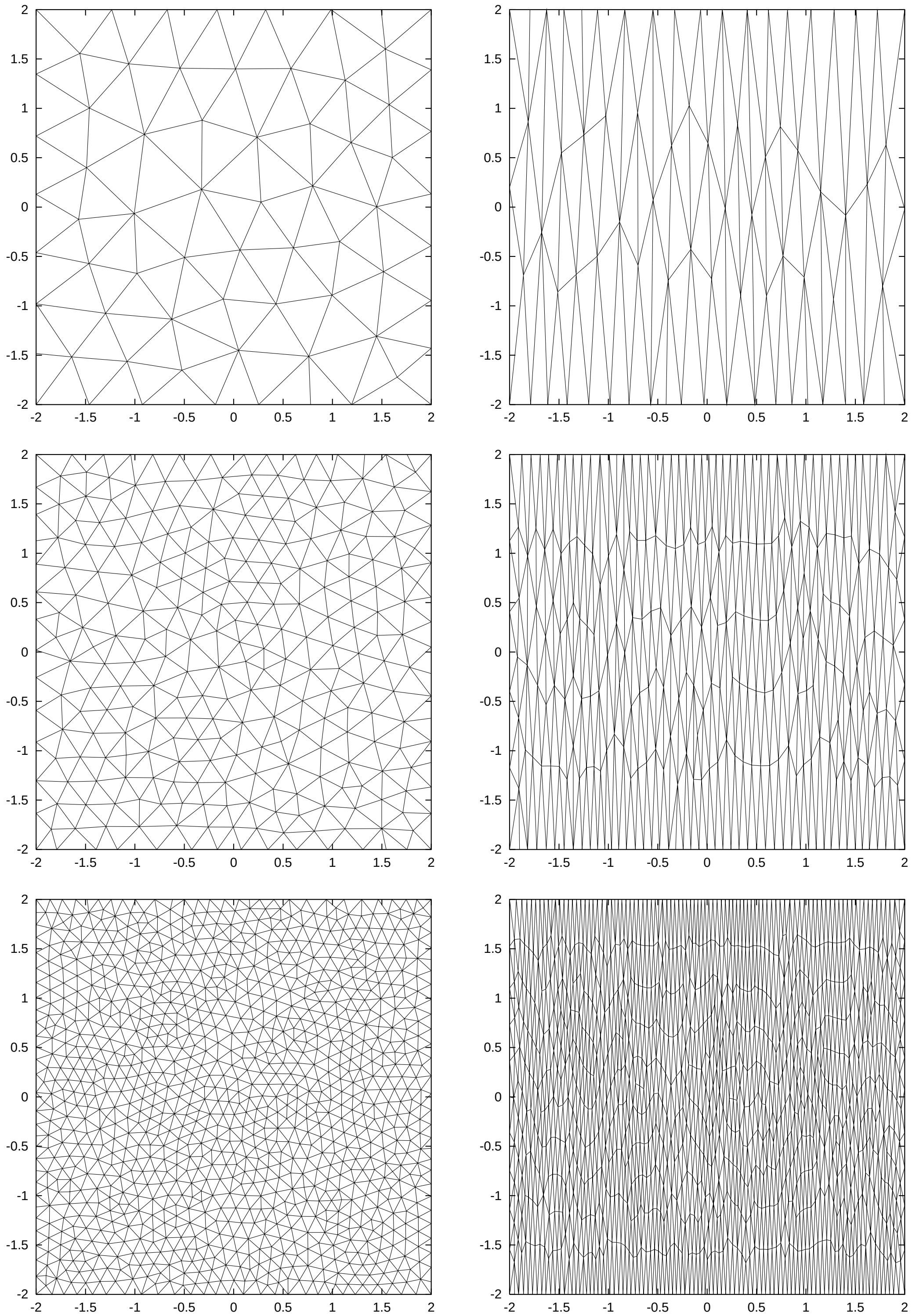

Figure 1.13: Uniform (isotropic) $\mathcal{T}^{\mathrm{I}}_{h_j}$(left) and anisotropic $\mathcal{T}^{\mathrm{A}}_{h_j}$(right) meshes for $j = 1, \ldots, 3$.

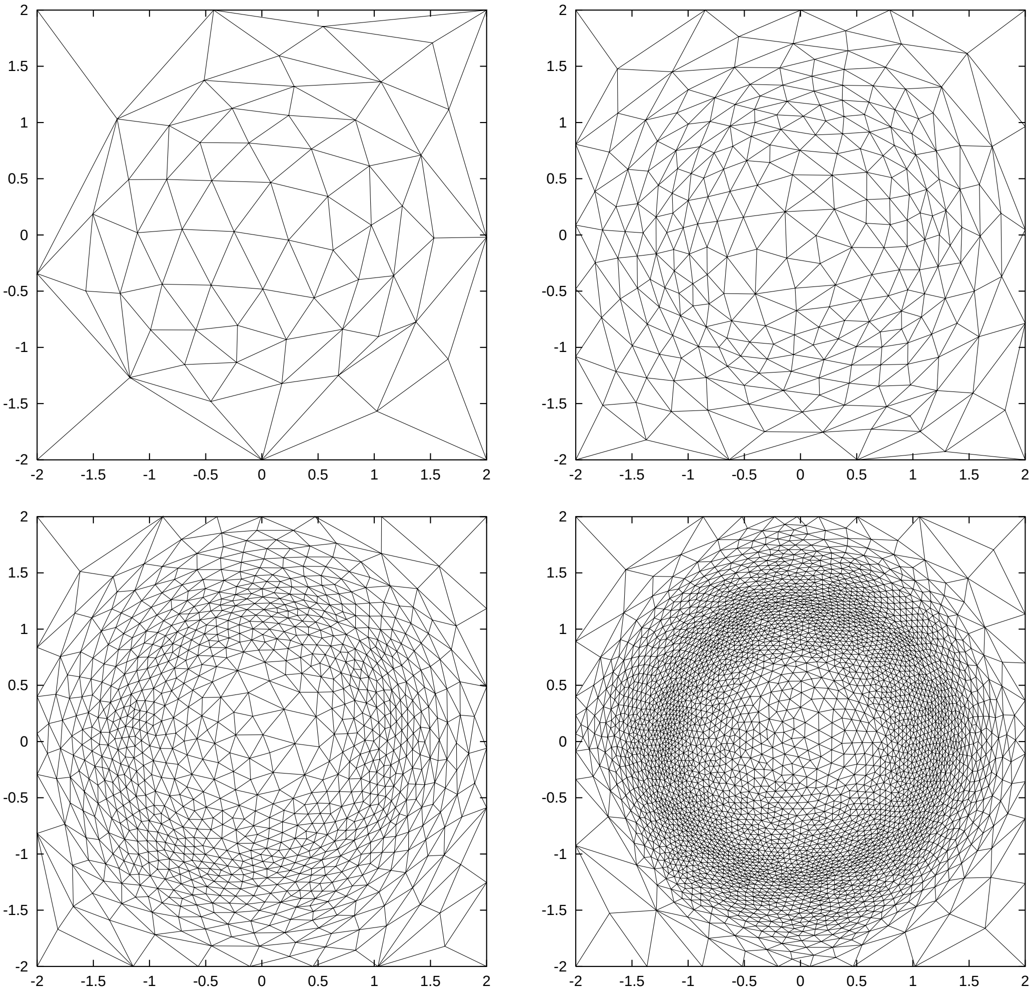

Figure 1.14: Adapted meshes $\mathcal{T}_{h_j}^{\mathrm{G}}$ for $j = 1, \ldots, 4$.

5.2 Efficiency of AMA

We apply AMA to the numerical solution of a two dimensional elliptic equation, where the exact solution is known and we demonstrate the practical usefulness of the proposed algorithm. Moreover, in the previous section we described the construction of a new triangulation if the matrices $\boldsymbol{M}(P_i), P_i \in \sigma_h$ are given. To define $\boldsymbol{M}(P_i)$ we use the relations (4.13) – (4.15), (4.20), (4.24) and (4.31), where the meaning of constants $N_{\mathcal{T}_h}$, ε_1 and p is specified in Section 4.2. Their determination will be demonstrated on the two following numerical examples.

In the rest of Section 5), following we are interested in the two following items:

- find some suitable algorithm how to choose the constants $N_{\mathcal{T}_h}$, ε_1 and p in order to obtain a suitable meshes for practical computation,
- compare the effectivity of AMA approach with a standard adaptive method.

The value $N_{\mathcal{T}_h}$ represents the number of elements of the initial mesh before the first adaptation. Let $\bar{l}$ be a characteristic length of a given domain. It is sufficient that the step size h of the mesh is z times smaller then $\bar{l}$, where we take z from the interval $\langle 1, 10 \rangle$. The d-dimensional measure of the equilateral element (with length of edges h) is equal to $C_1(d)h^d$ where $C_1(2) = \sqrt{3}/4$ and $C_1(3) = \sqrt{2}/12$. Taking into account the relation

$$N_{\mathcal{T}_h} C_1(d) \left(\frac{\bar{l}}{z}\right)^d \approx \text{meas}_d(\Omega), \tag{5.12}$$

we put

$$N_{\mathcal{T}_h} := C_1(d)^{-1} \left(\frac{z}{\bar{l}}\right)^d \text{meas}_d(\Omega). \tag{5.13}$$

Therefore, we can choose the value $N_{\mathcal{T}_h}$ according to (5.13). Numerical experiments show that the choice (5.13) is suitable.

To set the constants ε_1 and p we have applied AMA first for scalar elliptic equations (see [BD77]), where the exact solutions are known.

5.3 A singular diffusion problem

In [BD77], the following problem is formulated. Find $u : \overline{\Omega} \to I\!R^1$ satisfying

$$\Delta u = 0 \qquad \text{in } \Omega, \tag{5.14}$$

where

$$\Omega = \left\{(x_1, x_2) \in I\!R^2, x_1^2 + x_2^2 < 1\right\} \setminus \left\{(x_1, x_2) \in I\!R^2, x_1 \in (0,1), x_2 = 0\right\} \tag{5.15}$$

is the unit circle with a "crack" along the positive x_1 axis. The boundary conditions are $u = \sin(\theta/4)$ along the unit circumference (θ is the angle with respect to the positive x_1 axis), $u = 0$ along the upper surface of the crack, and $\partial u/\partial n = 0$ along the lower surface of the crack. The exact solution of weak problem is

$$u(r, \theta) = r^{1/4} \sin(\theta/4) \tag{5.16}$$

(r and θ are the polar coordinates), which belongs to $H^1(\Omega)\setminus H^2(\Omega)$, with a singularity in the origin. If h denotes the longest edge size in the triangulation, it is a classical result ([BD77], p. 4126) that

$$\|u - u_h\| \equiv \|\nabla(u - u_h)\|_{L^2(\Omega)} \leq \beta h^{\kappa}, \tag{5.17}$$

with $\kappa < \frac{1}{4}$ (remember that for regular problems ($u \in H^2(\Omega)$) $\kappa = 1$, see [Fei93], 2.9.41). Here u denotes the exact solution and u_h the piecewise linear finite element numerical solution of (5.14).

The fundamental question is the following: *It is possible to increase the convergence rate (5.17) with $\kappa < \frac{1}{4}$ if we use the anisotropically adapted grids?* For anisotropic meshes, it is natural to measure the rate of convergence not with respect to the mesh size h but with respect to the number of degrees of freedom, e.g. number of elements of $\mathcal{T}_h$. We define *average mesh size* by

$$h_A \equiv (\#\mathcal{T}_h)^{-1/2}, \tag{5.18}$$

where $\#\mathcal{T}_h$ denotes the number of elements of $\mathcal{T}_h$. Obviously, we have for uniform meshes

$$\frac{h}{h_A} \approx \text{constant independent of } h. \tag{5.19}$$

Uniform meshes

At first, we have solved (5.14) using uniform meshes. In order to generate uniform meshes, we applied the AMA algorithm described in Section 2.2 with isotropic matrices, i.e. with $\boldsymbol{M}(P_i) = c\boldsymbol{I}$, $P_i \in \sigma_h$, where c is the constant given by (4.20), where we prescribe the number of elements $N_{\mathcal{T}_h}$. The equation (5.14) is solved by the standard finite element method (FEM) with conforming piecewise linear elements. Then the approximate solution u_h is sought in the space

$$\bar{U}_h = \{u_h \in C(\Omega_h); u_h|_T \in P_1(T)\ \forall T \in \mathcal{T}_h, u_h|_{\Gamma_D} = u_D\}, \tag{5.20}$$

where $\Gamma_D \subset \partial\Omega$ is the part of the boundary, where the Dirichlet boundary condition u_D is prescribed.

Let $u_h \in \bar{U}_h$ be the approximate solution of the discrete version of problem (5.14) computed on triangulation $\mathcal{T}_h$. As we know the exact solution (5.16), we compute the error (5.15)

$$e_h \equiv \|u - u_h\| = \left(\sum_{T\in\mathcal{T}_h} \int_T |\nabla(u - u_h)|^2 \, \mathrm{d}x \right)^{1/2}, \tag{5.21}$$

where the integral is evaluated by the seven points numerical quadrature exact for the polynomials of degree less or equal to 3.

We define the operator $\bar{\Pi}_h$ using (2.11), where we put $\|\cdot\|_X = \|\cdot\|$, i.e. $\bar{\Pi}_h : w \in H^1(\Omega) \mapsto \bar{\Pi}_h w \in \bar{U}_h$ such that

$$\|w - \bar{\Pi}_h w\| = \min_{w_h \in \bar{U}_h} \|w - w_h\|. \tag{5.22}$$

Using the standard technique for seeking the minimum of a function of several variables, it is possible to show, that for each $w \in H^1(\Omega)$, there exists a uniquely determined function $\bar{\Pi}_h w \in \bar{U}_h$. Then we can compute the *minimal discretization error*

$$e_h^{\min} \equiv \|u - \bar{\Pi}_h u\| = \left(\sum_{T\in\mathcal{T}_h} \int_T |\nabla(u - \bar{\Pi}_h u)|^2 \, \mathrm{d}x \right)^{1/2}. \tag{5.23}$$

$N_{\mathcal{T}_h}$	$\#\mathcal{T}_h$	h	e_h	$e_h^{\min}$	CPU (s)
125	156	0.3079	1.0120	0.8094	7
500	628	0.1540	0.8333	0.6914	9
2000	2442	0.0789	0.7159	0.5958	23
8000	9831	0.0395	0.5837	0.4954	154
32000	39158	0.0199	0.4707	0.4036	2408

Table 1.1: Singular diffusion problem – uniform meshes.

The function $\bar{\Pi}_h u$ is the *best possible numerical solution* of (5.14) – (5.15) in the space $\bar{U}_h$ in the sense of the discretization error measured in the $\|\!|\cdot|\!\|$ norm.

Table 1.1 contains the results of computation on five different uniform meshes. The value $N_{\mathcal{T}_h}$ is the number of prescribed elements in (4.20) and $\#\mathcal{T}_h$ is the real number of triangles of $\mathcal{T}_h$. The value h in Table 1.1 denotes the step size of a uniform mesh

$$h \equiv \max_{T\in\mathcal{T}_h} \operatorname{diam}(T). \tag{5.24}$$

The values e_h and $e_h^{\min}$ are given by (5.21) and (5.23), respectively. The last column introduces the CPU time including the mesh generation process. The computation was performed on a workstation DEC Alpha EV56/500MHz.

Using the data from Table 1.1, we can set the rate of convergence. Let us suppose that

$$e_h = \beta h^{\kappa}, \tag{5.25}$$

where e_h is the discretization error given by (5.21), h the step size of the mesh given by (5.24), β a positive constant and κ the *rate of convergence*. To set κ we apply the *method of the least squares* (for more detail see, e.g., [FD98]). Therefore we obtain $\kappa = 0.28$ and $\beta = 1.49$, which corresponds with the theoretical result (5.17).

The smallest value of $e_h = 0.4707$ (computed for $N_{\mathcal{T}_h} = 32000$) is in agreement with low rate of convergence κ. We can easily estimate, how many triangles would be required, so that the discretization error would be significantly small, e.g. $e_h = 0.0567$ (=the smallest error obtained with AMA, see Section 5.3, Table 1.2). From the computed values $\kappa = 0.28$ and $\beta = 1.49$ and the relation (5.25), we obtain $h = 3.5 \cdot 10^{-15}$, which corresponds approximately to $5.8 \cdot 10^{29}$ equilateral triangles.

Anisotropic meshes

Using the AMA method, we obtain more suitable meshes for the resolution of (5.14) in the sense that the minimal discretization error $e_h^{\min}$ is smaller. As we know the exact solution we compute the Hessian matrices exactly, and we compare them with their numerical approximation given by (4.3) and (4.4). The matrices $\boldsymbol{H}(P_i)$ are given by (4.13) – (4.14). After simple but time consuming calculation we have

$$\boldsymbol{H}(P_i) = \frac{3}{16} r_i^{-7/4} \boldsymbol{I}, \quad P_i \in \sigma_h, \tag{5.26}$$

where $\boldsymbol{I}$ is the unit matrix and r_i is the radius of P_i.

$N_{\mathcal{T}_h}$	ε_1	p	$\#\mathcal{T}_h$	e_h	$e_h^{\min}$	CPU (s)
10	10^{10}	50.	759	0.1481	0.1376	122
10	10^{10}	100.	1577	0.1101	0.1022	245
10	10^{10}	150.	2404	0.0992	0.0906	405
10	10^{10}	200.	3194	0.0944	0.0835	610
10	10^{20}	50.	745	0.1378	0.1336	122
10	10^{20}	100.	1610	0.0860	0.0841	247
10	10^{20}	150.	2464	0.0683	0.0673	426
10	10^{20}	200.	3375	0.0567	0.0554	666
10	10^{30}	50.	745	0.1378	0.1336	128
10	10^{30}	100.	1598	0.0942	0.0880	222
10	10^{30}	150.	2452	0.0669	0.0656	400
10	10^{30}	200.	3376	0.0567	0.0554	649

Table 1.2: Singular diffusion problem – anisotropic mesh adaptation.

Let $\alpha(\|\bar{\boldsymbol{H}}(P_i)\|)$ be given by (4.22). Then the edge-optimal triangulation consists of almost equilateral triangles (provided that $\mathrm{diam}(T) \ll r$), whose areas depend on the radius of their barycentres r_T according

$$\mathrm{meas}(T) \approx \frac{3\sqrt{3}}{4}\left[c\left(1+\frac{3}{16}\hat{C}r_T^{-7/4}\right)\right]^{-1}. \tag{5.27}$$

It follows from (5.27) that for $r_T \to 0$ the triangles of the optimal triangulation collapse, i.e. $\mathrm{meas}(T) \to 0$ and the condition (4.23) is not satisfied.

On the other hand, let $\alpha(\|\bar{\boldsymbol{H}}(P_i)\|)$ be given by (4.24). Then for $\varepsilon_1 \gg 1$ we have

$$\mathrm{meas}(T) \approx \frac{3\sqrt{3}}{4}\left[c\left(1+\frac{\frac{3}{16}p\varepsilon_1 r_T^{-7/4}}{\varepsilon_1+\frac{3}{16}pr_T^{-7/4}}\right)\right]^{-1}, \tag{5.28}$$

which for $r_T \to 0$ gives

$$\mathrm{meas}(T) \to \frac{3\sqrt{3}}{4}\frac{1}{c(1+\varepsilon_1)}, \tag{5.29}$$

and the condition (4.23) is satisfied. From this point of view, relation (4.24) seems to be better than (4.22).

In order to solve (5.14), we use the same numerical scheme as for uniform meshes. We performed the mesh adaptation for various combinations of value of ε_1 and p, see Table 1.2. The meaning of values in Table 1.2 is similar as in Table 1.1, $\#\mathcal{T}_h$ denotes the real number of elements after last mesh adaptation and CPU is the time cost for all levels of adaptation including mesh generation.

Figure 1.15 shows the dependence of $h_T \equiv \mathrm{diam}(T)$, $T \in \mathcal{T}_h$ on r_T for the triangulation generated with $\varepsilon_1 = 10^{30}$ and $p = 200$. The separated points present the dependence of the value of diameter h_T on corresponding r_T of all $T \in \mathcal{T}_h$, while the dashed line presents the theoretical dependence given by

$$h_T = \frac{4}{\sqrt{3}}\left(\mathrm{meas}(T)\right)^{1/2}, \tag{5.30}$$

where $\mathrm{meas}(T)$ is given by (5.28). The good mutual correspondence is evident.

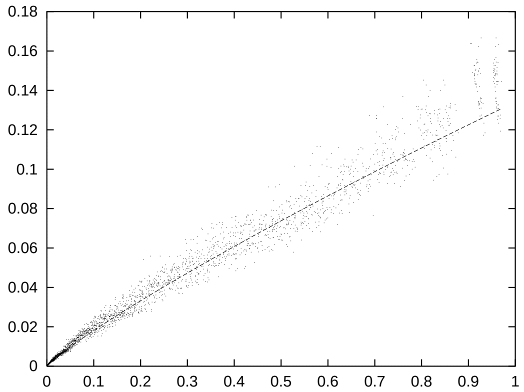

Figure 1.15: Singular diffusion problem – dependence of $h(T)$ on r_T for $\varepsilon_1 = 10^{30}$ and $p = 200$.

ε_1	κ	β
10^{10}	0.637	0.0783
10^{20}	1.172	0.8200
10^{30}	1.221	0.9038

Table 1.3: Singular diffusion problem – anisotropic mesh adaptation, rates of convergence.

Using the data from Table 1.2, we set the convergence rate for the anisotropic meshes. From (5.19) and (5.25), we have

$$e_h = \beta(\#\mathcal{T}_h)^{-\kappa/2}. \tag{5.31}$$

In order to set κ, we again apply the method of the least squares. The results are viewed in Table 1.3. We see that the rate of convergence is (for $\varepsilon_1 = 10^{20}$ and $\varepsilon_1 = 10^{30}$) comparable with the regular case ($u \in H^2(\Omega)$), i.e., $\kappa \approx 1$.

Figure 1.16 and Figure 1.17 show two examples of results from Table 1.2. The left figures present the used triangulation and the right ones the corresponding lines of constant values of the numerical solution u_h together with the best possible numerical solution $\bar{\Pi}_h u$. The second and third rows of figures represents the 10 times and 100 times zooms of the neighbourhood of the singularity point, respectively.

Isotropic mesh refinement

We compare AMA technique with a "clasical" adaptive method based on a posteriori error estimation. We define the *error indicator* (compare with (5.21))

$$g(T) = \left(\int_T |\nabla(u - u_h)|^2 \, \mathrm{d}x \right)^{1/2}, \quad T \in \mathcal{T}_h, \tag{5.32}$$

where u and u_h are the exact and numerical solutions, respectively. The error indicator (5.32) is the "best possible" a posteriori error indicator as it is based on the direct computation of the error (the exact solution is known).

All triangles $T \in \mathcal{T}_h$ with $g(T)$ greater than a given tolerance η are refined by the red-green technique [BSW83]. The computation was performed for 2 values of the tolerance η and for

η	No. of ref.	$\#\mathcal{T}_h$	e_h	$e_h^{\min}$	CPU (s)
$2E-04$	8	1474	0.2175	0.1906	50
$2E-04$	10	1538	0.1658	0.1485	66
$2E-04$	12	1602	0.1330	0.1225	82
$1E-04$	8	2127	0.2121	0.1843	78
$1E-04$	10	2181	0.1585	0.1403	102
$1E-04$	12	2255	0.1237	0.1122	128

Table 1.4: Singular diffusion problem – red-green refinement.

