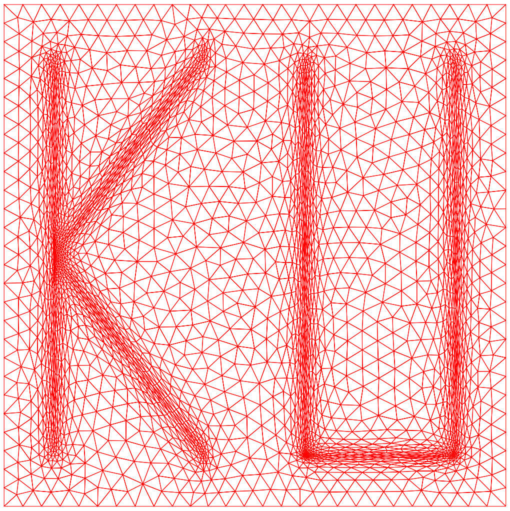
Anisotropic Mesh Adaptation and Movement

Weizhang Huang

Department of Mathematics, the University of Kansas
Lawrance, KS 66045, U.S.A.

July 6, 2005



These lecture notes were prepared for the workshop on
Adaptive Method, Theory and Application
organized by Zhiping Li, Tao Tang, Jinchao Xu, and Pingwen Zhang
June 20 – August 20, 2005
Peking University, Beijing, China.

The author's work on anisotropic meshes was supported partly by the NSF (U.S.A.) under grants DMS-0074240 and DMS-0410545 and by the University of Kansas General Research Fund.

Contents

1 Primaries **1**

1.1 Introduction . . . 1

1.2 Sobolev spaces . . . 2

1.3 Mesh terminology . . . 4

1.4 Two algebraic inequalities . . . 5

2 Basic principles in mesh adaptation **7**

2.1 Introduction . . . 7

2.2 Geometric meaning of SVD decomposition . . . 7

2.3 Alignment and equidistribution . . . 8

2.4 Alignment and equidistribution for finite element meshes . . . 11

3 Interpolation theory in Sobolev spaces **13**

3.1 Introduction . . . 13

3.2 Finite element terminology . . . 13

3.3 Element-wise estimate on interpolation error . . . 13

4 Isotropic error estimates **17**

4.1 Introduction . . . 17

4.2 Chain rule . . . 17

4.3 Isotropic error estimation on a general mesh . . . 19

4.4 Error bound on regular triangulations . . . 19

5 Anisotropic error estimates **21**

5.1 Introduction . . . 21

5.2 An anisotropic error bound . . . 21

5.3 Anisotropic error estimates independent of coordinate system . . . 22

5.3.1 Case $l = 1$ . . . 22

5.3.2 Case $l \geq 2$ . . . 23

5.4 Bibliographic notes . . . 24

6 Mesh quality measures and monitor functions **27**

6.1 Introduction . . . 27

6.2 Mesh quality measures in view of mesh adaptation . . . 28

6.3 The case with isotropic error estimation . . . 31

6.3.1 Mesh quality measures and monitor function . . . 31
6.3.2 Mesh assessment . . . 35
6.4 The case with anisotropic error estimation: $l = 1$ . . . 36
6.4.1 Mesh quality measures and monitor function . . . 36
6.4.2 Mesh assessment . . . 37
6.5 The case with anisotropic error estimation: $l \geq 2$ . . . 38
6.5.1 Mesh quality measures . . . 38
6.5.2 Mesh assessment . . . 40

7 Ansiotropic mesh adaptation: Refinement approach 41
7.1 Introduction . . . 41
7.2 Metric tensor . . . 42
7.2.1 Isotropic error estimation . . . 43
7.2.2 Anisotropic error estimation: $l = 1$ . . . 44
7.2.3 Anisotropic error estimation: $l \geq 2$ . . . 44
7.2.4 A remark on computation of metric tensor and monitor function . . . 44
7.3 Numerical experiment . . . 44

8 Ansiotropic mesh adaptation: Variational approach 51
8.1 Introduction . . . 51
8.2 Functional for mesh alignment . . . 52
8.3 Functional for equidistribution . . . 52
8.4 Mesh adaptation functional . . . 53
8.5 Mesh equation . . . 54
8.6 Numerical experiment . . . 55

9 Adaptive moving mesh methods: MMPDE approach 61
9.1 Introduction . . . 61
9.2 The MMPDE method . . . 63
9.2.1 MMPDE for mesh movement . . . 63
9.2.2 Discretization on a moving mesh . . . 64
9.2.3 Alternating solution procedure . . . 66
9.3 Example examples . . . 67

10 Adaptive moving mesh methods: GCL approach 73
10.1 Introduction . . . 73
10.2 GCL method . . . 73
10.3 Relation to the Lagrange method and the deformation map method . . . 75
10.4 Choice of w, v_{ref}, and ρ . . . 75
10.5 Numerical examples . . . 76

11 Conclusions and comments 83

Chapter 1

Primaries

1.1 Introduction

Many partial differential equations (PDEs) arising from science and engineering have a common feature that they have a small portion of the physical domain where small node separations are required to resolve large solution variations. Examples include problems having boundary layers, shock waves, ignition fronts, and/or sharp interfaces in fluid dynamics, the combustion and heat transfer theory, and groundwater hydrodynamics. Numerical solution of these PDEs using a uniform mesh may be formidable when the systems involve more than two spatial dimensions since the number of mesh nodes required can become very large. On the other hand, to improve efficiency and accuracy of numerical solution it is natural to put more mesh nodes in the region of large solution variation than the rest of the physical domain. With this basic idea of mesh adaptation, the number of mesh nodes required can be much smaller; thus significant economies can be gained.

Mesh adaptation lies in the ability to control the *size*, *shape*, and *orientation* of mesh elements throughout the domain. Traditionally, research has been concentrated mainly on isotropic mesh adaptation where where mesh elements are adjusted only in size according to an error estimate or indicator while their shape is kept to be or close to being equilateral; e.g., see books [51, 27, 113, 2] and references therein. Unfortunately, adaptive isotropic meshes often tend to use too many elements in the region of large solution error. This is especially true when problems have an anisotropic feature that the solution changes more significantly in one direction than the others. Full benefits of mesh adaptation can be taken by adjusting not only the size but also the shape and orientation of mesh elements according to the behavior of the physical solution. Often this results in an anisotropic mesh, a mesh having elements with large aspect ratio.

Mathematical studies of anisotropic meshes can be traced back to Synge [108], Zlámal [129], Babuška and Aziz [8], Jamet [70], and Barnhill and Gregory [12]. In the last decade, research has been intensified and progress has been made in developing strictly mathematically-based error estimates; e.g., see [3, 4, 5, 6, 7, 34, 33, 37, 38, 43, 49, 53, 61, 62, 76, 78, 97, 100, 103, 104, 105]. Particularly, D'Azevedo and Simpson [37, 38] introduce the concept of optimal triangles and derive estimates of linear interpolation error and its gradient for quadratic functions, and their results have motivated the formulation of the so-called metric tensor used in a number of anisotropic mesh generation codes.

On the practical side, a number of strategies and computer codes have been developed for anisotropic mesh generation, see the mesh generation website maintained by Robert Schneiders [101] and the meshing software survey conducted by Steve Owen [94]. They mainly fall into two categories, h-version and r-version. h-version methods typically employ local tools such as edge suppression, vertex suppression, vertex addition, edge swapping, and vertex reallocation (barycentering step) and generate unstructured anisotropic meshes as isotropic ones in the metric determined by a tensor specifying the size, shape and orientation of mesh elements on the whole physical domain. Various strategies have been used, including the Delaunay triangulation method [16, 17, 31, 95], the advancing front method [50], the bubble mesh method [122], the quadtree-based method [91], and the method combining local modification with smoothing or node movement [3, 18, 43, 53].

On the other hand, r-version methods generate adaptive meshes by dynamically reallocating node positions. Typically they employ a system of elliptic- or parabolic-type PDEs to produce a structured mesh [114] although they can also be used to generate an unstructured one [24]. A type of r-version method is variational methods. They formulate mesh generation PDEs through a functional and often incorporate the well-known equidistribution principle and mesh properties such as smoothness and orthogonality into the formulation. Examples of variational methods include those developed in [20, 45, 59, 73, 121]. (Also see the books [27, 72, 85, 114] and references therein). For time dependent problems, r-version methods are often referred to as adaptive moving mesh methods or moving mesh methods for short, where nodes are typically moved around to adapt to the moving feature of the physical solution. Moving mesh methods can be developed based on a (static) variational mesh generator, for example see [44, 64, 65, 67, 13, 82, 83]. Other types of moving mesh method include the moving finite element method (MFE) [90, 89] (also see the books by Baines [9] and Zegeling [124]) and the GCL method [25].

This series of lectures is devoted to the study of anisotropic meshes in both theory and algorithm. Focus is on basic principles of mesh adaptation, interpolation theory, anisotropic error estimates, monitor functions, variational mesh generation, and moving mesh methods. The moving finite element method or the MFE of Miller will not be discussed here. The interested reader is referred to the book by Baines [9].

1.2 Sobolev spaces

Throughout this lecture series, Ω is used to denote a simply connected, open, bounded domain in the n-dimensional space, $\Re^n$ $(n > 0)$. $\overline{\Omega}$ is the closure of Ω, and $|\Omega|$ denotes its volume (or n-dimensional measure). Given a multi-index $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ of non-negative integers, let $|\alpha| = \alpha_1 + \cdots + \alpha_n$ and

$$D^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} ... x_n^{\alpha_n}}.$$

Sometimes an l-th order partial derivative is also denoted by

$$D^{(i_1,...,i_l)} u = \frac{\partial^l u}{\partial x_{i_1} \cdots \partial x_{i_l}},$$

where $(i_1, ..., i_l)$ is an integer vector of l components with $1 \le i_1, ..., i_l \le n$. $D^l u$ or $D^{|\alpha|} u$ $(l = |\alpha|)$ is used to denote the set of all l-th order derivatives.

Lebesgue space $L^p(\Omega)$ $(1 \leq p < \infty)$ is defined as the vector space of the functions $u : \Omega \to \Re$ for which $|u|^p$ is Lebesgue integrable on Ω and $\int_\Omega |u(x)|^p dx < \infty$. It is a Banach space for the norm

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega |u(x)|^p dx \right)^{\frac{1}{p}}.$$

$L^\infty(\Omega)$ is the Banach space of the measurable functions which are defined on Ω and bounded outside a set of measure zero. It is equipped with the norm

$$\|u\|_{L^\infty(\Omega)} = \text{ess } \sup_{x \in \Omega} |u(x)|.$$

The Lebesgue spaces have the imbedding property: $L^1(\Omega) \hookleftarrow L^2(\Omega) \hookleftarrow \cdots \hookleftarrow L^\infty(\Omega)$. This is an immediate result of the following theorem. The interested reader is referred to Hardy et al. [55] for the proof. The theorem will be used frequently throughout this lecture series, particularly related to the equidistribution principle (2.4).

Theorem 1.2.1 *Given a weight function $w(x)$ with $\int_\Omega w dx = 1$, define*

$$M_r(f) = \left(\int_\Omega w|f|^r dx \right)^{\frac{1}{r}}$$

for arbitrary function f and real number r, with the limits that $M_0(f) = \exp(\int_\Omega w \log |f| dx)$ (geometric mean), $M_{+\infty} = \max |f|$, and $M_{-\infty} = \min |f|$. Then

$$M_r(f) < M_s(f) \tag{1.1}$$

for $-\infty \leq r < s \leq +\infty$ unless (a) $M_r(f) = M_s(f) = +\infty$ which can happen only if $r \geq 0$ or (b) $M_r(f) = M_s(f) = 0$ that can happen only if $s \leq 0$ or (c) $f =$ constant.

Sobolev spaces deal with function derivatives. For a multi-index $\alpha = (\alpha_1, ..., \alpha_n)$ and for a Lebesgue integrable function u on Ω, if there is a function v_α Lebesgue integrable on Ω and satisfying the condition

$$\int_\Omega u D^\alpha \psi dx = (-1)^{|\alpha|} \int_\Omega v_\alpha \psi dx \quad \forall \psi \in \mathcal{D}(\Omega),$$

then v_α is said to be a distributional derivative or a generalized derivative of order $|\alpha|$ of u. It is denoted by $v_\alpha = D^\alpha u$.

For a given integer $m \geq 0$ and a given real number $p \in [1, \infty]$, Sobolev space $W^{m,p}(\Omega)$ is defined as the vector space of the functions $u \in L^p(\Omega)$ such that for each multi-index α with $|\alpha| \leq m$, distributional derivative $D^\alpha u$ belongs to $L^p(\Omega)$. It is a Banach space for the norm

$$\|u\|_{W^{m,p}(\Omega)} = \left(\sum_{|\alpha| \leq m} \int_\Omega |D^\alpha u|^p dx \right)^{\frac{1}{p}}.$$

The semi-norm is denoted by

$$|u|_{W^{m,p}(\Omega)} = \left(\sum_{|\alpha| = m} \int_\Omega |D^\alpha u|^p dx \right)^{\frac{1}{p}} = \left(\int_\Omega \|D^m u\|_{l_p}^p dx \right)^{\frac{1}{p}},$$

where $\|\cdot\|_{l_p}$ denotes the l_p matrix norm. The scaled semi-norm will also be used,

$$\langle u\rangle_{W^{m,p}(\Omega)} = \left(\frac{1}{|\Omega|}\sum_{|\alpha|=m}\int_\Omega |D^\alpha u|^p dx\right)^{\frac{1}{p}} = \left(\frac{1}{|\Omega|}\int_\Omega \|D^m u\|_{l_p}^p dx\right)^{\frac{1}{p}}. \tag{1.2}$$

It is noted that

$$L^p(\Omega) = W^{0,p}(\Omega), \quad H^m(\Omega) = W^{m,2}(\Omega).$$

The reader is referred to Adams [1] for the Sobolev imbedding theorem which shows the imbedding characteristics of Sobolev spaces.

1.3 Mesh terminology

Consider a polyhedral domain Ω in $\Re^n$. Assume that a family of triangulations or meshes $\{\mathcal{T}_h\}$ is given on Ω, with $h > 0$ being the parameter characterizing the family. Let K be the generic element in $\mathcal{T}_h$. By convention, K is a close sub-domain of Ω. Also, $\overline{\Omega} \subset \bigcup_{K\in\mathcal{T}_h} K$, and the interiors of any two different elements should not be overlapped. Denote by N the number of elements of $\mathcal{T}_h$.

A *uniform* mesh of Ω is a triangulation whose elements are equilateral and of the same size. Thus, for a uniform mesh,

$$h_K \equiv \text{ diameter}(K) \approx \left(\frac{|\Omega|}{N}\right)^{\frac{1}{n}} = O(N^{-\frac{1}{n}}) \qquad \forall K \in \mathcal{T}_h. \tag{1.3}$$

For a family of uniform meshes, the parameter h can be defined as $h = \max_{K\in\mathcal{T}_h} h_K$. Thus,

$$h \to 0 \quad \Longleftrightarrow \quad N \to \infty. \tag{1.4}$$

A generalization of uniform meshes is a *regular* family of triangulations which satisfies the conditions:

(i) There exists a constant σ such that

$$\frac{h_K}{\rho_K} \le \sigma, \qquad \forall K \in \bigcup_h \mathcal{T}_h,$$

where $h_K =$ diam(K) and $\rho_K = \sup\{$ diam(B) : B is a ball contained in $K\}$ are the diameter and in-diameter, respectively, of K.

(ii)

$$h = \max_{K\in\mathcal{T}_h} h_K \to 0.$$

Note that both (1.3) holds for a regular family of triangulations. A *quasi-uniform* mesh is referred to a mesh for which (i) is satisfied and the variation of its element size is bounded by a constant. Thus, both (1.3) and (1.4) are true for a quasi-uniform mesh.

Throughout this series lectures, we make the following assumption

$$\boxed{\textbf{H1.}\ \{\mathcal{T}_h\} \text{ is an \textit{affine} family of triangulations for } \Omega.} \tag{1.5}$$

A family of triangulations is called an *affine* family of triangulations if there exits an element $\hat{K}$ such that any element K in the family of triangulations can be mapped from $\hat{K}$ through an affine mapping. In other words, for each element K of $T_h \in \{\mathcal{T}_h\}$, there exists an invertible affine mapping $F_K : \hat{K} \to K$ such that $K = F_K(\hat{K})$. The element $\hat{K}$ is called the reference or master element for the family of triangulations. Without loss of generality, we assume

$$\boxed{\textbf{H2.}\ \text{The reference element has been chosen to be equilateral with unitary size } |\hat{K}| = 1.} \tag{1.6}$$

From time to time, a mesh is considered to be the image of a computational mesh under an invertible coordinate transformation $x = x(\xi) : \Omega_c \to \Omega$. Domain Ω_c is called the computational domain which is artificially chosen for the purpose of mesh generation. Typically, the computational mesh is chosen to be uniform or quasi-uniform. Since generating the mesh is mathematically equivalent to determining the coordinate transformation, a coordinate transformation is viewed equivalently as a mesh.

The inverse coordinate transformation of $x = x(\xi)$ is denoted by $\xi = \xi(x)$. The Jacobian matrix and its determinant (i.e. the Jacobian) are denoted by

$$\boldsymbol{J} = \frac{\partial x}{\partial \xi}, \quad J = \det(\boldsymbol{J}).$$

1.4 Two algebraic inequalities

The following two inequalities will be used frequently. The first is a generalization of the well-known arithmetic-mean geometric-mean inequality and the second is Jensen's inequality.

Theorem 1.4.1 (Generalized arithmetic-mean geometric-mean inequality.) *Let $w_1, ..., w_m$ be m weights satisfying $w_i > 0$ and $\sum_i w_i = 1$. Then, for any positive numbers $a_1, ..., a_m$,*

$$\left(\sum_{i=1}^m w_i a_i^s \right)^{\frac{1}{s}} \le \left(\sum_{i=1}^m w_i a_i^t \right)^{\frac{1}{t}}$$

for any numbers $-\infty \le s < t \le \infty$, with equality iff (if and only if) $a_1 = \cdots = a_n$. Here, the following convention has been used,

$$\left(\sum_{i=1}^m w_i a_i^s \right)^{\frac{1}{s}} = \begin{cases} \prod_{i=1}^m a_i^{w_i} & \textit{for } s = 0, \\ \max_{1 \le i \le m} a_i & \textit{for } s = \infty, \\ \min_{1 \le i \le m} a_i & \textit{for } s = -\infty. \end{cases}$$

Note that the above theorem reduces to the well known arithmetic-mean geometric-mean inequality when $s = 0$, $t = 1$, and $w_1 = \cdots = w_m = \frac{1}{m}$.

Theorem 1.4.2 (Jensen's inequality.) *For any m positive numbers $a_1, ..., a_m$, the inequality*

$$\left(\sum_{i=1}^m a_i^s \right)^{\frac{1}{s}} \ge \left(\sum_{i=1}^m a_i^t \right)^{\frac{1}{t}}$$

holds for any numbers s and t with $0 < s \le t$.

Chapter 2

Basic principles in mesh adaptation

2.1 Introduction

Essential to mesh adaptation is the ability to control the size, shape, and orientation of mesh elements. This is often done in two steps. In the first steps, the element information on size, shape, and orientation is specified throughout the domain. The specification is often based on an error indicator, an error estimate, or a physical consideration. It typically utilizes a scalar function for isotropic mesh generation and a matrix-valued field for the anisotropic situation. In the second step, an algorithm is developed for generating the needed adaptive mesh according to the specification of element information.

In this chapter we study basic mathematical principles behind the specification of element information. Development of algorithms will be discussed later in Chapters 7, 8, 9, and 10.

2.2 Geometric meaning of SVD decomposition

A continuous view for mesh adaptation is to consider an adaptive mesh to be generated as the image of a computational mesh under a reversible coordinate transformation $x = x(\xi) : \Omega_c \to \Omega$, where Ω_c is the computational domain. Assuming that a uniform computational mesh is chosen, the size, shape, and orientation of mesh elements are then completely determined by $x = x(\xi)$. Consequently, the control of elements is equivalent to the control of $x = x(\xi)$. Unfortunately, it is not an easy task, if not impossible, to explicitly define a coordinate transformation having desired mesh concentration. For this reason, a closer look is needed for what determines the element size, shape, and orientation.

In the continuous viewpoint, mesh elements are represented by ellipsoids. The size, shape, and orientation of an element is clear: the size is its volume, the shape is determined by the relative lengths or ratios between the lengths of its semi-axes, and the orientation is specified by its principal axis vectors. Fig. 2.1 shows an ellipse in two dimensions. The size is its area πab, the shape is determined by the ratio a/b or b/a, and the orintation is specified by the vectors v_1 and v_2.

Let e be an arbitrary element and let e_c be the corresponding computational element. They are related by $e = x(e_c)$ under coordinate transformation $x = x(\xi)$. e_c is a ball since the computational mesh is assumed to be uniform. Being linearized about the center of e_c, ξ_0, the coordinate

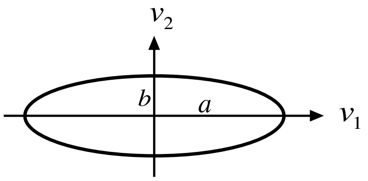


Figure 2.1: An ellipse – a mesh element in the continuous form in two dimensions.

transformation can be expressed as

$$x = x_0 + \boldsymbol{J}(\xi - \xi_0) + O(|\xi - \xi_0|^2),$$

where x_0 is the center of e and $\boldsymbol{J}$ is the Jacobian matrix of $x = x(\xi)$ calculated at ξ_0. To see how e_c is mapped into e, we consider the singular value decomposition (SVD) of $\boldsymbol{J}$,

$$\boldsymbol{J} = U\Sigma V^T,$$

where U and V are the orthogonal matrices associated with left and right singular vectors, respective, and Σ is the diagonal matrix consisting of the singular values. The geometric meaning of the SVD is illustrated in Fig. 2.2. Specifically, the computational element e_c is rotated by V, then mapped and compressed/expanded in the coordinate directions into a physical element by Σ, and finally rotated again by U and becomes e. Thus, U determines the orientation and Σ specifies the size and shape of physical element e, while V rotates the computational element and has no influence on the size, shape, and orientation of the physical element e.

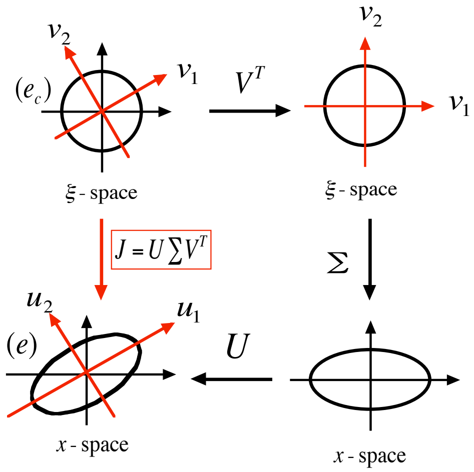


Figure 2.2: Geometric meaning of the singular value decomposition of $\boldsymbol{J}$. Here, $v_1, ..., v_n$ and $u_1, ..., u_n$ are the column vectors of V and U, respectively.

2.3 Alignment and equidistribution

The analysis in the previous section has shown that the size, shape, and orientation of mesh elements are determined by the left singular vectors U and the singular values Σ of the Jacobian matrix $\boldsymbol{J}$.

Thus, their control can be achieved by specifying $\boldsymbol{J}\boldsymbol{J}^T = U\Sigma^2 U^T$ or its inverse $\boldsymbol{J}^{-T}\boldsymbol{J}^{-1} = U\Sigma^{-2}U^T$. Let

$$\boldsymbol{J}^{-T}\boldsymbol{J}^{-1} = \left(\frac{\sigma}{|\Omega_c|}\right)^{-\frac{2}{n}} M(x), \tag{2.1}$$

where $M(x)$ an $n \times n$ symmetric positive definite matrix and σ is a constant defined as

$$\sigma = \int_\Omega \rho(x)dx, \qquad \rho(x) = \sqrt{\det(M(x))}. \tag{2.2}$$

Hence, through (2.1) $M(x)$ specifies the size, shape, and orientation of mesh elements on the whole domain. For this reason, $M(x)$ is referred to as *the monitor function*. The function $\rho(x) = \sqrt{\det(M(x))}$ is called *the adaptation fucntion*. The monitor function can be defined based on interpolation error estimates; see Chapter 6.

The following theorem gives a different perspective to the condition (2.1).

Theorem 2.3.1 *Given a monitor function M on Ω, the condition (2.1) is equivalent to the following two conditions:*

(i) **The alignment condition:**

$$tr\left(\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}\right) = n\ det\left(\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}\right)^{\frac{1}{n}}. \tag{2.3}$$

(ii) **The equidistribution condition:**

$$J\rho = \frac{\sigma}{|\Omega_c|}, \tag{2.4}$$

where $J = det(\boldsymbol{J})$, $\rho = \sqrt{det(M)}$, and σ is defined in (2.2).

The proof of the theorem is given later this section. It is remarked that the conditions (2.3) and (2.4) have been derived and used in [59] for developed a variational mesh adaptation method.

The equidistribution condition (2.4) is a multi-dimensional generalization of the well-known equdistribution principle [22, 39]. It is the most fundamental principle in mesh adaptation. Indeed, there are very few adaptive mesh algorithms that do not use its basic idea: evenly distribute an error function among all the mesh cells. In the current situation, $\rho(x)$ severs as the error function.

The following theorem shows that through alignment condition (2.3), the shape and orientation of mesh elements are determined, respectively, by the relative magnitude of the eigenvalues and the eigenvectors of M.

Theorem 2.3.2 *Given a monitor function M on Ω, the alignment condition (2.3) holds if and only if*

$$\boldsymbol{J}^{-T}\boldsymbol{J}^{-1} = \theta(x)M(x) \qquad \forall x \in \Omega \tag{2.5}$$

holds for some scalar function $\theta = \theta(x)$.

Proof. Let the eigenvalues of matrix $\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}$ be $\lambda_1, ..., \lambda_n$. Then, (2.3) is equivalent to

$$\sum_i \lambda_i = n \left(\prod_i \lambda_i \right)^{\frac{1}{n}} .$$

From the arithmetic-mean geometric-mean inequality (cf. Theorem 1.4.1), the above equation is equivalent to the conditions

$$\lambda_1 = \cdots = \lambda_n.$$

This is in turn equivalent to

$$\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T} = \theta(x) I, \tag{2.6}$$

where I is the identity matrix and $\theta(x) := \lambda_1 = \cdots = \lambda_n$. □

Proof of Theorem 2.3.1. Notice that (2.1) can be rewritten as

$$\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T} = \sigma^{-\frac{2}{n}} I. \tag{2.7}$$

Then the conclusion of the theorem follows from the observation that condition (2.7) is equivalent to the requirement (2.6), with θ being constant as implied by the equidistribution condition (2.4). □

It is interesting to know if there exists a coordinate transformation exactly satisfying (2.1). Generally speaking, n conditions are needed to determine a coordinate transformation $x = x(\xi)$. It is not difficult to see that (2.1) gives $n(n+1)/2$ conditions. Since $n(n+1)/2 > n$ when $n > 1$, (2.1) gives an over-determined system, meaning that in general there does not exist a coordinate transformation exactly satisfying (2.1).

Nevertheless, conditions (2.3) and (2.4), or equivalently condition (2.1), still play a fundamental role in mesh adaptation since they tell how mesh elements can be controlled via the monitor function $M = M(x)$. For convenience, hereafter *a mesh satisfying the alignment and equidistribution conditions exactly will be referred to as the mesh specified by M.* Obviously, in practice a mesh should be generated in such that it is as close as possible to the mesh specified by M.

A natural way to deal with the over-determined system is to use the least squares method. For example, one can define the (inverse) coordinate transformation $\xi = \xi(x)$ as a minimizer of the functional

$$I[\xi] = \int_\Omega \| \boldsymbol{J}^{-T}\boldsymbol{J}^{-1} - \sigma^{-\frac{2}{n}} M(x) \|_F^2 dx,$$

where $\| \cdot \|_F$ is the Frobenius matrix norm. Unfortunately, it leads to an undesired degenerate mesh equation. To see this more clearly, we take the 1D case as an example. In 1D, the above functional becomes

$$I[\xi] = \int_a^b \left(\xi_x^2 - \sigma^{-2} M(x) \right)^2 dx.$$

Its Euler-Lagrange equation is

$$\frac{\partial}{\partial x} \left[\left(\xi_x^2 - \sigma^{-2} M(x) \right) \xi_x \right] = 0,$$

which becomes degenerate when $\xi_x^2 = \sigma^{-2}M(x)$, the 1D form of (2.1). Thus, the least squares method does not work for the current situation.

Fortunately, the difficulty can be overcome by using (2.3) and (2.4) instead of (2.1). A method was proposed in [59], which is to be described in Chapter 8.

2.4 Alignment and equidistribution for finite element meshes

The analysis in the previous section has been given for a coordinate transformation. But it can be easily adopted for an affine family of triangulations. Indeed, one may have noticed that the discussion in the previous sections are mostly local, indicating that $x = x(\xi)$ can be replaced with $x = F_K(\xi)$ when $x \in K$. The size, shape, and orientation of element K is determined by $(F_K^{'})^{-T}(F_K^{'})^{-1}$. Let

$$(F_K^{'})^{-T}(F_K^{'})^{-1} = \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{n}} M(K), \tag{2.8}$$

where $F_K^{'}$ is the Jacobian matrix of F_K, $M(K)$ is a certain average of the monitor function on K, N is the number of elements in $\mathcal{T}_h$, and σ_h is a constant defined as

$$\sigma_h = \sum_{K \in \mathcal{T}_h} |K|\rho(K), \qquad \rho(K) = \sqrt{\det(M(K))}. \tag{2.9}$$

Similarly, the condition (2.8) can be decomposed into the following alignment and equidistribution conditions

$$\text{tr}\left((F_K^{'})^{-1}M^{-1}(F_K^{'})^{-T}\right) = n \det\left((F_K^{'})^{-1}M^{-1}(F_K^{'})^{-T}\right)^{\frac{1}{n}}, \tag{2.10}$$

$$|K|\rho = \frac{\sigma_h}{N}, \tag{2.11}$$

where we have used $|\det(F_K^{'})| = |K|$. Moreover, the condition (2.10) alone is equivalent to

$$(F_K^{'})^{-T}(F_K^{'})^{-1} = \theta(K)M(K) \tag{2.12}$$

for some scalar function $\theta = \theta(K)$.

Chapter 3

Interpolation theory in Sobolev spaces

3.1 Introduction

A classic error estimate in the interpolation theory in Sobolev spaces is presented in this chapter. The bound is very general and holds for any simplicial element in n-dimensional space. The result forms the base for other developments in later chapters, such as anisotropic error estimation, the definition of the monition function, and h- and r-version anisotropic mesh adaptation.

3.2 Finite element terminology

A *finite elenent* is defied as a triple (K, P_K, Σ_K), where K is a mesh element, P_K is a finite-dimensional linear space of functions defined on K, and Σ_K is a set of degrees of freedom which consists of the parameters uniquely determining a function in P_K.

Two finite elements are said to be *affine-equivalent* if their mesh elements, finite dimensional function spaces, and sets of degrees of freedom can be mapped to each other through affine mappings.

An *affine family of finite elements* is defined as a family of finite elements for which all its finite elements are affine-equivalent to a single finite element.

Example 3.2.1. A linear finite element in two dimensions is (K, P_K, Σ_K) where K is a triangular element with vertices a_i, $i = 1, 2, 3$; P_K consists of all linear functions defined on K, i.e.,

$$P_K = \{p \,|\, p = ax + by + c, \quad \forall a, b, c \in \Re\};$$

and Σ_K is defined as $\Sigma_K = \{p(a_i),\ i = 1, 2, 3\}$, i.e., each function in P_K is uniquely determined by its values at the vertices of K.

3.3 Element-wise estimate on interpolation error

The following theorem is a result in the interpolation theory in Sobolev spaces. The reader is referred to [35] for its proof.

Theorem 3.3.1 *Let $(\hat{K}, \hat{P}, \hat{\Sigma})$ be a finite element, where $\hat{K}$ is the reference element, $\hat{P}$ is a finite-dimensional linear space of functions defined on $\hat{K}$, and $\hat{\Sigma}$ is a set of degrees of freedom. Let s be the greatest order of partial derivatives occurring in $\hat{\Sigma}$. For some integers m, k, and l: $0 \le m \le l \le k+1$, and some numbers p, $q \in [1, \infty]$, if*

$$W^{l,p}(\hat{K}) \hookrightarrow C^s(\hat{K}), \tag{3.1}$$

$$W^{l,p}(\hat{K}) \hookrightarrow W^{m,q}(\hat{K}), \tag{3.2}$$

$$P_k(\hat{K}) \subset \hat{P} \subset W^{m,q}(\hat{K}), \tag{3.3}$$

where $P_k(\hat{K})$ is the space of polynomials of degree no more than k, then there exists a constant $C = C(\hat{K}, \hat{P}, \hat{\Sigma})$ such that, for all affine-equivalent finite elements (K, P_K, Σ_K),

$$|v - \Pi_{k,K} v|_{W^{m,q}(K)} \le C \|(F_K^{'})^{-1}\|^m \cdot |det(F_K^{'})|^{\frac{1}{q}} \cdot |\hat{v}|_{W^{l,p}(\hat{K})} \qquad \forall v \in W^{l,p}(K), \tag{3.4}$$

where $\Pi_{k,K} : W^{l,p}(K) \to P_K$ denotes the P_K-interpolation operator on K, $\hat{v} = v \circ F_K$ is the composite function defined on $\hat{K}$, and $\| \cdot \|$ denotes the l_2 matrix norm.

The theorem contains six parameters m, k, l, p, and q. They are summarized in Table 3.1.

Table 3.1: The parameters contained in Theorem 3.3.1

Parameter	Range	Physical meaning
k	Integer, $k \ge 0$	Degree of interpolating polynomial, $P_k \subset P_K$.
l	Integer, $0 \le l \le k+1$	Regularity of interpolated functions, $v \in W^{l,p}(K)$.
m	Integer, $0 \le m \le l$	Order of derivatives of error measured, $e \in W^{m,q}(K)$.
p	Real, $1 \le p \le \infty$	Regularity of interpolated functions, $v \in W^{l,p}(K)$.
q	Real, $1 \le q \le \infty$	Used in the norm of the error, $e \in W^{m,q}(K)$.

One may notice that the error bound in (3.4) is given in derivatives on $\hat{K}$. This is crucial to the study of anisotropic meshes since it allows to develop error bounds coupling mesh properties with solution derivatives on K. Also, (3.4) is not optimal when $m \ge 1$, but it greatly simplifies the discussion since there is no need to introduce conditions like the maximum angle condition.

It is instructive to spell out the conditions (3.1) – (3.3). By the Sobolev imbedding theorem [1], we have

$$\begin{array}{lll} \left\{ \begin{array}{ll} l > \frac{n}{p} + s & \text{for } p > 1 \\ l \ge n + s & \text{for } p = 1 \end{array} \right. & \Longrightarrow & W^{l,p}(\hat{K}) \hookrightarrow C^s(\hat{K}) \\ & & \\ \left\{ \begin{array}{ll} l \ge m & \text{for } p \ge q \\ l < \frac{n}{p} + m & \text{for } \frac{1}{q} = \frac{1}{p} - \frac{l-m}{n} \\ l = \frac{n}{p} + m & \text{for } 1 \le q < \infty \end{array} \right. & \Longrightarrow & W^{l,p}(\hat{K}) \hookrightarrow W^{m,q}(\hat{K}), \end{array} \tag{3.5}$$

where n is the dimenion of $\hat{K}$. Regarding (3.3), it is noted that $\hat{P}$ is often chosen as $P_k(\hat{K})$. If this is the case, condition (3.3) places no constraints on the parameters m, k, l, p, and q.

Example 3.3.1. Consider the widely used Lagrange interpolation ($s = 0$) with $p = q = 2$. Condition (3.5) becomes $0 \le m \le l \le k + 1$ and $l > n/2$. Thus, (3.4) holds for functions in $H^1(\hat{K}) \equiv W^{1,2}(\hat{K})$ in one dimension and $H^2(\hat{K}) \equiv W^{2,2}(\hat{K})$ in two and three dimensions.

Hereafter, we assume that

$$\boxed{\textbf{H3.}\ \text{Parameters } m,\ k,\ l,\ p, \text{ and } q \text{ have been chosen such that the result of Theorem 3.3.1 holds.}} \tag{3.6}$$

Chapter 4

Isotropic error estimates

4.1 Introduction

The goal is to derive concrete element-wise bounds on interpolation error using Theorem 3.3.1 of Chapter 3. The key is to estimate $|\hat{v}|_{W^{l,p}(\hat{K})}$ in (3.4) using the physical derivatives of v on the element K. This can be done in either the *isotropic* approach or the *anisotropic* approach. In the isotropic approach, the shape of elements is separated from the physical derivatives of v. On the contrary, the shape and orientation of elements are coupled with the physical derivatives of v in the anisotropic approach. Error estimates obtained using these approaches are referred to as isotropic and anisotropic error estimates, respectively.

Generally speaking, *isotropic meshes are associated with isotropic error estimates while anisotropic meshes are associated with anisotropic error estimates*; see Chapters 7 and 8. Isotropic error estimation has the advantage of simplicity whereas anisotropic error estimation takes the maximal benefit of mesh adaptation by allowing elements to adjust their shape and orientation to fit the geometry of the physical solution. Examples of isotropic meshes include uniform and regular meshes, and an example of anisotropic meshes is the well-known Shishkin-type mesh.

Traditionally error estimates have been derived under the assumption that the mesh is regular, quasi-uniform, or uniform; e.g., see [35]. Thus, traditional results are isotropic.

This chapter is devoted to isotropic error estimation, and anisotropic error estimation will be discussed in the next chapter.

4.2 Chain rule

The basic tools in the estimation are the coordinate transformation and the chain rule. To explain this, denote the physical (on K) and computational (on $\hat{K}$) coordinates by $x = (x_1, ..., x_n)^T$ and $\xi = (\xi_1, ..., \xi_n)^T$, respectively. In these coordinate systems, the affine mapping $F_K : \hat{K} \to K$ can be expressed as

$$x = x(\xi) := F_K(\xi), \quad \forall \xi \in \hat{K}. \tag{4.1}$$

The Jacobian matrix

$$F_K' = \frac{\partial x}{\partial \xi} = \frac{\partial(x_1, ..., x_n)}{\partial(\xi_1, ..., \xi_n)}$$

is constant on K and piecewise constant on the whole domain Ω. Affine mapping F_K can also be written in terms of $F_K^{'}$ as

$$x = F_K^{'}\xi + c, \quad \forall \xi \in \hat{K} \tag{4.2}$$

for some vector c.

Through the affine mapping, the length scales of K along the coordinate directions can be expressed as

$$h_{i,K} = \left(\sum_j \left| \frac{\partial x_i}{\partial \xi_j} \right|^2 \right)^{\frac{1}{2}}, \quad i = 1, ..., n, \tag{4.3}$$

where the sum is over the range $j = 1$ to n. Let

$$D^{(i_1,\ldots,i_l)}v = \frac{\partial^l v}{\partial x_{i_1} \cdots \partial x_{i_l}}, \qquad \hat{D}^{(i_1,\ldots,i_l)}v = \frac{\partial^l v}{\partial \xi_{i_1} \cdots \partial \xi_{i_l}}.$$

By changing integration variables it follows

$$\begin{aligned} |\hat{v}|^p_{W^{l,p}(\hat{K})} &\equiv \int_{\hat{K}} \sum_{i_1,\ldots,i_l} \left| \hat{D}^{(i_1,\ldots,i_l)}v \right|^p d\xi \\ &= \left| \det(F_K^{'}) \right|^{-1} \int_K \sum_{i_1,\ldots,i_l} \left| \hat{D}^{(i_1,\ldots,i_l)}v \right|^p dx. \end{aligned}$$

By the chain-rule, we get, for a given integer t, $0 \le t \le l$,

$$\begin{aligned} &\sum_{i_1,\ldots,i_l} |\hat{D}^{(i_1,\ldots,i_l)}\hat{v}|^p \\ &\quad = \sum_{i_2,\ldots,i_l} \sum_{i_1} \left| \sum_{j_1} \frac{\partial x_{j_1}}{\partial \xi_{i_1}} \hat{D}^{(i_2,\ldots,i_l)} D^{(j_1)} v \right|^p \\ &\quad \le C \sum_{j_1} h^p_{j_1,K} \sum_{i_2,\ldots,i_l} |\hat{D}^{(i_2,\ldots,i_l)} D^{(j_1)} v|^p \\ &\quad \le \cdots \qquad \text{(repeat it } t \text{ times)} \\ &\quad \le C \sum_{j_1,\ldots,j_{l-t}} h^p_{j_1,K} \cdots h^p_{j_{l-t},K} \sum_{i_{l-t+1},\ldots,i_l} |\hat{D}^{(i_{l-t+1},\ldots,i_l)} D^{(j_1,\ldots,j_{l-t})} v|^p, \end{aligned}$$

where the equivalence of vector norms (particularly between l_2 and l_p norms) has been used and C denotes the generic constant which may take different values at different occurrences. Thus,

$$\begin{aligned} |\hat{v}|^p_{W^{l,p}(\hat{K})} \le\ & C \left| \det(F_K^{'}) \right|^{-1} \\ & \times \sum_{j_1,\ldots,j_{l-t}} h^p_{j_1,K} \cdots h^p_{j_{l-t},K} \sum_{i_{l-t+1},\ldots,i_l} \int_K |\hat{D}^{(i_{l-t+1},\ldots,i_l)} D^{(j_1,\ldots,j_{l-t})} v|^p dx. \end{aligned} \tag{4.4}$$

This result will be used for both isotropic and anisotropic error estimation.

4.3 Isotropic error estimation on a general mesh

Taking $t = 0$ in (4.4), one gets

$$\begin{aligned}
|\hat{v}|^p_{W^{l,p}(\hat{K})} &\leq C \left|\det(F_K^{'})\right|^{-1} \sum_{j_1,\ldots,j_l} h^p_{j_1,K} \cdots h^p_{j_l,K} \int_K |D^{(j_1,\ldots,j_l)} v|^p dx \\
&\leq C \left|\det(F_K^{'})\right|^{-1} \|F_K^{'}\|^{pl} \sum_{j_1,\ldots,j_l} \int_K |D^{(j_1,\ldots,j_l)} v|^p dx \\
&= C \left|\det(F_K^{'})\right|^{-1} \|F_K^{'}\|^{pl} \cdot |v|^p_{W^{l,p}(K)}.
\end{aligned}$$

Inserting this into the bound (3.4) yields

$$|v - \Pi_{k,K} v|_{W^{m,q}(K)} \leq C \|(F_K^{'})^{-1}\|^m \cdot \|F_K^{'}\|^l \cdot |\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \cdot |v|_{W^{l,p}(K)}, \qquad \forall v \in W^{l,p}(K). \quad (4.5)$$

One may notice that the physical derivative term, $|v|_{W^{l,p}(K)}$, is not directly coupled with the Jacobian matrix, $F_K^{'}$. This feature makes the shape and orientation of element K independent from the solution behavior. For this reason, the bound (4.5) is referred to as an isotropic error bound.

4.4 Error bound on regular triangulations

It is instructive to see what the bound (4.5) looks like on regular triangulations, particularly uniform meshes. To this end, we first estimate the norm and determinant of $F_K^{'}$ in the following lemma.

Lemma 4.4.1 *The Jacobian matrix, $F_K^{'}$, of the affine mapping F_K between two simplicial elements $\hat{K}$ and K has the properties*

$$\|F_K^{'}\| \leq \frac{h_K}{\rho_{\hat{K}}}, \quad \|(F_K^{'})^{-1}\| \leq \frac{h_{\hat{K}}}{\rho_K}, \quad |\det(F_K^{'})| = \frac{|K|}{|\hat{K}|}, \quad (4.6)$$

where h_K and ρ_K are the diameter and in-diameter of K, respectively, and $h_{\hat{K}}$ and $\rho_{\hat{K}}$ are the corresponding quantities for $\hat{K}$.

Proof. Let $S(\xi_c, r)$ be the sphere of the biggest inscribed ball of $\hat{K}$ centered at ξ_c and with radius r. Obviously, the diameter of $S(\xi_c, r)$ is $\rho_{\hat{K}}$ and thus $\rho_{\hat{K}} = 2r$. For any point ξ on $S(\xi_c, r)$, denote by $\tilde{\xi}$ the conjugate point which is defined as the intersection of the sphere and the straight line passing through ξ_c and ξ. By definition, $\|\xi - \tilde{\xi}\| = \rho_{\hat{K}}$. It follows

$$\begin{aligned}
\|F_K^{'}\| &= \sup_{\xi \neq 0} \frac{\|F_K^{'}\xi\|}{\|\xi\|} \\
&= \sup_{\xi \in S(\xi_c,r)} \frac{\|F_K^{'}(\xi - \tilde{\xi})\|}{\|\xi - \tilde{\xi}\|} \\
&= \frac{1}{\rho_{\hat{K}}} \sup_{\xi \in S(\xi_c,r)} \|F_K^{'}\xi - F_K^{'}\tilde{\xi}\|. \quad (4.7)
\end{aligned}$$

Since both $F_K^{'}\xi$ and $F_K^{'}\tilde{\xi}$ are on K, $\|F_K^{'}\xi - F_K^{'}\tilde{\xi}\| \leq h_K$. Inserting this into (4.7) gives the first inequality in (4.6).

The second inequality can be obtained by interchanging the roles of $\hat{K}$ and K.

The third inequality in (4.6) comes from the change of variables in integration,

$$|K| = \int_K dx = \int_{\hat{K}} |\det(F_K^{'})| d\xi = |\det(F_K^{'})| \int_{\hat{K}} d\xi = |\det(F_K^{'})| \cdot |\hat{K}|.$$

□

Using this lemma and the assumption $|\hat{K}| = 1$ (cf. the hypothesis $H2$ (1.6)), it follows from (4.5) that for any element K in a regular, affine triangulation,

$$|v - \Pi_{k,K} v|_{W^{m,q}(K)} \leq C h_K^{l-m+\frac{n}{q}-\frac{n}{p}} \cdot |v|_{W^{l,p}(K)}, \qquad \forall v \in W^{l,p}(K) \tag{4.8}$$

where $|K| = O(h_K^n)$ has been used.

If it is further assumed that $\mathcal{T}_h$ is uniform, then $h = h_K$ for all $K \in \mathcal{T}_h$ and

$$\begin{aligned} |v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} &:= \left(\sum_{K \in \mathcal{T}_h} |v - \Pi_{k,K} v|^q_{W^{m,q}(\Omega)} \right)^{\frac{1}{q}} \\ &\leq C h^{l-m+\frac{n}{q}-\frac{n}{p}} \left(\sum_{K \in \mathcal{T}_h} |v|^q_{W^{l,p}(K)} \right)^{\frac{1}{q}} \end{aligned}$$

From the arithmetic-mean and geometric-mean inequality (cf. Theorem 1.4.1), for $p \geq q$ one gets

$$\left(\sum_{K \in \mathcal{T}_h} |v|^q_{W^{l,p}(K)} \right)^{\frac{1}{q}} \leq N^{\frac{1}{q}-\frac{1}{p}} \left(\sum_{K \in \mathcal{T}_h} |v|^p_{W^{l,p}(K)} \right)^{\frac{1}{p}} \leq C h^{\frac{n}{p}-\frac{n}{q}} \left(\sum_{K \in \mathcal{T}_h} |v|^p_{W^{l,p}(K)} \right)^{\frac{1}{p}}.$$

On the other hand, when $p \leq q$, Jensen's inequality (cf. Theorem 1.4.2) leads to

$$\left(\sum_{K \in \mathcal{T}_h} |v|^q_{W^{l,p}(K)} \right)^{\frac{1}{q}} \leq \left(\sum_{K \in \mathcal{T}_h} |v|^p_{W^{l,p}(K)} \right)^{\frac{1}{p}}.$$

Combining these results, it arrives

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \leq C h^{l-m-\max\{0, \frac{n}{p}-\frac{n}{q}\}} |v|_{W^{l,p}(\Omega)}, \qquad \forall v \in W^{l,p}(\Omega). \tag{4.9}$$

Particularly, when $p \geq q$,

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \leq C h^{l-m} |v|_{W^{l,p}(\Omega)}, \qquad \forall v \in W^{l,p}(\Omega) \tag{4.10}$$

which is a classic result and can be found in standard textbooks.

Chapter 5

Anisotropic error estimates

5.1 Introduction

The goal of this chapter is to obtain an anisotropic error bound where the physical derivatives are directly coupled with the size, shape, and orientation of mesh elements.

Generally speaking, anisotropic error estimation is more difficult and more complicated than isotropic error estimation since the former has to take consideration of directional changes of the solution. The benefit of so doing is a lower and oftentimes much lower error bound, especially when the physical solution exhibits an anisotropic feature that the solution changes more significantly in one direction than the others.

The results in this chapter have been first presented in a recent work [61]. Biographic notes on mathematical studies of anisotropic meshes are given in §5.4.

5.2 An anisotropic error bound

A general anisotropic error bound can be obtained simply by inserting (4.4) (taking $t = 0$) into (3.4),

$$\begin{aligned} |v - \Pi_{k,K} v|_{W^{m,q}(K)} &\leq C \|(F_K^{'})^{-1}\|^m |\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \\ &\quad \times \sum_{i_1,\ldots,i_l} h_{i_1,K} \cdots h_{i_l,K} \int_K \left| D^{(i_1,\ldots,i_l)} v \right|^p dx. \end{aligned} \tag{5.1}$$

It is instructive to see that for $l = 1$, (5.1) reduces to

$$|v - \Pi_{k,K} v|_{W^{m,q}(K)} \leq C \|(F_K^{'})^{-1}\|^m |\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \sum_i h_{i,K} \int_K \left| \frac{\partial v}{\partial x_i} \right|^p dx. \tag{5.2}$$

The bound (5.1) is anisotropic since it allows separate control of the length scales of K based on the solution derivatives in the corresponding coordinate directions. For example, (5.2) shows that $h_{i,K}$ can be chosen according to the magnitude of derivative $(\partial v)/(\partial x_i)$.

It is interesting to remark that an anisotropic error estimate similar to (5.1) has been developed by Apel and Dobrowolski [6] and Apel [4] under the maximal angle condition and the so-called coordinate condition. It reads as

$$\begin{aligned}|v-\Pi_{k,K}v|_{W^{m,q}(K)} \quad \leq \quad & C|\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \\ & \times \sum_{i_1,\ldots,i_{l-m}} h_{i_1,K}\cdots h_{i_{l-m},K} \left| D^{(i_1,\ldots,i_{l-m})}v \right|_{W^{m,p}(K)}. \end{aligned} \tag{5.3}$$

For the purpose of comparison, (5.1) and (5.3) are written as

$$\begin{aligned}|v-\Pi_{k,K}v|_{W^{m,q}(K)} \leq C|\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \sum_{i_1,\ldots,i_{l-m}} h_{i_1,K}\cdots h_{i_{l-m},K} \\ \times \sum_{i_{l-m+1},\ldots,i_l} \|(F_K^{'})^{-1}\|^m h_{i_{l-m+1},K}\cdots h_{i_l,K} \int_K \left| D^{(i_{l-m+1},\ldots,i_l)} D^{(i_1,\ldots,i_{l-m})}v\right|^p dx. \end{aligned} \tag{5.4}$$

and

$$\begin{aligned}|v-\Pi_{k,K}v|_{W^{m,q}(K)} \leq C|\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \sum_{i_1,\ldots,i_{l-m}} h_{i_1,K}\cdots h_{i_{l-m},K} \\ \times \sum_{i_{l-m+1},\ldots,i_l} \int_K \left| D^{(i_{l-m+1},\ldots,i_l)} D^{(i_1,\ldots,i_{l-m})}v\right|^p dx. \end{aligned} \tag{5.5}$$

One can see that if $\|(F_K^{'})^{-1}\|^m h_{i_{l-m+1},K}\cdots h_{i_l,K} = 1$, (5.4) reduces to (5.5). But generally speaking, $\|(F_K^{'})^{-1}\|^m h_{i_{l-m+1},K}\cdots h_{i_l,K} \geq 1$. Thus, the bound given in (5.1) is generally larger than that in (5.3). It should be pointed out, though, that the latter requires the maximal angle and coordinate conditions whereas the former requires no such a priori conditions on the mesh.