8, 10 and 12 level of the mesh refinement. It is necessary to prescribe the maximal number of levels of refinement, as the singularity causes that there exist some triangles with $g(T) > \eta$ for any level of refinement. The achieved results are viewed in Table 1.4 and one of them is visualized in Figure 1.18.

5.4 A Poisson problem with steep gradients

The previous test, though essential for checking the convergence properties of the adaptive algorithm, allows no evaluation of the performance of the optimizer in relation to the anisotropy. For problem (5.14), both eigenvalues of the Hessian matrix of the solution (5.16) differ only on sign (i.e. $\lambda_1 = -\lambda_2$) and thus the optimal triangulation consists of equilateral triangles.

Consider now a diffusion problem exhibiting steep gradients. Let $\Omega = (0,1) \times (0,1)$, and let u be the solution of

$$-\Delta u(x_1, x_2) = 90x_1^8(1 - x_2^{20}) + 380x_2^{18}(1 - x_1^{10}) \tag{5.33}$$

with boundary conditions $u = 0$ along the edges $\{(x_1, x_2) \in I\!R^2; x_1 = 1,\ x_2 \in (0,1)\}$ and $\{(x_1, x_2) \in I\!R^2; x_1 \in (0,1),\ x_2 = 1\}$, and $\partial u/\partial n = 0$ along $\{(x_1, x_2) \in I\!R^2; x_1 = 0,\ x_2 \in (0,1)\}$ and $\{(x_1, x_2) \in I\!R^2; x_1 \in (0,1),\ x_2 = 0\}$. The exact solution

$$u = (1 - x_1^{10})(1 - x_2^{20}) \tag{5.34}$$

exhibits two steep gradients along the right and top edges, the latter being stronger than the former.

Uniform meshes

The same numerical method as in previous section was applied. Table 1.5 contains the results of computation on four different uniform meshes. In contrast to the problem (5.14) – (5.15), the discretization error e_h of the problem (5.33) (given by (5.21)) differs from the minimal discretized error $e_h^{\min}$ (given by (5.23)) by the terms of order 10^{-4}. It is caused by the fact, that the problem (5.33) has a smooth solution (5.34) and the used numerical scheme gives solution u_h almost identical with $\bar{\Pi}_h u$. Therefore Table 1.5 – Table 1.7 do not contain the value $e_h^{\min}$ and Figure 1.19 - Figure 1.20 do not contains the isolines of $\bar{\Pi}_h u$.

If we again suppose the relation (5.25) we determine the rate of convergence as $\kappa = 1.007$ ($\beta = 11.19$), which corresponds to the theoretical results (5.17) ($\kappa = 1$ for regular problems).

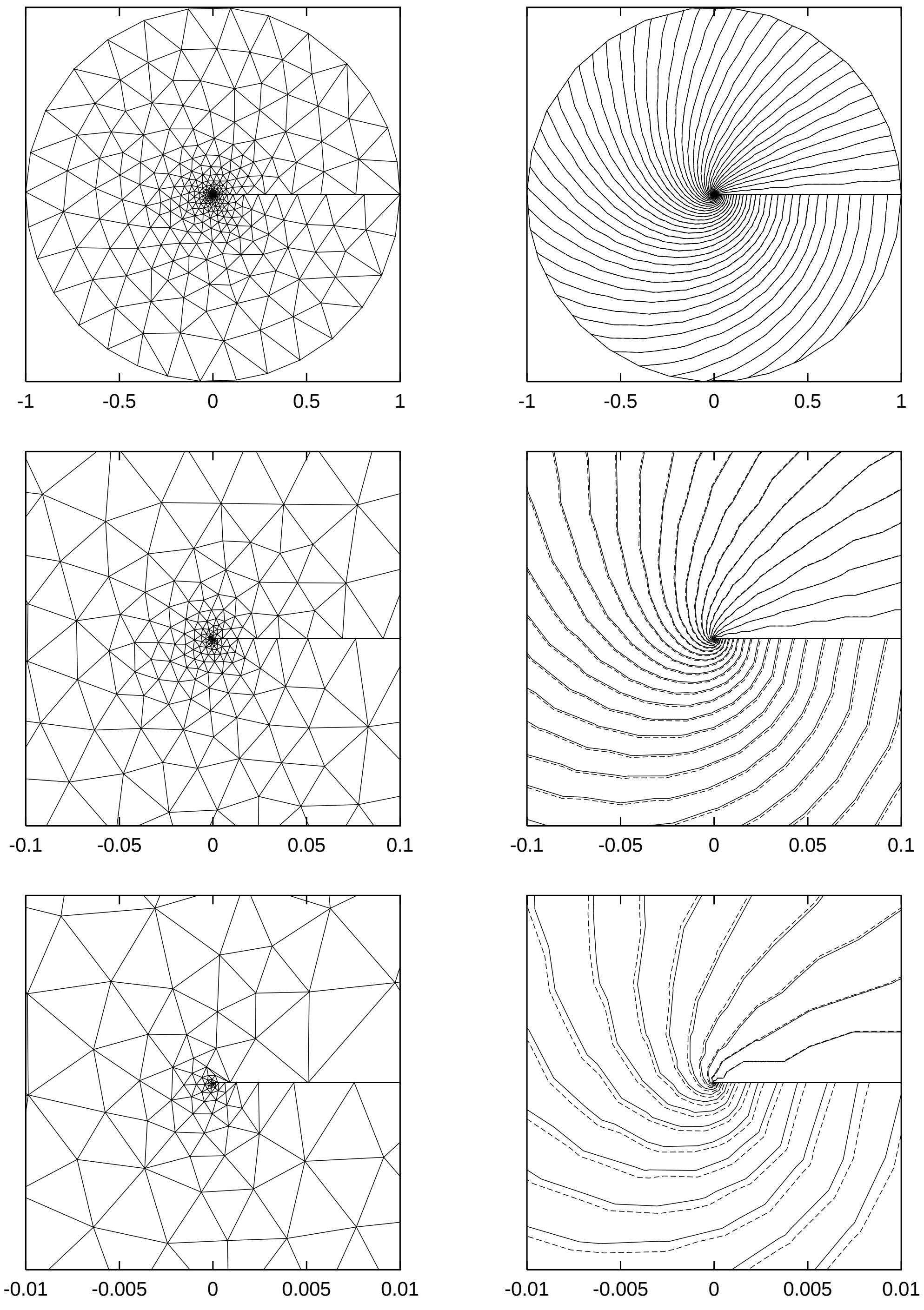

Figure 1.16: Singular diffusion problem, anisotropic mesh adaptation with $N_{\mathcal{T}_h} = 10$, $\varepsilon_1 = 10^{10}$, $p = 50$ ($\#\mathcal{T}_h = 759$) – mesh and isolines of u_h (full line) and $\bar{\Pi}_h u$ (dashed line) with zooms.

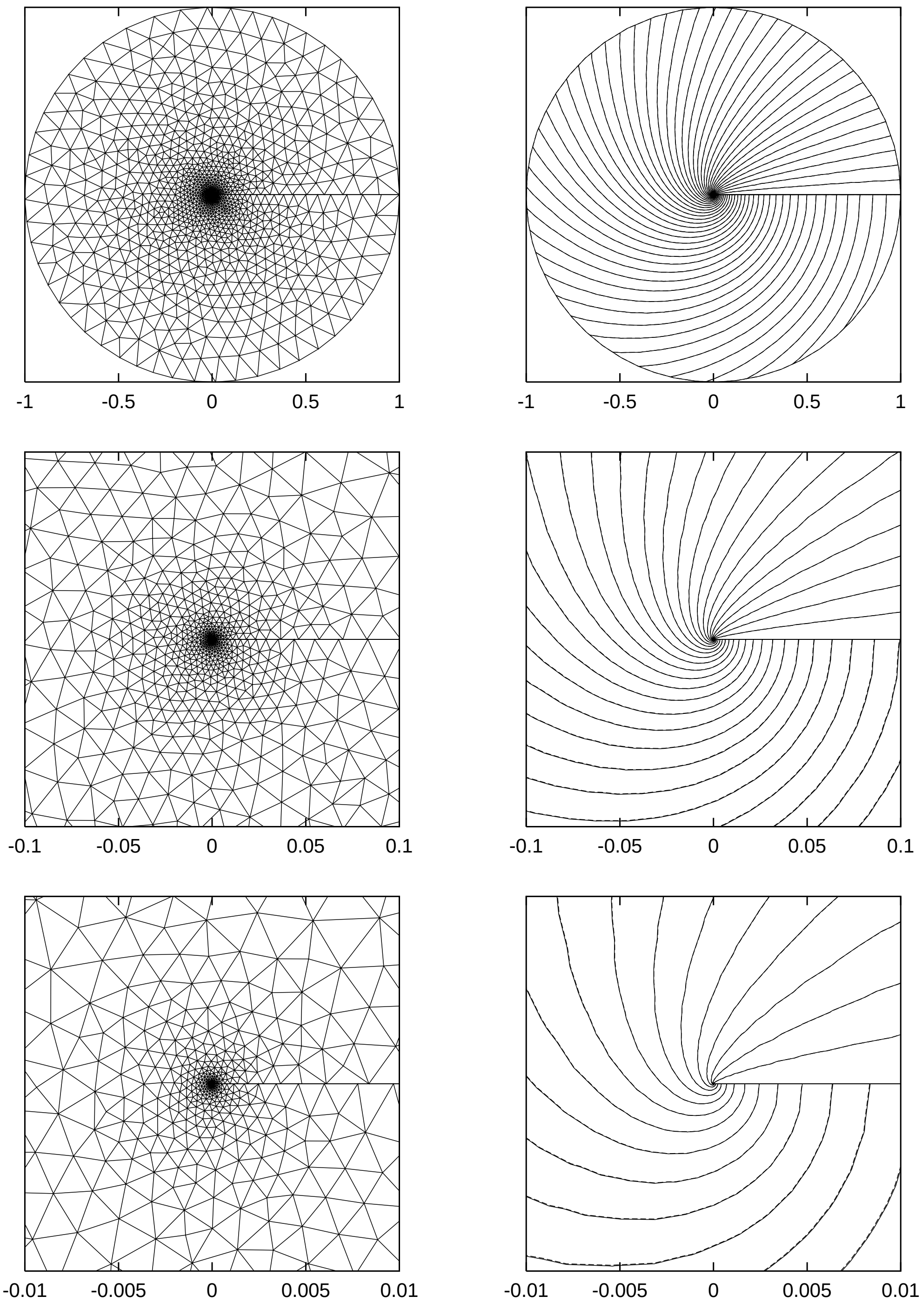

Figure 1.17: Singular diffusion problem, anisotropic mesh adaptation with $N_{\mathcal{T}_h} = 10$, $\varepsilon_1 = 10^{30}$, $p = 200$ ($\#\mathcal{T}_h = 3376$) – mesh and isolines of u_h (full line) and $\bar{\Pi}_h u$ (dashed line) with zooms.

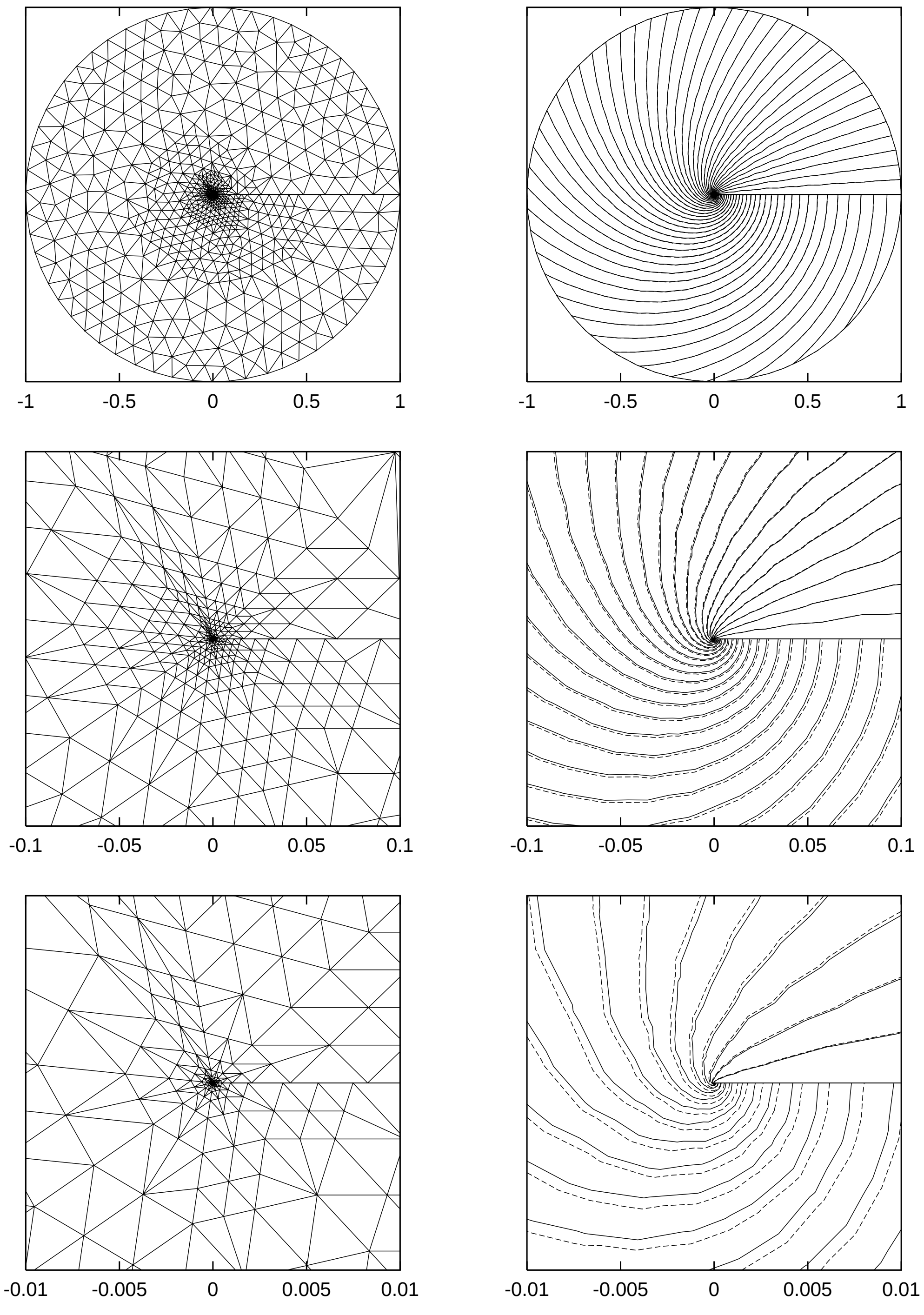

Figure 1.18: Singular diffusion problem, red-green refinement with $\eta = 2 \cdot 10^{-4}$ after 12 level of refinement ($\#\mathcal{T}_h = 1602$) – mesh and isolines of u_h (full line) and $\bar{\Pi}_h u$ (dashed line) with zooms.

$N_{\mathcal{T}_h}$	$\#\mathcal{T}_h$	h	e_h	CPU (s)
125	127	0.1783	1.9281	7
500	546	0.0894	0.9905	9
2000	2314	0.0484	0.5340	29
8000	9332	0.0226	0.2655	252
32000	18950	0.0132	0.1346	87904

Table 1.5: Poisson problem with steep gradients - uniform meshes.

$N_{\mathcal{T}_h}$	ε_1	p	$\#\mathcal{T}_h$	e_h	CPU (s)
1	10^{10}	100	361	0.18392	79
1	10^{20}	100	361	0.18392	79
1	10^{30}	100	361	0.18392	77
1	10^{20}	300	941	0.10783	152
1	10^{10}	300	945	0.10781	153
1	10^{30}	300	941	0.10783	146
1	10^{10}	600	1741	0.07753	287
1	10^{20}	600	1734	0.07666	295
1	10^{30}	600	1734	0.07666	301
1	10^{10}	1000	2883	0.06475	593
1	10^{20}	1000	2908	0.06426	581
1	10^{30}	1000	2908	0.06426	586

Table 1.6: Poisson problem with steep gradients – anisotropic meshes.

Anisotropic meshes

Table 1.6 contains results of the solution of (5.33) on meshes achieved by AMA (all quantities have the same meaning as in Section 5.3). Figure 1.19 shows one example of results from Table 1.6, where the triangulation and the corresponding contours of constant value of the numerical solution u_h are plotted. The expected grid alignment is easily viewed.

Isotropic mesh refinement

We have again applied the red-green refinement method introduced in Section 5.3. In this case, it is possible to apply the mesh refinement till the error indicator $g(T)$ is under the tolerance η for each triangle. The computation was performed for 3 values of the tolerance η. The achieved results are viewed in Table 1.7 and one of them visualized Figure 1.20.

η	No. of ref.	$\#\mathcal{T}_h$	e_h	$e_h^{\min}$	CPU (s)
10^{-3}	5	1041	0.3781	0.1906	35
10^{-4}	6	2778	0.2343	0.1485	49
10^{-5}	7	8583	0.1336	0.1225	175

Table 1.7: Poisson problem with steep gradients – red-green refinement.

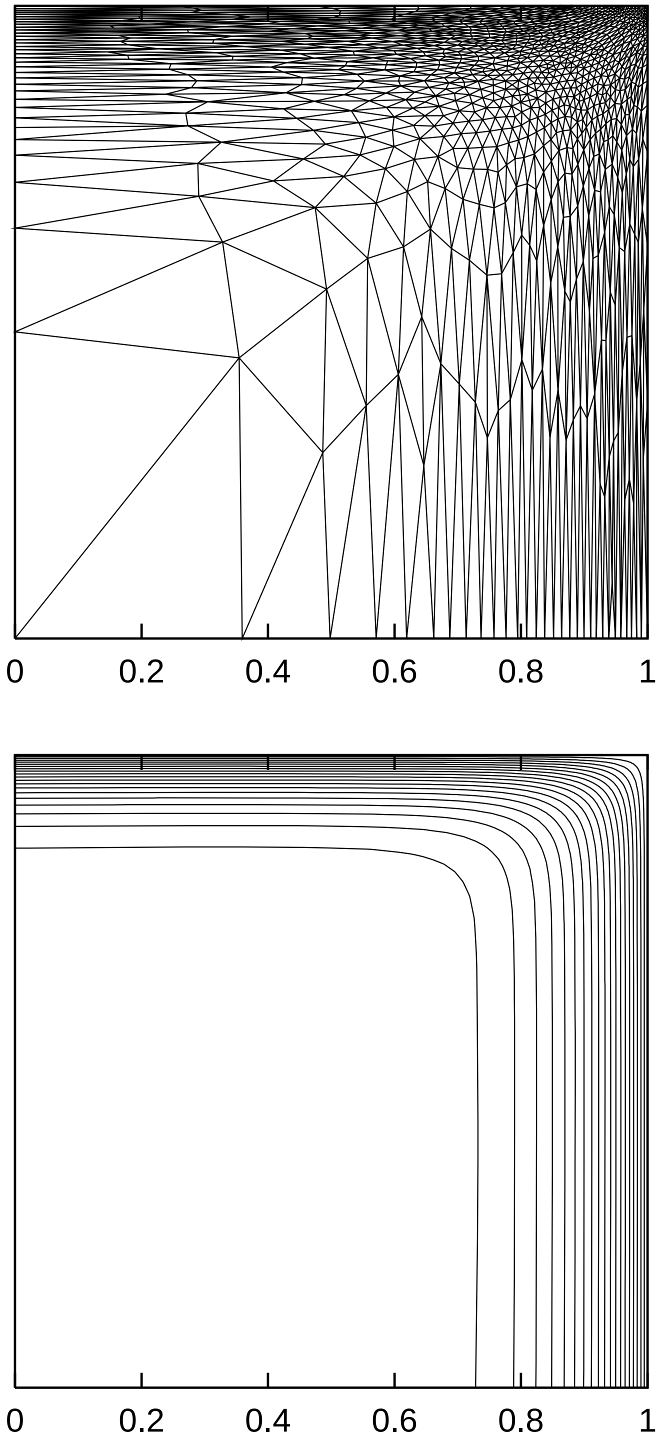

Figure 1.19: Poisson problem with steep gradients - anisotropic mesh adaptation: $N_{\mathcal{T}_h} = 1$, $\varepsilon_1 = 10^{30}$ and $p = 600$ ($\#\mathcal{T}_h = 1734$) - mesh and isolines of u_h.

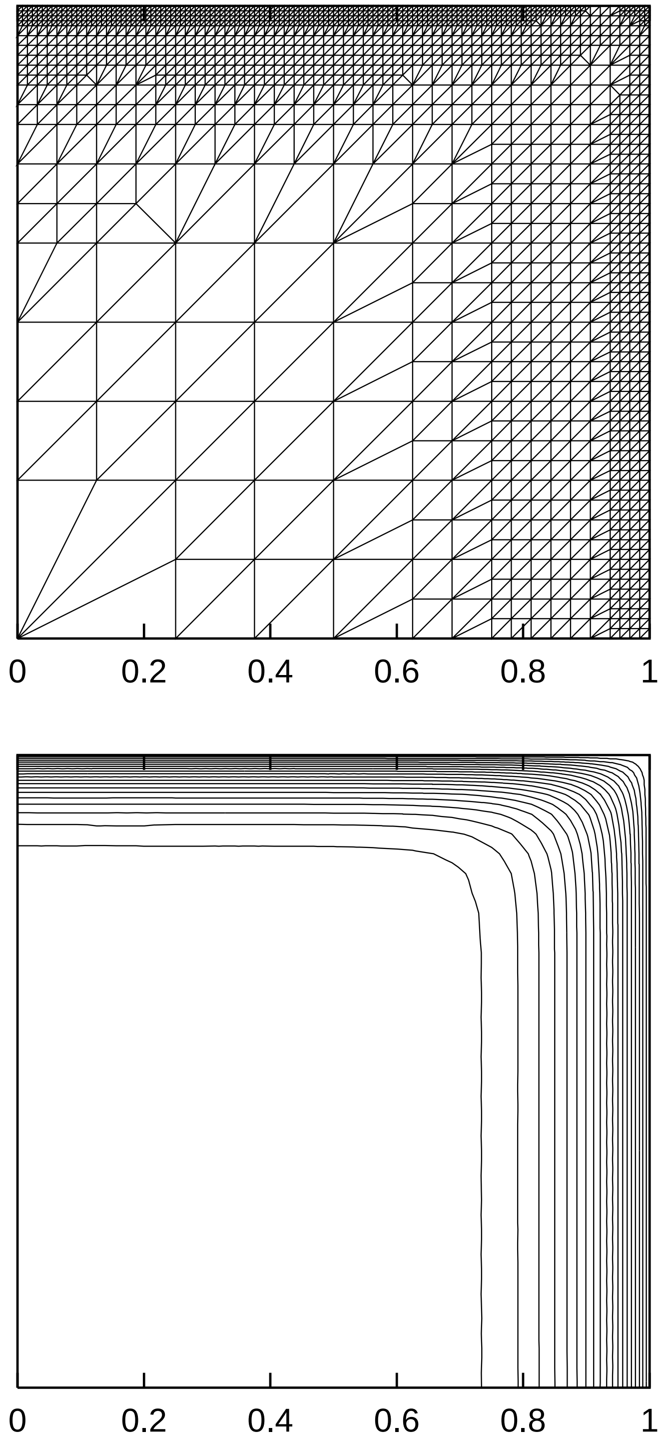

Figure 1.20: Poisson problem with steep gradients – red-green refinement with $\eta = 10^{-5}$ and $p = 1.0$ ($\#\mathcal{T}_h = 8583$) - mesh and isolines of u_h.