5.3 Anisotropic error estimates independent of coordinate system

The bound given in (5.1) is anisotropic and holds on a general simplicial element. But its coupling of the length scales of K with the directional derivatives of v in the coordinate directions is dependent on the coordinate system. This dependence makes the bound hard to use in mesh generation.

In this section an anisotropic error estimate independent of coordinate system is derived. The development is different from that in the previous section and considered for two separate cases $l = 1$ and $l \geq 2$.

5.3.1 Case $l = 1$

This case can happen for piecewise constant interpolation ($k = 0$) or a general k-th degree polynomial preserving interpolation but with functions having low regularity (i.e., functions in $W^{1,p}(\Omega)$ with $l < k+1$). For the current case, the conditions (3.5) require that $s = 0$ (where s is the maximal order of derivatives appearing $\Sigma_{\hat{K}}$), $0 \leq m \leq 1$, $q \leq p$, and $p \geq 1$ for $n = 1$ and $p > n$ for $n \geq 2$.

Once again the basic tool is the chain-rule. By it, we have

$$\begin{aligned}
\sum_i \left|\frac{\partial \hat{v}}{\partial \xi_i}\right|^p &= \sum_i \left|\sum_j \frac{\partial x_j}{\partial \xi_i}\frac{\partial \hat{v}}{\partial x_j}\right|^p \\
&= \sum_i \left|(F_K^{'} e_i)^T \nabla v\right|^p \\
&\le C\left(\sum_i \left|(F_K^{'} e_i)^T \nabla v\right|^2\right)^{\frac{p}{2}} \\
&= C\left[\mathrm{tr}\left((F_K^{'})^T \nabla v \nabla v^T F_K^{'}\right)\right]^{\frac{p}{2}},
\end{aligned}$$

where e_i is the i-th unit vector of $\Re^n$. Combining this result with (3.4) (taking $l = 1$) gives

$$|v - \Pi_{k,K} v|_{W^{m,q}(K)} \le C\|(F_K^{'})^{-1}\|^m \cdot |\det(F_K^{'})|^{\frac{1}{q}-\frac{1}{p}} \left[\int_K \left[\mathrm{tr}\left((F_K^{'})^T \nabla v \nabla v^T F_K^{'}\right)\right]^{\frac{p}{2}} dx\right]^{\frac{1}{p}}. \tag{5.6}$$

5.3.2 Case $l \ge 2$

For $l \ge 2$, taking $t = 2$ in (4.4) and using the chain-rule, one obtains

$$\begin{aligned}
&\sum_{i_1,\cdots,i_l} |\hat{D}^{(i_1,\cdots,i_l)} \hat{v}|^p \\
&\le C \sum_{j_1,\cdots,j_{l-2}} h_{j_1,K}^p \cdots h_{j_{l-2},K}^p \sum_{i_{l-1},i_l} \left| \sum_{j_{l-1},j_l} \frac{\partial x_{j_{l-1}}}{\partial \xi_{i_{l-1}}} \frac{\partial x_{j_l}}{\partial \xi_{i_l}} \frac{\partial^2 \left(D^{(j_1,\cdots,j_{l-2})} v\right)}{\partial x_{j_{l-1}} \partial x_{j_l}} \right|^p \\
&= C \sum_{j_1,\cdots,j_{l-2}} h_{j_1,K}^p \cdots h_{j_{l-2},K}^p \sum_{i_{l-1},i_l} \left| (F_K^{'} \boldsymbol{e}_{i_{l-1}})^T H(D^{(j_1,\cdots,j_{l-2})} v)(F_K^{'} \boldsymbol{e}_{i_l}) \right|^p,
\end{aligned} \tag{5.7}$$

where $H(D^{(j_1,\cdots,j_{l-2})} v)$ denotes the Hessian of function $D^{(j_1,\cdots,j_{l-2})} v$. Denote the eigen-decomposition of matrix $H(D^{(j_1,\cdots,j_{l-2})} v)$ by

$$H(D^{(j_1,\cdots,j_{l-2})} v)| = Q\,\mathrm{diag}(\lambda_1, \ldots, \lambda_n) Q^T,$$

where Q is the orthogonal matrix consisting of the (normalized) eigenvectors and the λ_i's are the eigenvalues. Define

$$|H(D^{(j_1,\cdots,j_{l-2})} v)| = Q\,\mathrm{diag}(|\lambda_1|, \ldots, |\lambda_n|) Q^T. \tag{5.8}$$

Lemma 5.3.1

$$|a^T H b| \le \frac{1}{2}(a^T |H| a + b^T |H| b) \qquad \forall a,\, b \in \Re^n. \tag{5.9}$$

Proof. Let $\Sigma = \mathrm{diag}(|\lambda_1|, \ldots, |\lambda_n|)$. Decompose Σ into

$$\Sigma = \Sigma_+ - \Sigma_-,$$

where Σ_+ and Σ_- are diagonal matrices with non-negative diagonal entries. Apparently, $|H| = Q(\Sigma_+ + \Sigma_-)Q^T$. Then, (5.9) follows from

$$\begin{aligned}
|a^T H b| &= |(Q^T a)^T \Sigma (Q^T b)| \\
&\leq |(Q^T a)^T \Sigma_+ (Q^T b)| + |(Q^T a)^T \Sigma_- (Q^T b)| \\
&= |(\Sigma_+^{\frac{1}{2}} Q^T a)^T (\Sigma_+^{\frac{1}{2}} Q^T b)| + |(\Sigma_-^{\frac{1}{2}} Q^T a)^T (\Sigma_-^{\frac{1}{2}} Q^T b)| \\
&\leq \frac{1}{2} \left(\|\Sigma_+^{\frac{1}{2}} Q^T a\|^2 + \|\Sigma_+^{\frac{1}{2}} Q^T b\|^2 + \|\Sigma_-^{\frac{1}{2}} Q^T a\|^2 + \|\Sigma_-^{\frac{1}{2}} Q^T b\|^2 \right) \\
&= \frac{1}{2} \left(a^T Q \Sigma_+ Q^T a + b^T Q \Sigma_+ Q^T b + a^T Q \Sigma_- Q^T a + b^T Q \Sigma_- Q^T b \right) \\
&= \frac{1}{2} \left(a^T |H| a + b^T |H| b \right). \qquad (5.10)
\end{aligned}$$

□

Using this lemma, (5.7) becomes

$$\begin{aligned}
\sum_{i_1,\ldots,i_l} |\hat{D}^{(i_1,\ldots,i_l)} \hat{v}|^p &\leq C \sum_{j_1,\ldots,j_{l-2}} h_{j_1,K}^p \cdots h_{j_{l-2},K}^p \sum_{i_{l-1},i_l} \Big[(F_K^{'} \boldsymbol{e}_{i_{l-1}})^T |H(D^{(j_1,\ldots,j_{l-2})} v)| (F_K^{'} \boldsymbol{e}_{i_{l-1}}) \\
&\qquad + (F_K^{'} \boldsymbol{e}_{i_l})^T |H(D^{(j_1,\ldots,j_{l-2})} v)| (F_K^{'} \boldsymbol{e}_{i_l}) \Big]^p \\
&\leq C \sum_{j_1,\ldots,j_{l-2}} h_{j_1,K}^p \cdots h_{j_{l-2},K}^p \sum_{i} \left((F_K^{'} \boldsymbol{e}_i)^T |H(D^{(j_1,\ldots,j_{l-2})} v)| (F_K^{'} \boldsymbol{e}_i) \right)^p \\
&\leq C \sum_{j_1,\ldots,j_{l-2}} h_{j_1,K}^p \cdots h_{j_{l-2},K}^p \left[\operatorname{tr} \left((F_K^{'})^T |H(D^{(j_1,\ldots,j_{l-2})} v)| F_K^{'} \right) \right]^p .
\end{aligned}$$

Combining this result with the error bound (3.4) and the fact that $h_{j_1,K} \cdots h_{j_{l-2},K} \leq C \|F_K^{'}\|^{l-2}$, we obtain

$$\begin{aligned}
|v - \Pi_{k,K} v|_{W^{m,q}(K)} &\leq C \|(F_K^{'})^{-1}\|^m \cdot \|F_K^{'}\|^{l-2} \cdot |\det(F_K^{'})|^{\frac{1}{q} - \frac{1}{p}} \\
&\qquad \times \left[\int_K \left[\operatorname{tr} \left((F_K^{'})^T |H(D^{l-2} v)| F_K^{'} \right) \right]^p dx \right]^{\frac{1}{p}}, \qquad (5.11)
\end{aligned}$$

where

$$|H(D^{l-2} v)| \equiv \sum_{i_1,\ldots,i_{l-2}} |H(D^{(i_1,\ldots,i_{l-2})} v)|. \qquad (5.12)$$

It is noted that the bounds in (5.6) and (5.11) are independent of the coordinate system because the terms such as gradient and Hessian of v and the norm, trace, and determinant of $F_K^{'}$ are all coordinate-independent. Moreover, the Jacobian matrix $F_K^{'}$ is directly coupled with the gradient or Hessian of function v in bounds (5.6) and (5.11). Thus, in mesh generation process the choice of the shape and orientation of K should also be determined by the gradient or Hessian of v.

5.4 Bibliographic notes

Mathematical studies of anisotropic meshes can be traced back to Synge [108], Zlámal [129], Babuška and Aziz [8], Jamet [70], and Barnhill and Gregory [12]. Research has been intensified in the last decade in developing strictly mathematically-based error estimates; e.g., see

[3, 4, 5, 6, 7, 34, 33, 37, 38, 43, 49, 53, 61, 62, 76, 78, 97, 100, 103, 104, 105]. For example, D'Azevedo [37] and D'Azevedo and Simpson [38] introduce the concept of optimal triangles defined through a mapping from the reference element to an arbitrary element and derive estimates of linear interpolation error and its gradient for quadratic functions. Apel and Dobrowolski [6] and Apel [4] obtain a general anisotropic estimate of interpolation error, (5.3), under the maximal angle and coordinate system conditions. A number of semi-a posteriori anisotropic error estimators are obtained by Siebert [104], Dobrowolski et al. [41], Kunert [78, 79], and Kunert and Verfeurth [80]. Particularly, Kunert [78, 79] and Kunert and Verfeurth [80] introduce the so-called matching functions to measure the correspondence of the mesh to the anisotropic feature of the physical solution. A priori and semi-a posteriori error bounds for linear elements are then obtained in terms of these matching functions. Formaggia and Perrotto [49] obtain estimates for the L^2 and H^1 interpolation error on linear finite elements in terms of the eigenvalues and eigenvectors of the Jacobian matrix of the affine mapping from the reference element to a generic element. Their results do not require any a priori condition and therefore are convenient to use within an automatic mesh adaptation procedure. Recently, Picasso [97] combines the results of Formaggia and Perrotto [49] with the Zienkiewicz-Zhu gradient recovery technique [127, 128] to obtain an a posteriori error indicator for numerical solution of elliptic and parabolic PDEs. Motivated by work [59, 68] on variational mesh generation, Huang [61] develops a general anisotropic estimate on interpolation error that has been presented in this chapter. More recently, Chen et al. [33] show that some anisotropic error estimates formally obtained in [68] are optimal.

Chapter 6

Mesh quality measures and monitor functions

6.1 Introduction

Mesh quality measures and monitor functions are defined in this chapter. The mesh quality measures include three element-wise measures on geometry, alignment, and equidistribution and an overall quality measure. The quality of a mesh is assessed using the overall quality measure and a measure on the roughness of a function.

There are several reasons why assessment of an existing mesh should be studied. First, it is always useful to know the aspect ratio and size of mesh elements as well as how well they are aligned with the physical solution in the numerical solution of PDEs. Second, in view of mesh adaptation, particularly from the analysis of Chapter 2), it is important to know how closely the alignment and equidistribution conditions, (2.3) and (2.4), are satisfied by a mesh. As will be shown in §6.2, this requirement leads to the alignment and equidistribution measures. Third, many existing mesh adaptation algorithms produce meshes but without knowing their quality. Finally, as will demonstrated in later chapters, a better understanding of the correspondence of a mesh to the solution behavior will help design more robust and effective mesh adaptation algorithms.

As a matter of fact, mesh assessment has been extensively studied in the context of finite elements; see the review paper [5] and references therein. It should be pointed out that most of existing work is restricted to non-adaptive, isotropic meshes. Classic examples of mesh measures include Zlámal's minimal angle condition [129], Babuška and Aziz's maximal angle condition [8], and the aspect radio. Liu and Joe [86] study the shape quality measures for tetrahedron elements. Berzins [14] proposes a mesh quality indicator which takes into account both the shape of elements and the local solution behavior. Kunert [77] uses the so-called matching function to measure the correspondence of a mesh to the anisotropic feature of the solution.

The development in this chapter follows the approach used in [61] but there is some difference. In the current development for the case of anisotropic error estimation, the geometric quality measure is bounded by a term involving the alignment quality measure. As a result, the overall mesh quality measure (cf. (6.42) or (6.49)) does not involve the geometric quality measure. The advantage of this treatment is that the monitor function can now be defined naturally for anisotropic error estimates. The disadvantage is that the error estimates (cf. (6.43) and (6.50)) are slightly larger than those obtained in [61].

6.2 Mesh quality measures in view of mesh adaptation

Given a monitor function M, the analysis in Chapter 2 shows that the alignment and equidistribution conditions (2.3) and (2.4) define a precise control of the size, shape, and orientation of mesh elements. It is thus natural to define quantities to measure how closely they are satisfied. This results in the *alignment* and *equidistribution* measures.

Consider first the alignment condition (2.3). It is equivalent to

$$\mathrm{tr}\left(\boldsymbol{J}^T M \boldsymbol{J}\right) = n \, \det\left(\boldsymbol{J}^T M \boldsymbol{J}\right)^{\frac{1}{n}}. \tag{6.1}$$

Recall the arithmetic-geometric mean inequality (cf. Theorem 1.4.1) that this condition requires all the eigenvalues of $\boldsymbol{J}^T M \boldsymbol{J}$ to be equal to each other. The alignment measure can be defined as

$$Q_{ali}(x) = \left[\frac{\mathrm{tr}\left(\boldsymbol{J}^T M \boldsymbol{J}\right)}{n \, \det\left(\boldsymbol{J}^T M \boldsymbol{J}\right)^{\frac{1}{n}}} \right]^{\frac{n}{2(n-1)}}. \tag{6.2}$$

$Q_{ali}(x)$ measures how closely (6.1) is satisfied by a mesh. To explain this, denote the eigenvalues of $\boldsymbol{J}^T M \boldsymbol{J}$ by λ_i^2, $i = 1, ..., n$. Let $\lambda_{\max} = \max_i \lambda_i$ and $\lambda_{\min} = \min_i \lambda_i$. In terms of the eigenvalues, $Q_{ali}(x)$ can be expressed as

$$Q_{ali}(x) = \left[\frac{\sum_i \lambda_i^2}{n \left(\prod_i \lambda_i^2\right)^{\frac{1}{n}}} \right]^{\frac{n}{2(n-1)}}.$$

The arithmetic-geometric mean inequality implies that $Q_{ali}(x) \geq 1$. Also,

$$Q_{ali}(x) \leq \left[\frac{n \lambda_{\max}^2}{n \left(\lambda_{\max}^2 \lambda_{\min}^{2(n-1)}\right)^{\frac{1}{n}}} \right]^{\frac{n}{2(n-1)}} \leq \frac{\lambda_{\max}}{\lambda_{\min}}.$$

A refined version of the arithmetic-geometric mean inequality [75] reads as

$$\frac{1}{n(n-1)} \sum_{i<j} (\lambda_i - \lambda_j)^2 \leq \frac{1}{n} \sum_i \lambda_i^2 - \left(\prod_i \lambda_i^2 \right)^{\frac{1}{n}} \leq \frac{1}{n} \sum_{i<j} (\lambda_i - \lambda_j)^2. \tag{6.3}$$

Using the left inequality we have

$$\begin{aligned}
Q_{ali}^{\frac{2(n-1)}{n}}(x) - 1 &\geq \frac{1}{n(n-1)} \frac{\sum_{i<j} (\lambda_i - \lambda_j)^2}{\left(\prod_i \lambda_i^2\right)^{\frac{1}{n}}} \\
&\geq \frac{1}{n(n-1)} \frac{(\lambda_{\max} - \lambda_{\min})^2}{\lambda_{\min}^{\frac{2}{n}} \lambda_{\max}^{\frac{2(n-1)}{n}}} \\
&\geq \frac{1}{n(n-1)} \left[\left(\frac{\lambda_{\max}}{\lambda_{\min}} \right)^{\frac{1}{n}} - \left(\frac{\lambda_{\min}}{\lambda_{\max}} \right)^{\frac{n-1}{n}} \right]^2 \\
&\geq \frac{1}{n(n-1)} \left[\left(\frac{\lambda_{\max}}{\lambda_{\min}} \right)^{\frac{1}{n}} - 1 \right]^2
\end{aligned}$$

or

$$\frac{\lambda_{\max}}{\lambda_{\min}} \le \left[1+\sqrt{n(n-1)\left(Q_{ali}^{\frac{2(n-1)}{n}}(x)-1\right)}\right]^n .$$

Summarizing the above results we obtain

Theorem 6.2.1

$$1 \le Q_{ali}(x) \le \sqrt{\frac{\lambda_{\max}(\boldsymbol{J}^T M \boldsymbol{J})}{\lambda_{\min}(\boldsymbol{J}^T M \boldsymbol{J})}} \le \left[1+\sqrt{n(n-1)\left(Q_{ali}^{\frac{2(n-1)}{n}}(x)-1\right)}\right]^n , \tag{6.4}$$

where $\lambda_{\max}(\boldsymbol{J}^T M\boldsymbol{J})$ and $\lambda_{\min}(\boldsymbol{J}^T M\boldsymbol{J})$ denote the maximum and minimum eigenvalues of $\boldsymbol{J}^T M\boldsymbol{J}$.

Thus, $Q_{ali}(x) = 1$ if and only if $\lambda_{\max} = \lambda_{\min}$. The latter equality implies $\boldsymbol{J}^T M\boldsymbol{J} = \theta^{-1}(x)I$ or $\boldsymbol{J}^{-T}\boldsymbol{J}^{-1} = \theta(x)M(x)$ for some scalar function $\theta = \theta(x)$. From the analysis in Chapter 2, one can conclude that when $Q_{ali}(x) = 1$, the shape and orientation of mesh elements are completely controlled by $M(x)$. Inequality (6.4) also shows that the more largely the eigenvalues are different from each other, the larger the difference the ratio $\lambda_{\max}/\lambda_{\min}$ and therefore the quantity $Q_{ali}(x)$ are. In this sense, $Q_{ali}(x)$ indeed provides a measure on how well the mesh are aligned with the monitor function $M(x)$. It is useful to mention that $Q_{ali}(x)$ characterizes the shape of elements and has the range $[1,\infty)$.

The alignment quality measure can be also defined based on the inverse of the Jacobian matrix,

$$\hat{Q}_{ali}(x) = \left[\frac{\operatorname{tr}\left(\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}\right)}{n\ \det\left(\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}\right)^{\frac{1}{n}}}\right]^{\frac{n}{2(n-1)}} . \tag{6.5}$$

It is not difficult to show that $\hat{Q}_{ali}(x)$ has the same properties as $Q_{ali}(x)$ does. Particularly,

$$1 \le \hat{Q}_{ali}(x) \le \sqrt{\frac{\lambda_{\max}(\boldsymbol{J}^T M \boldsymbol{J})}{\lambda_{\min}(\boldsymbol{J}^T M \boldsymbol{J})}} \le \left[1+\sqrt{n(n-1)\left(\hat{Q}_{ali}^{\frac{2(n-1)}{n}}(x)-1\right)}\right]^n . \tag{6.6}$$

We now consider the equidistribution condition (2.4). Define the equidistribution measure as

$$Q_{eq}(x) = \frac{J\rho|\Omega_c|}{\sigma}, \tag{6.7}$$

where $\sigma = \int_\Omega \rho(x)dx$ and $\rho = \sqrt{\det(M)}$ is the adaptation function. $Q_{eq}(x)$ has a range $(0,\infty)$, with the average value being 1: $(1/|\Omega_c|)\int_{\Omega_c} Q_{eq}(x(\xi))d\xi = 1$. As a consequence, $\max_x Q_{eq}(x) = 1$ implies $Q_{eq}(x) \equiv 1$, $\forall x \in \Omega$, which in turn means that the equidistribution condition (2.4) holds exactly. Moreover, the farther $J\rho|\Omega_c|$ deviates from constant σ, the larger $\max_x Q_{eq}$ is. Hence, Q_{eq} measures how closely the equidistribution condition (2.4) is satisfied by the mesh. It is remarked that Q_{eq} characterizes the size of mesh elements.

In addition to alignment and equidistribution, it is also useful to know how skewed an element is. The aspect ratio is a natural measure. A measure easier to compute is the so-called geometric quality measures defined as

$$\begin{aligned}
Q_{geo}(x) &= \left[\frac{\mathrm{tr}\left(\boldsymbol{J}^T\boldsymbol{J}\right)}{n\ \det\left(\boldsymbol{J}^T\boldsymbol{J}\right)^{\frac{1}{n}}}\right]^{\frac{n}{2(n-1)}} = \left[\frac{\|\boldsymbol{J}\|_F}{\sqrt{n}\,|J|^{\frac{1}{n}}}\right]^{\frac{n}{(n-1)}}, && (6.8)\\
\hat{Q}_{geo}(x) &= \left[\frac{\mathrm{tr}\left(\boldsymbol{J}^{-1}\boldsymbol{J}^{-T}\right)}{n\ \det\left(\boldsymbol{J}^{-1}\boldsymbol{J}^{-T}\right)^{\frac{1}{n}}}\right]^{\frac{n}{2(n-1)}} = \left[\frac{\|\boldsymbol{J}^{-1}\|_F}{\sqrt{n}\,|J|^{-\frac{1}{n}}}\right]^{\frac{n}{(n-1)}}, && (6.9)
\end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius matrix norm. One may notice that Q_{geo} simply is Q_{ali} with $M = I$ and $\hat{Q}_{geo}$ is defined based on the inverse Jacobian matrix. Similar to Theorem 6.2.1, we have

Theorem 6.2.2

$$1 \le Q_{geo}(x) \le \sqrt{\frac{\lambda_{\max}(\boldsymbol{J}^T\boldsymbol{J})}{\lambda_{\min}(\boldsymbol{J}^T\boldsymbol{J})}} \le \left[1 + \sqrt{n(n-1)\left(Q_{geo}^{\frac{2(n-1)}{n}}(x) - 1\right)}\right]^n, \tag{6.10}$$

$$1 \le \hat{Q}_{geo}(x) \le \sqrt{\frac{\lambda_{\max}(\boldsymbol{J}^T\boldsymbol{J})}{\lambda_{\min}(\boldsymbol{J}^T\boldsymbol{J})}} \le \left[1 + \sqrt{n(n-1)\left(\hat{Q}_{geo}^{\frac{2(n-1)}{n}}(x) - 1\right)}\right]^n, \tag{6.11}$$

where $\lambda_{\max}(\boldsymbol{J}^T\boldsymbol{J})$ and $\lambda_{\min}(\boldsymbol{J}^T\boldsymbol{J})$ denote the maximum and minimum eigenvalues of $\boldsymbol{J}^T\boldsymbol{J}$.

Observing that $\sqrt{\lambda_{\max}(\boldsymbol{J}^T\boldsymbol{J})/\lambda_{\min}(\boldsymbol{J}^T\boldsymbol{J})}$ is actually the element aspect ratio (in the continuous form, cf. Chapter 2), one can conclude that Q_{geo} and $\hat{Q}_{geo}$ are equivalent to the aspect ratio. Q_{geo} and $\hat{Q}_{geo}$ have a range of $[1, \infty)$ and characterize the shape of mesh elements.