5.5 Discussion of the results

From the results presented in Sections 5.3 and 5.4, we can conclude the following:

- The computations show that the discretization error e_h (5.21) is close to $e_h^{\min}$ (5.23) for both uniform and adapted meshes. This is given by the fact, that the used numerical method (conforming linear finite element method) gives sufficiently accurate solution for a given triangulation. This underlines the importance of the necessary condition (2.13).
- The global mesh refinement leads to an enormous requirement on the memory and CPU-time. It is absolutely impractical.
- Applying AMA for singular diffusion problem (with nonregular exact solution – $u \in H^1(\Omega) \setminus H^2(\Omega)$), we achieve the rate of convergence $\kappa \approx 1$, which is the theoretical results for regular case ($u \in H^1(\Omega)$).
- In order to achieve the same level of the computational error, the grids adapted by AMA have a smaller number of elements than the grids adapted by the red-green refinement (RGR). The difference is remarkable namely for the second example.
- On the other hand, the CPU-time is higher for AMA than for RGR. It is caused by the simplicity of considered problem – scalar linear equation. Here the mesh adaption algorithm requires more time than the numerical solution of the Poisson equation. The top efficiency of AMA plays a role in practical problems, where the CPU-time rests the same for the mesh adaptation but it is much higher for the numerical solution of the considered problem.
- Observing results from Table 1.2 and Table 1.6, we see that whereas the increase of ε_1 do not cause essential improvement of results, then the increase of p leads to the decrease of the computational error (and naturally the increase of the number of elements). So that we propose the following strategy how to set the parameters $N_{\mathcal{T}_h}$, ε_1 and p:
  1. prescribe $N_{\mathcal{T}_h}$ according to (5.13)
  2. set ε_1 sufficiently high, e.g. 10^{20}, see (4.30)
  3. start with a small p and increase successively its value till the number of elements is acceptable from the point of view of the computer memory and CPU time.

6 Application of AMA for compressible flow simulation

In this section, we present the use of AMA for the solution of two and three dimensional fluid dynamics problems. We deal with the simulation of inviscid and viscous compressible flow, i.e., we seek a numerical solution of the system of the Euler and Navier-Stokes equations.

The application of AMA for inviscid flow simulation is straightforward, see Section 6.3. On the other hand, the sufficient accurate viscous flow simulation requires additional mesh adaptation techniques which are determined from the physical properties of the flow, namely the presence of boundary layers and wakes.

At first, we present the system of the Navier-Stokes equations and we briefly describe the used numerical method for its solution, which we developed several years ago. Then the application for inviscid and viscous flow are given.

6.1 Governing equations

The complete system describing the motion of compressible viscous flow in a bounded domain $\Omega \subset I\!R^d$ and time interval $(0, \boldsymbol{T})$ is written in the dimensionless form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = \sum_{s=1}^{d} \frac{\partial \boldsymbol{R}_s(\boldsymbol{w}, \nabla \boldsymbol{w})}{\partial x_s} \quad \text{in } Q_{\boldsymbol{T}} = \Omega \times (0, \boldsymbol{T}), \tag{6.1}$$

where

$$\begin{aligned}
\boldsymbol{w} &= (\rho, \rho v_1, \dots, \rho v_d, e)^T, \ \boldsymbol{w} = \boldsymbol{w}(x,t), \quad x \in \Omega, \ t \in (0, \boldsymbol{T}), \\
\boldsymbol{f}_s(\boldsymbol{w}) &= (\rho v_s, \rho v_s v_1 + \delta_{s1} p, \dots, \rho v_s v_d + \delta_{sd} p, (e+p)\, v_s)^T, \ s = 1, \dots, d \\
\boldsymbol{R}_s(\boldsymbol{w}) &= (0, \tau_{s1}, \dots, \tau_{sd}, \sum_{r=1}^{d} \tau_{sr} v_r + q_s)^T, \ s = 1, 2 \\
\tau_{sr} &= \frac{1}{Re}\left(\frac{\partial v_s}{\partial x_r} + \frac{\partial v_r}{\partial x_s} - \frac{2}{3} \mathrm{div}\, \boldsymbol{v}\, \delta_{sr}\right), \ q_s = \frac{\gamma}{Re\, Pr} \frac{\partial \theta}{\partial x_s}, \ s, r = 1, \dots, d \\
p &= (\gamma - 1)\,(e - \rho|\boldsymbol{v}|^2/2), \ e = \rho(\theta + |\boldsymbol{v}|^2/2).
\end{aligned} \tag{6.2}$$

Notation is the following: t – time coordinate, $x = (x_1, \dots, x_d)$ – Cartesian coordinates, ρ – density, p – pressure, θ – temperature, e – total energy, $\boldsymbol{v} = (v_1, \dots, v_d)$ – velocity, δ_{ij} – Kronecker delta, γ – Poisson adiabatic constant, Pr – Prandtl number, Re – Reynolds number. We assume that $\gamma > 1$, Pr and Re are given constants.

The above system is equipped with the initial condition $\boldsymbol{w}(x, 0) = \boldsymbol{w}_0(x), \ x \in \Omega$ and boundary conditions, see [FJS]: At inlet we prescribe $\rho, v_1, \dots, v_d$ and use "do-nothing" condition $\sum_{r=1}^{d}\left(\sum_{s=1}^{d} \tau_{sr} n_s\right) v_r + \frac{\gamma}{Re\,Pr} \partial\theta/\partial n = 0$, on fixed walls we assume that $v_1 = \ldots = v_d = 0, \ \partial\theta/\partial n = 0$ and at outlet we set $-p n_j + \sum_{i=1}^{d} \tau_{sr} n_s = 0, \ s = 1, \dots, d$, and $\partial\theta/\partial n = 0$. (Here $\partial/\partial n$ is the derivative in the direction of the unit outer normal $\boldsymbol{n} = (n_1, \dots, n_d)$ to the boundary $\partial\Omega$.)

6.2 Numerical methods

Our aim is to obtain the steady-state solution of the problem (6.1) – (6.2) by a time marching method. For a flow with a high Reynolds number, the viscous effects are small, and therefore we can consider the viscous terms $\boldsymbol{R}_s,\ s = 1, \dots, d$ as a perturbation of the inviscid system (the Euler equations). We split (6.1) into the inviscid system

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = 0, \tag{6.3}$$

and purely viscous system

$$\frac{\partial \boldsymbol{w}}{\partial t} = \sum_{s=1}^{d} \frac{\partial \boldsymbol{R}_s(\boldsymbol{w}, \nabla \boldsymbol{w})}{\partial x_s} \tag{6.4}$$

and discretize them separately.

The time discretization of (6.3) and (6.4) is carried out with the use of a partition $0 = t_0 < t_1 < t_2 \dots < t_N = T$ of the time interval $(0, T)$. The inviscid system (6.3) is discretized by the cell-centered *finite volume* (FV) *method* on $\mathcal{D}_h = \{D_i\}_{i \in J}$. Here D_i are the finite volumes (which can be triangular, barycentric and dual finite volumes or their 3D analogues), $\mathcal{D}_h$ is a partition of a polygonal approximation Ω_h of the domain Ω and J is a suitable index set. The purely viscous system (6.4) is discretized by the piecewise linear conforming or nonconforming *finite element* (FE) *method* on a triangulation ($d = 2$) or a tetrahedrization ($d = 3$) $\mathcal{F}_h$ of Ω_h, compatible with $\mathcal{D}_h$ in some sense. Then the complete system (6.1) is discretized via *operator inviscid-viscous splitting* (see e. g. [DA96], [FF96], [FFD96]). One time step $t_k \to t_{k+1}$ consists of two fractional substeps: inviscid FV step on the mesh $\mathcal{D}_h$ and viscous FE step on the mesh $\mathcal{F}_h$.

1. *Inviscid FV step on* $\mathcal{D}_h$:
   Assume that the values $\boldsymbol{w}_i^k,\ i \in J$, approximating the solution on the finite volumes D_i at time t_k are known. Compute the values $w_i^{k+1/2},\ i \in J$, from the FV formula

$$\boldsymbol{w}_i^{k+1/2} = \boldsymbol{w}_i^k - \frac{\tau_k}{|D_i|} \sum_{j \in S(i)} H(\boldsymbol{w}_i^k, \boldsymbol{w}_j^k, \boldsymbol{n}_{ij})\, |\Gamma_{ij}|, \tag{6.5}$$

   where $S(i)$ denotes the set of neighbours volumes of D_i and Γ_{ij} denotes the common face between D_i and D_j.

2. *Viscous FE step on* $\mathcal{F}_h$:
   Define the finite element function $\boldsymbol{w}_h^{k+1/2}$ with values $\boldsymbol{w}_h^{k+1/2}(P_i) = \boldsymbol{w}_i^{k+1/2}$ at the vertices $P_i,\ i \in J$ of $\mathcal{F}_h$. At the vertices $P_i \in \partial\Omega_h$, the viscous Dirichlet boundary conditions and suitable extrapolation are used. Compute the finite element function $\boldsymbol{w}_h^{k+1}$ as the solution of the following problem:

   (a) $\boldsymbol{w}_h^{k+1}$ satisfies the viscous Dirichlet boundary conditions,

   (b)

$$(\boldsymbol{w}_h^{k+1}, \varphi_h)_h = (\boldsymbol{w}_h^{k+1/2}, \varphi_h)_h - \tau_k\, a_h(\boldsymbol{w}_h^{k+1/2}, \varphi_h) \tag{6.6}$$

   for all test functions $\varphi_h = (\phi_1, ..., \phi_4)$ such that ϕ_j $(j = 1, ..., 4)$ is continuous in Ω, linear on each $T \in \mathcal{F}_h$ and vanishes on the part of $\partial\Omega$, where the j-th component w_j of the state vector $\boldsymbol{w}$ satisfies the Dirichlet boundary condition.

In (6.5), H is the numerical flux approximating

$$H(\boldsymbol{w}_i^k, \boldsymbol{w}_j^k, \boldsymbol{n}_{ij}) \approx \sum_{s=1}^{d} f_s(\boldsymbol{w}(x, t_k))(\boldsymbol{n}_{ij})_s, \tag{6.7}$$

where $(\boldsymbol{n}_{ij})_1 \ldots, (\boldsymbol{n}_{ij})_d$ are the components of the unit outer normal $\boldsymbol{n}_{ij}$ on Γ_{ij} pointed out of D_i. For the evaluation of the numerical flux, the Osher-Solomon scheme [OS82], [FŠ98] was used.

In (6.6), the forms $(\boldsymbol{w}, \varphi)_h$ and $a_h(\boldsymbol{w}, \varphi)$ denote the approximations

$$\begin{aligned} (\boldsymbol{w}, \varphi)_h &\approx \int_\Omega \boldsymbol{w}\, \varphi \, dx, \\ a_h(\boldsymbol{w}, \varphi) &\approx \int_\Omega \sum_{s=1}^{d} \boldsymbol{R}_s(\boldsymbol{w}, \nabla \boldsymbol{w})\, \partial\varphi/\partial x_s \, dx, \end{aligned} \tag{6.8}$$

which were obtained with the aid of numerical quadrature using the vertices of $T \in \mathcal{F}_h$ as integration points.

We developed three different methods. The first method uses the dual finite volume and conforming FE (see [FFL95]), the second one uses triangular FV and conforming FE (see [FFD96]) and the third one uses the barycentric FV and nonconforming FE (see [DFFK02]).

6.3 AMA for inviscid flow simulation

The motion of inviscid compressible flow is described by the system of the Euler equations (6.3). The numerical solution is performed by the finite volume method, which is in fact the first step of the method presented in Section 6.2, i.e., *Inviscid FV step on the mesh $\mathcal{D}_h$*. All results in Section 6.3 was computed by the FV method on triangular or tetrahedral meshes, so that we put $\mathcal{D}_h := \mathcal{T}_h$, where $\mathcal{T}_h$ was obtained by AMA technique.

In order to apply AMA for inviscid flow simulation, we chose the density of the flow ρ as the *significant* quantity. In order to compute the Hessian matrices we simply put $u_h := \rho_h$ in (4.4). Then the whole computational process from Section 4.3 is applied.

AMA technique is applied for three 2D benchmark cases and two 3D problems. The first example deals with the transonic flow around NACA0012 profile. It shows that the multilevel computational process described in Section 4.3 terminates after a finite number of levels of adaptation. The second one compares the AMA technique with two other adaptive methods for the case of the flow through the forward facing step. The third example is a benchmark with complicated geometry of shock waves. The last two examples show the application of AMA for threedimensional problems.

Our aim is to obtain a steady state solution (except 2D and 3D forward facing steps, which are unsteady). It is achieved by a time stabilization technique for $\boldsymbol{T} \to \infty$, i.e. we apply the computational process till the following residuum criterion is satisfied

$$\frac{1}{\tau_k} \left\| \frac{\rho^{k+1} - \rho^k}{\rho^k} \right\|_{L^1(\Omega)} < \mathsf{TOL}, \tag{6.9}$$

where ρ^k is the value of the density considered on the time level t_k, τ_k is the time step and $\mathsf{TOL} > 0$ is a given tolerance. All computations were performed on a workstation DEC Alpha EV56/500MHz.

k = level of adaptation	$\#\mathcal{T}_{h_k}$	$\Delta\#\mathcal{T}_{h_k}$	CPU-time (s) adaptation	solver
0	714	–	54	156
1	1483	769	167	289
2	7472	5989	192	1545
3	12254	4785	185	2689
4	14522	2268	231	3352
5	15619	1097	168	3521
6	16270	651	172	3683
7	16525	255	154	3765

Table 1.8: NACA0012, mesh adaptation levels, $\#\mathcal{T}_{h_k}$ = number of elements of $\mathcal{T}_{h_k}$, $\Delta\#\mathcal{T}_{h_k} \equiv \#\mathcal{T}_{h_k} - \#\mathcal{T}_{h_{k-1}}$ and CPU-times for mesh adaptation and computation of the Euler equations itself.

Transonic flow around NACA 0012 profile

We present the simulation of the inviscid flow around NACA 0012 profile with the inlet Mach number ($= |\boldsymbol{v}|/(\gamma p/\rho)^{1/2}$) $M_{\rm in} = 0.8$ and angle of attack $\alpha = 1°25'$. We performed seven levels of AMA algorithm with $c = 200$, $\varepsilon_1 = 10^6$ and $p = 1000$. Table 1.8 contains the number of elements $\#\mathcal{T}_{h_k}$, $k = 0, \dots, 7$ of all levels of adaptation and moreover CPU times needed for the mesh adaptation (left column) and for the solution of the Euler equations itself (right column). Figures 1.21–1.28, show the triangulations and the isolines of the Mach number of all levels of adaptation. Wee see that from very poor results obtained for the initial triangulation, we obtain good results with very sharp shock waves. Figures 1.29 show the distributions of the Mach number along the profile. Finally, the convergence of residuum is plot at Figure 1.30.

The results for the last two levels of refinement (Figure 1.27 and Figure 1.28) are almost identical, which corresponds to the assertion that the AMA algorithm "converges" (see Section 4.3). This is also viewed in Table 1.8. The results from "old" mesh are interpolated on the new one in order to save the CPU time. It is vied at Figure 1.30 that for these cases the residuum decreases very fast.

Transonic flow through the forward facing step

We deal with the numerical simulation of inviscid flow through a classical benchmark example – the forward facing step, see [WC84]. The geometry of the problem is viewed in Figure 1.31. The computations of the inviscid Euler equations were performed with the initial conditions: $\rho_0 = 1.4$, $\boldsymbol{v} = (3, 0)$, $p = 1$ and the boundary conditions taken from the initial conditions at inlet and outlet part of $\partial\Omega$. On the impermeable walls, we naturally put $v_n \equiv \boldsymbol{v} \cdot \boldsymbol{n} = 0$. We investigate the solution at the time $\boldsymbol{T} = 2.5$. We compare the efficiency of AMA with two other adaptive methods, a shock indicator [Dol98b], [FDF94] and a residual indicator [Kli00].

The shock indicator (SHOCK) is based on physical properties of shock waves. It checks the density jumps up in the direction of a flow. We define the shock indicator

$$g(i) = \frac{1}{\mathrm{diam}(T_i)} \max_{j \in s(i)} \left[- (\rho_i - \rho_j)\, \boldsymbol{v}_i \cdot \boldsymbol{n}_{ij} \right]^+, \tag{6.10}$$

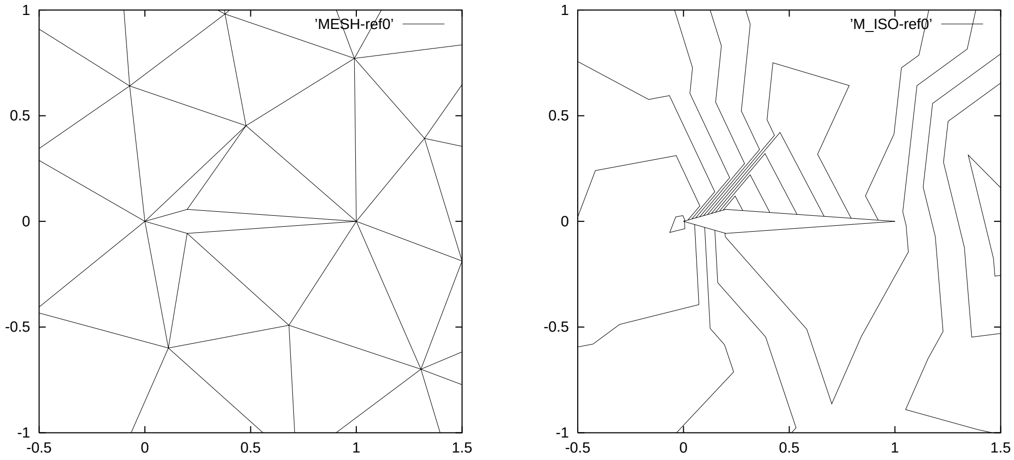

Figure 1.21: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, basic mesh.

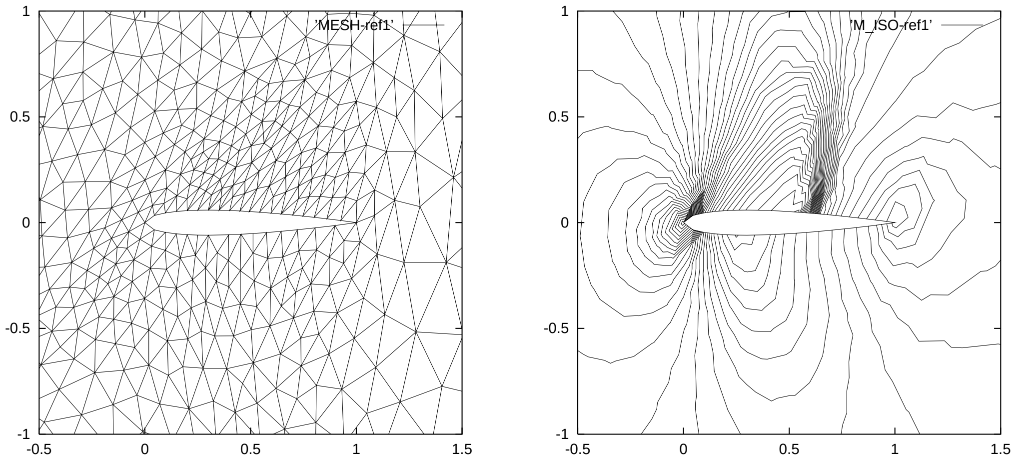

Figure 1.22: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after first adaptation.

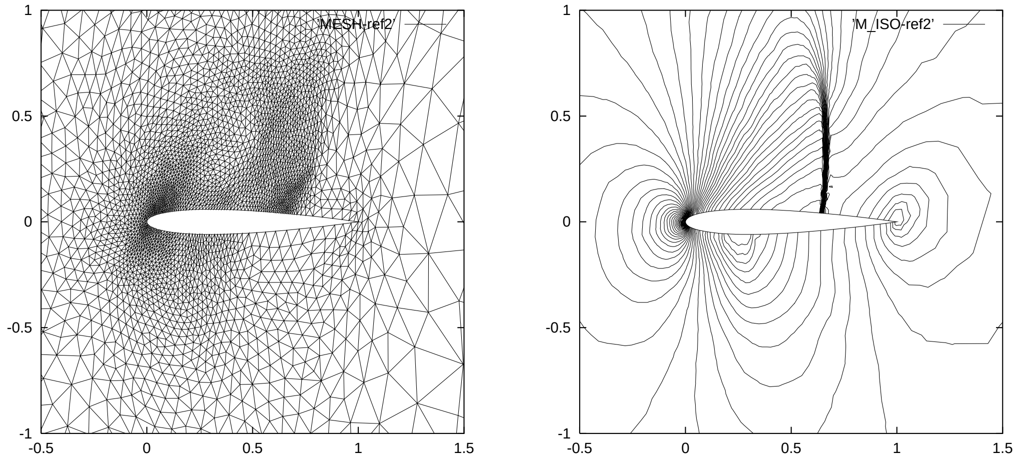

Figure 1.23: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after second adaptation.

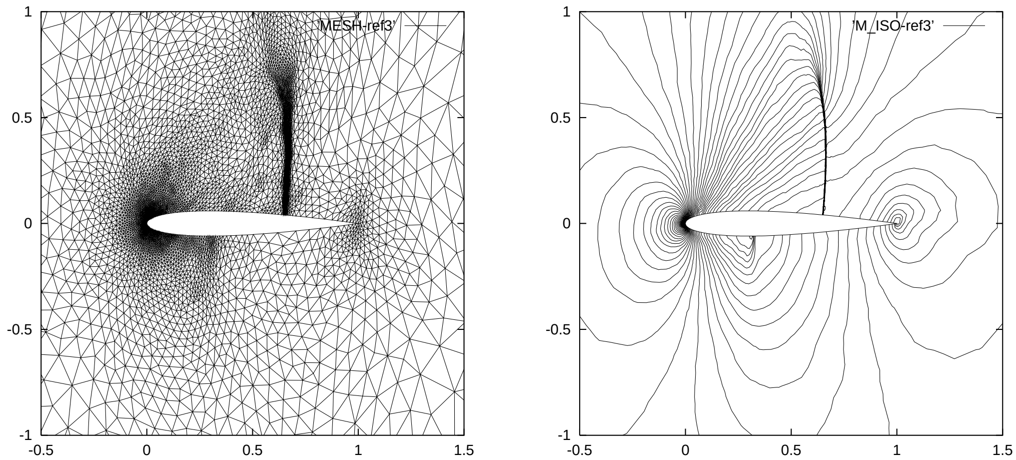

Figure 1.24: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after third adaptation.