In the following we explore the relations between Q_{ali} and $\hat{Q}_{ali}$ with Q_{geo} and $\hat{Q}_{geo}$. Let the eigen-decomposition of M be

$$M = Q\mathrm{diag}(\lambda_1, ..., \lambda_n)Q^T,$$

where Q is an orthogonal matrix and $\lambda_i > 0$. Let

$$(Q^T\boldsymbol{J})^T = [a_1, ..., a_n],$$

where a_i's are column vectors. Then,

$$\begin{aligned}
\mathrm{tr}\left(\boldsymbol{J}^T M\boldsymbol{J}\right) &= \mathrm{tr}\left((Q^T\boldsymbol{J})^T\mathrm{diag}(\lambda_1, ..., \lambda_n)(Q^T\boldsymbol{J})\right)\\
&= \mathrm{tr}\left(\sum_i \lambda_i a_i a_i^T\right)\\
&= \sum_i \lambda_i\|a_i\|^2.
\end{aligned}$$

It follows

$$\begin{aligned}\operatorname{tr}\left(\boldsymbol{J}^T\boldsymbol{J}\right) &= \operatorname{tr}\left((Q^T\boldsymbol{J})^T(Q^T\boldsymbol{J})\right)\\ &= \sum_i \|a_i\|^2\\ &= \sum_i \lambda_i^{-1}\left(\lambda_i\|a_i\|^2\right)\\ &\le \left(\sum_i \lambda_i^{-1}\right)\cdot\left(\sum_i \lambda_i\|a_i\|^2\right)\end{aligned}$$

or

$$\operatorname{tr}\left(\boldsymbol{J}^T\boldsymbol{J}\right) \le \operatorname{tr}(M^{-1})\cdot\operatorname{tr}\left(\boldsymbol{J}^T M\boldsymbol{J}\right). \tag{6.12}$$

Thus,

$$\frac{\operatorname{tr}\left(\boldsymbol{J}^T\boldsymbol{J}\right)}{n|J|^{\frac{2}{n}}} \le \frac{\operatorname{tr}(M^{-1})}{\rho^{-\frac{2}{n}}}\cdot\frac{\operatorname{tr}\left(\boldsymbol{J}^T M\boldsymbol{J}\right)}{n|J|^{\frac{2}{n}}\rho^{\frac{2}{n}}}$$

or

$$Q_{geo}^{\frac{2(n-1)}{n}} \le Q_{ali}^{\frac{2(n-1)}{n}}\cdot\frac{\operatorname{tr}(M^{-1})}{\rho^{-\frac{2}{n}}}, \tag{6.13}$$

where we recall that $\rho = \sqrt{\det(M)}$. Similarly, one can get

$$\hat{Q}_{geo}^{\frac{2(n-1)}{n}} \le \hat{Q}_{ali}^{\frac{2(n-1)}{n}}\cdot\frac{\operatorname{tr}(M)}{\rho^{\frac{2}{n}}}. \tag{6.14}$$

Note that these inequalities are not sharp. But they are good enough for our purpose – to bound the geometry quality measures using the alignment measures.

6.3 The case with isotropic error estimation

The mesh quality measures have been considered in the previous section intuitively in view of mesh adaptation and for a given monitor function. They can be studied more concretely based on an error bound. This is done in this and next two sections based on the isotropic and anisotropic error estimates obtained in Chapters 4 and 5.

6.3.1 Mesh quality measures and monitor function

The development is based on the isotropic error bound (4.5). Taking q power on the both sides and summing overall the elements, one gets

$$\begin{aligned}|v-\Pi_k v|^q_{W^{m,q}(\Omega)} &\equiv \sum_K |v-\Pi_{k,K}v|^q_{W^{m,q}(K)}\\ &\le C\sum_K |K|\cdot\|(F_K^{'})^{-1}\|^{mq}\cdot\|F_K^{'}\|^{lq}\cdot\langle v\rangle^q_{W^{l,p}(K)}, \quad \forall v\in W^{l,p}(\Omega)\end{aligned} \tag{6.15}$$

where we have used $|K| = |\det(F_K^{'})|$ and

$$\langle v\rangle_{W^{l,p}(K)} = \left(\frac{1}{|K|}\int_K \sum_{i_1,\ldots,i_l}\left|D^{(i_1,\ldots,i_l)}v\right|^p dx\right)^{\frac{1}{p}} = \left(\frac{1}{|K|}\int_K \|D^l v\|^p_{l_p}dx\right)^{\frac{1}{p}}.$$

We use a continuous form to simplify the derivation. For this purpose, we assume that an affine, quasi-uniform mesh $\mathcal{T}_{c,h}$ can be defined on Ω such that it has the same connectivity as $\mathcal{T}_h$ does. When associated with this quasi-uniform mesh, Ω will be considered as the "computational" domain Ω_c, and the coordinate on it is denoted by ξ. Then, a piecewise linear, global coordinate transformation $x = x(\xi) : \Omega_c \to \Omega$ can be defined as

$$x(\xi) := F_K(F_{K_c}^{-1}(\xi)), \quad \forall \xi \in K_c, \forall K_c \in \mathcal{T}_{c,h}$$

where K and K_c are the corresponding elements on Ω and Ω_c, respectively, and $F_K : \hat{K} \to K$ and $F_{K_c} : \hat{K}_c \to K$ are linear mappings. The definition is illustrated in Fig. 6.1.

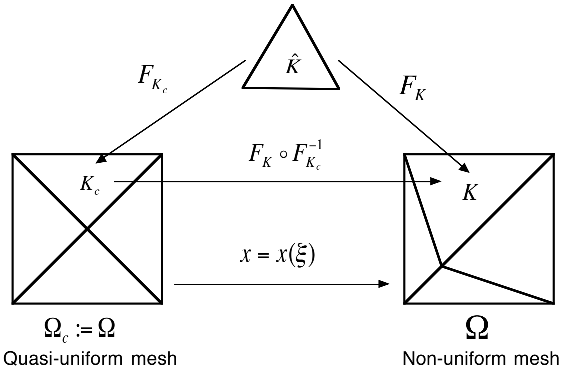


Figure 6.1: The definition of the piecewise linear coordinate transformation $x = x(\xi)$ is illustrated.

The local and global mappings can be connected as follows. First recall that

$$\boldsymbol{J} = \frac{\partial x}{\partial \xi}, \quad J = \det(\boldsymbol{J}),$$

and N is the number of the elements in $\mathcal{T}_h$. By assumption, $|\Omega_c| = O(1)$ and $|\hat{K}| = 1$. It is not difficult to get

$$F_K^{'} = \boldsymbol{J} \cdot F_{K_c}^{'}, \quad |K| \approx N^{-1} J, \quad \|F_K^{'}\| \approx N^{-\frac{1}{n}} \|\boldsymbol{J}\|. \tag{6.16}$$

Using these relations, from (6.15) we have

$$\begin{aligned} |v - \Pi_k v|^q_{W^{m,q}(\Omega)} &\leq CN^{-\frac{(l-m)q}{n}} \sum_K |K| \cdot \|\boldsymbol{J}^{-1}\|^{mq} \cdot \|\boldsymbol{J}\|^{lq} \cdot \left(\frac{1}{|K|} \int_K \|D^l v\|^p_{l_p} dx \right)^{\frac{q}{p}} \\ &\to CN^{-\frac{(l-m)q}{n}} \int_\Omega \|\boldsymbol{J}^{-1}\|^{mq} \cdot \|\boldsymbol{J}\|^{lq} \cdot \|D^l v\|^q_{l_p} dx, \end{aligned}$$

where the limit is taken as $\max_K \operatorname{diam}(K) \to 0$. Hereafter, for simplicity this asymptotical bound will be denoted by

$$|v - \Pi_k v|^q_{W^{m,q}(\Omega)} \overset{<}{\to} CN^{-\frac{(l-m)q}{n}} \int_\Omega \|\boldsymbol{J}^{-1}\|^{mq} \cdot \|\boldsymbol{J}\|^{lq} \cdot \|D^l v\|^q_{l_p} dx. \tag{6.17}$$

In the following development the factors $\|\boldsymbol{J}^{-1}\|$ and $\|\boldsymbol{J}\|$ are replaced with $\det(\boldsymbol{J})$ via the geometric quality measurea and then $\det(\boldsymbol{J})$ is replaced with $\|D^l v\|_{l_p}$ through the equidistribution measure. Specifically, from the geometric measures (6.8) and (6.9),

$$\begin{aligned}\|\boldsymbol{J}\| &\le \|\boldsymbol{J}\|_F \le \sqrt{n}\, Q_{geo}^{\frac{n-1}{n}} \cdot |J|^{\frac{1}{n}},\\ \|\boldsymbol{J}^{-1}\| &\le \|\boldsymbol{J}^{-1}\|_F \le \sqrt{n}\, \hat{Q}_{geo}^{\frac{n-1}{n}} \cdot |J|^{-\frac{1}{n}}.\end{aligned}$$

Inserting these into (6.17) yields

$$|v - \Pi_k v|^q_{W^{m,q}(\Omega)} \overset{\le}{\to} CN^{-\frac{(l-m)q}{n}} \int_\Omega \hat{Q}_{geo}^{\frac{mq(n-1)}{n}} \cdot Q_{geo}^{\frac{lq(n-1)}{n}} \cdot |J|^{\frac{(l-m)q}{n}} \cdot \|D^l v\|^q_{l_p} dx. \tag{6.18}$$

We are now in a position to define the monitor function. *The idea is to define M such that the right-hand side of (6.18) has a lowest bound attained on an M-specified mesh.* (An M-specified mesh is a mesh that satisfies the alignment and equidistribution conditions (2.3) and (2.4).) Let

$$B[\xi] = \int_\Omega \hat{Q}_{geo}^{\frac{mq(n-1)}{n}} \cdot Q_{geo}^{\frac{lq(n-1)}{n}} \cdot |J|^{\frac{(l-m)q}{n}} \cdot \|D^l v\|^q_{l_p} dx. \tag{6.19}$$

Since this bound involves only the geometric quality measures, it is natural to generate the mesh such that its elements are close to being equilateral. Thus, choose

$$M = \theta(x) I \tag{6.20}$$

for some scalar function $\theta = \theta(x)$. For a mesh specified by this monitor function, the alignment condition (2.3) implies that $Q_{geo}(x) \equiv 1$ and $\hat{Q}_{geo}(x) \equiv 1$. Thus,

$$B[\xi] = \int_\Omega |J|^{\frac{(l-m)q}{n}} \cdot \|D^l v\|^q_{l_p} dx. \tag{6.21}$$

To completely determine M, we need to define $\rho \equiv \sqrt{\det(M)}$. To this end, we prove the following theorem.

Theorem 6.3.1 (Optimality of equidistribution) *Given a real number $s > 0$ and a positive function $\rho = \rho(x)$ defined on Ω. Then the functional has a lower bound,*

$$I[\xi] \equiv \int_\Omega \rho(|J|\rho)^s dx \ge \sigma \left(\frac{\sigma}{|\Omega_c|}\right)^s, \tag{6.22}$$

where $\sigma = \int_\Omega \rho dx$, among all invertible coordinate transformation $\xi = \xi(x) : \Omega \to \Omega_c$. The functional attains its lower bound on an equidistributing mesh which satisfies $Q_{eq}(x) \equiv 1$ or the equidistribution condition (2.4).

Proof. Taking $t = -1$, $w = \rho/\sigma$ and $f = \rho$ in Theorem 1.2.1, it follows

$$\left(\int_\Omega (|J|\rho)^s \frac{\rho}{\sigma} dx\right)^{\frac{1}{s}} \ge \left(\int_\Omega (|J|\rho)^{-1} \frac{\rho}{\sigma} dx\right)^{-1} = \frac{\sigma}{|\Omega_c|},$$

which yields (6.22).

It is obvious that an equidistributing mesh for ρ gives the lower bound. □

This theorem states that the equidistributing mesh for ρ is an optimal mesh in the sense that it minimizes the functional $I[\xi]$. It can also be used to define the optimal function ρ. Indeed, by comparing $I[\xi]$ with $B[\xi]$ given in (6.21), one can get

$$\rho = \|D^l v\|_{l_p}^{\frac{nq}{n+(l-m)q}}. \tag{6.23}$$

With this choice, we have

$$M = \|D^l v\|_{l_p}^{\frac{2q}{n+(l-m)q}} I. \tag{6.24}$$

Unfortunately, the monitor function defined in (6.24) is not always positive definite since the term on the right-hand side can vanish locally. To avoid the difficulty, we regularize (6.18) with a positive parameter α_{iso} in such a way that

$$\begin{aligned}
&|v-\Pi_k v|^q_{W^{m,q}(\Omega)}\\
\overset{<}{\to}\quad & CN^{-\frac{(l-m)q}{n}} \int_\Omega \hat{Q}_{geo}^{\frac{mq(n-1)}{n}} \cdot Q_{geo}^{\frac{lq(n-1)}{n}} \cdot |J|^{\frac{(l-m)q}{n}} \cdot \left(\alpha + \|D^l v\|_{l_p}\right)^q dx\\
=\quad & CN^{-\frac{(l-m)q}{n}} \alpha_{iso}^q \int_\Omega \hat{Q}_{geo}^{\frac{mq(n-1)}{n}} \cdot Q_{geo}^{\frac{lq(n-1)}{n}} \cdot |J|^{\frac{(l-m)q}{n}} \cdot \left(1 + \frac{1}{\alpha_{iso}}\|D^l v\|_{l_p}\right)^q dx.
\end{aligned} \tag{6.25}$$

Following the same procedure, one obtains the adaptation and monitor functions as

$$\rho = \rho_{iso}(x) \equiv \left[1 + \frac{1}{\alpha_{iso}}\|D^l v\|_{l_p}\right]^{\frac{nq}{n+q(l-m)}}, \tag{6.26}$$

$$M = M_{iso}(x) \equiv \left[1 + \frac{1}{\alpha_{iso}}\|D^l v\|_{l_p}\right]^{\frac{2q}{n+q(l-m)}} I. \tag{6.27}$$

The parameter α_{iso} is often referred to as the intensity parameter in the context of mesh adaptation since it controls the intensity of mesh concentration. It is suggested in [58] that α_{iso} be chosen such that (i) the monitor function M_{iso} is invariant under the scaling transformation of v and (ii) $\sigma \equiv \int_\Omega \rho_{iso} dx \le C$ for some constant C. For the current situation,

$$\begin{aligned}
\sigma &\equiv \int_\Omega \rho_{iso}(x)dx\\
&\le C_1 \int_\Omega \left[1 + \alpha_{iso}^{-\frac{nq}{n+q(l-m)}} \|D^l v\|_{l_p}^{\frac{nq}{n+q(l-m)}}\right] dx\\
&= C_1 \left[|\Omega| + \alpha_{iso}^{-\frac{nq}{n+q(l-m)}} \int_\Omega \|D^l v\|_{l_p}^{\frac{nq}{n+q(l-m)}} dx\right].
\end{aligned}$$

Thus, by choosing

$$\alpha_{iso} = \left[\frac{1}{|\Omega|}\int_\Omega \|D^l v\|_{l_p}^{\frac{nq}{n+q(l-m)}} dx\right]^{\frac{n+q(l-m)}{nq}}, \tag{6.28}$$

we have $\sigma \le 2C_1|\Omega|$ and $\rho_{iso}(x)$ (and therefore $M(x)$) is invariant under the scaling transformation of v.

Theorem 6.3.1 and inequality (6.25) imply that the interpolation error has a bound on an M-specified mesh as

$$|v-\Pi_k v|_{W^{m,q}(\Omega)} \overset{<}{\to} CN^{-\frac{(l-m)}{n}} \alpha_{iso}$$

or

$$|v - \Pi_k v|_{W^{m,q}(\Omega)} \overset{<}{\to} C N^{-\frac{(l-m)}{n}} |v|_{W^{l,\frac{nq}{n+q(l-m)}}(\Omega)},$$

where $\sigma \le C$ has been used.

On a general mesh, from the definitions for Q_{eq} (6.7) and ρ (6.26) we can rewrite (6.25) into

$$|v - \Pi_k v|^q_{W^{m,q}(\Omega)} \overset{<}{\to} C N^{-\frac{(l-m)q}{n}} \alpha^q_{iso} \int_\Omega \hat{Q}_{geo}^{\frac{mq(n-1)}{n}} \cdot Q_{geo}^{\frac{lq(n-1)}{n}} \cdot Q_{eq}^{\frac{(l-m)q}{n}} \cdot \rho_{iso} dx, \tag{6.29}$$

where once again $\sigma \le C$ has been used. Define the overall mesh quality measure as

$$Q_{mesh,iso} = \left[\frac{1}{\sigma} \int_\Omega \left(\hat{Q}_{geo}^{\frac{m(n-1)}{n}} \cdot Q_{geo}^{\frac{l(n-1)}{n}} \cdot Q_{eq}^{\frac{(l-m)}{n}} \right)^q \rho_{iso} dx \right]^{\frac{1}{q}}, \tag{6.30}$$

which is the weighted (with weight function ρ_{iso}) L^q norm of $\hat{Q}_{geo}^{\frac{m(n-1)}{n}} Q_{geo}^{\frac{l(n-1)}{n}} Q_{eq}^{\frac{(l-m)}{n}}$. With this definition, we obtain an interpolation error bound as

$$|v - \Pi_k v|_{W^{m,q}(\Omega)} \overset{<}{\to} C\, N^{-\frac{(l-m)}{n}} Q_{mesh,iso}\, \alpha_{iso} \tag{6.31}$$

or

$$|v - \Pi_k v|_{W^{m,q}(\Omega)} \overset{<}{\to} C\, N^{-\frac{(l-m)}{n}} Q_{mesh,iso}\, |v|_{W^{l,\frac{nq}{n+q(l-m)}}(\Omega)}. \tag{6.32}$$

6.3.2 Mesh assessment

An existing adaptive mesh is assessed by comparing the interpolation error thereon to its counterpart on a uniform mesh with the same number of elements.

Recall that the interpolation error is bounded by

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \le C\, N^{-\frac{(l-m)}{n}} |v|_{W^{l,p}(\Omega)} \tag{6.33}$$

on a uniform mesh of N elements (cf. (4.10)). Let

$$Q_{soln,iso} = \frac{\langle v \rangle_{W^{l,p}(\Omega)}}{\langle v \rangle_{W^{l,\frac{nq}{n+q(l-m)}}(\Omega)}}. \tag{6.34}$$

Upon the assumption $q \le p$, we have $nq/(n + q(l-m)) \le p$ and therefore $Q_{soln,iso} \ge 1$. Theorem 1.2.1 implies that when $nq/(n + q(l-m)) < p$, $Q_{soln,iso} = 1$ if and only if $D^l v$ is constant. The rougher $D^l v$ is, the larger $Q_{soln,iso}$ is and the more difficult v is approximated numerically. Thus, $Q_{soln,iso}$ measures how rough v is. For this reason, $Q_{soln,iso}$ is referred to as the *roughness* measure.

The bound (6.50) can now be rewritten as

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \overset{<}{\to} C\, N^{-\frac{(l-m)}{n}} |v|_{W^{l,p}(\Omega)} \frac{Q_{mesh,iso}}{Q_{soln,iso}}. \tag{6.35}$$

Thus, the overall mesh quality should be considered good if $Q_{mesh,iso} \ll Q_{soln,iso}$ or the adaptive mesh leads to a much smaller error than that on a uniform mesh. On the other hand, when the solution is smooth (i.e., $Q_{soln,iso}$ is small), an adaptive mesh cannot be expected to have a significant improvement in accuracy over a uniform mesh. In that case, an adaptive mesh with $Q_{mesh,iso} = O(Q_{soln,iso}) = O(1)$ should be considered to have a good quality. To summarize, *a mesh has a good overall quality if $Q_{mesh,iso} = O(1)$ or $Q_{mesh,iso} \ll Q_{soln,iso}$.*

6.4 The case with anisotropic error estimation: $l = 1$

The mesh quality measures and monitor function for this case are developed based on the anisotropic error bound given in (5.6).

6.4.1 Mesh quality measures and monitor function

Taking q power on both sides of (5.6) and summing over all the elements, we have

$$\begin{aligned} |v - \Pi_k v|^q_{W^{m,q}(\Omega)} &\leq C \sum_K |K| \cdot \|(F_K^{'})^{-1}\|^{mq} \\ & \qquad \times \left[\frac{1}{|K|} \int_K \left[\operatorname{tr}\left((F_K^{'})^T \nabla v \nabla v^T F_K^{'} \right) \right]^{\frac{p}{2}} dx \right]^{\frac{q}{p}} . \end{aligned}$$

From (6.16), the above inequality can be written in a continuous form as

$$|v - \Pi_k v|^q_{W^{m,q}(\Omega)} \overset{<}{\to} C\, N^{-\frac{(1-m)q}{n}} \int_\Omega \|\boldsymbol{J}^{-1}\|^{mq} \left[\operatorname{tr}\left(\boldsymbol{J}^T \nabla v \nabla v^T \boldsymbol{J} \right) \right]^{\frac{q}{2}} dx.$$

The right-hand-side term is regularized with a positive constant $\alpha_{ani,1} > 0$ in a way that

$$|v - \Pi_k v|^q_{W^{m,q}(\Omega)} \overset{<}{\to} C\, \alpha^q_{ani,1}\, N^{-\frac{(1-m)q}{n}} \int_\Omega \|\boldsymbol{J}^{-1}\|^{mq} \left[\operatorname{tr}\left(\boldsymbol{J}^T \left[I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right] \boldsymbol{J} \right) \right]^{\frac{q}{2}} dx. \quad (6.36)$$

As for (6.12), it can be shown

$$\operatorname{tr}\left(\boldsymbol{J}^{-1} \boldsymbol{J}^{-T} \right) \leq \operatorname{tr}\left(\boldsymbol{J}^{-1} \left[I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right]^{-1} \boldsymbol{J}^{-T} \right) \cdot \operatorname{tr}\left(I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right).$$

Taking the monitor function into the form

$$M(x) = \theta(x) \left[I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right], \quad (6.37)$$

where $\theta = \theta(x)$ is a to-be-determined scalar function, from the definitions of the alignment measures we have

$$\begin{aligned} \operatorname{tr}\left(\boldsymbol{J}^T \left[I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right] \boldsymbol{J} \right) &= Q_{ali}^{\frac{2(n-1)}{n}}\, n\, |J|^{\frac{2}{n}} \det\left(I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right)^{\frac{1}{n}}, \\ \operatorname{tr}\left(\boldsymbol{J}^{-1} \left[I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right]^{-1} \boldsymbol{J}^{-T} \right) &= \hat{Q}_{ali}^{\frac{2(n-1)}{n}}\, n\, |J|^{-\frac{2}{n}} \det\left(I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right)^{-\frac{1}{n}}. \end{aligned}$$

Combining the above results, we get

$$\begin{aligned} |v - \Pi_k v|^q_{W^{m,q}(\Omega)} &\overset{<}{\to} C\, \alpha^q_{ani,1}\, N^{-\frac{(1-m)q}{n}} \int_\Omega dx\, \hat{Q}_{ali}^{\frac{mq(n-1)}{n}}\, Q_{ali}^{\frac{q(n-1)}{n}}\, |J|^{\frac{(1-m)q}{n}} \\ & \qquad \times \det\left(I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right)^{\frac{(1-m)q}{2n}} \left[\operatorname{tr}\left(I + \frac{1}{\alpha^2_{ani,1}} \nabla v \nabla v^T \right) \right]^{\frac{mq}{2}}. \quad (6.38) \end{aligned}$$

Notice that

$$\begin{aligned}
\det\left(I + \frac{1}{\alpha_{ani,1}^2}\nabla v \nabla v^T\right) &= 1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2, \\
\mathrm{tr}\left(I + \frac{1}{\alpha_{ani,1}^2}\nabla v \nabla v^T\right) &= 1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2.
\end{aligned}$$

Thus,

$$\begin{aligned}
|v - \Pi_k v|_{W^{m,q}(\Omega)}^q \; &\lesssim \; C\,\alpha_{ani,1}^q\, N^{-\frac{(1-m)q}{n}} \int_\Omega dx\, \hat{Q}_{ali}^{\frac{mq(n-1)}{n}}\; Q_{ali}^{\frac{q(n-1)}{n}}\; |J|^{\frac{(1-m)q}{n}} \\
&\quad \times \left(1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2\right)^{\frac{(1-m)q}{2n} + \frac{mq}{2}}.
\end{aligned}$$

From Theorem 6.3.1), the optimal adaptation function can be defined as

$$\rho = \rho_{ani,1}(x) \equiv \left(1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2\right)^{\frac{(1-m)q+nmq}{2(n+(1-m)q)}}, \tag{6.39}$$

and the monitor function is given by

$$M = M_{ani,1}(x) \equiv \left(1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2\right)^{\frac{mq-1}{n+(1-m)q}} \left[I + \frac{1}{\alpha_{ani,1}^2}\nabla v \nabla v^T\right]. \tag{6.40}$$

The intensity parameter $\alpha_{ani,1}$ can be defined using the same considerations for α_{iso} in the previous section, viz., scaling invariance and $\sigma = \int_\Omega \rho dx \le C$ for some constant C. This gives

$$\alpha_{ani,1} = \left[\frac{1}{|\Omega|}\int_\Omega \|\nabla v\|^{\frac{(1-m)q+nmq}{n+q(1-m)}}\right]^{\frac{n+q(1-m)}{(1-m)q+nmq}}. \tag{6.41}$$

Defining the overall mesh quality measure as

$$Q_{mesh,ani,1} = \left[\frac{1}{\sigma}\int_\Omega \left(\hat{Q}_{ali}^{\frac{m(n-1)}{n}}\; Q_{ali}^{\frac{(n-1)}{n}}\; Q_{eq}^{\frac{(1-m)}{n}}\right)^q \rho_{ani,1} dx\right]^{\frac{1}{q}}, \tag{6.42}$$

from (6.38) the interpolation error on a general mesh can be bounded as

$$|v - \Pi_k v|_{W^{m,q}(\Omega)} \lesssim C\, N^{-\frac{(1-m)}{n}}\, Q_{mesh,ani,1}\, |v|_{W^{1,\frac{(1-m)q+nmq}{n+q(1-m)}}(\Omega)}. \tag{6.43}$$

6.4.2 Mesh assessment

Once again, an existing adaptive mesh is assessed by comparing the interpolation error thereon to its counterpart on a uniform mesh of the same number of elements.

For the current situation $l = 1$, a bound of interpolation error on a uniform mesh is given in (6.33) with $l = 1$. From (6.33) and (6.43), the solution roughness can be defined as

$$Q_{soln,ani,1} = \frac{\langle v\rangle_{W^{1,p}(\Omega)}}{\langle v\rangle_{W^{1,\frac{(1-m)q+nmq}{n+q(1-m)}}(\Omega)}}. \tag{6.44}$$

From (6.43) the error bound reads as

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \stackrel{<}{\rightarrow} C\, N^{-\frac{(1-m)}{n}}\, |v|_{W^{1,p}(\Omega)}\, \frac{Q_{mesh,ani,1}}{Q_{soln,ani,1}}.$$

Thus, a mesh has a good overall quality when $Q_{mesh,ani,1} = O(1)$ or $Q_{mesh,ani,1} \ll Q_{soln,ani,1}$.