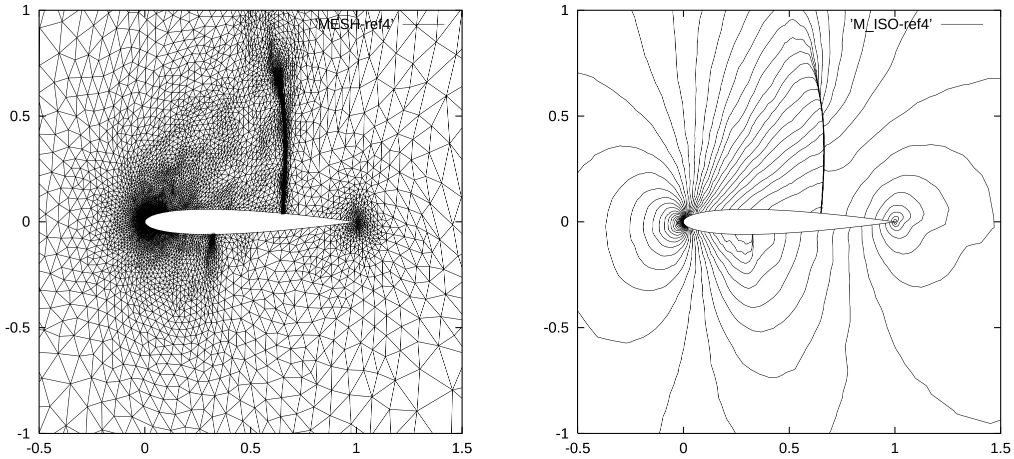

Figure 1.25: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after fourth adaptation.

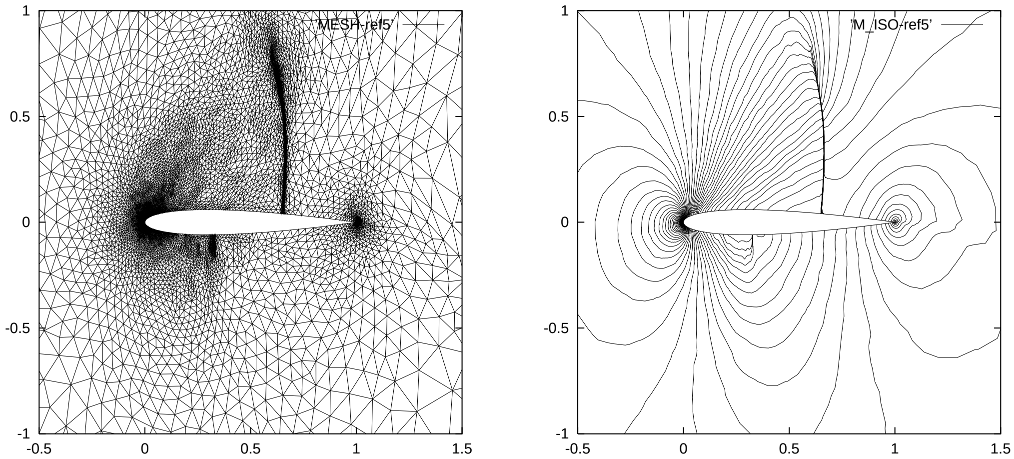

Figure 1.26: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after fifth adaptation.

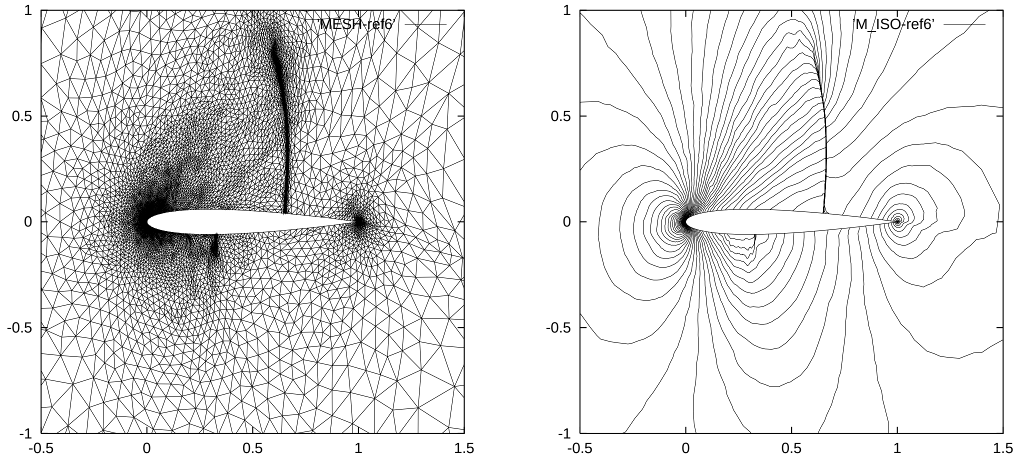

Figure 1.27: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after sixth adaptation.

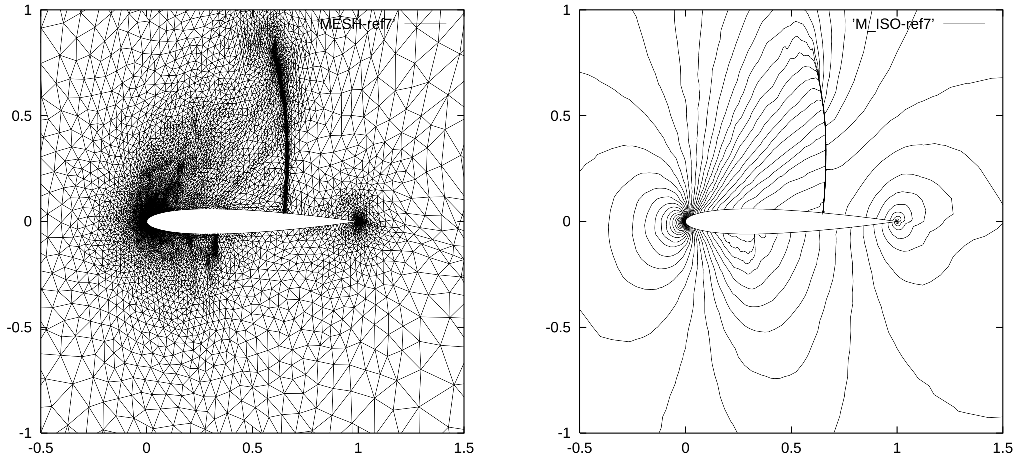

Figure 1.28: Inviscid flow around NACA 0012, triangulation and isolines of Mach number, mesh after seventh adaptation.

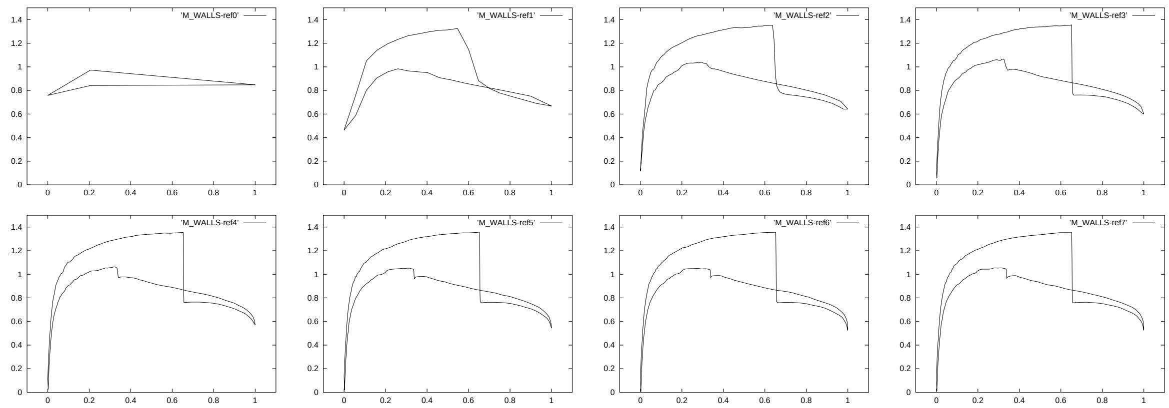

Figure 1.29: Inviscid flow around NACA 0012, Mach number distributions along the profile for all levels of mesh adaptation.

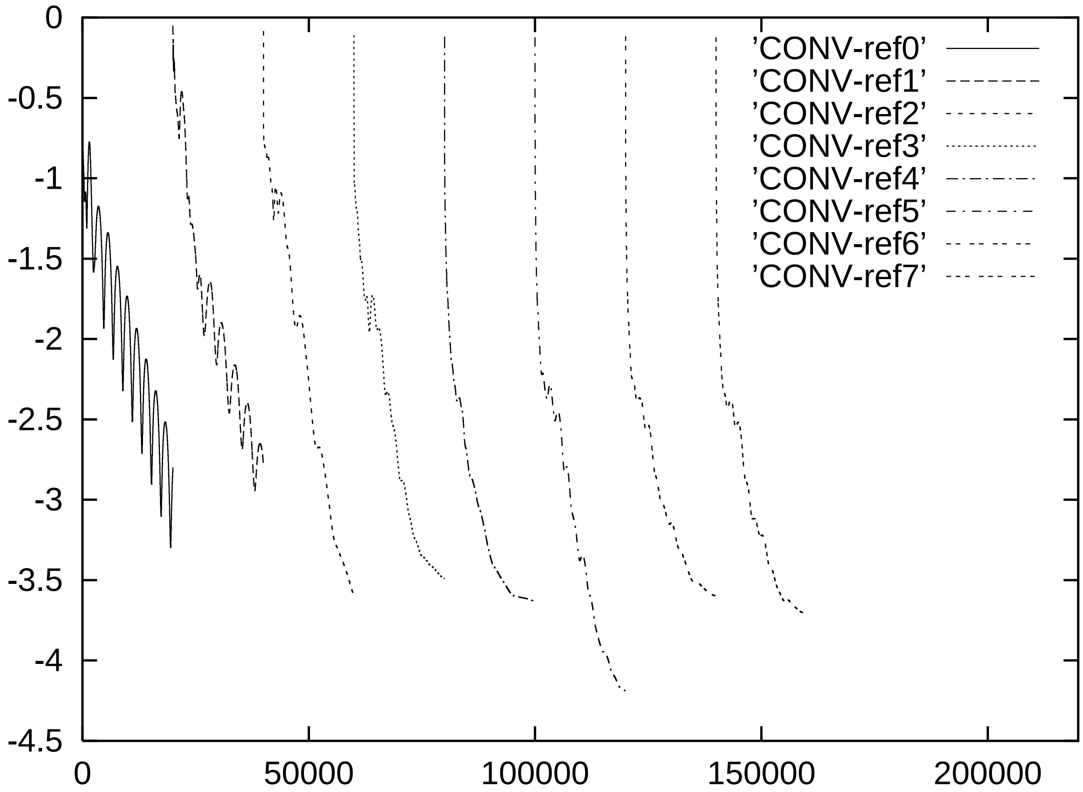

Figure 1.30: Inviscid flow around NACA 0012, history of convergence for all levels of mesh adaptation.

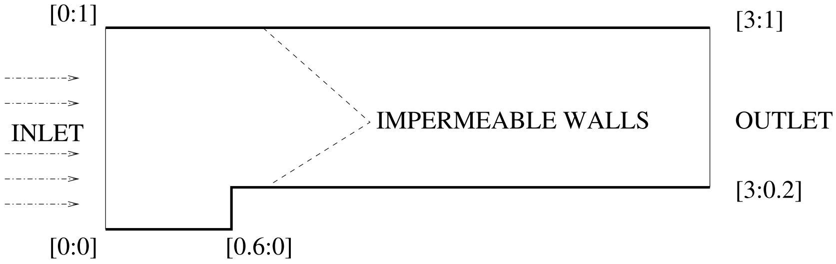

Figure 1.31: Inviscid benchmark: forward facing step.

where $s(i)$ is a set of neighbouring triangles of T_i, ρ_i and $\boldsymbol{v}_i$ is the density and velocity corresponding to T_i and $\boldsymbol{n}_{ij}$ is the unit outer normal to the common edge between triangles T_i and T_j. Moreover we put $a^+ = \max(a, 0)$. If the value of $g(i)$ is higher then a given tolerance η then T_i is refined by the red–greed refinement technique, see [BSW83].

The residual indicator (RESID) is based on the weak formulation of the Euler equations. We define the value

$$r(i) = \sup_{\varphi_h \in V_i} \frac{|(\mathcal{A}(\boldsymbol{w}_h), \varphi_h)|}{\|\varphi_h\|_{H_0^1(T_i)}}, \tag{6.11}$$

where $(\mathcal{A}(\boldsymbol{w}_h), \varphi_h)$ is a weak form of the residuum of the Euler equations and supremum is taken over a special set of test functions V_i, see [Kli00]. Again if the value of $r(i)$ is higher then a given tolerance η then T_i is refined by the red–greed refinement technique.

The mesh adaptation for all three adaptive technique was applied several times till the quality of the solution was approximately the same. The quality of the solution is measured only as the sharpness of shock waves in "eye" norm. It is not of course sufficient for practical computation but here we present only a qualitative comparison of three different numerical methods, see Figures 1.32 – 1.34. The following table compare the number of triangles ($\#\mathcal{T}_h$) and CPU-time of all adaptive methods.

	$\#\mathcal{T}_h$			CPU-time			
level	AMA	SHOCK	RESID	AMA adapt	AMA comput	SHOCK	RESID
0	590	590	590	0.9	2.5	2.4	2.5
1	1 680	1 680	2 113	3.1	22.5	12.4	15.5
2	2 813	4 842	6 253	5.8	118.3	67.9	86.9
3	3 480	12 769	10 249	11.3	488.4	366.4	260.8
4	3 727	31 117	14 261	13.1	559.6	1 768.9	712.0
5			17 814				1 719.2
subtotal				34.2	1 191.3		
total	3 727	31 117	17 814		1 225.5	2 218.0	2 796.9

We observe that $\#\mathcal{T}_h$ for AMA is several times smaller than for other adaptive methods and as well as the CPU-time is better for AMA. The CPU-time of AMA is shown in the table separately for the mesh adaptation itself and for the numerical solution of the Euler equations. We see that the anisotropic mesh generation does not cause any essential increasing of CPU-time.

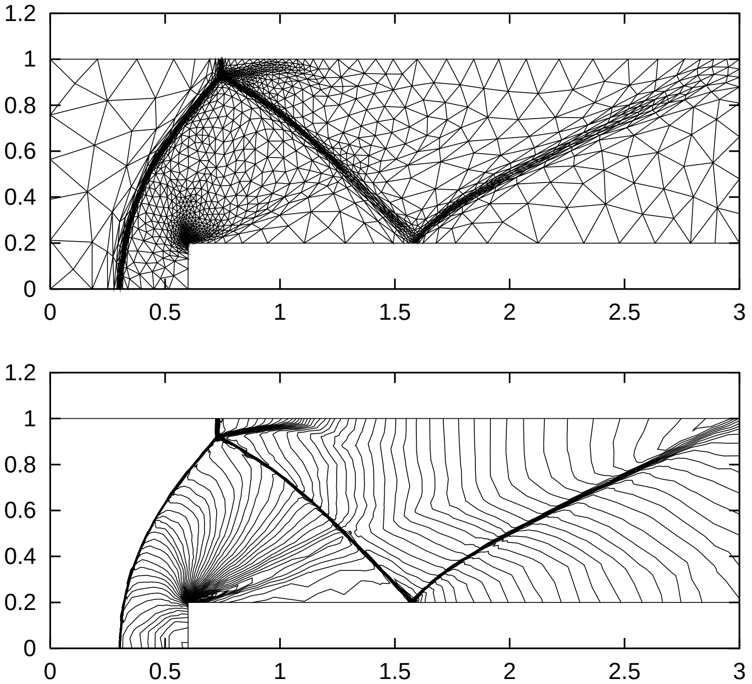

Figure 1.32: Triangulation and isolines of Mach number for AMA.

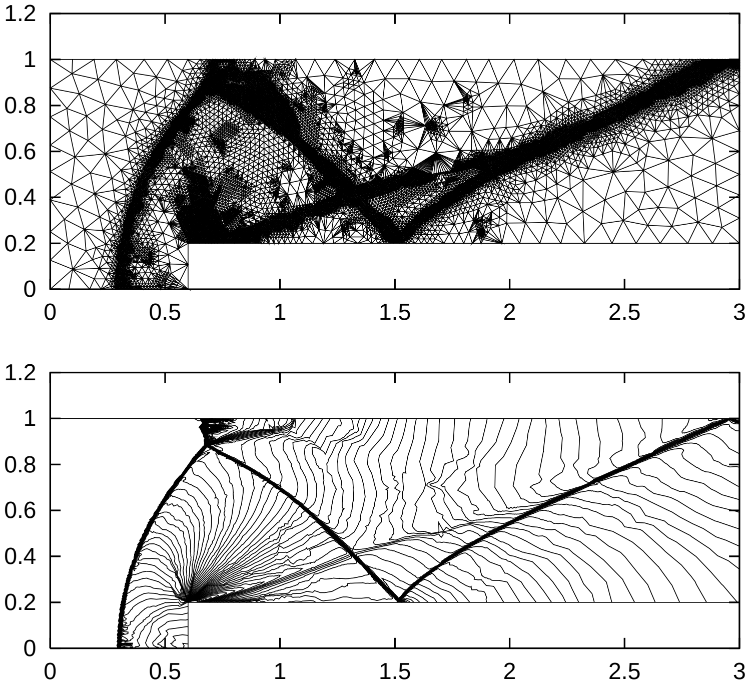

Figure 1.33: Triangulation and isolines of Mach number for shock indicator (6.10).

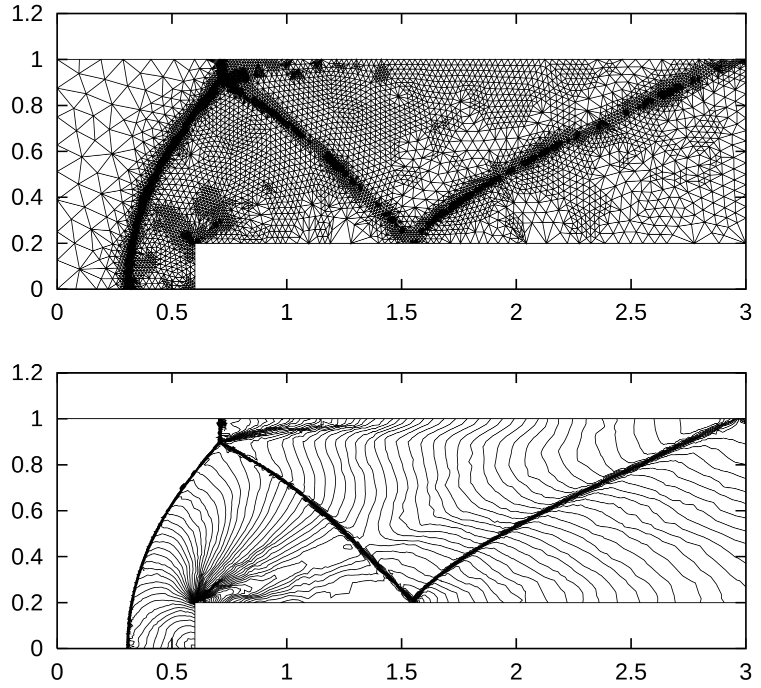

Figure 1.34: Triangulation and isolines of Mach number for residual indicator (6.11).

2D supersonic scramjet inlet

This example consists of an internal supersonic flow at Mach number $M_{\text{in}} = 3$ in a scramjet inlet. Figure 1.35 shows the geometry of the problem. The supersonic inlet and several obstacles with sharp angles give the solution with a few shock waves, see [DHM95]. Although the configuration is symmetric, the nonsymmetric mesh of the whole domain has been computed to observe if the solution remains symmetric or not.

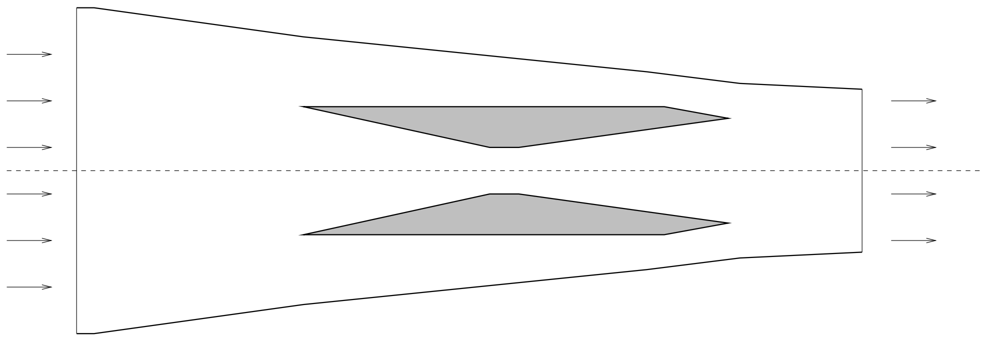

Figure 1.35: Geometry of the 2D supersonic scramjet inlet problem.

Using five automatic mesh adaptations we have obtained the final mesh. Figures 1.36 and 1.37 show the final triangular mesh (52285 elements) and the corresponding isolines of Mach number, respectively. We see rather complicated geometry of shock waves and it is interesting to notice that there are no oscillations in the solution and that the finite volume solver is robust and accurate even for such meshes with very anisotropic triangles. Moreover the numerical solution stays symmetric.

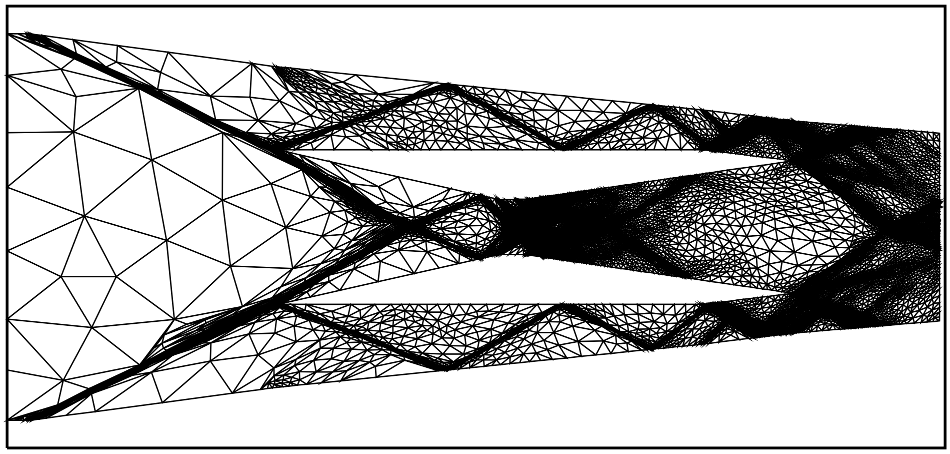

Figure 1.36: Final triangulation for a supersonic scramjet inlet.

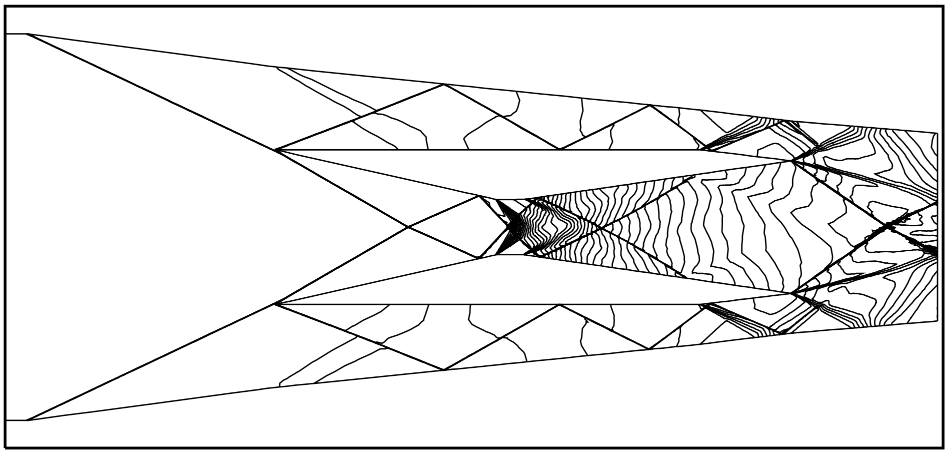

Figure 1.37: The corresponding isolines of Mach number for a supersonic scramjet inlet.

Transonic flow through 3D GAMM channel

The three-dimensional transonic inviscid flow through the channel (with 25 % spherical bump on the lower wall) of air with inlet Mach number $M_{\mathrm{in}} = 0.67$ was solved. The size of the channel is $x_1 \in [0,2]$, $x_2 \in [0,1.5]$ and $x_3 \in [0,1]$. The direction of the flow is parallel with x_1 axis and goes from left to right. Figure 1.38 shows the final tetrahedral mesh (with 8786 elements) achieved after five mesh adaptations and the corresponding isolines of Mach number.

Despite the small number of elements used in the computation, the anisotropic mesh adaptation method leads to satisfactory results.

Supersonic flow through 3D forward facing step

This example is a 3D extension of the plane case, see Figure 1.31. The size in x_3–direction is $[0, 0.25]$ and we consider the same initial and boundary conditions, i.e. $\rho_0 = 1.4$, $\boldsymbol{v} = (3, 0, 0)$, $p = 1$. In fact it is only "kvasi" 3D example, as the solution does not depend on x_3 variable but it is interesting observe meshes generated by AMA.

Figure 1.39 shows the final tetrahedral mesh (with 41709 elements) achieved after three mesh adaptations and the corresponding isolines of Mach number. There are elements having size in x_3–direction equal to the size of the computational domain (=0.25), which corresponds to the fact that the solution is independent of x_3.

6.4 AMA for viscous flow simulation

In previous section we present the application of AMA technique for the computation of inviscid flow (i.e. $\boldsymbol{R}_i = 0,\ i = 1, \ldots, d$ in (6.1)), where the density ρ was chosen as the significant component of the state vector $\boldsymbol{w}$. This choice seems to be suitable because the triangulation was adapted along discontinuities (shock waves), which they were numerically captured with high accuracy.

Next possibility is to use the local Mach number $M = |\boldsymbol{v}|/\sqrt{\gamma p/\rho}$ as the significant component. The choice of M gives for inviscid flow similar results as for the density, because in subdomains $\Omega^s \subset \Omega$, where the components of $\boldsymbol{w}$ are smooth, the flow is isentropic and the density ρ is the function of M. On the other hand both ρ and M are discontinuous on the shock waves $(\Omega \setminus \Omega^s)$. The choice M as the significant component is more suitable for viscous flow simulation [CDBG+96] because the velocity (and then Mach number) is essentially smaller in boundary layers and wakes than in the external flow and then the high gradient of Mach number will be detected by AMA. The Mach number flow field is viewed in Figure 1.40.

The numerical examples show that the simple use of the Mach number as a significant component do not give yet satisfactory results for high speed flow with a small viscosity, because boundary layers and wakes are not satisfactory captured. Therefore an additional requirement has to be added to the mesh adaptation algorithm.

In the following we describe the viscous modification of AMA for the case $d = 2$ for simplicity. The extension for $d = 3$ is straightforward and it is presented in Section 6.5.

Boundary layer

At first we focus on the capturing of boundary layers. We present their physical properties, namely the dependence of the boundary thickness on the Reynolds number. Then we include this property into AMA method.

Let us introduce a local coordinate system $\tilde{x}, \tilde{y}$ at the point lying on the impermeable wall Γ_W, where the condition $(\boldsymbol{v} = 0)$ is prescribed. The axis $\tilde{x}$ is the tangent to Γ_W and $\tilde{y}$ is perpendicular to Γ_W. It is known fact that the velocity does not change very much in the direction $\tilde{x}$ whereas the dependency of the velocity with respect to $\tilde{y}$ is very high. To capture precisely the boundary layer we have to generate a suitable triangulation along Γ_W, Figure 1.41. We denote by h_X and h_Y the size of the triangle in the direction $\tilde{x}$ and $\tilde{y}$, respectively,

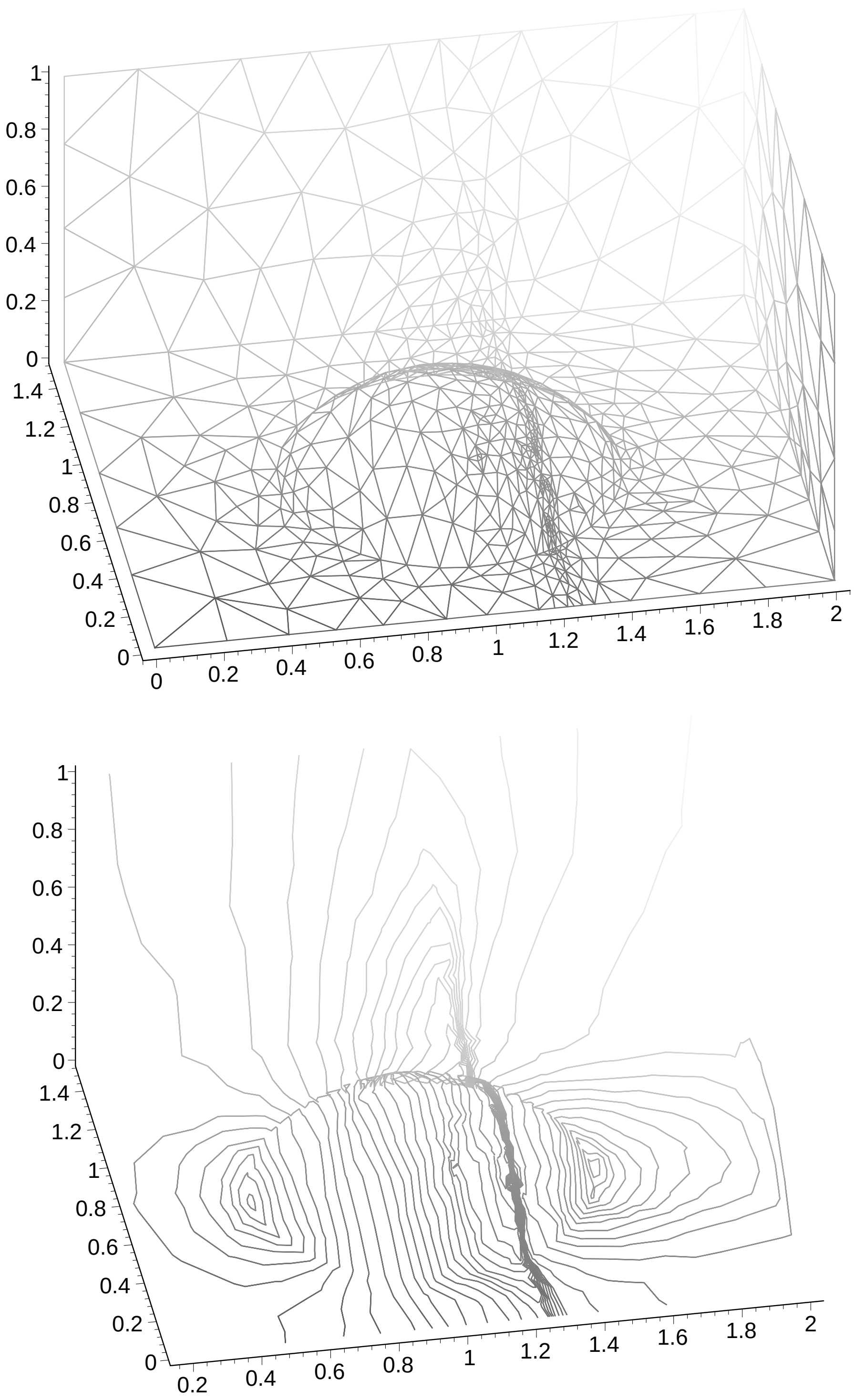

Figure 1.38: Final tetrahedrization (top) and the corresponding isolines of Mach number (bottom) for 3D GAMM channel.

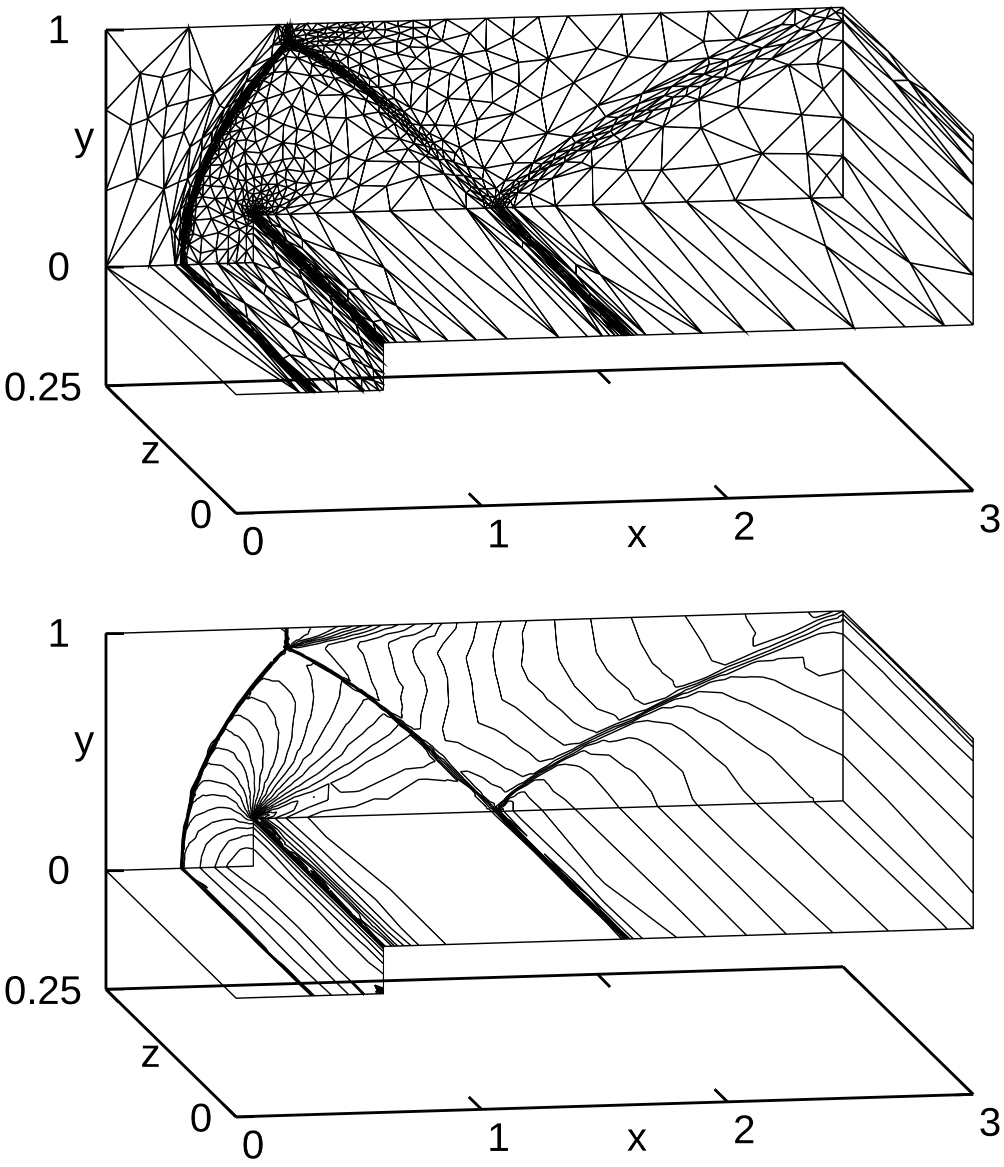

Figure 1.39: Final tetrahedrization (top) and the corresponding isolines of Mach number (bottom) for 3D forward facing step.

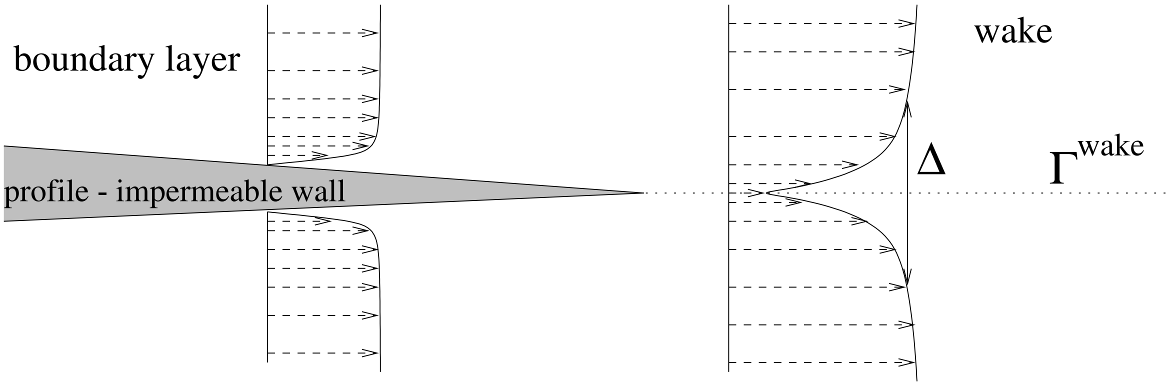

Figure 1.40: Example of Mach number flow field in a boundary layer and in a wake, the wake line Γ^{wake} and the width of the wake Δ.

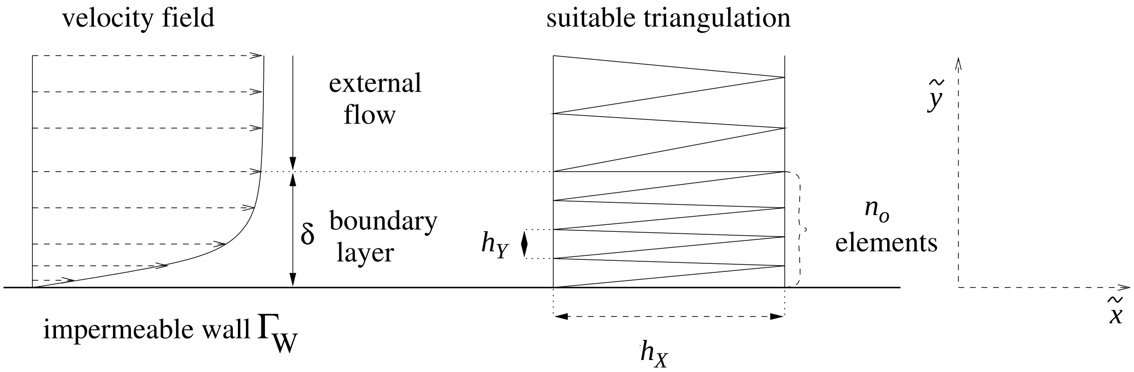

Figure 1.41: Illustration of the boundary layer with thickness δ and the sketch of a suitable triangulation with $n_0 = 4$.

see Figure 1.41. The suitable triangulation for satisfactory capturing of a boundary layer is sketched at the middle part of Figure 1.41. Triangles are thin and long and are alignment along Γ_W. The size h_Y has to be enough small to capture a boundary layer and it should successively increases with the distance from Γ_W. The size h_X has to be as high as possible so that the triangulation has as small as possible number of elements. The thickness of the boundary layer δ depends on the flow viscosity according to the relation

$$\delta \approx \frac{1}{\sqrt{Re}}, \tag{6.12}$$

where Re is the Reynolds number. Then to well simulate the boundary layer we put

$$h_Y(0) = \frac{n_0}{\sqrt{Re}}, \tag{6.13}$$

where $n_0 > 1$ is a suitable constant and its meaning is viewed in Figure 1.41. Numerical tests lead us to the choice $n_0 \in \langle 3, 10 \rangle$. For triangles laying at the distance r_y from Γ_W we put

$$h_Y(r_y) = \frac{n_0}{\sqrt{Re}} + c_y^B r_y, \tag{6.14}$$

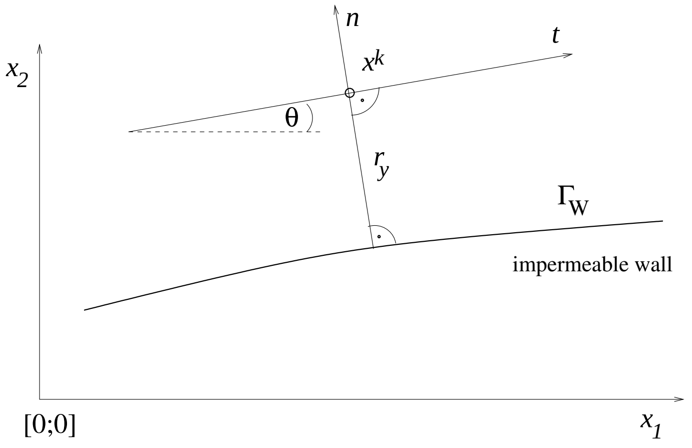

Figure 1.42: Construction of matrix $\boldsymbol{M}$ for boundary layer.

where c_y^B is a suitable constant.

On the other hand the size h_X of the triangle in the direction parallel with Γ_W is restricted only by *the minimal angle condition* required e.q. for FEM. Therefore we put

$$h_X = c_\alpha^B h_Y \tan \alpha_0, \tag{6.15}$$

where α_0 is the given minimal angle value and $0 < c_\alpha^B < 1$ is an "security" constant.

The relations (6.13) and (6.15) guarantee the sufficiently precise capturing of a boundary layer.

Boundary layer in the context of AMA

The previous consideration will be formulated in terms of AMA. Let $x^k \in I\!R^2$ be a point in the vicinity of fixed wall and $\boldsymbol{n}$ denotes the normal from x^k to this wall. We denote by θ the angle between the axis x_1 and the tangent $\boldsymbol{t}$ and by r_y the distance of x^k to Γ_W, see Figure 1.42.

According (6.15) and (6.14), it is suitable require that the length edge in the direction $\boldsymbol{n}$ and $\boldsymbol{t}$ are h_X and h_Y, respectively. Then using results from Appendix, for the point x^k, we define the matrix

$$\boldsymbol{M}^B(x_k) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{1}{h_X^2} & 0 \\ 0 & \frac{1}{h_Y^2} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{6.16}$$

where θ is the angle between direction parallel with the wall and the axis x (see Figure 1.42) and h_X and h_Y are given by (6.15) and (6.14), respectively. For the matrix $\boldsymbol{M}^B(x^k)$, there exists an ellipse ϵ_k^B whose lengths and orientation are given by (9.3) and in virtue of (9.11) the norms (corresponding to $\boldsymbol{M}^B(x^k)$) of all chords of ϵ_k^B going through x^k are equal to $\sqrt{3}$. Moreover, the length of the chord going through x^k parallel with and $\boldsymbol{n}$ is equal to h_X and h_Y, respectively, which corresponds with our imagination of suitable triangulation for capturing a boundary layer.

As it is mentioned in Section 2.4, it is suitable for the practical implementation to come to vertex associated matrices. We define $\boldsymbol{M}^B(P_i)$, $P_i \in \sigma_h$ in the same way as $\boldsymbol{M}^B(x^k)$. We

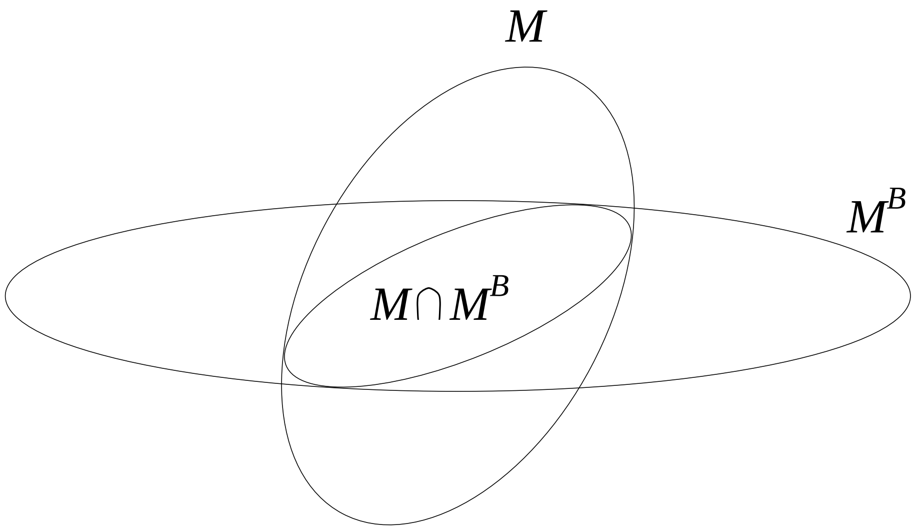

Figure 1.43: Intersection of two metrics.

have for each point $P_i \in \sigma_h$ two matrixes $\boldsymbol{M}(P_i)$ (given by (4.15) checking the fulfilling the necessary condition (2.13)) and $\boldsymbol{M}^B$ (given by (6.16) checking the relations (6.14) – (6.15)). To satisfy both of this condition we use the so-called *intersection of matrices*, see [DHM95].