6.5 The case with anisotropic error estimation: $l \geq 2$

The procedure is the same as in the previous section but the development is now based on the bound (5.11).

6.5.1 Mesh quality measures

Taking q power on both sides of (5.11) and summing over all the elements gives

$$|v - \Pi_k v|^q_{W^{m,q}(\Omega)} \leq C \sum_K |K| \cdot \|(F_K^{'})^{-1}\|^{mq} \cdot \|F_K^{'}\|^{q(l-2)} \left\langle \mathrm{tr}\left((F_K^{'})^T |H(D^{l-2}v)| F_K^{'} \right) \right\rangle^q_{L^p(K)} .$$

Rewriting in a continuous form and regularizing the right-hand side term with $\alpha_{ani,2} > 0$, we get

$$\begin{aligned} |v - \Pi_k v|^q_{W^{m,q}(\Omega)} &\stackrel{<}{\rightarrow} C\, \alpha^q_{ani,2}\, N^{-\frac{(l-m)q}{n}} \\ &\quad \times \int_\Omega \|\boldsymbol{J}^{-1}\|^{mq} \cdot \|\boldsymbol{J}\|^{q(l-2)} \left[\mathrm{tr}\left(\boldsymbol{J}^T \left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right] \boldsymbol{J} \right) \right]^q dx. \end{aligned}$$

As for (6.16), we have

$$\begin{aligned} \mathrm{tr}\left(\boldsymbol{J}^{-1}\boldsymbol{J}^{-T}\right) &\leq \mathrm{tr}\left(\boldsymbol{J}^{-1} \left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right]^{-1} \boldsymbol{J}^{-T} \right) \cdot \mathrm{tr}\left(I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right), \\ \mathrm{tr}\left(\boldsymbol{J}^T\boldsymbol{J}\right) &\leq \mathrm{tr}\left(\boldsymbol{J}^T \left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right] \boldsymbol{J} \right) \cdot \mathrm{tr}\left(\left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right]^{-1} \right). \end{aligned}$$

Taking

$$M = \theta(x) \left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right],$$

where $\theta = \theta(x)$ is a scalar function to be determined, it follows from the definitions of the alignment measures that

$$\begin{aligned} \|\boldsymbol{J}^{-1}\|_F^2 &\leq \hat{Q}_{ali}^{\frac{2(n-1)}{n}}\, |J|^{-\frac{2}{n}}\, n \det\left(I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right)^{-\frac{1}{n}} \mathrm{tr}\left(I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right), \\ \|\boldsymbol{J}\|_F^2 &\leq Q_{ali}^{\frac{2(n-1)}{n}}\, |J|^{\frac{2}{n}}\, n \det\left(I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right)^{\frac{1}{n}} \mathrm{tr}\left(\left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right]^{-1} \right), \\ \mathrm{tr}\left(\boldsymbol{J}^T \left[I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right] \boldsymbol{J} \right) &= Q_{ali}^{\frac{2(n-1)}{n}}\, |J|^{\frac{2}{n}}\, n \det\left(I + \frac{1}{\alpha_{ani,2}} |H(D^{l-2}v)| \right)^{\frac{1}{n}}. \end{aligned}$$

Combining the above results leads to

$$\begin{aligned} |v - \Pi_k v|^q_{W^{m,q}(\Omega)} \quad &\overset{<}{\to} \quad C\,\alpha^q_{ani,2}\,N^{-\frac{(l-m)q}{n}} \int_\Omega dx\,\hat{Q}_{ali}^{\frac{mq(n-1)}{n}}\;Q_{ali}^{\frac{ql(n-1)}{n}}\;|J|^{\frac{(l-m)q}{n}} \\ &\times \det\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)^{\frac{(l-m)q}{2n}} \left[\operatorname{tr}\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)\right]^{\frac{mq}{2}} \\ &\times \left[\operatorname{tr}\left(\left[I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right]^{-1}\right)\right]^{\frac{(l-2)q}{2}} . \end{aligned} \tag{6.45}$$

From this, the optimal adaptation and monitor functions can be defined as

$$\begin{aligned} \rho = \rho_{ani,2} \quad \equiv \quad &\det\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)^{\frac{(l-m)q}{2(n+(l-m)q)}} \\ &\times \left[\operatorname{tr}\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)\right]^{\frac{mnq}{2(n+(l-m)q)}} \\ &\times \left[\operatorname{tr}\left(\left[I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right]^{-1}\right)\right]^{\frac{(l-2)nq}{2(n+(l-m)q)}} \end{aligned} \tag{6.46}$$

and

$$M = M_{ani,2}(x) \equiv \rho^{\frac{2}{n}}_{ani,2} \det\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)^{-\frac{1}{n}} \left[I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right]. \tag{6.47}$$

For the current case, the intensity parameter $\alpha_{ani,2}$ has to be defined implicitly, i.e.,

$$\int_\Omega \rho_{ani,2} dx = 2|\Omega| \; . \tag{6.48}$$

Defining the overall mesh quality measure as

$$Q_{mesh,ani,2} = \left[\frac{1}{\sigma_h}\int_\Omega \left(\hat{Q}_{ali}^{\frac{m(n-1)}{n}}\;Q_{ali}^{\frac{l(n-1)}{n}}\;Q_{eq}^{\frac{(l-m)}{n}}\right)^q \rho_{ani,2} dx\right]^{\frac{1}{q}} . \tag{6.49}$$

From (6.45) the error is bounded by

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \overset{<}{\to} C N^{-\frac{(l-m)}{n}} \alpha_{ani,2}\, Q_{mesh,ani,2}. \tag{6.50}$$

From (6.46) and (6.48), an estimate can be obtained for $\alpha_{ani,2}$:

$$\alpha_{ani,2} \le C\left[\int_\Omega \operatorname{tr}\left(|H(D^{l-2}v)|\right)^{\frac{lnq}{2(n+(l-m)q)}} dx\right]^{\frac{2(n+(l-m)q)}{lnq}} \approx C|v|_{W^{l,\frac{lnq}{2(n+(l-m)q)}}(\Omega)} . \tag{6.51}$$

Note that this estimate is very rough, particularly when $l > 2$.

6.5.2 Mesh assessment

An existing adaptive mesh is assessed by comparing the interpolation error thereon to its counterpart on a uniform mesh of the same number of elements.

For the current situation $l \geq 2$, a bound of interpolation error on a uniform mesh is given in (6.33). From (6.33) and (6.50), the solution roughness can be defined as

$$Q_{soln,ani,2} = \frac{\langle v \rangle_{W^{l,p}(\Omega)}}{\alpha_{ani,2}}. \tag{6.52}$$

The error bound (6.50) reads as

$$|v - \Pi_{k,K} v|_{W^{m,q}(\Omega)} \overset{<}{\to} C\, N^{-\frac{(l-m)}{n}} \, |v|_{W^{l,p}(\Omega)} \, \frac{Q_{mesh,ani,2}}{Q_{soln,ani,2}}.$$

Thus, a mesh has a good overall quality when $Q_{mesh,ani,2} = O(1)$ or $Q_{mesh,ani,2} \ll Q_{soln,ani,2}$.

Chapter 7

Ansiotropic mesh adaptation: Refinement approach

7.1 Introduction

The objective of this chapter is to develop a general formula of the metric tensor for use with h-version meshing strategies and software for the generation of unstructured anisotropic meshes.

Refinement or h-version adaptation has been the most popular approach in use for generating unstructured meshes in finite element computation because of its reliability and simplicity in concept. Typically with the approach, a local error estimate is computed and then local minimization tools, such as edge suppression, vertex suppression, vertex addition, edge swapping, and vertex reallocation, are employed to generate the needed adaptive mesh; e.g., see [51, 31, 42, 43].

For the generation of adaptive anisotropic meshes, the common practice with the refinement approach is to generate them as quasi-uniform meshes in the metric determined by a tensor (or a matrix-valued function) that specifies the size, shape, and orientation of elements throughout the domain. Such a metric tensor plays a similar role as the monitor function used and defined in Chapters 2 and 6. Their relation is discussed in this chapter.

A number of meshing strategies have been developed in the last decade for generating anisotropic meshes according to a given metric tensor. Examples are the Delaunay triangulation method [16, 17, 31, 95], the advancing front method [50], the bubble mesh method [122], and the method combining local modification with smoothing or node movement [3, 18, 43, 53]. About a dozen of computer codes, mostly in two dimensions, have been developed or modified with the anisotropic mesh option; e.g., see the meshing software survey by Owen [94]. Among these meshing strategies and computer codes, the metric tensor is commonly defined based on the Hessian of the physical solution and largely motivated by by the results of D'Azevedo [37] and D'Azevedo and Simpson [38] on linear interpolation for quadratic functions on triangles. For example, Castro-Díaz et al. [31], Habashi et al. [53], and Remacle et al. [98] define their metric tensor as $\mathcal{M} = |H(v)|$. (Here we use the calligraphic letter $\mathcal{M}$, compared to M for the monitor function.) The metric tensor $\mathcal{M}$ is further modified by imposing the maximal and minimal edge lengths to guarantee its positive definiteness and avoid unrealistic metric. In his two dimensional anisotropic mesh generation code

BAMG, Hecht [57] uses

$$\mathcal{M} = \frac{1}{\epsilon_0 \cdot \text{Coef}^2} \cdot \frac{|H(v)|}{\max\{\text{CutOff}, |v|\}} \tag{7.1}$$

for the relative error and

$$\mathcal{M} = \frac{1}{\epsilon_0 \cdot \text{Coef}^2} \cdot \frac{|H(v)|}{\sup(v) - \inf(v)} \tag{7.2}$$

for the absolute error, where ϵ_0, Coef, and CutOff are the user specified parameters used for setting the level of the linear interpolation error (with default value 10^{-2}), the value of a multiplicative coefficient on the mesh size (with default value 1), and the limit value of the relative error evaluation (with default value 10^{-5}), respectively. In [52], George and Hecht define the metric tensor for various norms of the interpolation error as

$$\mathcal{M} = \left(\frac{c_0}{\epsilon_0}\right)^{\nu} Q \begin{pmatrix} |\lambda_1|^{\nu} & 0 & 0 \\ 0 & |\lambda_2|^{\nu} & 0 \\ 0 & 0 & |\lambda_3|^{\nu} \end{pmatrix} Q^T, \tag{7.3}$$

where λ_i's are the eigenvalues of $|H(v)|$, Q consists of the corresponding normalized eigenvectors, c_0 is a constant, ϵ_0 is a given error threshold, and $\nu = 1$ for the L^{∞} norm and the H^1 semi-norm and $\nu = 1/2$ for L^2 norm of the error. It is emphasized that the above definitions are based on either the results of [37] or heuristic considerations.

A general formula for the metric tensor for use in anisotropic (and isotropic) mesh generation in any spatial dimension has been developed in [62]. It is presented in this chapter based on the monitor function defined in Chapter 6. To be specific, the formulation is targeted for use with a public-domain c++ code BAMG (Bidimensional Anisotropic Mesh Generator) developed by Hecht [57]. But it should be emphasized that the strategy of formulating the metric tensor applies straightforwardly to other meshing algorithms and codes.

BAMG is a Delaunay-type triangulator which allows the user to supply a metric tensor or a solution defined on a background mesh. Its internal metric tensors are defined in equations (7.1) and (7.2). The user defined metric tensor should be given in such a way that the elements of the desired mesh are isotropic and have a unitary volume in the given metric. Once the metric is given, BAMG employs five local minimization tools, edge suppression, vertex suppression, vertex addition, edge swapping, and vertex reallocation (barycentering step) to generate the needed anisotropic mesh.

7.2 Metric tensor

Both the monitor function M and metric tensor $\mathcal{M}$ play the same role in mesh generation, i.e., they are used to specify the size, shape, and orientation of mesh elements throughout the physical domain. The only difference lies in the way they specify the size of elements. Indeed, M specifies the element size through the equidistribution condition (2.4) (also see (2.11)) while $\mathcal{M}$ used in BAMG determines the element size through the unitary volume requirement

$$\int_K \sqrt{\det(\mathcal{M})}\, dx = 1, \quad \forall K \in \mathcal{T}_h. \tag{7.4}$$

One can easily guess that M and $\mathcal{M}$ are related by

$$\mathcal{M}(x) = \theta\, M(x), \quad \forall x \in \Omega \tag{7.5}$$

where θ is a positive constant to be determined. In the following $\mathcal{M}$ is defined for three cases, isotropic, anisotropic with $l = 1$, and anisotropic with $l \geq 2$.

7.2.1 Isotropic error estimation

For this case, the monitor function M_{iso} is given in the equation (6.27). It follows from (7.5) that

$$\mathcal{M}_{iso}(x) = \theta \, M_{iso}(x).$$

Inserting this into the unitary condition (7.4) leads to

$$\theta^{\frac{n}{2}} \int_K \rho_{iso} dx = 1,$$

where $\rho_{iso} = \sqrt{\det(M_{iso})}$ is given in (6.26). Summing the above equation over all the elements of $\mathcal{T}_h$, one gets

$$\theta^{\frac{n}{2}} \sigma = N,$$

where N is the number of elements of $\mathcal{T}_h$ and $\sigma = \int_\Omega \rho_{iso} dx$. Thus,

$$\theta = \left(\frac{N}{\sigma}\right)^{\frac{2}{n}}.$$

It follows from (6.27) that

$$\mathcal{M}_{iso,N}(x) = \left(\frac{N}{\sigma}\right)^{\frac{2}{n}} M_{iso}(x) = \left(\frac{N}{\sigma}\right)^{\frac{2}{n}} \left(1 + \frac{1}{\alpha_{iso}} \|D^l v\|_{l_p}\right)^{\frac{2q}{n+q(l-m)}} I, \tag{7.6}$$

where $\sigma = \int_\Omega \rho_{iso} dx$, α_{iso} is defined in (6.28), and ρ_{iso} is given in (6.26).

Sometimes it is more convenient to use an error level ϵ_0 instead of the number of elements N in the metric tensor. Consider the case where an error level ϵ_0 is given for the $W^{m,q}$ semi-norm of interpolation error. From (6.28) one can see that the error bound is asymptotically proportional to $N^{-\frac{(l-m)}{n}} \alpha_{iso}$ provided that the overall mesh quality measure $Q_{mesh,iso}$ is bounded by a relatively small number. It is thus reasonable to set

$$\epsilon_0 = N^{-\frac{(l-m)}{n}} \alpha_{iso}.$$

Hence,

$$N = \left(\frac{\alpha_{iso}}{\epsilon_0}\right)^{\frac{n}{l-m}}.$$

Inserting this into (7.6) yields

$$\mathcal{M}_{iso,\epsilon_0}(x) = \left(\frac{1}{\sigma}\left(\frac{\alpha_{iso}}{\epsilon_0}\right)^{\frac{n}{l-m}}\right)^{\frac{2}{n}} \left(1 + \frac{1}{\alpha_{iso}} \|D^l v\|_{l_p}\right)^{\frac{2q}{n+q(l-m)}} I. \tag{7.7}$$

Note that (7.7) holds only for $m < l$.

7.2.2 Anisotropic error estimation: $l = 1$

In this case, the metric tensor can be obtained from the monitor function (6.40) as in the previous subsection. Indeed, one has

$$\begin{aligned}
\mathcal{M}_{ani,1,N}(x) &= \left(\frac{N}{\sigma}\right)^{\frac{2}{n}} \left(1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2\right)^{\frac{mq-1}{n+(1-m)q}} \left[I + \frac{1}{\alpha_{ani,1}^2}\nabla v \nabla v^T\right], \qquad (7.8)\\
\mathcal{M}_{ani,1,\epsilon_0}(x) &= \left(\frac{1}{\sigma}\left(\frac{\alpha_{ani,1}}{\epsilon_0}\right)^{\frac{n}{1-m}}\right)^{\frac{2}{n}} \left(1 + \frac{1}{\alpha_{ani,1}^2}\|\nabla v\|^2\right)^{\frac{mq-1}{n+(1-m)q}} \left[I + \frac{1}{\alpha_{ani,1}^2}\nabla v \nabla v^T\right], (7.9)
\end{aligned}$$

where $\sigma = \int_\Omega \rho_{ani,1} dx$, $\alpha_{ani,1}$ is defined in (6.41), and $\rho_{ani,1}$ is given in (6.39). It is noted that (7.9) holds only for $m = 0$.

7.2.3 Anisotropic error estimation: $l \geq 2$

In this case, the monitor function is given in (6.47). The metric tensors are

$$\begin{aligned}
\mathcal{M}_{ani,2,N}(x) &= \left(\frac{N}{\sigma}\right)^{\frac{2}{n}} \rho_{ani,2}^{\frac{2}{n}} \det\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)^{-\frac{1}{n}} \left[I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right] (7.10)\\
\mathcal{M}_{ani,2,\epsilon_0}(x) &= \left(\frac{1}{\sigma}\left(\frac{\alpha_{ani,2}}{\epsilon_0}\right)^{\frac{n}{l-m}}\right)^{\frac{2}{n}} \rho_{ani,2}^{\frac{2}{n}} \det\left(I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right)^{-\frac{1}{n}}\\
&\quad \times \left[I + \frac{1}{\alpha_{ani,2}}|H(D^{l-2}v)|\right], \qquad (7.11)
\end{aligned}$$

where $\sigma = \int_\Omega \rho_{ani,2} dx$, $\alpha_{ani,2}$ is defined in (6.48), and $\rho_{ani,2}$ is given in (6.46). It is noted that (7.11) holds only for $m < l$.

7.2.4 A remark on computation of metric tensor and monitor function

The formulas of the metric tensor developed in this section depend on several factors, including the function regularity (through parameters l and p), the dimension of space (n), and the norm used to measure interpolation error (m and q). Moreover, the formulas involve derivatives (of order $l \leq k+1$) of the physical solution, which is unknown in general. Fortunately, adaptive computation is often carried out in an iterative fashion and approximations of the nodal values of the physical solution on a current mesh are always available. A gradient recovery technique such as those of Zienkiewicz and Zhu [127, 128] and Zhang and Naga [126] can then be used for computing the needed derivatives, although their convergence has been analyzed only on isotropic meshes.

This remark also applies to the computation of the monitor function defined in Chapter 6.

7.3 Numerical experiment

In this section some two dimensional numerical results are presented for illustrative purpose. The results are obtained with BAMG [57] in an iterative fashion: Starting from a coarse mesh, the nodal values of the solution are obtained either from an analytical expression (for the problems with given analytical solutions) or by solving a PDE via a finite element method. Then the first and second order derivatives of the solution used in the metric tensor are obtained by using a linear least-squares fitting to the nodal values of the solution and the computed first order derivatives, respectively. This is followed by the computation of the metric tensor according to the formulas given in the previous section. Finally, a new mesh according to the computed metric tensor on the current mesh (as the background mesh) is obtained by called BAMG. The process is repeated twenty times in the computation.

Other parameters chosen in the computation are: $k = 1$ (for linear interpolation or linear finite elements), $l = 2$, $p = q = 2$, and $m = 0$ (with the error measured in the L^2 norm) or $m = 1$ (with the error measured in the H^1 semi-norm).

In the results presented below, e denotes the error either for linear interpolation or in the linear finite element solution. Variables *nbv* and *nbt* denote the actual numbers of vertices and elements of a mesh, respectively. Note that the quantity *nbt* is different from N used in the formulas of the metric tensor in the previous section. The former is the actual number of the elements in a computed mesh whereas the latter is only a user prescribed, target number of elements.

Example 7.3.1. This example is to generate an adaptive mesh for the function

$$v(x, y) = \tanh(60y) - \tanh(60(x - y) - 30), \qquad \forall (x, y) \in \Omega \equiv (0, 1) \times (0, 1). \tag{7.12}$$

This function exhibits a strong anisotropic feature, simulating the interaction of a boundary layer (on $y = 0$) with an oblique shock wave (along $y = x - 0.5$). It has been used as a test example by a number of researchers; e.g. see [54].

Fig. 7.1 shows two typical adaptive meshes obtained with metric tensors $\mathcal{M}_{ani,2,N}$ and $\mathcal{M}_{iso,N}$. They both have correct mesh concentration. Moreover, the metric tensor $\mathcal{M}_{ani,2,N}$ leads to an anisotropic mesh (Fig. 7.1 (a) and (b)) whereas $\mathcal{M}_{iso,N}$ yields an isotropic one (Fig. 7.1 (c) and (d)). This can be seen more clearly in Fig. 7.2 where the mesh quality measures are shown as the meshes are refined. Indeed, the geometric quality measure, $\|Q_{geo}\|_\infty$, is about 2 for meshes obtained with $\mathcal{M}_{iso,N}$ (with $(l, m) = (2, 0)$ and $(l, m) = (2, 1)$), indicating that the elements are close to being equilateral. On the other hand, $\|Q_{geo}\|_\infty \gg 1$ for meshes obtained with $\mathcal{M}_{ani,2,N}$, implying that some of the elements have large aspect ratio. Furthermore, Fig. 7.1 shows an isotropic mesh requires ten times more elements than an anisotropic mesh to attain nearly the same level of interpolation error. This demonstrates that a significant gain can be achieved by using an isotropic mesh which allows not only the size but also the shape and orientation of elements to adapt to the features of the physical solution. The advantage will not deteriorate as the meshes are refined, as can be seen from Fig. 7.3 where $\|e\|_{L^2(\Omega)}$ and $|e|_{H^1(\Omega)}$ are shown as function of *nbt* for isotropic and anisotropic adaptive meshes. For comparison purpose, the error is also shown in the figure for a uniform mesh. It can be seen that both isotropic and anisotropic adaptive meshes lead to much smaller interpolation error than a uniform mesh although they have different levels of accuracy improvements.

Example 7.3.2. This example is to generate an adaptive mesh for the function

$$v(x, y) = \tanh\left(-100\left(y - \frac{1}{2} - \frac{1}{4}\sin(2\pi x)\right)^2\right). \tag{7.13}$$

Compared to Example 8.3.1, this function exhibits a weaker anisotropic feature. Two typical adaptive meshes obtained with $\mathcal{M}_{iso,N}$ and $\mathcal{M}_{ani,2,N}$ are shown in Fig. 7.4. The advantage of an anisotropic mesh over an isotropic one is clear.

Example 7.3.3. This example is to solve the PDE

$$-\epsilon\Delta v + \left(1 + e^{\frac{x+y-0.85}{2\epsilon}}\right)^{-1}(v_x + v_y) = -\frac{1}{2\epsilon}\left(1 + e^{\frac{x+y-0.85}{2\epsilon}}\right)^{-2} e^{\frac{x+y-0.85}{2\epsilon}} \tag{7.14}$$

defined on $\Omega \equiv (0,1) \times (0,1)$. The Dirichlet boundary condition is chosen such that the exact solution is given by

$$v(x,y) = \left(1 + e^{\frac{x+y-0.85}{2\epsilon}}\right)^{-1}. \tag{7.15}$$

The solution exhibits a sharp layer along the line $x + y - 0.85 = 0$ when ϵ is small. In the computation, ϵ is taken as 0.005, and the PDE is diecretized using linear triangular finite elements. It is emphasized that for this example, the metric tensors are computed using a computed solution of the PDE rather than the exact solution.

Two adaptive meshes of almost the same number of elements are shown in Fig. 7.5. One can see that the anisotropic mesh gives a solution error (in the L^2 norm) nearly ten times smaller than that on the isotropic mesh. In Fig. 7.6 $\|e\|_{L^2(\Omega)}$ and $|e|_{H^1(\Omega)}$ are shown as function of *nbt* for isotropic and anisotropic adaptive meshes and a uniform mesh.