Intersection of matrices

We say, that the matrix $\boldsymbol{M} \cap \boldsymbol{M}^B$ is the intersection of matrices $\boldsymbol{M}$ and $\boldsymbol{M}^B$, if the corresponding ellipse is the subset of ellipses corresponding to $\boldsymbol{M}$ and $\boldsymbol{M}^B$ and has the maximal possible area, see Figure 1.43. We apply the transformation of coordinates to the system in which both $\boldsymbol{M}$ and $\boldsymbol{M}^B$ are diagonal, i.e. let $\boldsymbol{P}$ be a matrix such that

$$\tilde{\boldsymbol{M}} = \boldsymbol{P}^T \boldsymbol{M} \boldsymbol{P} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \qquad \tilde{\boldsymbol{M}}^B = \boldsymbol{P}^T \boldsymbol{M}^B \boldsymbol{P} = \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix}, \tag{6.17}$$

where $\boldsymbol{P}^T$ denotes the matrix transpose to $\boldsymbol{P}$. Then putting

$$(\boldsymbol{M}, \boldsymbol{M}^B) = (\boldsymbol{P}^{-1})^T \begin{pmatrix} \max(\lambda_1, \mu_1) & 0 \\ 0 & \max(\lambda_2, \mu_2) \end{pmatrix} \boldsymbol{P}^{-1} \tag{6.18}$$

we obtain the desired matrix. Here $\boldsymbol{P}^{-1}$ means the inverse matrix to $\boldsymbol{P}$. The matrix $\boldsymbol{P}$ is not defined uniquely and it can be set using the following relations. Let $\boldsymbol{M}$ and $\boldsymbol{M}^B$ are written in the form

$$\boldsymbol{M} = \begin{pmatrix} a_1 & b_1 \\ b_1 & c_1 \end{pmatrix}, \qquad \boldsymbol{M}^B = \begin{pmatrix} a_2 & b_2 \\ b_2 & c_2 \end{pmatrix}, \tag{6.19}$$

where $a_i > 0, c_i > 0, a_i c_i > b_i^2,\ i = 1, 2$. Then the matrix $\boldsymbol{P}$ is given by

$$\boldsymbol{P} = \begin{pmatrix} p_1 & p_2 \\ p_3 & p_4 \end{pmatrix} \tag{6.20}$$

with components given for example by the following relations

$$\begin{array}{ll}
\textbf{if} & \exists C > 0 \text{ such that } a_1 = Ca_2,\ b_1 = Cb_2,\ c_1 = Cc_2 \\
\textbf{then} & \\
& a_1 \neq -b_1 \quad \Rightarrow \quad \begin{array}{l} p_1 = -\frac{b_1+c_1}{a_1+b_1} \\ p_2 = 1 \\ p_3 = 1 \\ p_4 = 1 \end{array} \\
& \\
& a_1 = -b_1 \quad \Rightarrow \quad \begin{array}{l} p_1 = 1 \\ p_2 = 1 \\ p_3 = 1 \\ p_4 = -\frac{a_1+b_1}{b_1+c_1} \end{array} \\
\textbf{else} & \\
& \begin{array}{l} d = (a_1c_2 - a_2c_1)^2 - 4(a_1b_2 - b_1a_2)(b_1c_2 - c_1b_2) \\ p_2 = \frac{a_2c_1 - a_1c_2 - \sqrt{d}}{2(a_1b_2 - b_1a_2)} \\ p_1 = -\frac{b_2p_2+c_2}{p_2a_2+b_2} \\ p_3 = 1 \\ p_4 = 1 \end{array}
\end{array} \tag{6.21}$$

It can be proved that if does not exist C such that $a_1 = Ca_2,\ b_1 = Cb_2,\ c_1 = Cc_2$ then d from (6.21) is always nonnegative.

The application of the viscous correction is obvious. We use the AMA technique described in Section 4.3 but the matrix $\boldsymbol{M}(P_i)$ is replaced by the matrix $\boldsymbol{M}(P_i) \cap \boldsymbol{M}^B(P_i)$ for nodes in vicinity of Γ_W.

Wake

A viscous flow around a profile gives rise to a wake behind the trailing edge of the profile. Its flow field is viewed in Figure 1.40. In order to capture a wake accurately, it is necessary to adapt the triangulation behind the profile. We use a similar approach for nodes in vicinity of a wake as for them in vicinity of boundary layers. We define *the wake line* Γ^{wake} starting at the trailing edge and parallel with the flow field behind the profile, see Figure 1.40. Its position is known a priori (e.q. for symmetric problems) or it is set a posteriori from the results computed on the previous level of adaptation. Similarly as for a boundary layer we introduce a local coordinate system $\tilde{x}, \tilde{y}$ at the point lying on Γ^{wake}. The axis $\tilde{x}$ and $\tilde{y}$ are parallel and perpendicular with Γ^{wake}, respectively. Instead of the relations (6.15) and (6.14) we define the following relations

$$h_Y(r_y) = \frac{n_0}{\sqrt{Re}} + c_y^W \max(0, r_y - \Delta/2), \tag{6.22}$$

$$h_X = c_\alpha^W h_Y \tan \alpha_0, \tag{6.23}$$

where Δ means the width of the wake (Figure 1.40) and c_y^W and c_α^W are constants similar as c_y^B and c_α^B, respectively. The magnitude of Δ depends on the considered problem.

Then we define the matrix $\boldsymbol{M}^W$ similarly as in (6.16) and use the intersection of matrix $\boldsymbol{M} \cap \boldsymbol{M}^W$ from Section 6.4.

6.5 AMA for viscous flow – case $d = 3$

As the extension the two dimensional version is straightforward we present here only the relations which have different form. The relation (6.14), (6.15), (6.22), (6.23) have the same form. Further the relations (6.16) is replaced by

$$\boldsymbol{M}^B(x_k) = \boldsymbol{R}^{\mathrm{T}} \begin{pmatrix} \frac{1}{h_X^2} & 0 & 0 \\ 0 & \frac{1}{h_X^2} & 0 \\ 0 & 0 & \frac{1}{h_Y^2} \end{pmatrix} \boldsymbol{R}, \tag{6.24}$$

where $\boldsymbol{R}$ is the rotation matrix

$$\boldsymbol{R} = \begin{pmatrix} \cos\varphi \sin\theta\, \cos 2\varphi + \sin 2\varphi\, \sin\varphi & \sin 2\varphi\, \cos\varphi\, \sin\theta - \sin\varphi\, \cos 2\varphi & \cos\theta\, \cos\varphi \\ \sin\varphi\, \sin\theta\, \cos 2\varphi - \sin 2\varphi\, \cos\varphi & \sin 2\varphi\, \sin\varphi\, \sin\theta - \cos\varphi\, \cos 2\varphi & \cos\theta\, \sin\varphi \\ -\cos\theta\, \cos 2\varphi & -\cos\theta\, \sin 2\varphi & \sin\theta \end{pmatrix}, \tag{6.25}$$

where φ and θ are the spherical coordinates of a chosen the direction tangent to the solid wall, see Appendix.

The intersection of matrices is analogue to (6.17). We apply the transformation of coordinates to the system in which both $\boldsymbol{M}$ and $\boldsymbol{M}^B$ are diagonal.

6.6 Mesh gradation control

The application of AMA technique presented in Sections 4.3, 6.4 and 6.4 can lead to triangulations with high gradation. It means that the sizes of neighbour triangles very differ, see Figure 1.44 left. Too high mesh gradation gives rise some troubles for different numerical solvers (FEM, FVM). The uncontrolled mesh gradation is caused by the fact, that matrixes $\boldsymbol{M}_i$ and $\boldsymbol{M}_j$ very differ for two nearby nodes P_i and P_j. Therefore our aim is to control the mesh gradation of the triangulation generated by AMA. It is illustrated in Figure 1.44, where is shown the passage from the coarse part of the triangulation to the fine one (from left to right). The left and right figures are the triangulation generated without and with mesh gradation control, respectively. The use of mesh gradation control technique gives triangulations with better properties for numerical solution using FEM, but naturally the number of elements is higher.

We have applied two approaches from [BHF97], the H-variation and H-shock. We present only the final relations, for more detail see [BHF97]. Both H-variation and H-shock lead to similar grids.

H-variation

Let P_1P_2 be an edge of the triangulation $\mathcal{T}_h$ and $\boldsymbol{M}_{P_1}$ and $\boldsymbol{M}_{P_2}$ the matrices corresponding to nodes P_1 and P_2, respectively. Then we put

$$\begin{aligned} \boldsymbol{M}_{P_1} &:= \boldsymbol{M}_{P_1} \cap \boldsymbol{M}_{P_2}(1 + \alpha \|P_1P_2\|_{\boldsymbol{M}_{P_1}})^{-2}, \quad \alpha > 0, \\ \boldsymbol{M}_{P_2} &:= \boldsymbol{M}_{P_2} \cap \boldsymbol{M}_{P_1}(1 + \alpha \|P_1P_2\|_{\boldsymbol{M}_{P_2}})^{-2}, \end{aligned} \tag{6.26}$$

where $\|P_1P_2\|_{\boldsymbol{M}_{P_1}}$ and $\|P_1P_2\|_{\boldsymbol{M}_{P_2}}$ are the norms of the edge P_1P_2 corresponding to $\boldsymbol{M}_{P_1}$ and $\boldsymbol{M}_{P_2}$, respectively. The operator $\cap$ means the intersection of matrices, Section 6.4. We use the value $\alpha = 2$.

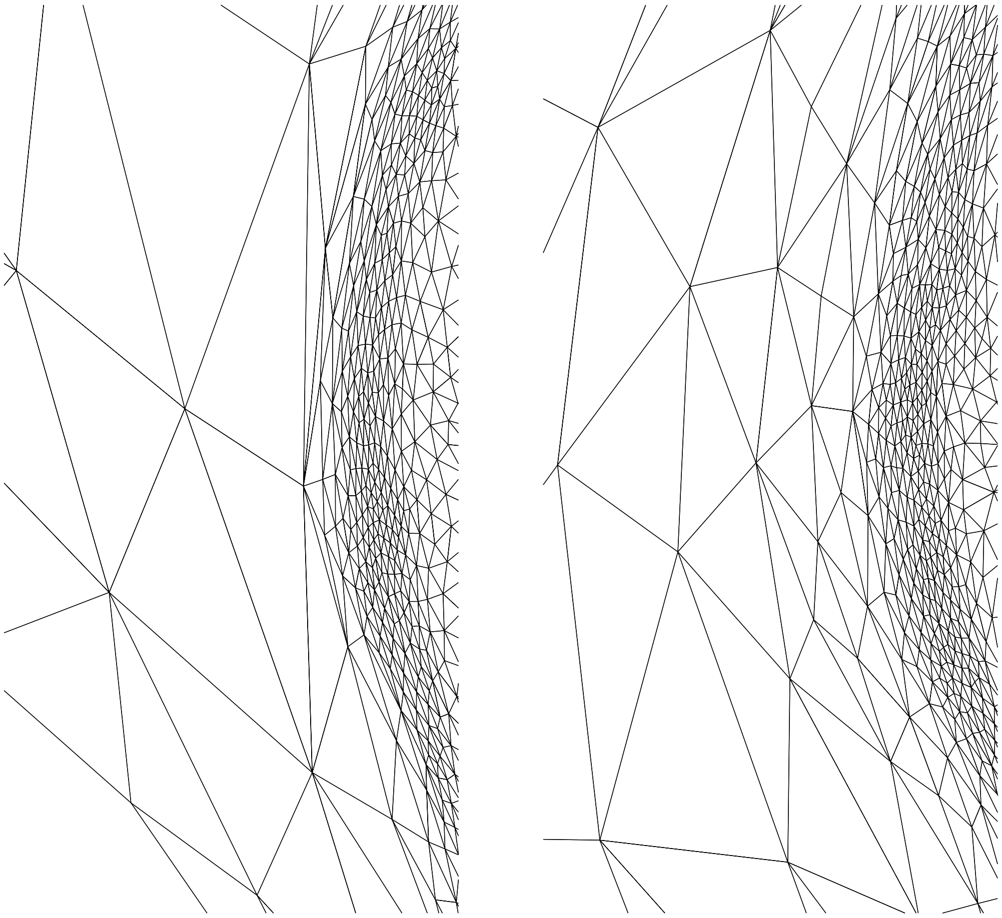

Figure 1.44: Mesh gradation control, triangulation without (left) and with (right) mesh gradation control.

H-shock

Let P_1P_2 be an edge of the triangulation. Let h_{P_1} and h_{P_2} be the norm of the unit vector in the direction P_1P_2 corresponding to matrixes $\boldsymbol{M}_{P_1}$ and $\boldsymbol{M}_{P_2}$, respectively. Let $l_{P_1P_2}$ be the norm of P_1P_2 corresponding to matrix $\bar{\boldsymbol{M}}_{P_1P_2} = (\boldsymbol{M}_{P_1} + \boldsymbol{M}_{P_2})/2$. Then H-shock associated to the edge P_1P_2 is the value

$$c_{P_1P_2} = \left(\frac{h_{P_2}}{h_{P_1}}\right)^{1/l_{P_1P_2}}. \tag{6.27}$$

If $c_{P_1P_2} \geq \beta$ then we replace $\boldsymbol{M}_{P_2}$ by $\eta^{-2}\boldsymbol{M}_{P_2}$, where $\eta = \left(\frac{\beta}{c_{P_1P_2}}\right)^{l_{P_1P_2}}$. We use the value $\beta = 1.5$.

6.7 Numerical examples of viscous flow simulation

The described method (solver and mesh adaptation) was applied to the numerical simulation of flow past a turbine cascade shown in Figure 1.45. The goal was to obtain the steady state solution with the aid of the time stabilization for $\boldsymbol{T} \to \infty$. The computational results are compared with a wind tunnel experiment (by courtesy of the Institute of Thermodynamics of the Czech Academy of Sciences in Prague, see [ŠŠ90]). The experiment and computations were performed for the following data: angle of attack $= 19^\circ\, 18'$, inlet Mach number $= 0.32$, outlet Mach number $= 1.18$, $\gamma = 1.4$, Reynolds number $Re = 1.5 \cdot 10^6$, Prandtl number $Pr = 0.72$.

Figure 1.46 represents the wind tunnel interferogram showing density isolines. We compare the results achieved by the combined barycentric FV - nonconforming FE method (Section 6.2, we set $\mathcal{F}_h := \mathcal{T}_h$) using triangulations $\mathcal{T}_h$ generated by AMA (Section 4.3) without and with viscous correction (Section 6.4). As the starting triangulation and numerical solution we use the results of inviscid flow simulation [Dol98b]. In Figure 1.47 and Figure 1.48 the final triangular mesh and the corresponding isolines of Mach number with zooms of trailing edge obtained with the aid of anisotropic mesh refinement with and without viscous correction are plotted, respectively. The viscous correction yields to the mesh refinement along profile and then the boundary layers are captured with high accuracy and the shock waves are not smeared as for the example without viscous mesh correction. Figure 1.49 shows the pressure distributions along the profile compared with the measurement. We see that a good agreement of computational results with experiment was achieved, if the viscous mesh correction is applied.

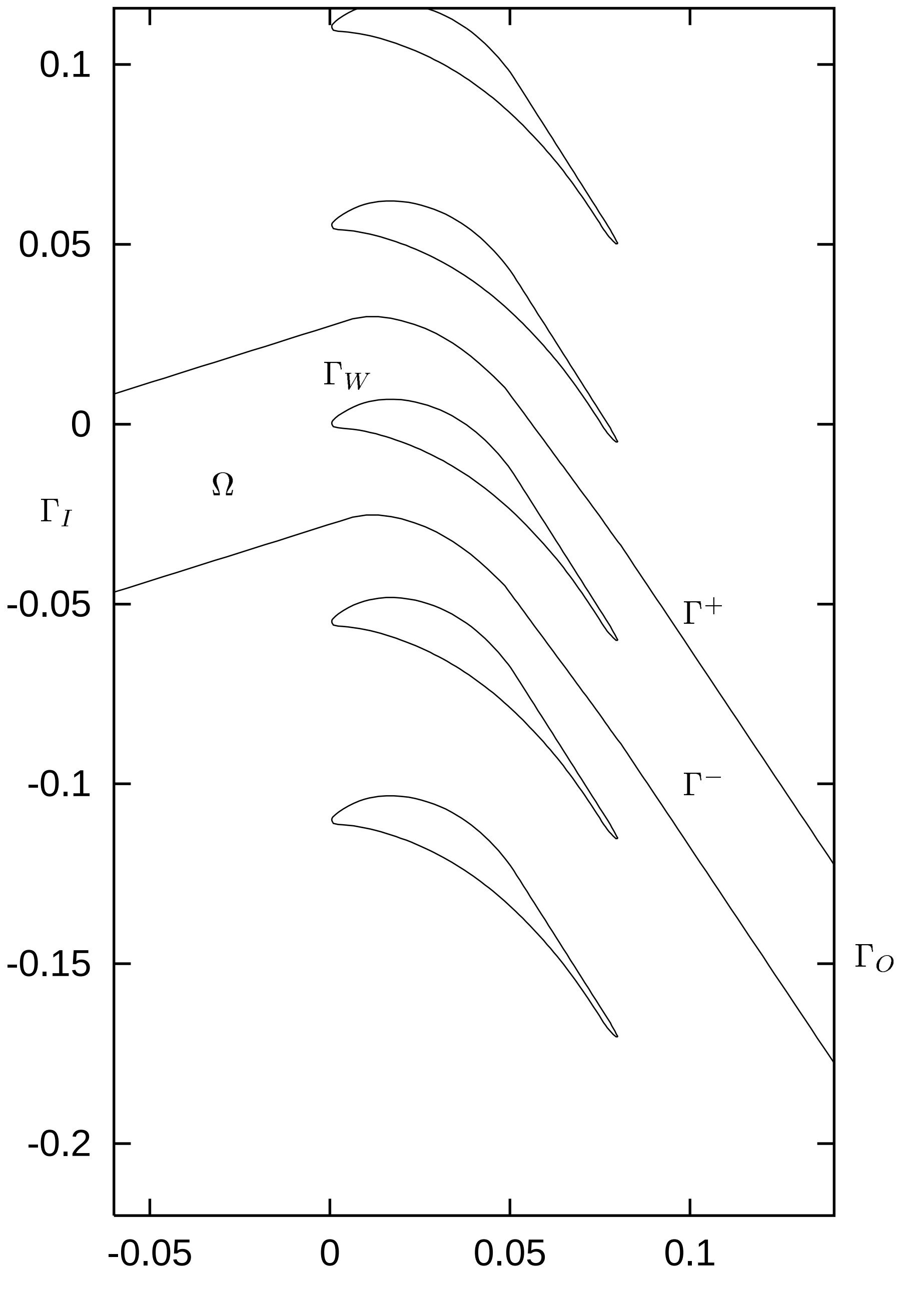

Figure 1.45: Cascade of profiles with the computational domain Ω and the boundary parts Γ_I (inlet), Γ_O (outlet), Γ_W (impearmeable walls) and the artificial periodical cuts Γ^+ and Γ^-, where the periodic boundary conditions are prescribed.

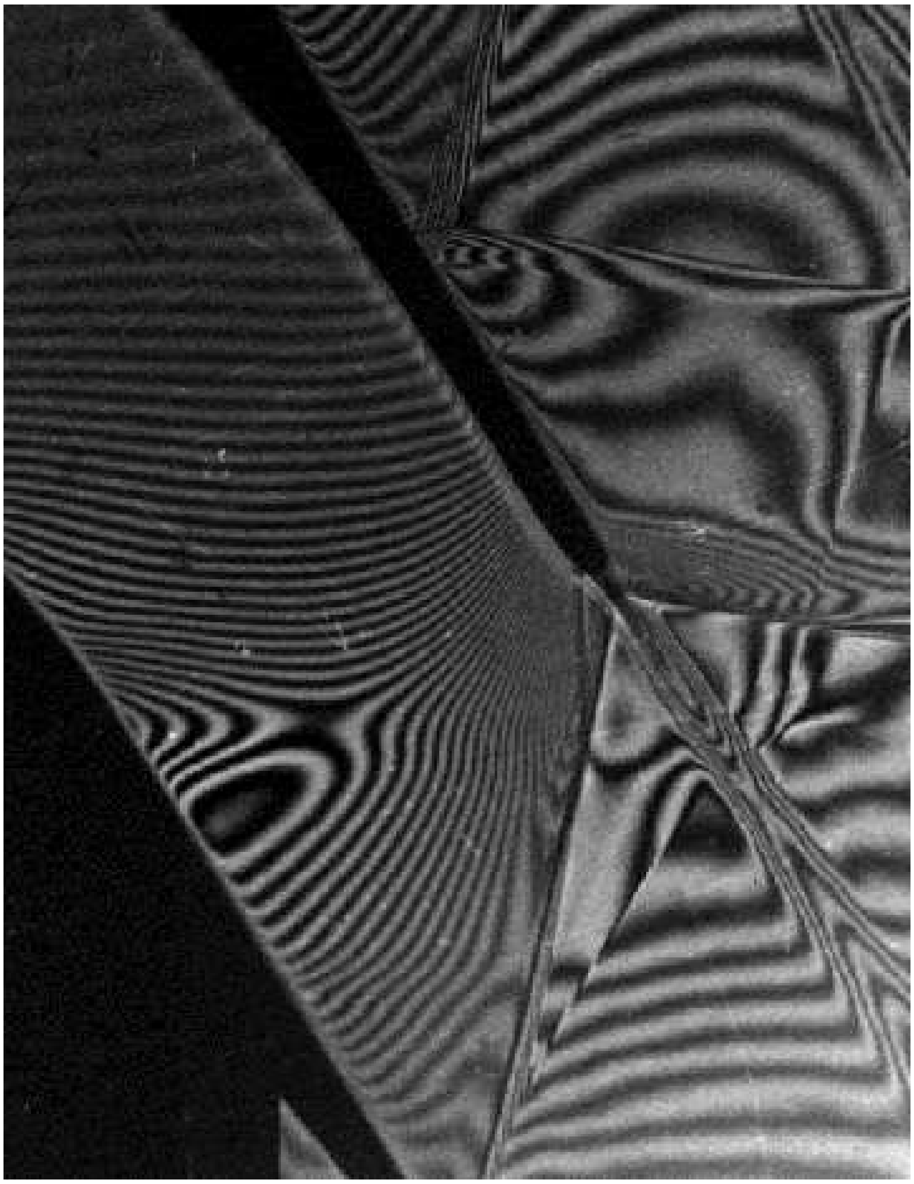

Figure 1.46: The wind tunnel interferogram showing density isolines (Courtesy of the Institute of Thermodynamics, Czech Academy of Science, Prague).

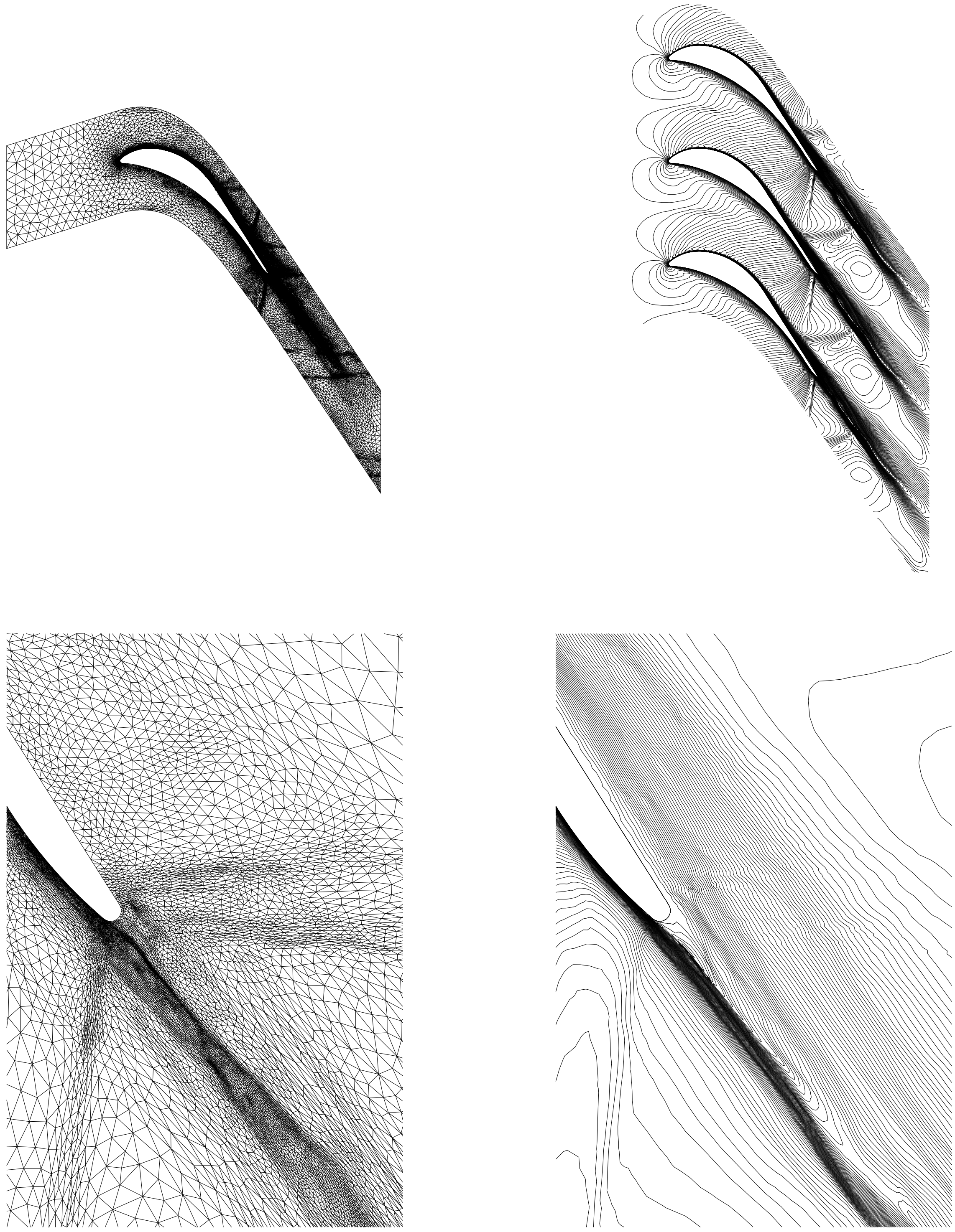

Figure 1.47: Triangulation (64 483 elements) and the corresponding isolines of Mach number with zooms achieved by AMA without viscous correction.

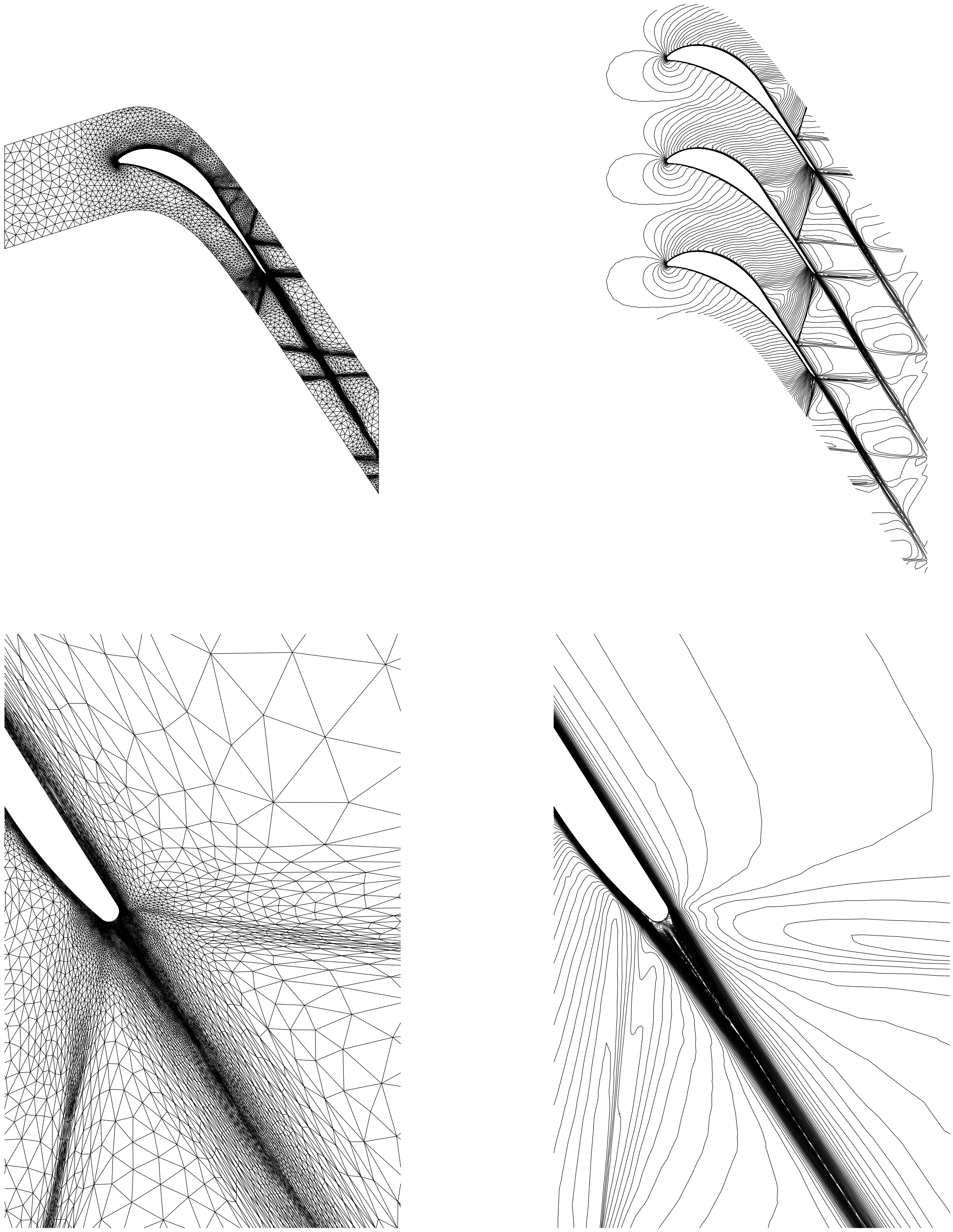

Figure 1.48: Triangulation (38 086 elements) and the corresponding isolines of Mach number with zooms achieved by AMA with viscous correction.

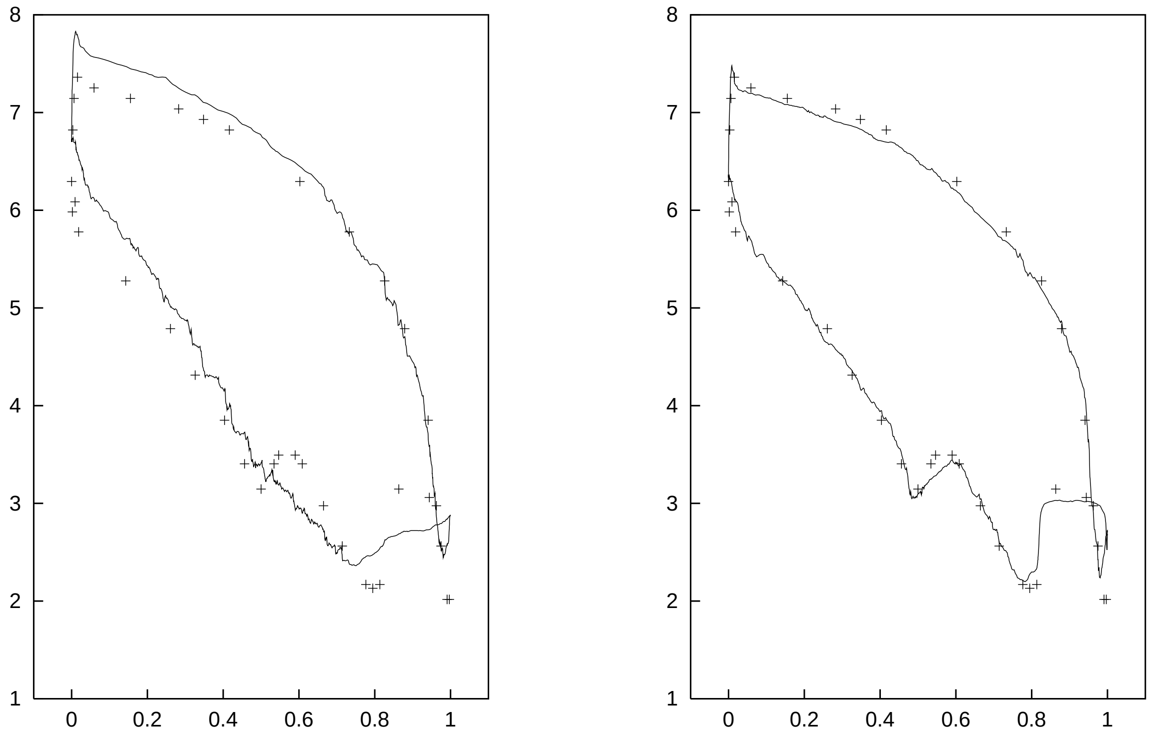

Figure 1.49: Distribution of the pressure (full line) in comparison with measurement (separated nodes) without (left) and with (right) AMA viscous correction.

7 Application for heat conduction problems

Although the AMA technique was originally developed for computational fluid dynamics problems, its generality allow us to use it for very different practical problems. We present the results presented in [BDM] where a computer simulation of the thermoregulation of premature infants was presented.

The motivation of this interest is the following. Suddenly occuring lack of oxygen is known to be an important cause of injury to the developing brain of premature infants [MWJG95]. The neuronal loss suffered in such a critical situation evolves over several days [GW92]. One of the factors that influence the degree and distribution of neuronal loss is the cerebral temperature. Clinical studies have shown that lowering the cerebral temperature can prevent much damage [BDGG89]. In order to investigate if the temperature within the brain of a premature infant in an incubator can be lowered by manipulating the environmental parameters, Fischer et al. [FLM00] employed a finite volume technique identical with the numerical method described within this paper but without an adaptive routine to solve a variation of the bio-heat equation which reads as

$$\partial_t \theta(x,t) = \operatorname{div}(\lambda(x)\nabla\theta(x,t)) + q(x,t), \qquad x \in \Omega,\ t \in (0,\boldsymbol{T}) \tag{7.1}$$

equipped by the boundary and initial conditions

$$\begin{aligned} \theta(x,t) &= \theta_{\mathrm{D}}(x), \quad x \in \partial\Omega,\ t \in (0,\boldsymbol{T}), \\ \theta(x,0) &= \theta_0(x), \quad x \in \overline{\Omega}, \end{aligned} \tag{7.2}$$

and which features strong source terms given by

$$q(x,t) = q_M(x) + q_B(x,t),$$

where θ is temperature and λ is heat conduction coefficient. Thereby, $q_M(x)$ is the metabolical heat production and $q_B(x,t)$ is a model of heat transfer within the human body due to blood flow. For the initial condition we choose $\theta_0(x) = 310.15\,K$. Furthermore, the boundary temperature distribution is given in the form

$$\theta_{\mathrm{D}}(x) = \begin{cases} 299.15\,K, & \text{if } x \in \partial\Omega_H, \\ 309.15\,K, & \text{otherwise}, \end{cases} \tag{7.3}$$

where $\partial\Omega_H$ represents the boundary of the head region.

This model is based on the assumption of a steady state situation. Concerning the grid used within the work of [FLM00], it should be noted that structured subgrids were used in the boundary region in order to capture steep heat gradients. For more details and for a theoretical discussion of the method see [FLM00].

In the following, we discuss the simulation of heat transfer with respect to an infant of 1 kg by the use of the described adaptive scheme. It is evident to start from a coarse primary grid and to proceed by computing the corresponding steady state solution. If a steady state solution is reached, the grid is adapted by using the described AMA method and again a steady state calculation is started. We now investigate different levels of refinement computed and used by the method together with the corresponding numerical solutions. Within the Figures 1.50 and 1.51 the evolution the grid undergoes is displayed, and it can be seen that a suitable solution is reached three refinement steps after starting from a quite coarse triangulation. In the following table the numbers of triangles used within the different steps of refinement is given together with the corresponding number of the mesh used in [FLM00] which is denoted as the reference mesh.

Mesh	Number of triangles
First Mesh	1822
Second Mesh	3673
Third Mesh	6731
Fourth Mesh	8956
Reference Mesh	37351

8 Conclusion

We present an efficient adaptive technique which can be used for the numerical solution of different problems of the mathematical engineering and computational physics. Although the AMA method is not based on a posteriori error estimation, the presented numerical examples confirm its high efficiency (i.e., the possibility to compute a sufficiently accurate solution without a high increasment of degree of freedom) and its generality (CFD, thermoregulation).

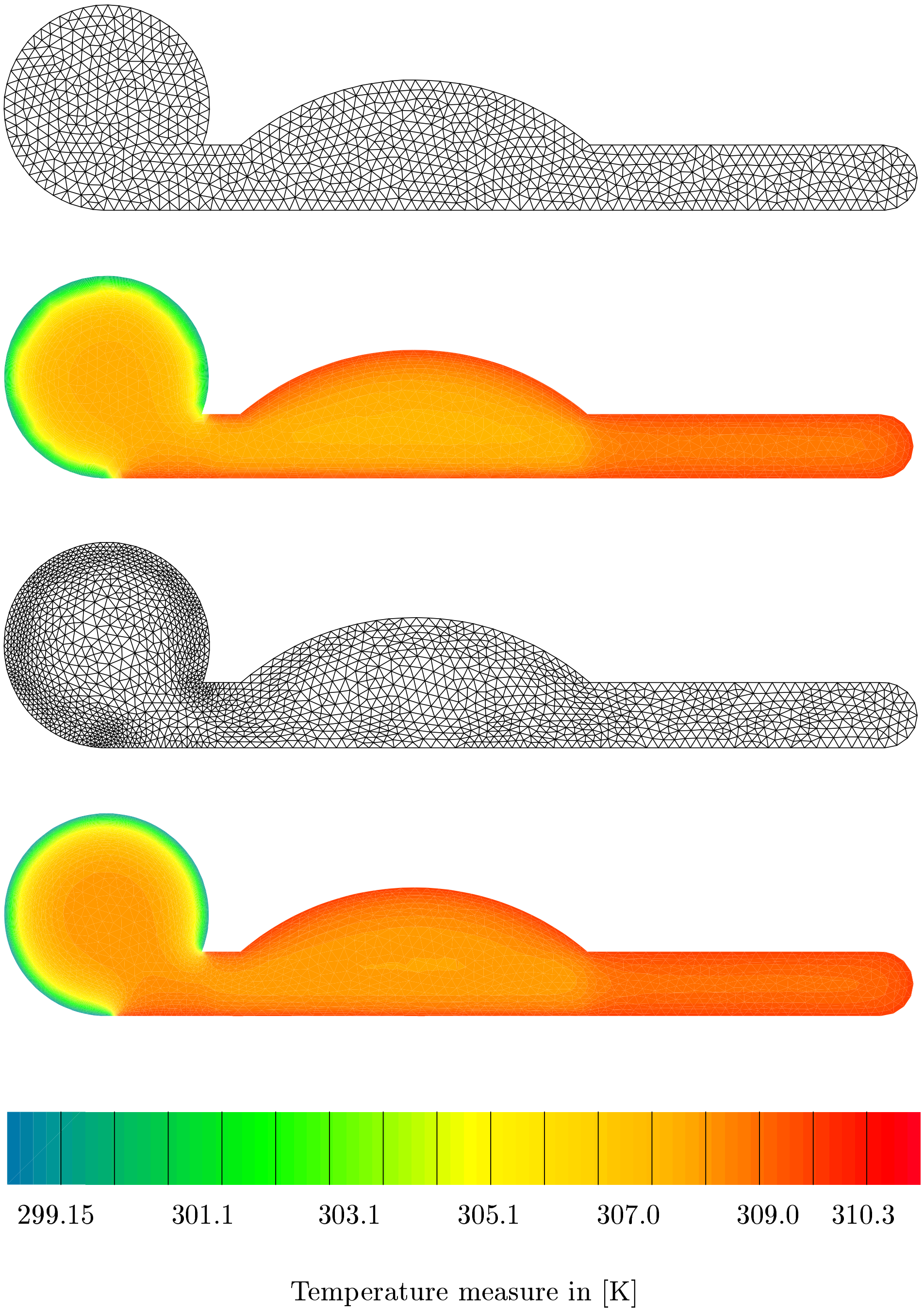

Figure 1.50: Sequence of triangulations and corresponding temperature distributions (part I).

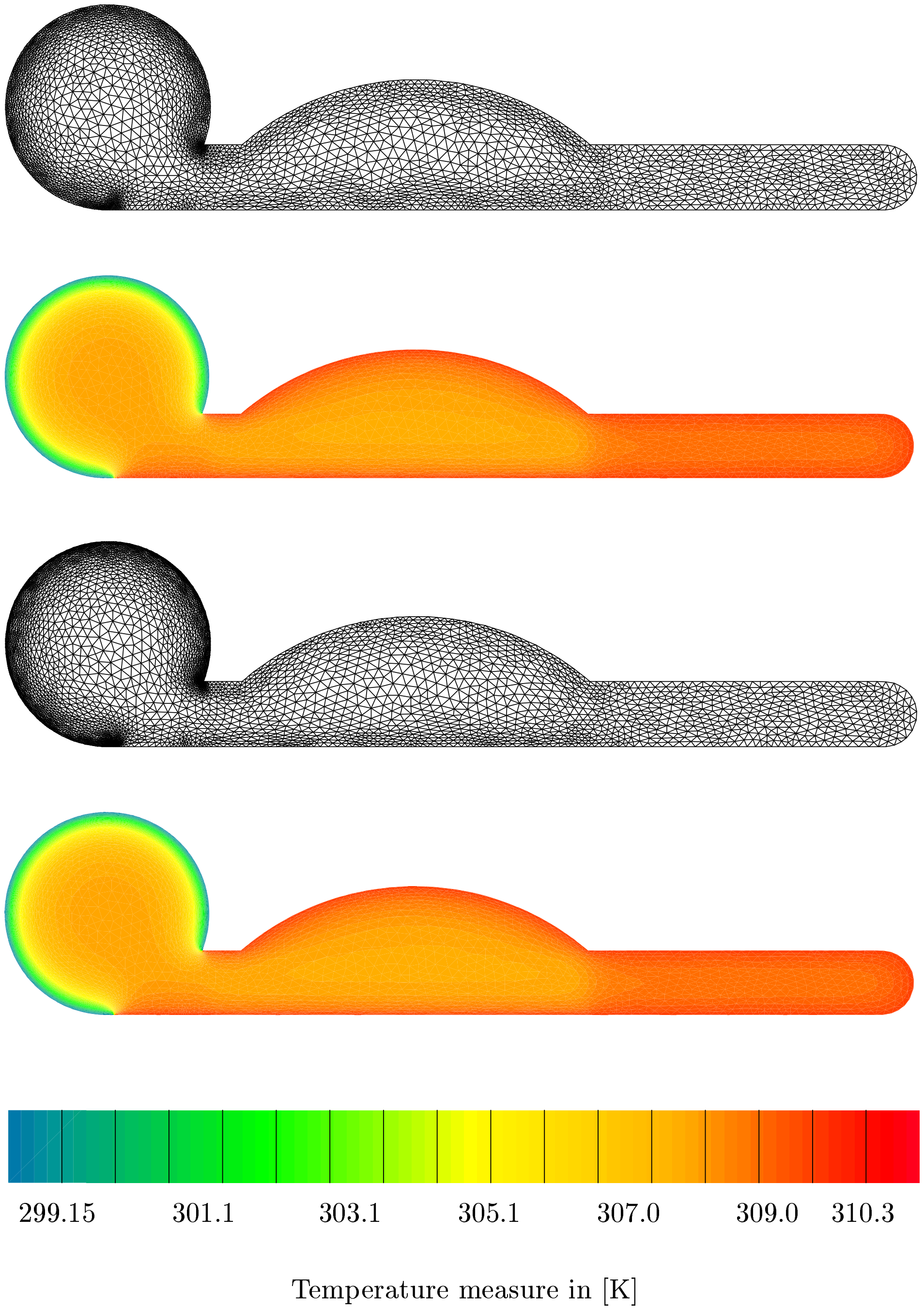

Figure 1.51: Sequence of triangulations and corresponding temperature distributions (part II).

9 Appendix

9.1 2D case

Let M be a symmetric and positive definite 2×2 matrix

$$M = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \qquad ac - b^2 > 0, \tag{9.1}$$

and let $\boldsymbol{v} = (v_1, v_2)$ be a vector in $I\!R^2$. We define *the norm* of the vector $\boldsymbol{v}$ *corresponding* to the matrix M as

$$\|\boldsymbol{v}\|_M \equiv \left(\boldsymbol{v}^{\mathrm{T}} M \boldsymbol{v}\right)^{\frac{1}{2}} = \left[\begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}\right]^{\frac{1}{2}} = \left(av_1^2 + 2bv_1v_2 + cv_2^2\right)^{\frac{1}{2}}. \tag{9.2}$$

If $M = I$ (=identical matrix) then $\|\boldsymbol{v}\|_M = |\boldsymbol{v}|$, where $|\boldsymbol{v}| = (v_1^2 + v_2^2)^{1/2}$ is the Euclidean norm of $\boldsymbol{v}$.

We decompose M in the following way:

$$M \equiv \begin{pmatrix} a & b \\ b & c \end{pmatrix} = R^{\mathrm{T}} \Lambda R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{9.3}$$

where λ_1, λ_2 are the eigenvalues of M and $\theta \in [0, 2\pi)$.

Let Ox_1x_2 be a Cartesian coordinate system. The set of all $x = (x_1, x_2) \in I\!R^2$ satisfying the inequality

$$x^{\mathrm{T}} M x = ax_1^2 + 2bx_1x_2 + cx_2^2 \le 1 \tag{9.4}$$

forms an ellipse ϵ_M with the centre O and with the axes lengths $r_1 = 1/\sqrt{\lambda_1}, r_2 = 1/\sqrt{\lambda_2}$. The angle between the axis x and the major axis of ϵ_M is θ, see Figure 1.52. Let P_0 be the centre of ϵ_M, then for any point P lying on the boundary of ϵ_M the norm of the vector $P_0P = (P - P_0)$ corresponding to M is constant and $\|P_0P\|_M = 1$.

We define *the optimal triangle* T_M *corresponding to* M such that $T_M \subset \epsilon_M$ and $|T_M| \ge |T|$ for any triangle $T \subset \epsilon_M$. (Here $|T|$ means the area of the triangle T). If $M = I$ (= unit matrix) then ϵ_M is an unit circle and T_M is an equilateral triangle with the lengths of edges equal to $\sqrt{3}$.

We express analytically the coordinates of the optimal triangle T_M. Let us define a new Cartesian coordinate system $O\tilde{x}_1\tilde{x}_2$ with the centre at the centre of ϵ_M and with the axes parallel with the axes of the ellipse (Figure 1.53).