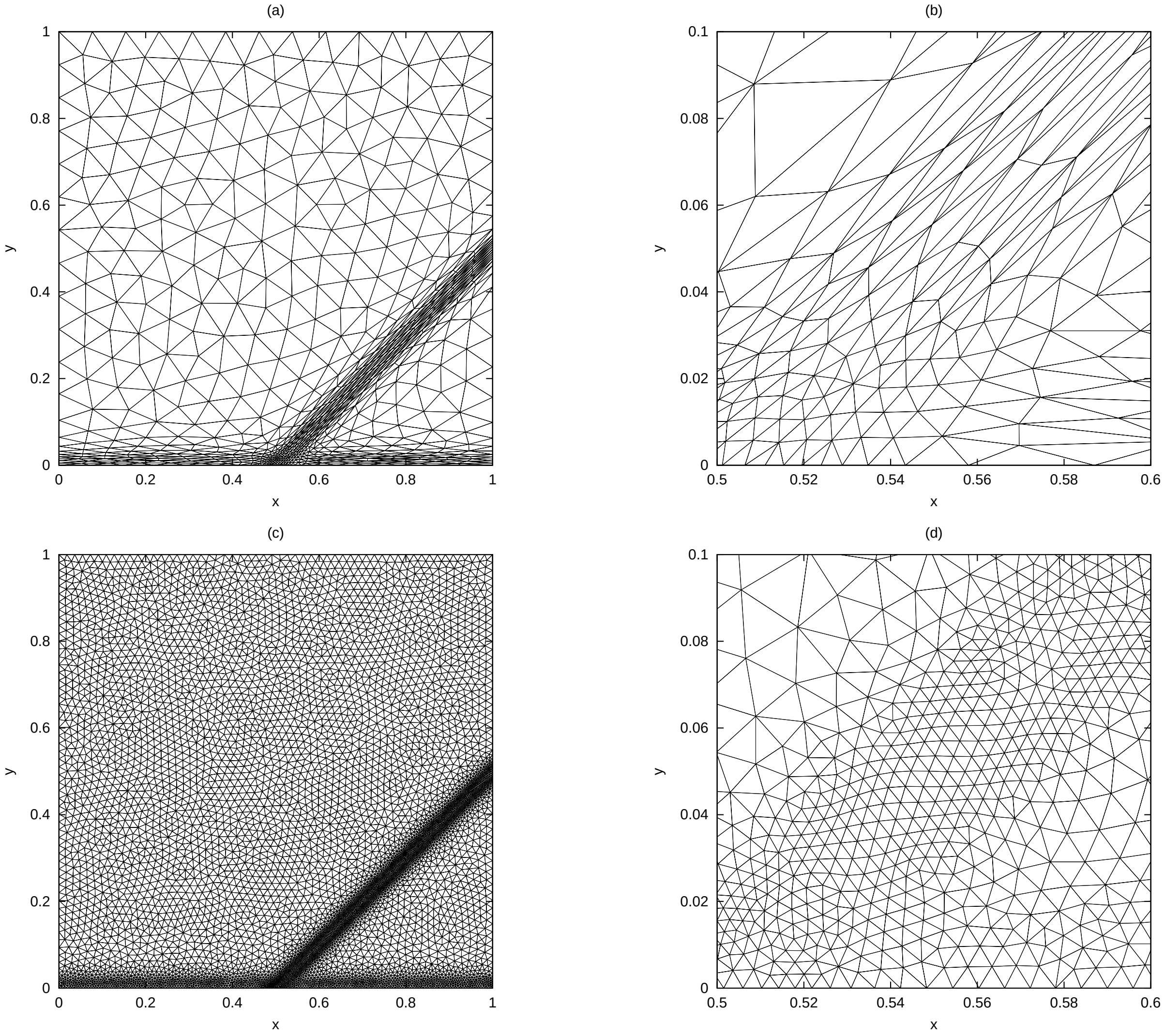


Figure 7.1: Example 7.3.1. (a) An anisotropic mesh obtained with $\mathcal{M}_{ani,2,N}$ and $(l,m) = (2,1)$: $nbv = 645$, $nbt = 1187$, $|e|_{H^1} = 0.88$, and $\|e\|_{L^2} = 1.3 \times 10^{-3}$. (b) Close-up of the mesh in (a) near $(x,y) = (0.5,0)$. (c) An isotropic mesh obtained with $\mathcal{M}_{iso,N}$ and $(l,m) = (2,1)$: $nbv = 7328$, $nbt = 14291$, $|e|_{H^1} = 0.85$, and $\|e\|_{L^2} = 1.0 \times 10^{-3}$. (d): Close-up of the mesh in (c) near $(x,y) = (0.5,0)$.

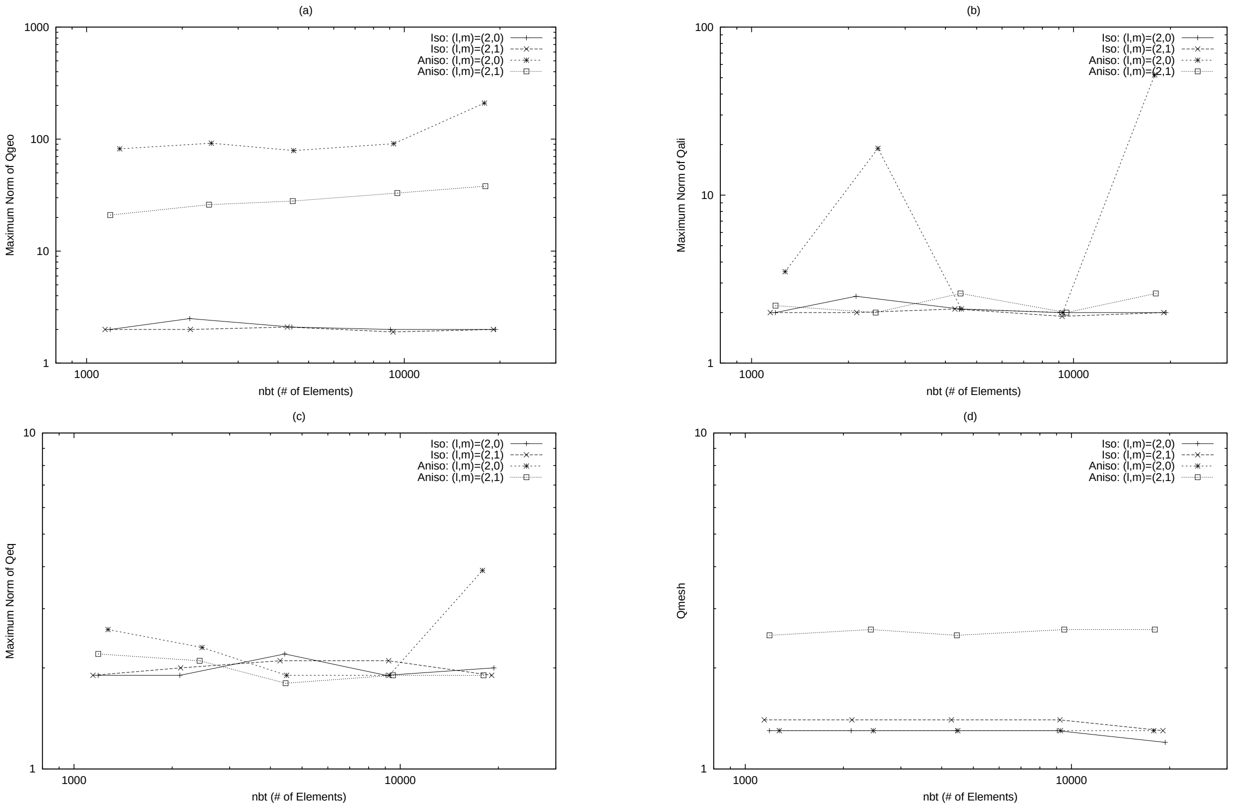


Figure 7.2: Example 7.3.1. The mesh quality measures, $\|Q_{geo}\|_\infty$, $\|Q_{ali}\|_\infty$, $\|Q_{eq}\|_\infty$, and Q_{mesh} are depicted as functions of *nbt* for meshes generated with various metric tensors.

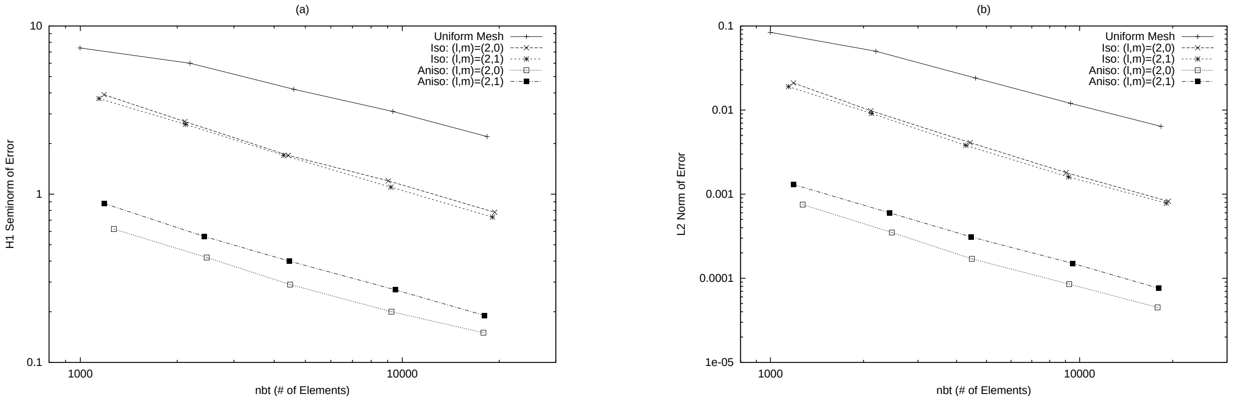
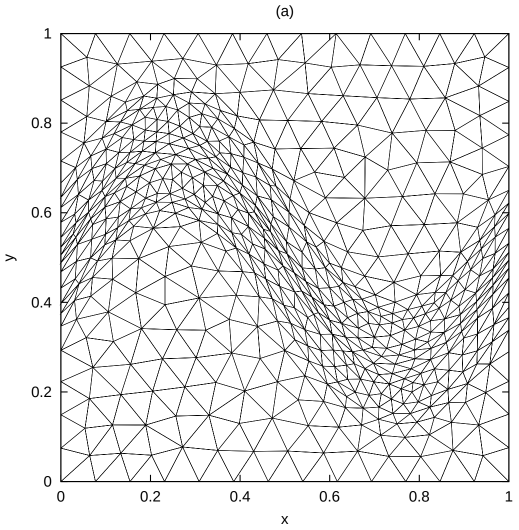


Figure 7.3: Example 7.3.1. The H^1 semi-norm and the L^2 norm of interpolation error are plotted as functions of the number of elements (*nbt*) in (a) and (b), respectively.



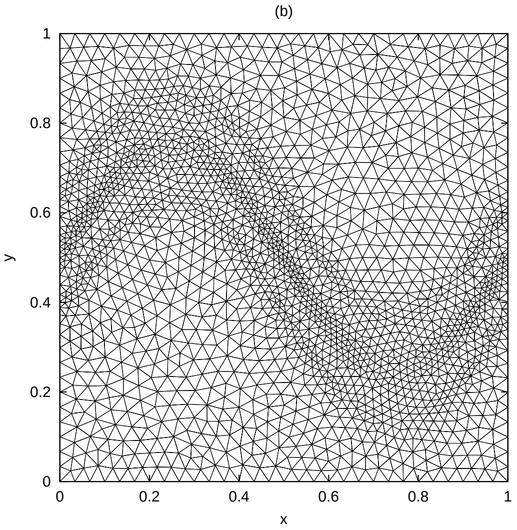
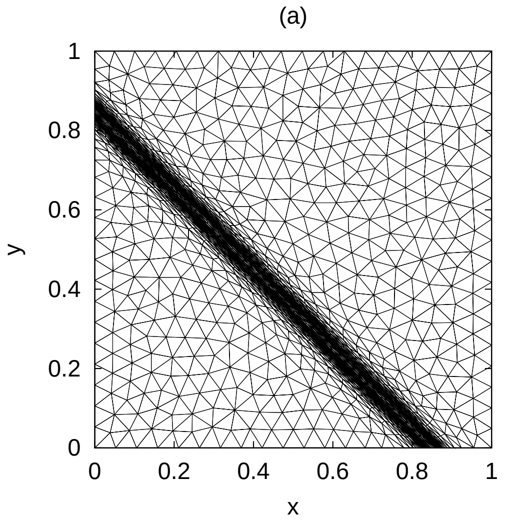


Figure 7.4: Example 7.3.2. (a) An anisotropic mesh obtained with $\mathcal{M}_{ani,2,N}$ and $(l, m) = (2, 1)$: $nbv = 583$, $nbt = 1094$, $|e|_{H^1} = 0.90$, and $\|e\|_{L^2} = 5.2 \times 10^{-3}$. (b) An isotropic mesh obtained with $\mathcal{M}_{iso,N}$ and $(l, m) = (2, 1)$: $nbv = 2036$, $nbt = 3928$, $|e|_{H^1} = 0.86$, and $\|e\|_{L^2} = 3.8 \times 10^{-3}$.



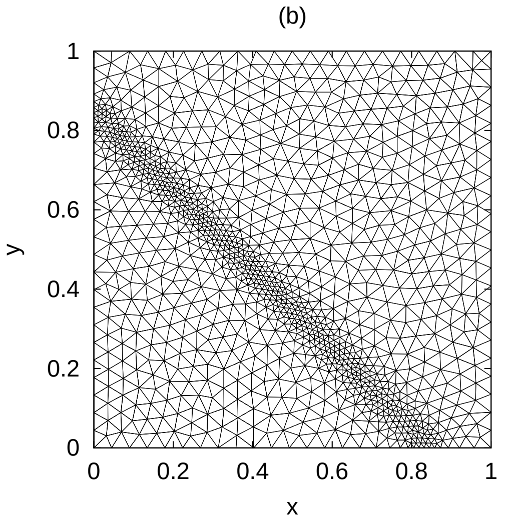


Figure 7.5: Example 7.3.3. (a) An anisotropic mesh obtained with $\mathcal{M}_{ani,2,N}$, $(l, m) = (2, 1)$, and $N = 1000$: $nbv = 1261$, $nbt = 2388$, $|e|_{H^1} = 0.2$, and $\|e\|_{L^2} = 2.0 \times 10^{-4}$. (b) An isotropic mesh obtained with $\mathcal{M}_{iso,N}$, $(l, m) = (2, 1)$, and $N = 1000$: $nbv = 1214$, $nbt = 2324$, $|e|_{H^1} = 1.1$, and $\|e\|_{L^2} = 3.3 \times 10^{-3}$.

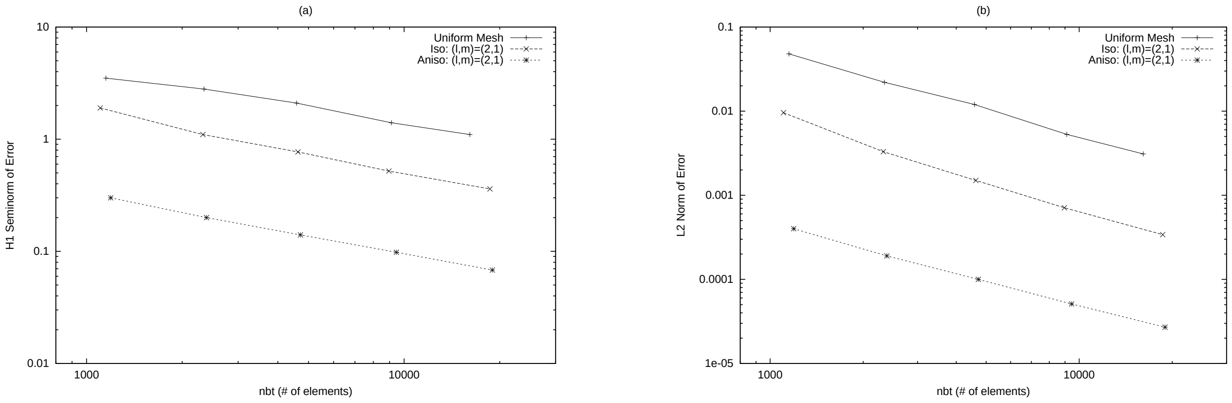


Figure 7.6: Example 7.3.3. The H^1 semi-norm and the L^2 norm of the solution error are plotted as functions of the number of elements (*nbt*) in (a) and (b), respectively.

Chapter 8

Ansiotropic mesh adaptation: Variational approach

8.1 Introduction

In the variational approach of mesh adaptation, adaptive meshes are generated as images of a computational mesh under a coordinate transformation from the computational domain to the physical domain. Such a coordinate transformation is determined by the so-called adaptation functional which is commonly designed to measure the difficulty in the numerical approximation of the physical solution. The functional often involves mesh properties and employs a monitor function to control mesh quality and mesh concentration.

In the last two decades, the variational approach has received considerable attention from scientists and engineers; e.g., see [19, 20, 45, 54, 59, 68, 71, 73, 74, 106, 120] and books [46, 72, 85, 114] and references therein. The approach is particularly suitable for finite difference computations. Its implementation is simple, requiring no expertise in data structure that is necessary for the implementation of mesh refinement. A variational method has typically been used for generating structured meshes, but it can also be used for generating unstructured meshes; e.g., see [24].

The key to the development of variational methods is to formulate the adaptation functional. Upon the well-posedness consideration, people usually do not directly use standard error estimates since they lead to non-convex functionals in two and higher dimensions. Instead, most of the existing variational methods have been developed based on other considerations such as geometric ones. For example, Brackbill and Saltzman [20] develop a very popular method by combining mesh concentration, smoothness, and orthogonality. Dvinsky [45] uses the energy of harmonic mappings as his mesh adaptation functional. Brackbill [19] combines directional control with Dvinsky's harmonic mapping method. Knupp [71, 73] and Knupp and Robidoux [74] develop functionals based on the idea of conditioning the Jacobian matrix of the coordinate transformation. A functional has been developed recently by Huang [59] based on the alignment (also called isotropy or conformal) and equidistribution considerations.

In this chapter attention is focused on the functional developed in [59]. It includes the harmonic map method as a special example. Moreover, the terms involved in the functional have the same dimension and this makes it relatively easy to balance among the terms. Furthermore and most importantly, the functional is developed directly based on the alignment and equidistribution conditions (2.3) and (2.4).

8.2 Functional for mesh alignment

We first consider the alignment condition (2.3). Let the eigenvalues of matrix $\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}$ be λ_1, ..., λ_n. By the arithmetic-mean geometric-mean inequality (cf. Theorem 1.4.1) the desired coordinate transformation can be obtained by minimizing the difference between the two sides of the inequality

$$\left(\prod_i \lambda_i\right)^{\frac{1}{n}} \le \frac{1}{n}\sum_i \lambda_i. \tag{8.1}$$

Notice that

$$\sum_i \lambda_i = \operatorname{tr}(\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}) = \sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i,$$

$$\prod_i \lambda_i = \det(\boldsymbol{J}^{-1}M^{-1}\boldsymbol{J}^{-T}) = \frac{1}{(J\rho)^2},$$

where $J = \det(\boldsymbol{J})$ and $\rho = \sqrt{\det(M)}$. Then, (8.1) becomes

$$\left(\frac{1}{(J\rho)^2}\right)^{\frac{1}{n}} \le \frac{1}{n}\sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i$$

or equivalently

$$\frac{n^{\frac{n}{2}}}{J} \le \rho\left(\sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i\right)^{\frac{n}{2}}. \tag{8.2}$$

Integrating the above inequality over the physical domain yields

$$n^{\frac{n}{2}}\int_{\Omega_c} d\xi \le \int_{\Omega} \rho\left(\sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i\right)^{\frac{n}{2}} dx.$$

Hence, the adaptation functional associated with mesh alignment for the inverse coordinate transformation $\xi = \xi(x)$ can be defined as

$$I_{ali}[\xi] = \frac{1}{2}\int_{\Omega} \rho\left(\sum_i (\nabla\xi_i)^T M^{-1}\nabla\xi_i\right)^{\frac{n}{2}} dx. \tag{8.3}$$

It is remarked that functional (8.3) can also be derived from the concept of conformal norm in the context of differential geometry [59]. Moreover, in two dimensions ($n = 2$), (8.3) gives the energy of a harmonic mapping [45].

8.3 Functional for equidistribution

We now consider the equidistribution condition (2.4). Taking $w = \rho$, $f = 1/(J\rho)$, $r = 1$, and $s = \gamma$ for any real number $\gamma > 1$ in Theorem 1.2.1, one gets

$$\int_{\Omega} \frac{\rho}{J\rho} dx = \int_{\Omega_c} d\xi \le \left(\int_{\Omega} \frac{\rho}{(J\rho)^{\gamma}} dx\right)^{1/\gamma}, \tag{8.4}$$

with equality if and only if the equidistribution condition (2.4) holds. Thus, the adaptation functional according to the equidistribution condition (2.4) can be defined as

$$I_{eq}[\xi] = \int_\Omega \frac{\rho}{(J\rho)^\gamma} dx, \tag{8.5}$$

where $\gamma > 1$ is a parameter.

8.4 Mesh adaptation functional

Note that neither of the functionals defined in the previous sections can alone lead to a robust mesh adaptation method because either of them represents only one of the mesh control conditions (2.3) and (2.4). It is necessary and natural to combine them together.

To this end, taking γ power on both sides of (8.2) and multiplying with ρ one gets

$$n^{n\gamma/2} \int_\Omega \frac{\rho}{(J\rho)^\gamma} dx \leq \int_\Omega \rho \left(\sum_i (\nabla \xi_i)^T M^{-1} \nabla \xi_i \right)^{\frac{n\gamma}{2}} dx. \tag{8.6}$$

On the other hand, taking γ power of (8.4) gives

$$\left(\int_{\Omega_c} d\xi \right)^\gamma \leq \int_\Omega \frac{\rho}{(J\rho)^\gamma} dx. \tag{8.7}$$

For a given value $\theta \in [0, 1]$, a balance of the differences between the two sides of (8.6) and of (8.7) is

$$\begin{aligned} & \theta \left[\int_\Omega \rho \left(\sum_i (\nabla \xi_i)^T M^{-1} \nabla \xi_i \right)^{\frac{n\gamma}{2}} dx - n^{n\gamma/2} \int_\Omega \frac{\rho}{(J\rho)^\gamma} dx \right] \\ + \quad & (1-\theta) n^{n\gamma/2} \left[\int_\Omega \frac{\rho}{(J\rho)^\gamma} dx - \left(\int_{\Omega_c} d\xi \right)^\gamma \right]. \end{aligned} \tag{8.8}$$

Thus, the adaptation functional is

$$I[\xi] = \theta \int_\Omega \rho \left(\sum_i (\nabla \xi_i)^T M^{-1} \nabla \xi_i \right)^{\frac{n\gamma}{2}} dx + (1 - 2\theta) n^{\frac{n\gamma}{2}} \int_\Omega \frac{\rho}{(J\rho)^\gamma} dx, \tag{8.9}$$

where $\theta \in [0, 1]$ and $\gamma > 1$ are real parameters, $\rho = \sqrt{\det(M)}$, and $J = \det(\boldsymbol{J})$. The first term of the functional corresponds to the alignment requirement while the second term represents the equidistribution requirement. By design, these two terms have the same dimension. The balance between them is controlled by a dimensionless parameter θ. When $\theta = 1/2$, only the first term remains.

Regarding well posedness, it is noted that the first term of the functional is convex when $n\gamma/2 \geq 1$, and the existence, uniqueness, and the maximal principle for its minimizer are guaranteed; e.g., see [99]. It is unclear if this result can apply to the whole functional.

8.5 Mesh equation

The coordinate transformation is governed by the Euler-Lagrange equation of functional (8.9). For simplicity, (8.9) is rewritten as

$$I[\xi] = \theta \int_\Omega \left(\sum_i (\nabla \xi_i)^T \bar{M}^{-1} \nabla \xi_i \right)^{\frac{n\gamma}{2}} dx + (1-2\theta) n^{\frac{n\gamma}{2}} \int_\Omega \frac{\rho}{(J\rho)^\gamma} dx, \tag{8.10}$$

where

$$\bar{M} = \rho^{-\frac{n\gamma}{2}} M.$$

The Euler-Lagrange equation can be written as

$$-\nabla \cdot \left[\frac{\theta n \gamma \beta^{\frac{n\gamma}{2}-1}}{2} \bar{M}^{-1} \nabla \xi_i + \frac{(1-2\theta)\gamma n^{\frac{n\gamma}{2}} \rho}{2} \left(\frac{1}{J\rho} \right)^\gamma \frac{\partial x}{\partial \xi_i} \right] = 0, \quad i = 1, ..., n \tag{8.11}$$

where $\beta = \sum_i (\nabla \xi_i)^T \bar{M}^{-1} \nabla \xi_i$.

Practically it is more convenient to compute $x = x(\xi)$ instead of its inverse $\xi = \xi(x)$ since the former explicitly defines the location of mesh points. Interchanging the roles of the dependent and independent variables, we have the conservative form

$$\sum_j \frac{\partial}{\partial \xi_j} J(\boldsymbol{a}_i)^T \left[\theta \frac{n\gamma}{2} \beta^{\frac{n\gamma}{2}-1} \bar{M}^{-1} \nabla \xi_i + \frac{(1-2\theta)\gamma n^{\frac{n\gamma}{2}} \rho}{2} \left(\frac{1}{J\rho} \right)^\gamma \frac{\partial x}{\partial \xi_i} \right] = 0, \quad i = 1, ..., n \tag{8.12}$$

and the non-conservative form

$$\begin{aligned} & \theta \left[\sum_{ij} \left((\boldsymbol{a}^i)^T \bar{M}^{-1} \boldsymbol{a}^j \right) \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} - \sum_i \left((\boldsymbol{a}^i)^T \sum_j \frac{\partial (\bar{M}^{-1})}{\partial \xi_j} \boldsymbol{a}^j \right) \frac{\partial x}{\partial \xi_i} \right] \\ + \quad & \frac{\theta(n\gamma - 2)}{2\beta} \left[2 \sum_{ij} \left((\bar{M}^{-1} \boldsymbol{a}^i)(\bar{M}^{-1} \boldsymbol{a}^j)^T \sum_k \boldsymbol{a}^k (\boldsymbol{a}^k)^T \right) \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} \right. \\ & \left. - \sum_i \left(\sum_j \left((\boldsymbol{a}^i)^T \bar{M}^{-1} \boldsymbol{a}^j \sum_k (\boldsymbol{a}^k)^T \frac{\partial (\bar{M}^{-1})}{\partial \xi_j} \boldsymbol{a}^k \right) \right) \frac{\partial x}{\partial \xi_i} \right] \\ + \quad & \frac{(1-2\theta) 2 (\gamma - 1) n^{\frac{n\gamma}{2}} \rho}{n \beta^{\frac{n\gamma}{2}-1} (J\rho)^\gamma} \left[\sum_{ij} \left(\boldsymbol{a}_i (\boldsymbol{a}^j)^T \right) \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} + \sum_i \left(\frac{1}{\rho} \frac{\partial \rho}{\partial \xi_i} \right) \frac{\partial x}{\partial \xi_i} \right] = 0, \end{aligned} \tag{8.13}$$

where $\boldsymbol{a}_i \equiv (\partial x)/(\partial \xi_i)$ and $\boldsymbol{a}^i \equiv \nabla \xi_i$ are the covariant and contravariant base vectors that are related by

$$\boldsymbol{a}^i = \frac{1}{J} \boldsymbol{a}_j \times \boldsymbol{a}_k \qquad \text{with } (i,j,k) \text{ cyclic.}$$

The mesh equation (8.13) can be written into

$$\sum_{i,j} C_{i,j} \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} + \sum_i b_i \frac{\partial x}{\partial \xi_i} = 0, \tag{8.14}$$

where $C_{i,j}$'s are $n \times n$ matrix-valued functions and b_i's are scalar ones. Given a monitor function M, the discretization of (8.14) is standard, and either finite differences or finite elements can be used. Care should be taken, however, when solving the resulting algebraic system because (8.14) is highly nonlinear in general. (Recall that the coefficients $C_{i,j}$ and b_i involve the derivatives of x with respect to ξ and the monitor function defined as function x.) In the iterative solution process, the coefficients are typically calculated at the previous iterate and a relaxation or a quasi-time continuation is used. Often a quasi-time method is helpful; see the moving mesh PDE approach to be discussed in Chapter 9.