The problem of finding a triangle contained in ϵ_M with the maximal area has 4 solutions, see Figure 1.54. These four triangles have the following vertices:

$$\tilde{V}_1^{(1)} = \left[\frac{1}{\sqrt{\lambda_1}}, 0\right], \quad \tilde{V}_2^{(1)} = \left[-\frac{1}{2}\frac{1}{\sqrt{\lambda_1}}, \frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_2}}\right], \quad \tilde{V}_3^{(1)} = \left[-\frac{1}{2}\frac{1}{\sqrt{\lambda_1}}, -\frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_2}}\right], \tag{9.5}$$

$$\tilde{V}_1^{(2)} = \left[-\frac{1}{\sqrt{\lambda_1}}, 0\right], \quad \tilde{V}_2^{(2)} = \left[\frac{1}{2}\frac{1}{\sqrt{\lambda_1}}, -\frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_2}}\right], \quad \tilde{V}_3^{(2)} = \left[\frac{1}{2}\frac{1}{\sqrt{\lambda_1}}, \frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_2}}\right], \tag{9.6}$$

$$\tilde{V}_1^{(3)} = \left[\frac{1}{\sqrt{\lambda_2}}, 0\right], \quad \tilde{V}_2^{(3)} = \left[-\frac{1}{2}\frac{1}{\sqrt{\lambda_2}}, \frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_1}}\right], \quad \tilde{V}_3^{(3)} = \left[-\frac{1}{2}\frac{1}{\sqrt{\lambda_2}}, -\frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_1}}\right], \tag{9.7}$$

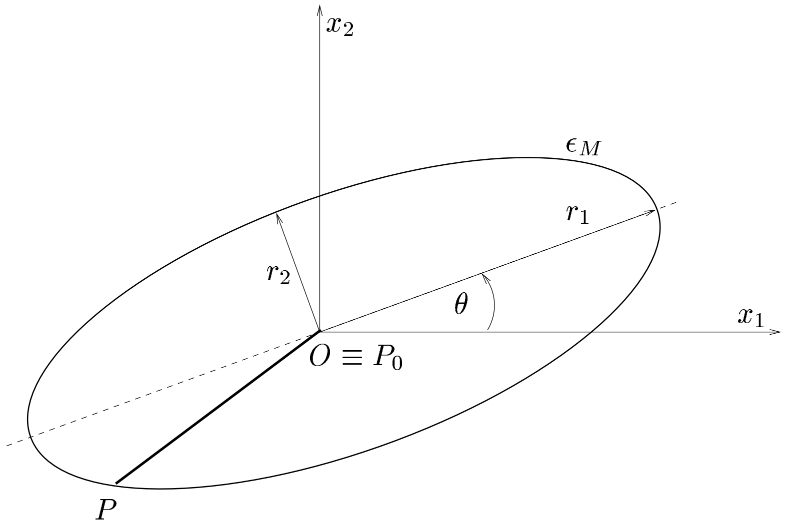

Figure 1.52: An ellipse ϵ_M corresponding to M with semiaxes length r_1, r_2, P_0 centre of ϵ_M, P any point of ϵ_M.

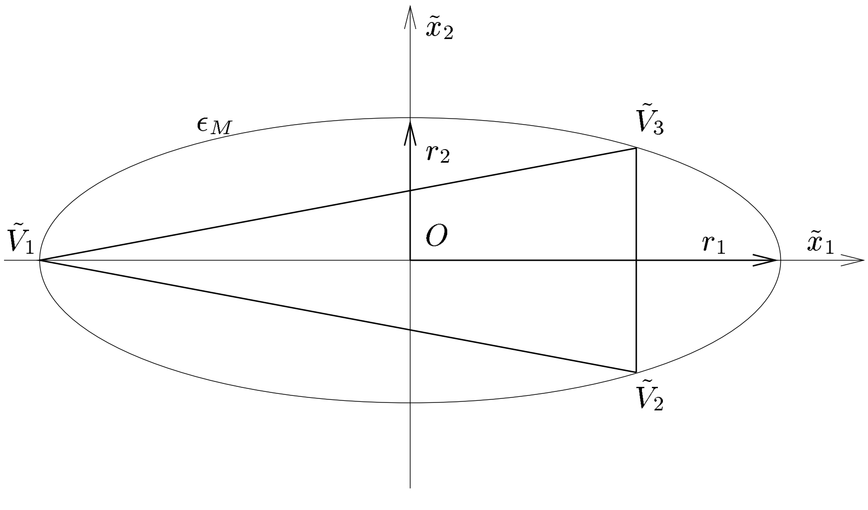

Figure 1.53: An ellipse ϵ_M corresponding to M in the new Cartesian system and the optimal triangle $T_M = \tilde{V}_1\tilde{V}_2\tilde{V}_3$.

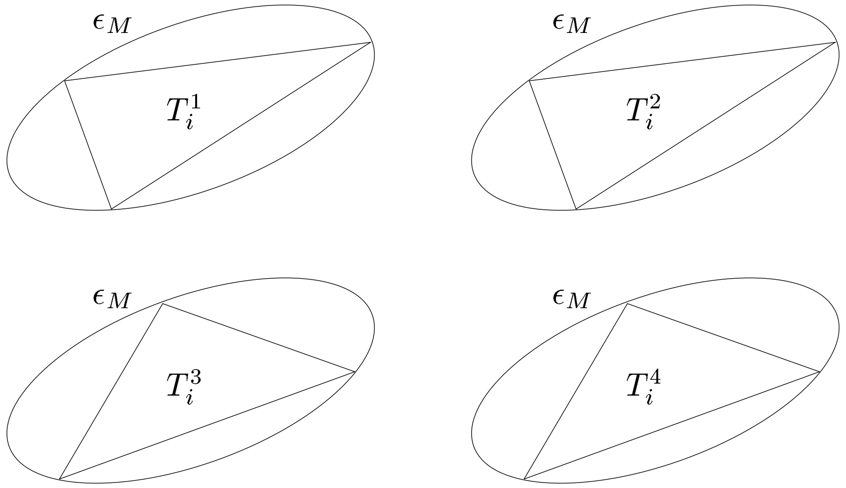

Figure 1.54: Ellipse ϵ_M and the corresponding optimal elements $T_i^1, \ldots, T_i^4$.

$$\tilde{V}_1^{(4)} = \left[-\frac{1}{\sqrt{\lambda_2}}, 0\right], \quad \tilde{V}_2^{(4)} = \left[\frac{1}{2}\frac{1}{\sqrt{\lambda_2}}, -\frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_1}}\right], \quad \tilde{V}_3^{(4)} = \left[\frac{1}{2}\frac{1}{\sqrt{\lambda_2}}, \frac{\sqrt{3}}{2}\frac{1}{\sqrt{\lambda_1}}\right]. \tag{9.8}$$

Then the area of all these triangles $\tilde{V}_1^{(l)}\tilde{V}_2^{(l)}\tilde{V}_3^{(l)}$, $l = 1, \ldots, 4$ is equal to

$$|T_M| = \frac{3\sqrt{3}}{4}\frac{1}{\sqrt{\lambda_1\lambda_2}}. \tag{9.9}$$

Figure 1.53 shows the optimal triangle $\tilde{V}_1^{(1)}\tilde{V}_2^{(1)}\tilde{V}_3^{(1)}$. The lengths of edges parallel with the direction $\tilde{x}_2$ are smaller than the lengths of edges parallel with the direction $\tilde{x}_1$.

Now, we can go back to the original coordinate system by the transformation

$$V = R\tilde{V}, \tag{9.10}$$

where V is a point in the system Ox_1x_2, $\tilde{V}$ is a point in the system $O\tilde{x}_1\tilde{x}_2$, R is the matrix of the rotation from (9.3). Applying (9.10) to the points $\tilde{V}_1, \tilde{V}_2, \tilde{V}_3$ (here we use the transformation only for the first triangle given by (9.5) and we omit the superscript $^{(1)}$) we obtain the points V_1, V_2, V_3 which are the vertices of the optimal triangle T_M in the original system:

$$\begin{aligned}
V_1 &= \left[\frac{\cos\theta}{\sqrt{\lambda_1}}, \frac{\sin\theta}{\sqrt{\lambda_2}}\right] \\
V_2 &= \left[-\frac{\sqrt{3}}{2}\frac{\sqrt{\lambda_2}\cos\theta + \sqrt{\lambda_1}\sin\theta}{\sqrt{\lambda_1\lambda_2}}, \frac{\sqrt{3}}{2}\frac{\sqrt{\lambda_1}\cos\theta - \sqrt{\lambda_2}\sin\theta}{\sqrt{\lambda_1\lambda_2}}\right], \\
V_3 &= \left[-\frac{\sqrt{3}}{2}\frac{-\sqrt{\lambda_2}\cos\theta + \sqrt{\lambda_1}\sin\theta}{\sqrt{\lambda_1\lambda_2}}, \frac{\sqrt{3}}{2}\frac{\sqrt{\lambda_1}\cos\theta + \sqrt{\lambda_2}\sin\theta}{\sqrt{\lambda_1\lambda_2}}\right].
\end{aligned}$$

Now we compute the norms corresponding to M of the edges of T_M. We put $\boldsymbol{v}_1 = V_1 - V_2$, $\boldsymbol{v}_2 = V_2 - V_3$, $\boldsymbol{v}_3 = V_3 - V_1$. After some elementary but long calculation we find that

$$\|\boldsymbol{v}_i\|_M = \left(\boldsymbol{v}_i^{\mathrm{T}} M \boldsymbol{v}_i\right)^{\frac{1}{2}} = \sqrt{3}, \qquad i = 1, 2, 3, \tag{9.11}$$

i.e. the norms of all edges of the optimal triangle T_M are constant and equal to $\sqrt{3}$. The relation (9.11) is valid also for triangles which we obtain applying (9.10) to (9.6) – (9.8). This is a very interesting and important property in the context of AMA.

9.2 3D case

It is possible to proceed in the similar way as in 2D case. We present here only the main items. Let M be a symmetric and positive definite 3×3 matrix

$$M = \begin{pmatrix} a & b & c \\ b & d & e \\ c & d & f \end{pmatrix}, \tag{9.12}$$

and let $\boldsymbol{v} = (v_1, v_2, v_3)$ be a vector in $I\!R^3$. We define analogously as in 2D case *the norm* of the vector $\boldsymbol{v}$ *corresponding* to the matrix M as

$$\|\boldsymbol{v}\|_M \equiv \left(\boldsymbol{v}^{\mathrm{T}} M \boldsymbol{v}\right)^{\frac{1}{2}}. \tag{9.13}$$

We decompose M in the following way:

$$M = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} = R^T \Lambda R = \tag{9.14}$$

$$= \begin{pmatrix} \cos\varphi \sin\theta \cos 2\varphi + \sin 2\varphi \sin\varphi & \sin\varphi \sin\theta \cos 2\varphi - \sin 2\varphi \cos\varphi & -\cos\theta \cos 2\varphi \\ \sin 2\varphi \cos\varphi \sin\theta - \sin\varphi \cos 2\varphi & \sin 2\varphi \sin\varphi \sin\theta - \cos\varphi \cos 2\varphi & -\cos\theta \sin 2\varphi \\ \cos\theta \cos\varphi & \cos\theta \sin\varphi & \sin\theta \end{pmatrix}$$

$$\times \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

$$\times \begin{pmatrix} \cos\varphi \sin\theta \cos 2\varphi + \sin 2\varphi \sin\varphi & \sin 2\varphi \cos\varphi \sin\theta - \sin\varphi \cos 2\varphi & \cos\theta \cos\varphi \\ \sin\varphi \sin\theta \cos 2\varphi - \sin 2\varphi \cos\varphi & \sin 2\varphi \sin\varphi \sin\theta - \cos\varphi \cos 2\varphi & \cos\theta \sin\varphi \\ -\cos\theta \cos 2\varphi & -\cos\theta \sin 2\varphi & \sin\theta \end{pmatrix},$$

where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of M and $\theta \in [\pi/2, \pi/2], \varphi \in [0, 2\pi)$. The inequality

$$x^{\mathrm{T}} M x = a x_1^2 + d x_2^2 + f x_3^2 + 2b x_1 x_2 + 2c x_1 x_3 + 2e x_2 x_3 \leq 1, \qquad x = (x_1, x_2, x_3), \tag{9.15}$$

defines an ellipsoid ϵ_M with the axes lengths $r_i = 1/\sqrt{\lambda_i},\ i = 1, 2, 3$ and the values φ and θ are the spherical coordinates of the the direction of major axis, see Figure 1.55. Similarly as in 2D case we define the *optimal tetrahedron* T_M *corresponding to* M such that $T_M \subset \epsilon_M$ and $|T_M| \geq |T|$ for any tetrahedron $T \subset \epsilon_M$. (Here $|T|$ means the volume of the tetrahedron T).

It is possible to prove that there exists 24 optimal tetrahedra for each matrix M. Moreover, if $\boldsymbol{v}_i, i = 1, \ldots, 6$ are the edges of the optimal tetrahedron T_M than

$$\|\boldsymbol{v}_i\|_M = \sqrt{\frac{8}{3}}, \qquad i = 1, \ldots, 6, \tag{9.16}$$

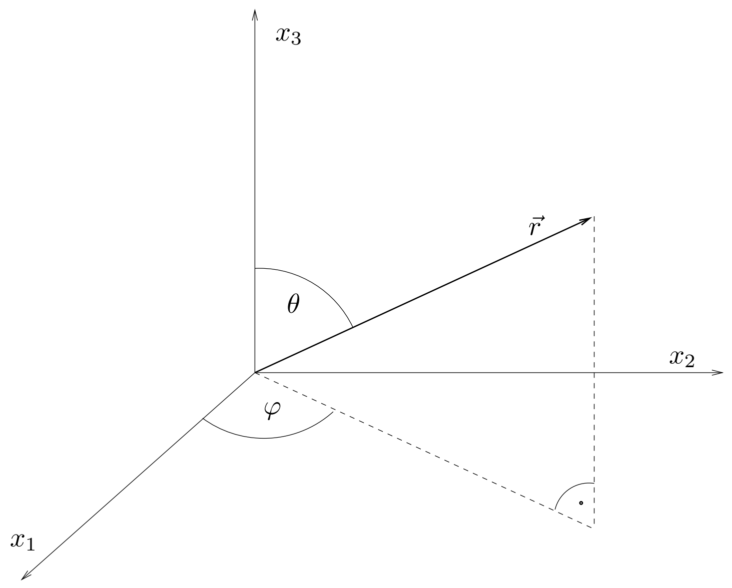

Figure 1.55: Spherical coordinates φ and θ of the direction $\vec{r}$.

Bibliography

[ADFF98] Ph. Angot, V. Dolejší, M. Feistauer, and J. Felcman. Analysis of a combined barycentric finite volume – nonconforming finite element method for nonlinear convection-diffusion problems. *Appl. Math.*, 43(4):263–310, 1998.

[Ape99] T. Apel. *Anisotropic Finite Elements: Local Estimates and Applications.* Teubner, Stuttgart - Leipzig, 1999.

[BD77] G. C. Buscaglia and E. A. Dari. Anisotropic mesh optimization and its application in adaptivity. *International Journal for Numerical Methods in Engineering*, 40:4119–4136, 1977.

[BDGG89] R. Busto, W. D. Dietrich, M. Y. Globus, and M. D. Ginsberg. The importance of brain temperature in cerebral ischemic injury. *Stroke*, 20:1114–1134, 1989.

[BDM] M. Breuss, V. Dolejší, and A. Meister. On an adaptive method for heat conduction problems with boundary layers. *Preprint Hamburger Beiträge zur Angenwandten Mathematik.* (submitted to Computing and Visualization in Science).

[BHF97] H. Borouchaki, F. Hecht, and P. J. Frey. Mesh gradation control. In *Proceedings of 6th International Meshing Roundtable*, Park City, Utah, USA, 1997.

[BSW83] R. Bank, A. Sherman, and A. Weiser. Refinement algorithm and data structures for regular local mesh refinement. In R. Stepleman et al., editor, *Scientific Computing*, volume 44, pages 3–17. IMACS North–Holland, Amsterdam, 1983.

[CDBG+96] M. J. Castro-Díaz, H. Borouchaki, P. L. George, F. Hecht, and B. Mohammadi. Anisotropic Adaptive Mesch Generation in Two Dimensions for CFD. In J.-A. Désidéri, C. Hirsch, P. Le Tallec, M. Pandolfi, and J. Périaux, editors, *Computational Fluid Dynamics '96*, pages 181–186, Paris, 1996. Wiley, Chichester.

[Cia79] P. G. Ciarlet. *The Finite Elements Method for Elliptic Problems.* North-Holland, Amsterdam, New York, Oxford, 1979.

[CKe00] B. Cockburn, G. E. Karniadakis, and C.-W. Shu (eds.). Discontinuous Galerkin methods. In *Lecture Notes in Computational Science and Engineering 11.* Springer, Berlin, 2000.

[Coc] B. Cockburn. An introduction to the discontinuous Galerkin method for convection-dominated problems.

[DA96] V. Dolejší and Ph. Angot. Finite Volume Methods on Unstructured Meshes for Compressible Flows. In F. Benkhaldoun and R. Vilsmeier, editors, *Finite Volumes for Complex Applications (Problems and Perspectives)*, pages 667–674. Hermes, Rouen, 1996.

[DF] V. Dolejší and J. Felcman. Anisotropic mesh adaptation and its application for scalar diffusion equations. (submitted to Numerical Methods for Partial Differential Equations).

[DF02] V. Dolejší and J. Felcman. Anisotropic mesh adaptation for transonic and supersonic flow simulation. In A. Handličová, Z. Krivá, K. Mikola, and D. Ševčovič, editors, *Algoritmy 2002,* 16^{th} *Conference on Scientific Computing*, pages 78–85. Slovak University of Technology, Bratislava, 2002.

[DFFK02] V. Dolejší, M. Feistauer, J. Felcman, and A. Kliková. Error estimates for barycentric finite volumes combined with nonconforming finite elements applied to nonlinear convection–diffusion problems. *Applications of Mathematics*, 47(4):301–340, 2002.

[DHM95] M. J. Castro Díaz, F. Hecht, and B. Mohammadi. New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. In *Proceedings of the 4th Annual International Meshing Roundtable.* Sandia National Laboratories, 1995.

[Dol98a] V. Dolejší. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Computing and Visualization in Science*, 1(3):165–178, 1998.

[Dol98b] V. Dolejší. *Sur des méthodes combinant des volumes finis et des éléments finis pour le calcul d'écoulements compressibles sur des maillages non structurés.* PhD thesis, Charles Univeristy Prague and Université Mediterannée Aix–Marseille II, 1998.

[Dol01] V. Dolejší. Anisotropic mesh adaptation technique for viscous flow simulation. *East-West Journal of Numerical Mathematics*, 9(1):1–24, 2001.

[DS89] E. F. D'Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM. J. Sci. Statist. Comput.*, 10:1063–1075, 1989.

[DS91] E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Numer. Math.*, 59:321–348, 1991.

[FD98] J. Felcman and V. Dolejší. Investigation of accuracy for higher order finite volume schemes. *Engineering Mechanics*, 5(5):327–337, 1998.

[FDF94] J. Felcman, V. Dolejší, and M. Feistauer. Adaptive Finite Volume Method for the Numerical Solution of the Compressible Euler Equations. In J. Périaux S. Wagner, E. H. Hirschel and R. Piva, editors, *Computational Fluid Dynamics '94*, pages 894–901. John Wiley and Sons, Stuttgart, 1994.

[Fei93] M. Feistauer. *Mathematical Methods in Fluid Dynamics.* Longman Scientific & Technical, Harlow, 1993.

[FF96] M. Feistauer and J. Felcman. Convection-Diffusion Problems and Compressible Navier–Stokes Equations. In J. R. Whitemen, editor, *The Mathematics of Finite Elements and Applications*, pages 175–194. John Wiley & Sons, 1996.

[FFD96] M. Feistauer, J. Felcman, and V. Dolejší. Numerical simulation of compresssible viscous flow through cascades of profiles. *ZAMM*, 76(S4):297–300, 1996.

[FFL95] M. Feistauer, J. Felcman, and M. Lukáčová. Combined finite element–finite volume solution of compressible flow. *J. Comput. Appl. Math.*, 63:179–199, 1995.

[FJS] M. Feistauer, J.Felcman, and I. Straškraba. Mathematical and computational methods for compressible flow.

[FLM00] B. Fischer, M. Ludwig, and A. Meister. An application of the finite volume method to the thermoregulation of infants. Hamburg Beiträge zur Angewandten Mathematik, Reihe F, Computational Fluid Dynamics and Data Analysis 12, 2000.

[FOG97] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Methods Eng*, 40(21):3979–4002, 1997.

[FŠ98] J. Felcman and P. Šolín. On the construction of the Osher–Solomon scheme for 3D Euler equations. *East–West J. Numer. Math.*, 6(1):43–64, 1998.

[FVD+96] M. Fortin, M.-G. Vallet, J. Dompierre, Y. Bourgault, and W. G. Habashi. Anisotropic Mesh Adaptation: Theory, Validation and Applications. In J.-A. Désidéri, C. Hirsch, P. Le Tallec, M. Pandolfi, and J. Périaux, editors, *Computational Fluid Dynamics '96*, pages 174–180, Paris, 1996. Wiley, Chichester.

[GW92] P. D. Gluckman and C. E. Williams. When and why do brain cells die? *Dev. Med. Child Neurol.*, 34:1010–1014, 1992.

[HFaY+96] W. G. Habashi, M. Fortin, D. Ait ali Yahia, S. Boivin, Y.Bourgault, J. Dompierre, M. P. Robichaud, A. Tam, and M. G. Vallet. Anisotropic mesh optimization: Towards a solver-independent and mesh-independent CFD. In *Compressible Fluid Dynamics*, VKI Lecture Series 1996-06. von Karman Institute for Fluid Dynamics - Concordia University, August 1996.

[Kli00] A. Kliková. *Finite volume – finite element solution of compressible flow.* PhD thesis, Charles Univeristy, Prague, 2000.

[KR94] D. Kröner and M. Rokyta. Convergence of upwind finite volume schemes for scalar conservation laws in 2D. *SIAM J. Numer. Anal.*, 31:324–343, 1994.

[Krö97] D. Kröner. *Numerical Schemes for Conservation Laws.* Wiley Teubner, Stuttgart, 1997.

[MWJG95] E. C. Mallard, C. E. Williams, B. M. Johnston, and P. D. Gluckman. Neuronal damage in the developing brain following intrauterine asphyxia. *Reprod. Fertil. Dev.*, 7:647–653, 1995.

[OS82] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comp.*, 38:339–374, 1982.

[Ran98] R. Rannacher. Adaptive finite element methods. In C. Zenger H. Bulgak, editor, *NATO–Summer School: Error Control and Adaptivity in Scientific Computing*, pages 247–278. Kluwer, Academic Publishers, 1998.

[Sim94] R. B. Simpson. Anisotropic mesh transformations and optimal error control. *Applied Numer. Math.*, 14:183–198, 1994.

[ŠŠ90] M. Štastný and P. Šafařík. Experimental analysis data on the transonic flow past a plane turbine. Technical report, ASME Paper 90-GT-313, New York, 1990.

[Tem77] R. Temam. *Navier–Stokes Equations. Theory and numerical analysis.* North–Holland, Amsterdam–New York–Oxford, 1977.

[Ver96] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques.* Wiley-Teubner Series Advances in Numerical Mathematics. Chichester: John Wiley & Sons. Stuttgart, 1996.

[WC84] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. of Comput. Phys.*, 54:115–173, 1984.

[ZZ92a] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a-posteriori error estimates. Part 1: The recovery technique. *Internat. J. Numer. Methods Engrg.*, 33:1331–1364, 1992.

[ZZ92b] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a-posteriori error estimates. Part 2: Error estimated and adaptivity. *Internat. J. Numer. Methods Engrg.*, 33:1365–1382, 1992.