The monitor function is given in Chapter 6. See the remark in §7.2.4 on its computation based on nodal values of the computed solution.

8.6 Numerical experiment

The results presented in this section are obtained using $\theta = 0.1$ in the functional (8.9) and $\omega = 0$ in monitor functions (6.47).

Example 8.6.1. This example is to generate an adaptive mesh for the function

$$v(x,y) = \tanh(60y) - \tanh(60(x-y) - 30), \qquad \forall (x,y) \in \Omega \equiv (0,1) \times (0,1). \tag{8.15}$$

This example is used in Chapter 7 for h-version methods. Results obtained using monitor functions M_{iso} (6.27) and $M_{ani,2}$ (6.47) with $(l,m) = (2,1)$) are plotted in Figs. 8.1, 8.2, and 8.2. It is interesting to note that in the current situation, the monitor function $M_{ani,2}$ (6.47) associated with anisotropic error estimation does not lead to a significant improvement in accuracy over M_{iso} associated with isotropic error estimation. This is in sharply contrast with the unstructured mesh situation (cf. Example 7.3.1)) where a nearly ten times smaller error is resulted from using an anisotropic metric tensor.

Example 8.6.2. This example is to generate an adaptive mesh for for a given analytical solution

$$\begin{aligned} v(x,y) &= \tanh(30(x^2+y^2-\frac{1}{8})) + \tanh(30((x-0.5)^2+(x-0.5)^2-\frac{1}{8})) \\ &+ \tanh(30((x-0.5)^2+(x+0.5)^2-\frac{1}{8})) + \tanh(30((x+0.5)^2+(x-0.5)^2-\frac{1}{8})) \\ &+ \tanh(30((x+0.5)^2+(x+0.5)^2-\frac{1}{8})) \end{aligned} \tag{8.16}$$

defined in $[-2,2] \times [-2,2]$. An adaptive mesh is expected to concentrate around five circles. Two adaptive meshes obtained with monitor functions M_{iso} and $M_{ani,2}$ are shown in Fig. 8.4. They are very similar. This may largely be attributed to the isotropic feature of the function. Linear interpolation error and geometric quality measure are listed in Table 8.1) for isotropic and anisotropic and uniform meshes.

Example 8.6.3. This example is to generate 3D adaptive meshes for

$$v(x,y,z) = \tanh(100((x-0.5)^2+(y-0.5)^2+(z-0.5)^2) - 0.0625) \tag{8.17}$$

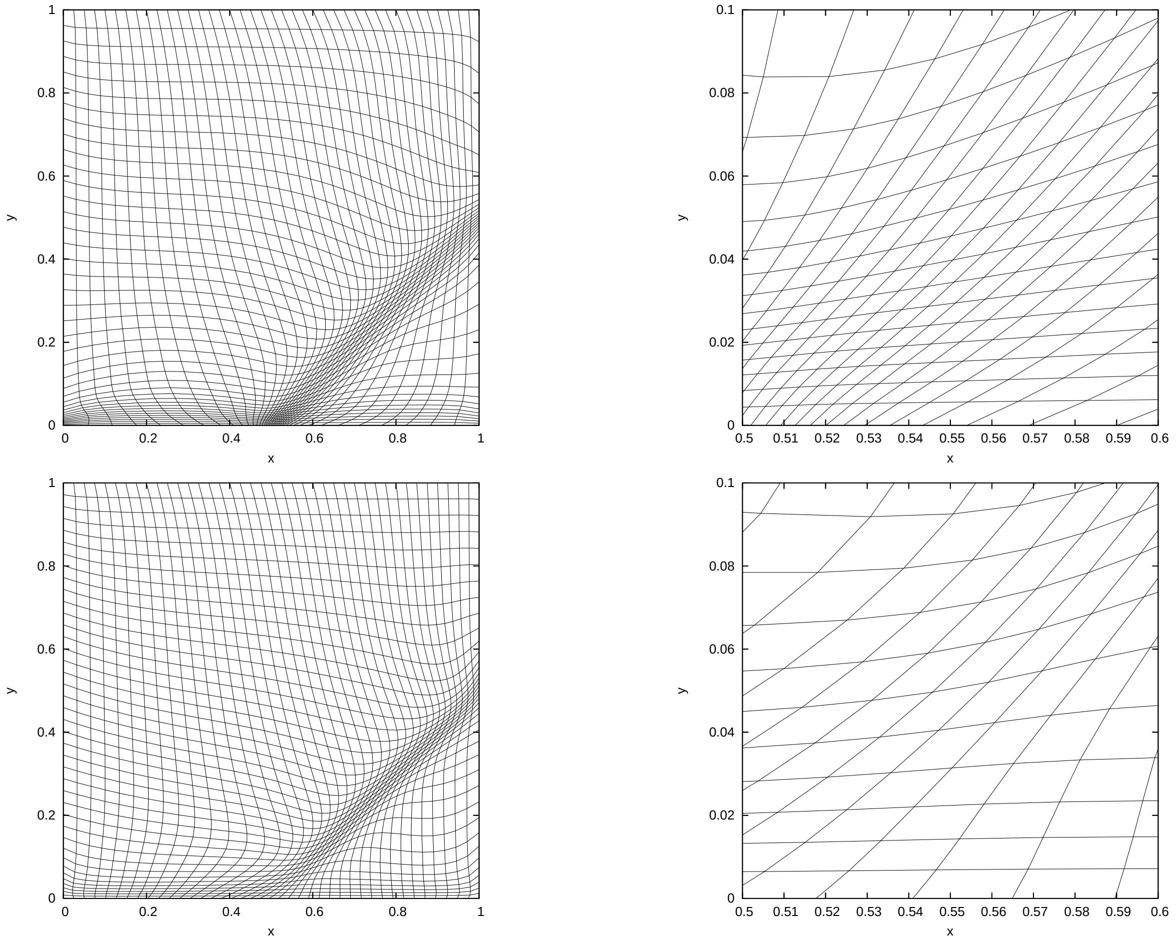


Figure 8.1: Example 8.6.1. (a) A 41×41 anisotropic mesh obtained with $M_{ani,2}$ and $(l, m) = (2, 1)$: $|e|_{H^1} = 1.25$, and $\|e\|_{L^2} = 2.38 \times 10^{-3}$. (b) Close-up of the mesh in (a) near $(x, y) = (0.5, 0)$. (c) A 41×41 isotropic mesh obtained with M_{iso} and $(l, m) = (2, 1)$: $|e|_{H^1} = 1.65$, and $\|e\|_{L^2} = 3.94 \times 10^{-3}$. (d): Close-up of the mesh in (c) near $(x, y) = (0.5, 0)$.

defined in the unit cube. An adaptive mesh for this function is expected to concentrate near the sphere centered at $(0, 0, 0)$ with radius 0.25. Results are shown in Table 8.2 and Fig. 8.5.

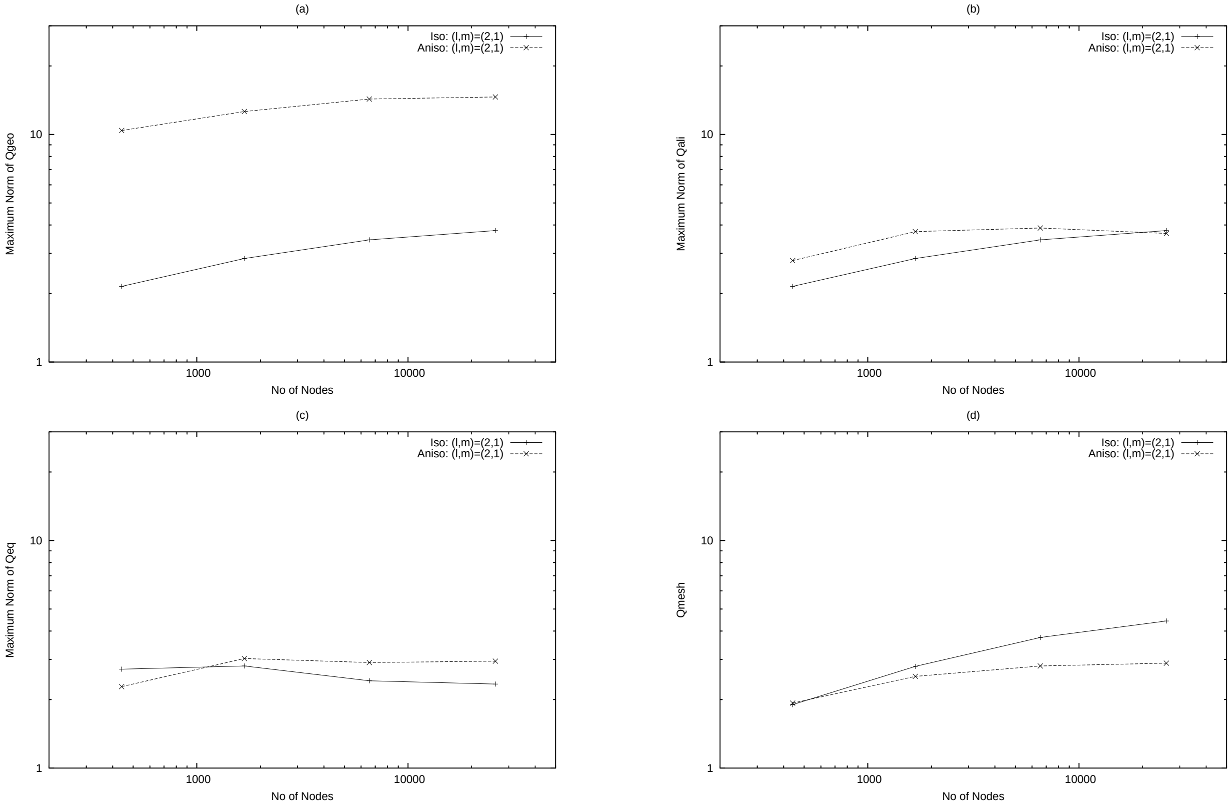


Figure 8.2: Example 8.6.1. Mesh quality measures as function of the number of nodes.

Example 8.6.4. The example is to generate 3D adaptive meshes for

$$\begin{aligned} v(x,y,z) &= \tanh(30.0(x^2+y^2+z^2-0.1875)) \\ &+ \tanh(30((x-0.5)^2+(y-0.5)^2+(z-0.5)^2-0.1875)) \\ &+ \tanh(30((x-0.5)^2+(y+0.5)^2+(z-0.5)^2-0.1875)) \\ &+ \tanh(30((x+0.5)^2+(y-0.5)^2+(z-0.5)^2-0.1875)) \\ &+ \tanh(30((x+0.5)^2+(y+0.5)^2+(z-0.5)^2-0.1875)) \\ &+ \tanh(30((x-0.5)^2+(y-0.5)^2+(z+0.5)^2-0.1875)) \\ &+ \tanh(30((x-0.5)^2+(y+0.5)^2+(z+0.5)^2-0.1875)) \\ &+ \tanh(30((x+0.5)^2+(y-0.5)^2+(z+0.5)^2-0.1875)) \\ &+ \tanh(30((x+0.5)^2+(y+0.5)^2+(z+0.5)^2)-0.1875)) \qquad (8.18) \end{aligned}$$

defined in the cube $(-2,2)\times(-2,2)\times(-2,2)$. An adaptive mesh is shown in Fig. 8.6.

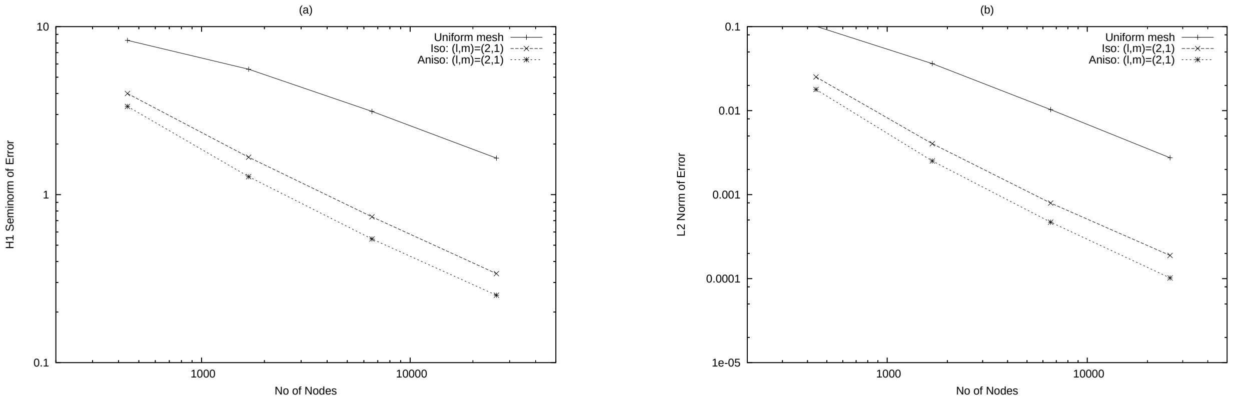


Figure 8.3: Example 8.6.1. The $H61$ semi-norm and L^2 norm of linear interpolation error are plotted as function of the number of nodes.

Table 8.1: Linear interpolation error and geometric quality for adaptive meshes obtained for Example 8.6.2.

mesh size	$\lvert e\rvert_{H^1}$	$\lVert e\rVert_\infty$	$\lVert Q_{geo}\rVert_\infty$
Isotropic mesh with M_{iso} and $(l,m) = (2,1)$			
21×21	11.1	7.98e-1	1.33
41×41	5.30	3.20e-1	1.64
81×81	3.04	2.06e-1	2.61
161×161	1.62	5.63e-2	3.20
Anisotropic mesh with $M_{ani,2}$ and $(l,m) = (2,1)$			
21×21	11.5	7.17e-1	1.82
41×41	4.75	2.89e-1	2.29
81×81	2.24	1.02e-1	4.28
161×161	1.07	1.64e-2	3.92
Uniform mesh			
21×21	12.6	9.04e-1	
41×41	7.83	7.31e-1	
81×81	5.09	3.08e-1	
161×161	2.64	9.12e-2	

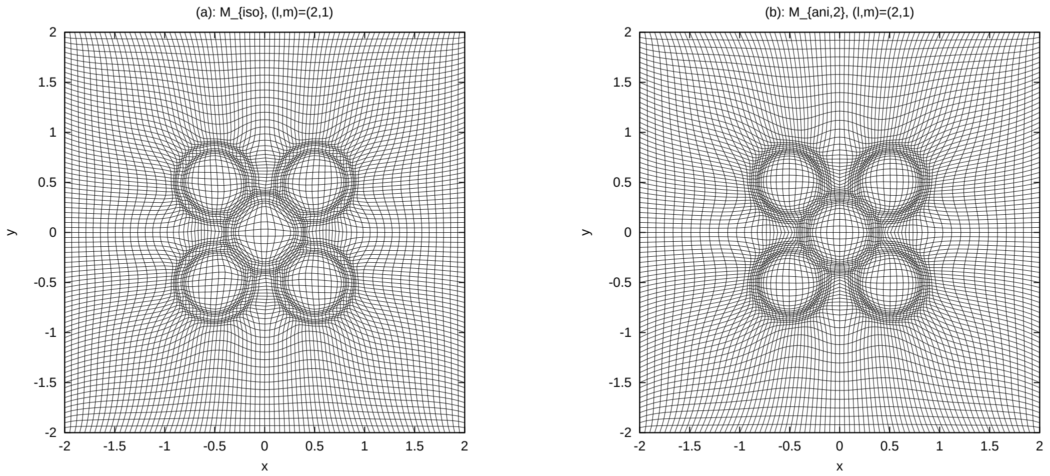


Figure 8.4: Example 8.6.2. Adaptive meshes of size 81 × 81 are obtained using the variational method with (a) $M_{ani,2}$ and $(l, m) = (2, 1)$ and (b) M_{iso} and $(l, m) = (2, 1)$.

Table 8.2: Numerical results of Example 8.6.3 obtained using the variational method. e denotes the linear interpolation error.

mesh size	$\|e\|$	$\|e\|_{H^1}$	$\|Q_{geo}\|_\infty$	$\|Q_{ali}\|_\infty$	$\|Q_{eq}\|_\infty$	Q_{mesh}	Q_{soln}
Isotropic mesh with M_{iso} and $(l, m) = (2, 1)$							
$17 \times 17 \times 17$	4.786e-2	4.049	1.2	1.2	1.3	1.2	1.5
$33 \times 33 \times 33$	8.270e-3	1.709	1.5	1.5	1.6	1.4	2.1
$65 \times 65 \times 65$	1.394e-3	0.650	1.7	1.7	1.9	1.6	2.5
Anisotropic mesh with $M_{ani,2}$ and $(l, m) = (2, 1)$							
$17 \times 17 \times 17$	4.719e-2	3.925	1.3	1.3	1.4	1.2	17
$33 \times 33 \times 33$	7.474e-3	1.608	2.0	2.2	1.8	1.7	50
$65 \times 65 \times 65$	1.671e-3	0.751	2.7	3.0	2.5	2.1	72
Uniform mesh							
$17 \times 17 \times 17$	6.872e-2	4.933					
$33 \times 33 \times 33$	2.493e-2	3.256					
$65 \times 65 \times 65$	7.190e-3	1.813					

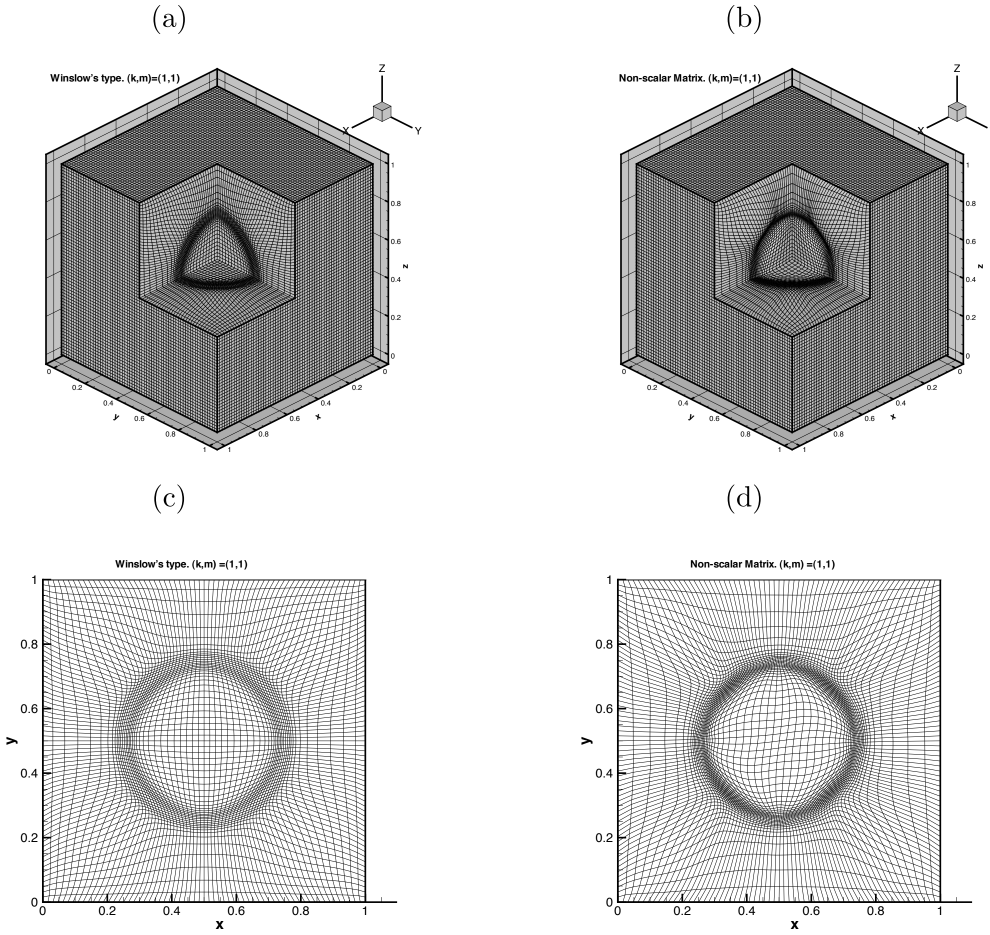


Figure 8.5: Example 8.6.3. Adaptive meshes of size $65 \times 65 \times 65$. (a): Cutaway plot of the mesh obtained with Winslow-type monitor function. (b): Cutaway plot of the mesh obtained with non-scalar monitor function. (c): Plane projection of slice at $K_z = 32$ of the mesh in (a). (d): Plane projection of slice at $K_z = 32$ of the mesh in (b).

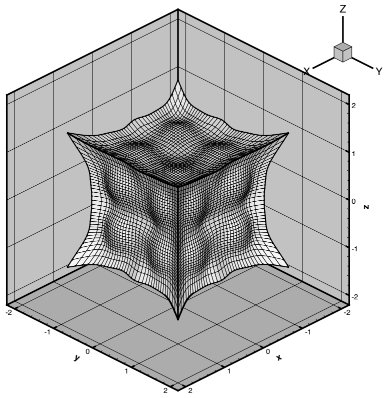


Figure 8.6: Example 8.6.4. The cutaway plot with Ix, Jy and kz between 4 to 38 for an adaptive mesh of size $41 \times 41 \times 41$ obtained with $M_{ani,2}$ and $(l, m) = (2, 1)$.

Chapter 9

Adaptive moving mesh methods: MMPDE approach

9.1 Introduction

Adaptive moving mesh methods, or moving mesh methods for short, are a type of adaptive mesh method specially designed for the numerical solution of time dependent PDEs. In a moving mesh method, the mesh nodes are typically moved or reallocated continuously, via a mesh equation, to adapt to the evolutionary feature of the physical solution. For this reason, moving mesh methods are often referred to as dynamical or r-version adaptive methods (as compared to h- and p-version methods in the context of finite elements).

Generally speaking, a moving mesh method can be distinguished from three aspects, the mesh movement strategy, the PDE discretization method, and the solution procedure for the coupled system of the physical and mesh equations. Mesh movement can be controlled by an elliptic/parabolic system of PDEs or an error-based direct minimization process. Mesh movement strategies are typically designed in the guidance of the equidistribution principle (cf. (2.4)) which requires an error function to be evenly distributed among all the mesh cells.

Discretization of the physical PDE on a moving mesh using finite differences or finite elements can be done in either the quasi-Lagrange approach or the rezoning approach. In the quasi-Lagrange approach, a physical time derivative is transformed into a time derivative along mesh trajectories, supplemented with a convective term reflecting mesh movement. The new time derivative and the extra convection term are typically treated in the same way as other terms in the physical PDE during the discretization process. On the other hand, in the rezoning approach, the physical solution is first interpolated from the old mesh to the new one and then the physical PDE is discretized on the new mesh, with the new mesh being considered fixed for the current time step. Interpolation of the physical solution is crucial to the success of this approach. A conservative interpolation scheme which preserves some quantities of the solution is often necessary.

In the rezoning approach, the physical PDE and the meshing process are naturally decoupled. On the other hand, they form a coupled system in the quasi-Lagrange approach. Solving the system can be done either simultaneously or alternately. Simultaneous solution has the advantage that the method of lines approach (MOL) and many well developed ODE solvers and computer codes can be used for integrating the system. On the other hand, alternating solution decouples the process of solving the physical PDE from the meshing process. Structures in each of the physical PDE and the mesh PDE can be fully explored and this often result in a significant improvement in efficiency. Moreover, with alternating solution the mesh generation part can be coded as a module and used with many existing PDE solvers. It should be pointed out, however, that the mesh is often generated based on the solution information at a previous time step in an alternating solution procedure. The lag in time would require use of a smaller time step size in the integration of the system. Another drawback is that the solution procedure has the risk of causing instability in the integration because it does not have a mechanism built in to force the system back to the track once the mesh is incorrectly generated at one time step. Comparably this risk is smaller with the simultaneous solution method because the physical solution and the mesh are forced to satisfy the physical PDE and the mesh equation simultaneously.

Moving mesh methods can be classified according to the mesh movement strategy into two groups [26], *velocity-based* methods and *location-based* ones. The first group is referred to as the velocity based method since it targets directly the mesh velocity and obtains mesh point locations by integrating the velocity field. Methods in this group are more or less motivated by the Lagrange method in fluid dynamics where the mesh coordinates, defined to follow fluid particles, are obtained by integrating flow velocity. A major effort in the development of these methods has been to avoid mesh tangling, an undesired property of the Lagrange method. This type of method includes those developed in [123, 90, 89, 111, 96, 84, 25]. The method of Yanenko et al. [123] is of Lagrange-type. Anderson and Rai [111] move the mesh based on attraction and repulsion pseudo-forces between nodes motivated by a spring model in mechanics. The moving finite element method (MFE) of Miller [90, 89] has aroused considerable interest. It computes the solution and the mesh simultaneously by minimizing the residual of the PDEs written in a finite element form. Penalty terms are added to avoid possible singularities in the mesh movement equations; see [28, 29]. A way of treating the singularities but without using penalty functions has been proposed by Wathen and Baines [118]. Liao and Anderson [84] and Cai et al. [23] use a deformation map. Cao, Huang, and Russell [25] develops the GCL method based on the Geometric Conservation Law (see Chapter 10). The similar idea has been used by Baines et al. [119, 10] for fluid flow problems.

The second group of moving mesh methods is referred to as the location based method because it controls directly the location of mesh points. Methods in this group typically employ an adaptation functional (cf. Chapter 8) and determines the mesh or the coordinate transformation as a minimizer of the functional. For example, the method of Dorfi and Drury [44] can be linked to a functional associated with equidistribution principle [64]. The moving mesh PDE (MMPDE) method developed in [64, 63, 66, 67, 24] moves the mesh through the gradient flow equation of an adaptation functional which includes the energy of a harmonic mapping [45] as a special example. A combination of the MMPDE method with local refinement is studied in [81]. Tang et al. [82, 83, 109] also use the energy of a harmonic mapping as their adaptation functional, but discretize the physical PDE in the rezoning approach.

So far a number of moving mesh methods and a variety of variants have been developed and successfully applied to practical problems; see review articles [115, 112, 47, 48, 56, 116, 26] and books [114, 72, 9, 124, 27, 85]. Particularly, Hawken, Gottlieb, and Hansen [56] give an extensive overview and references on moving mesh methods before 1990. In addition to the references cited above, we would also like to bring the reader's attention to the recent interesting work [21, 11, 107, 117, 13, 32, 87, 109, 125, 40, 69] on moving mesh methods and their applications.

9.2 The MMPDE method

A distinct feature of the MMPDE method is that it employs a parabolic-type PDE (called a moving mesh PDE or MMPDE) to move the mesh nodes around so that they adapt to the evolutionary feature of the physical solution. Such an MMPDE is defined as the gradient flow equation of an adaption functional. (The functional (8.9) is used in this chapter.) In the MMPDE method, the physical PDE is discretized using the quasi-Lagrange approach. The coupled system of the physical PDE and the mesh PDE is typically solved simultaneously in one dimension but alternately in higher dimensions for better efficiency.

9.2.1 MMPDE for mesh movement

The MMPDE is defined as the gradient flow equation of $I[\xi]$ given in (8.9) with the monitor function M defined in Chapter 6, i.e.,

$$\frac{\partial \xi}{\partial t} = -\frac{1}{\tau p(x,t)} \frac{\delta I}{\delta \xi}, \tag{9.1}$$

where $p = p(x,t)$ is a balancing factor, $\tau > 0$ is a parameter specified by the user for adjusting the time scale of mesh movement, and $\frac{\delta I}{\delta \xi}$ is the functional derivative of I with respect to the unknown function ξ. It is not difficult to show that $\frac{\delta I}{\delta \xi}$ is proportional to the left-hand-side term of the Euler-Lagrange equation (8.11). By interchanging the roles of dependent and independent variables, equation (9.1) can be transformed to

$$\begin{aligned}
\tau p(x,t)\frac{\partial x}{\partial t} = \quad & \theta \left[\sum_{ij} \left((\boldsymbol{a}^i)^T \bar{M}^{-1} \boldsymbol{a}^j \right) \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} - \sum_i \left((\boldsymbol{a}^i)^T \sum_j \frac{\partial (\bar{M}^{-1})}{\partial \xi_j} \boldsymbol{a}^j \right) \frac{\partial x}{\partial \xi_i} \right] \\
+ \quad & \frac{\theta(n\gamma - 2)}{2\beta} \left[2 \sum_{ij} \left((\bar{M}^{-1}\boldsymbol{a}^i)(\bar{M}^{-1}\boldsymbol{a}^j)^T \sum_k \boldsymbol{a}^k (\boldsymbol{a}^k)^T \right) \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} \right. \\
& \left. - \sum_i \left(\sum_j \left((\boldsymbol{a}^i)^T \bar{M}^{-1} \boldsymbol{a}^j \sum_k (\boldsymbol{a}^k)^T \frac{\partial (\bar{M}^{-1})}{\partial \xi_j} \boldsymbol{a}^k \right) \right) \frac{\partial x}{\partial \xi_i} \right] \\
+ \quad & \frac{(1-2\theta)2(\gamma-1) n^{\frac{n\gamma}{2}} \rho}{n \beta^{\frac{n\gamma}{2}-1} (J\rho)^\gamma} \left[\sum_{ij} \left(\boldsymbol{a}_i (\boldsymbol{a}^j)^T \right) \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} + \sum_i \left(\frac{1}{\rho} \frac{\partial \rho}{\partial \xi_i} \right) \frac{\partial x}{\partial \xi_i} \right].
\end{aligned} \tag{9.2}$$

It can be written into

$$\frac{\partial x}{\partial t} = \frac{1}{\tau p(x,t)} \left[\sum_{i,j} C_{i,j} \frac{\partial^2 x}{\partial \xi_i \partial \xi_j} + \sum_i b_i \frac{\partial x}{\partial \xi_i} \right], \tag{9.3}$$

where $C_{i,j}$'s are $n \times n$ matrices and b_i's are scalar. It is recommended in [58] to choose the balancing factor as

$$p(x,t) = \left(\sum_i \left((\boldsymbol{a}^i)^T \bar{M}^{-1} \boldsymbol{a}^i \right)^2 + \sum_i b_i^2 \right)^{1/2}.$$

But it is unclear what the optimal value for τ is. Experience shows that choosing τ in the range $[10^{-3}, 0.1]$ works for most of problems.

When the computational domain Ω_c has a simple geometry and a tensor-product type mesh is chosen thereon, the spatial discretization of the MMPDE (9.3) via finite differences is straightforward [65, 67]. However, when Ω_c has a simplex geometry, finite elements can be used for (9.3); see [24]. For time integration, low accuracy schemes can be used. For example, the backward Euler method has been used in [65, 67], with the coefficients $C_{i,j}$ and b_i calculated at the previous time step. The linear algebraic system resulting from the semi-implicit discretization has been solved using a preconditioned conjugate residual method. The preconditioner is constructed as an ILU preconditioner using level-1 fill-ins.

One may notice that the monitor function M in Chapter 6 involves the derivatives of the solution of the physical PDE. As in Chapters 7 and 8, these derivatives can be computed using a gradient recovery technique, such as those in [126, 127, 128], based on approximations of the nodal values of the physical solution at the previous time step. Since the so recovered derivatives are often non-smooth, to obtain a smoother mesh and also make the MMPDE easier to integrate, it is common practice in moving mesh methods to smooth the computed monitor function. For example, the following averaging scheme can be applied several times to the monitor function,

$$M_{j,k,l} \leftarrow \frac{\sum_{\hat{j}=j-1}^{j+1} \sum_{\hat{k}=k-1}^{k+1} \sum_{\hat{l}=l-1}^{l+1} 2^{-|j-\hat{j}|-|k-\hat{k}|-|l-\hat{l}|} M_{\hat{j},\hat{k},\hat{l}}}{\sum_{\hat{j}=j-1}^{j+1} \sum_{\hat{k}=k-1}^{k+1} \sum_{\hat{l}=l-1}^{l+1} 2^{-|j-\hat{j}|-|k-\hat{k}|-|l-\hat{l}|}}.$$

The MMPDE has to be completed with boundary conditions. The most straightforward way is to fix the boundary points. They can also be moved using a lower dimensioanl MMPDE with the corresponding monitor function being projecting to faces or lines.

9.2.2 Discretization on a moving mesh

Consider the PDE in the general form

$$v_t = F(t, x, v), \quad \forall x \in \Omega \tag{9.4}$$

where F is a differential operator. We are concerned with the quasi-Lagrange approach of the spatial discretization on a moving mesh.

PDE (9.4) can be discretized via finite differences or finite elements by first transforming from the physical domain to the computational domain. Indeed, let $\hat{v}(\xi, t) = v(x(\xi, t), t)$. Differentiating this with respect to t while keeping ξ fixed gives

$$\frac{\partial \hat{v}}{\partial t} = \frac{\partial v}{\partial t} + \nabla v \cdot \frac{\partial x}{\partial t} = \frac{\partial v}{\partial t} + \nabla_\xi \hat{v} \cdot \left(\frac{\partial \xi}{\partial x} \frac{\partial x}{\partial t} \right), \tag{9.5}$$

where $\frac{\partial \xi}{\partial x}$ is the Jacobian matrix of $\xi = \xi(x, t)$ and ∇ and ∇_ξ denote the gradient operators with respect to x and ξ, respectively. Note that $\frac{\partial \hat{v}}{\partial t}$ is the time derivative along the mesh trajectories and the convective term $\nabla_\xi \hat{v} \cdot \left(\frac{\partial \xi}{\partial x} \frac{\partial x}{\partial t} \right)$ reflects the mesh movement. With relation (9.5), PDE (9.4) can be written as

$$\frac{\partial \hat{v}}{\partial t} - \nabla_\xi \hat{v} \cdot \left(\frac{\partial \xi}{\partial x} \frac{\partial x}{\partial t} \right) = F(t, x(\xi, t), \hat{v}(\xi, t)), \qquad \xi \in \Omega_c. \tag{9.6}$$

The finite difference or finite element discretization for (9.6) on the computational domain is standard. The resulting ODE system can be integrated by any time marching scheme.

PDE (9.4) can also be discretized directly on a moving mesh using finite elements. Let $\{\mathcal{T}_h(t)\}$ be an affine family of moving meshes. Consider a discretization with an affine family of finite elements of an arbitrary order $k \geq 0$. By definition (for the affine family of finite elements) any finite element $(K(t), P_K, \Sigma_K)$ is affine-equivalent to the master finite element $(\hat{K}, P_{\hat{K}}, \Sigma_{\hat{K}})$. Consequently, for any basis function ϕ_i whose support contains $K(t)$ there exists a basis function $\hat{\phi}_{\hat{i}} \in P_{\hat{K}}$ such that

$$\phi_i(x,t) = \hat{\phi}_{\hat{i}}(F_K^{-1}(x,t)), \qquad \forall x \in K(t) \tag{9.7}$$

where $\xi = F_K^{-1}(x,t)$ denotes the inverse of affine mapping $x = F_K(\xi, t) : \hat{K} \to K(t)$. Differentiating (9.7) with respect to t while keeping x fixed, we have

$$\frac{\partial \phi_i}{\partial t} = \nabla_\xi \hat{\phi}_{\hat{i}} \cdot \frac{\partial F_K^{-1}}{\partial t}. \tag{9.8}$$

By differentiating the equality $\xi = F_K^{-1}(F_K(\xi,t),t)$ with respect to t we have

$$0 = \frac{\partial F_K^{-1}}{\partial t} + \frac{\partial F_K^{-1}}{\partial x}\frac{\partial F_K}{\partial t}. \tag{9.9}$$

Combining (9.8) with (9.9) gives rise to

$$\begin{aligned} \frac{\partial \phi_i}{\partial t} &= -\nabla_\xi \hat{\phi}_{\hat{i}} \cdot \frac{\partial F_K^{-1}}{\partial x}\frac{\partial F_K}{\partial t} \\ &= -\left(\frac{\partial F_K^{-1}}{\partial x}\right)^T \nabla_\xi \hat{\phi}_{\hat{i}} \cdot \frac{\partial F_K}{\partial t} \\ &= -\nabla \hat{\phi}_{\hat{i}} \cdot \frac{\partial F_K}{\partial t}, \end{aligned}$$

where we have used $\nabla = \left(\frac{\partial F_K^{-1}}{\partial x}\right)^T \nabla_\xi$. From (9.7) it follows

$$\frac{\partial \phi_i}{\partial t} = -\nabla \phi_i \cdot \frac{\partial F_K}{\partial t}. \tag{9.10}$$

We now proceed with the direct finite element discretization. Since Dirichlet boundary conditions can be incorporated by modifying the final algebraic equations, without loss of generality we assume here that only Neumann boundary conditions are present. More specifically, we assume that PDE (9.4), supplemented with boundary conditions, has the weak form

$$\int_\Omega v_t \phi dx = \mathcal{F}(t, v, \phi) \tag{9.11}$$

where $\mathcal{F}$ is a functional resulting from the right-hand side term of (9.4) and ϕ is an admissible test function. Approximate v by

$$v^h = \sum_i v_i(t)\phi_i(x,t). \tag{9.12}$$

Differentiating it with respect to t and using (9.8) we have

$$\begin{aligned} v_t^h &= \sum_i \frac{dv_i}{dt}(t)\phi_i(x,t) + \sum_i v_i(t)\frac{\partial \phi_i}{\partial t}(x,t) \\ &= \sum_i \frac{dv_i}{dt}(t)\phi_i(x,t) - \sum_i v_i(t)\nabla\phi_i(x,t)\cdot\frac{\partial F_K}{\partial t} \\ &= \sum_i \frac{dv_i}{dt}(t)\phi_i(x,t) - \nabla v^h \cdot \dot{X}(x,t), \end{aligned} \tag{9.13}$$

where $\dot{X}(x,t)$ denotes a linear polynomial interpolating mesh speed at the vertices and satisfies

$$\dot{X}(x,t) = \frac{\partial F_K}{\partial t}(F_K^{-1}(x,t),t), \qquad \forall x \in K \text{ and } \forall K \in \mathcal{T}_h.$$

It is interesting to point out that (9.13) is a finite element analog of the continuous relation (9.5).

Inserting (9.12) and (9.13) into (9.9) and taking ϕ to be each (say ϕ_k) of the basis functions, we obtain the ODE system

$$\int_\Omega \left(\sum_i \frac{dv_i}{dt}(t)\phi_i(x,t) - \nabla v^h\cdot\dot{X}(x,t)\right)\phi_k dx = \mathcal{F}(t,v^h,\phi_k), \tag{9.14}$$

which can be integrated using any time marching scheme.

9.2.3 Alternating solution procedure

In principle, the physical PDE (9.6) (or (9.14)) and the MMPDE (9.3) can be integrated in time either simultaneously or alternately. However, alternating solution seems more realistic in multi-dimensions since it voids the highly nonlinear coupling of the mesh and physical solution and preserves many structures such as ellipticity and sparsity in each of the mesh and physical PDEs. For illustrative purpose, we give an alternating procedure in the following. Here, Δt and Δt_{mesh} are the time step sizes associated with the physical PDE and the MMPDE, respectively.

Alternating Solution Procedure: Assume that the physical solution v^n, the mesh x^n, and a time step size Δt_n are given at time $t = t_n$.

(i) Compute the monitor function $M^n(x) = M(t_n, x)$ using v^n and x^n. The solution derivatives used in M (cf. Chapter 6) are calculated using a gradient recovery technique based on the nodal values of the computed solution.

(ii) Integrate the MMPDE (9.3) over the time period $[t_n, t_n+\Delta t_n]$ using variable step size $\Delta t_{mesh,n}$ and monitor function $M^n(x)$. The MMPDE is discretized in time using the backward Euler scheme with the coefficients $C_{i,j}$ and b_i being calculated at t_n. More than one sub-step may be used for the integration to reach $t = t_n + \Delta t_n$. When this happens, the monitor function is updated for each sub-step via linear interpolation. The obtained mesh is denoted by x^{n+1}.

(iii) Integrate the physical PDE (9.6) with a fixed or variable step size. The equation can be discretized in time using, say, the Singly Diagonally Implicit Runge-Kutta scheme [30]. The mesh and mesh velocity are calculated using linear interpolation:

$$x(t) = \frac{t - t_n}{\Delta t_n}x^{n+1} + \frac{t_n + \Delta t_n - t}{\Delta t_n}x^n. \tag{9.15}$$

(iv) When a variable step size is used in step (iii), the physical PDE may actually be integrated over a smaller step $\hat{\Delta} t_n < \Delta t_n$. In this case, the mesh at the actual new time level $t_{n+1} = t_n + \hat{\Delta} t_n$ should be updated as $x^{n+1} := x(t_{n+1})$ using (9.15).

(v) Go to the next time step with the step size predicted by the physical PDE solver.

9.3 Example examples

Example 9.3.1. The first example is an initial-boundary value problem (IBVP) of the well-known Burgers equation,

$$v_t = \epsilon v_{xx} - \left(\frac{v^2}{2} \right)_x, \quad x \in (0,1),\, t > 0 \tag{9.16}$$

subject to the boundary conditions

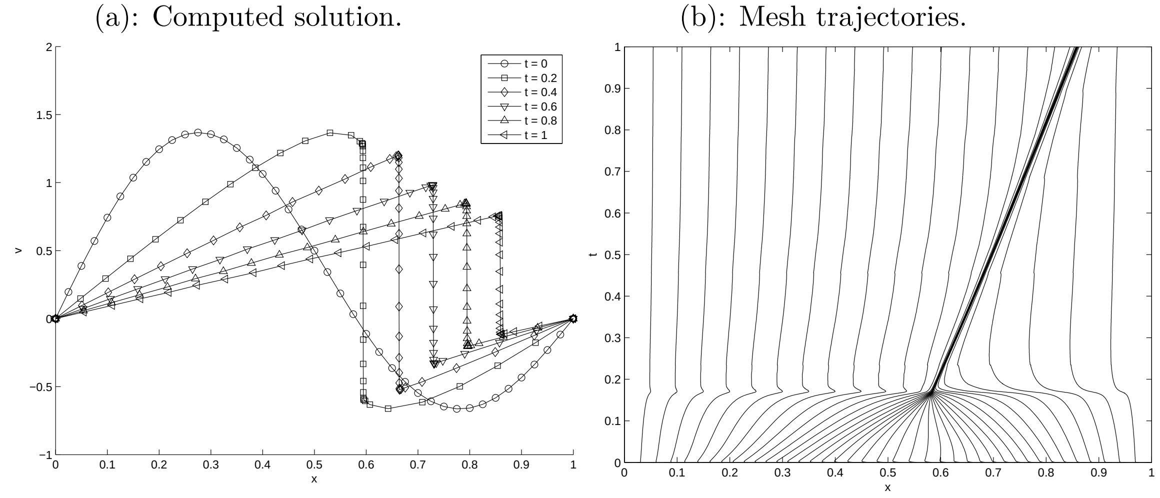
$$v(0,t) = v(1,t) = 0 \tag{9.17}$$

and the initial condition

$$v(x,0) = \sin(2\pi x) + \frac{1}{2}\sin(\pi x). \tag{9.18}$$

Here $\epsilon > 0$ is a physical parameter. It is known that, for small ϵ, the solution of the IBVP starts with a smooth profile and then develops a steep front, which propagates toward the right end and eventually dies out due to the homogeneous Dirichlet boundary condition at the right end. The difficulty in the numerical solution of this model problem lies in the resolution of the sheep front and its propagation.

Fig. 9.1 shows a computed solution at various time instants and the corresponding mesh trajectories. This result is obtained with the finite difference method. For mesh movement, $\tau = 10^{-2}$ and the monitor function is chosen for the case $(l,m) = (2,1)$. For comparison, the computed solution with a uniform mesh of 2001 nodes is shown in Fig. 9.2.


(a): Computed solution.

(b): Mesh trajectories.

Figure 9.1: Example 9.3.1. (a): Computed solutions at $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0 obtained with an adaptive moving mesh of 41 points for Burgers' equation with $\epsilon = 10^{-4}$. (b): The mesh trajectories.

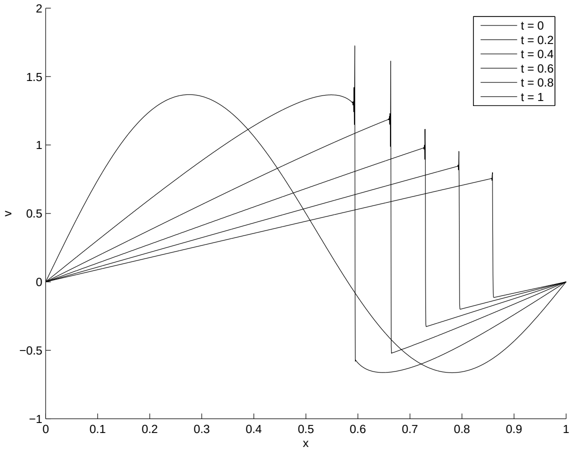


Figure 9.2: Example 9.3.1. Computed solutions at $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0 obtained with a uniform mesh of 2001 points for Burgers' equation with $\epsilon = 10^{-4}$.

Example 9.3.2. The second example is the two dimensional Burgers equation

$$\frac{\partial v}{\partial t} = 0.005\Delta v - v\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y}, \quad t \in (0.25, 1.5],\, x \in \Omega \equiv (0,1) \times (0,1). \tag{9.19}$$

The initial and Dirichlet boundary conditions are chosen so that the problem has exact solution $v(x, y, t) = (1 + e^{100(x+y-t)})^{-1}$. This solution describes a straight-line wave (v is constant along line $x + y = c$) moving in the direction $\theta = 45^o$.

Adaptive moving meshes and solution error obtained with a MMPDE finite difference method are shown in Figs. 9.3 and 9.4.

Example 9.3.3. The third example is a model of multiphase flow and transport in groundwater environment [69]. It simulates the dissolution of nonaqueous phase liquids (NAPLs) into the aqueous phase [88]. The physical process is described by two PDEs, one for the volumetric fraction of NAPL or NAPL content,

$$\frac{\partial \theta_n}{\partial t} = -\frac{k_{na}\,(C_a^* - C_a)}{\rho_n}, \tag{9.20}$$

and the other for the NAPL dissolved in water,

$$\frac{\partial(\theta_a C_a)}{\partial t} = \nabla \cdot (D\nabla C_a - q_a C_a) + k_{na}\,(C_a^* - C_a), \tag{9.21}$$

where the subscripts "a" and "n" represent the aqueous and nonaqueous phases, respectively, the superscript "$*$" indicates an equilibrium condition with the companion phase involved in the mass transfer, $\theta = \theta(t, x, y)$ is the volumetric fraction, $C_a = C_a(t, x, y)$ is the concentration of the NAPL dissolved in water, ρ is density, k_{na} is a mass transfer coefficient representing a mass transfer process referenced to a loss by the nonaqueous phase and a gain by the aqueous phase, q_a is water flux, and D is the dispersivity. It is noted that $\theta_n + \theta_a = n$, where n is the porosity considered to be constant here. The reader is referred to [88] for the derivations of the governing equations, the corresponding initial and boundary conditions and the physical parameters. The physical scenario is that the aqueous phase is being flushed from the left boundary and the dissolved NAPL is being

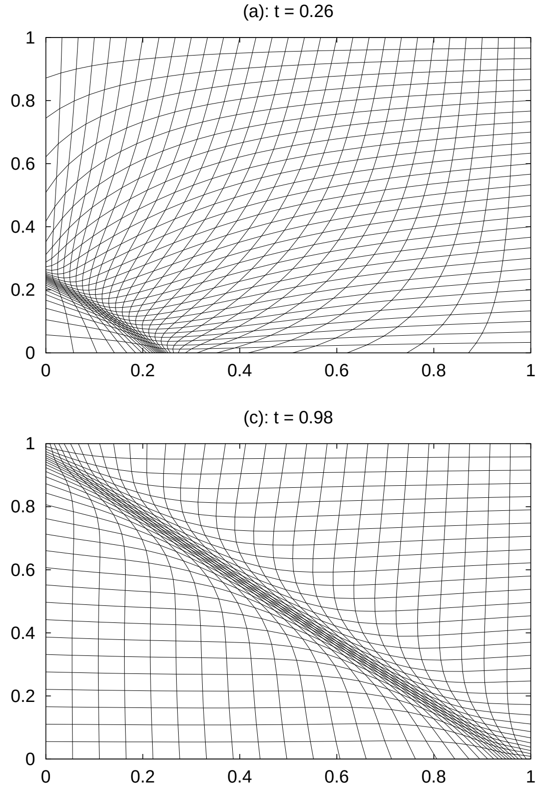


Figure 9.3: Example 9.3.2. A typical moving mesh at time instants (a) $t = 0.26$, (b) $t = 0.5$, (c) $t = 0.98$, and $t = 1.46$.

eluted from the right boundary. The left and right boundary conditions are a specified flux for the aqueous phase, while the top and bottom boundary conditions are no-flow. The initial residual NAPL saturation and other parameters are homogeneous, a typical laboratory condition, except that a perturbation in the residual NAPL saturation near the left boundary where a portion of the boundary is NAPL free, indicating that a clean water is flushing in. The development of this perturbation into a dissolution profile is then observed.

The obtained results are shown in Fig. 9.5 for mesh evolution, NAPL, and dissolved NAPL in water. It can be observed that a clean inflow from the west boundary washes out NAPL and reduces dissolved NAPL in water in a channel zone with time. The movements of the front and the boundary of the channel are captured correctly with the adaptive mesh.

Example 9.3.4. Our next example is a combustion problem considered in [67, 92]. The mathematical model is a system of coupled nonlinear reaction-diffusion equations

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla^2 u &= -\frac{R}{\alpha\delta} u e^{\delta(1-1/T)}, \\ \frac{\partial T}{\partial t} - \frac{1}{Le}\nabla^2 T &= \frac{R}{\delta Le} u e^{\delta(1-1/T)}, \end{aligned} \tag{9.22}$$

where u and T represent the dimensionless concentration and temperature of a chemical which is undertaking a one-step reaction. We consider the J-shape solution domain shown in Fig. 9.6. The

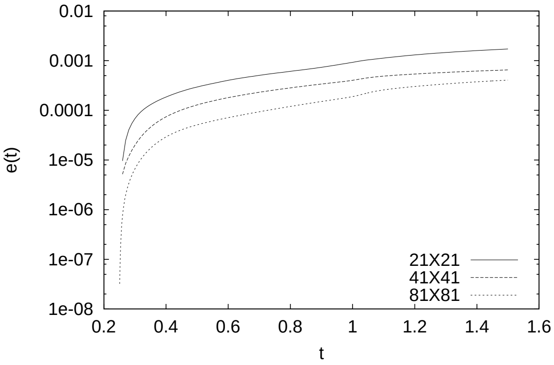


Figure 9.4: Example 9.3.2. The global L^2 error $e(t) = \int_0^t \|u - u^{compt}\|_2 dt$ is plotted as function of t for three meshes.

initial and boundary conditions are

$$\begin{array}{ll} u|_{t=0} = T|_{t=0} = 1, & \text{in } \Omega, \\ u|_{\partial\Omega} = T|_{\partial\Omega} = 1, & \text{for } t > 0 \end{array} \qquad (9.23)$$

and the physical parameters are set to be $Le = 0.9, \alpha = 1, \delta = 20$, and $R = 5$. A result obtained with a MMPDE finite element method [24] is shown in Fig. 9.6.

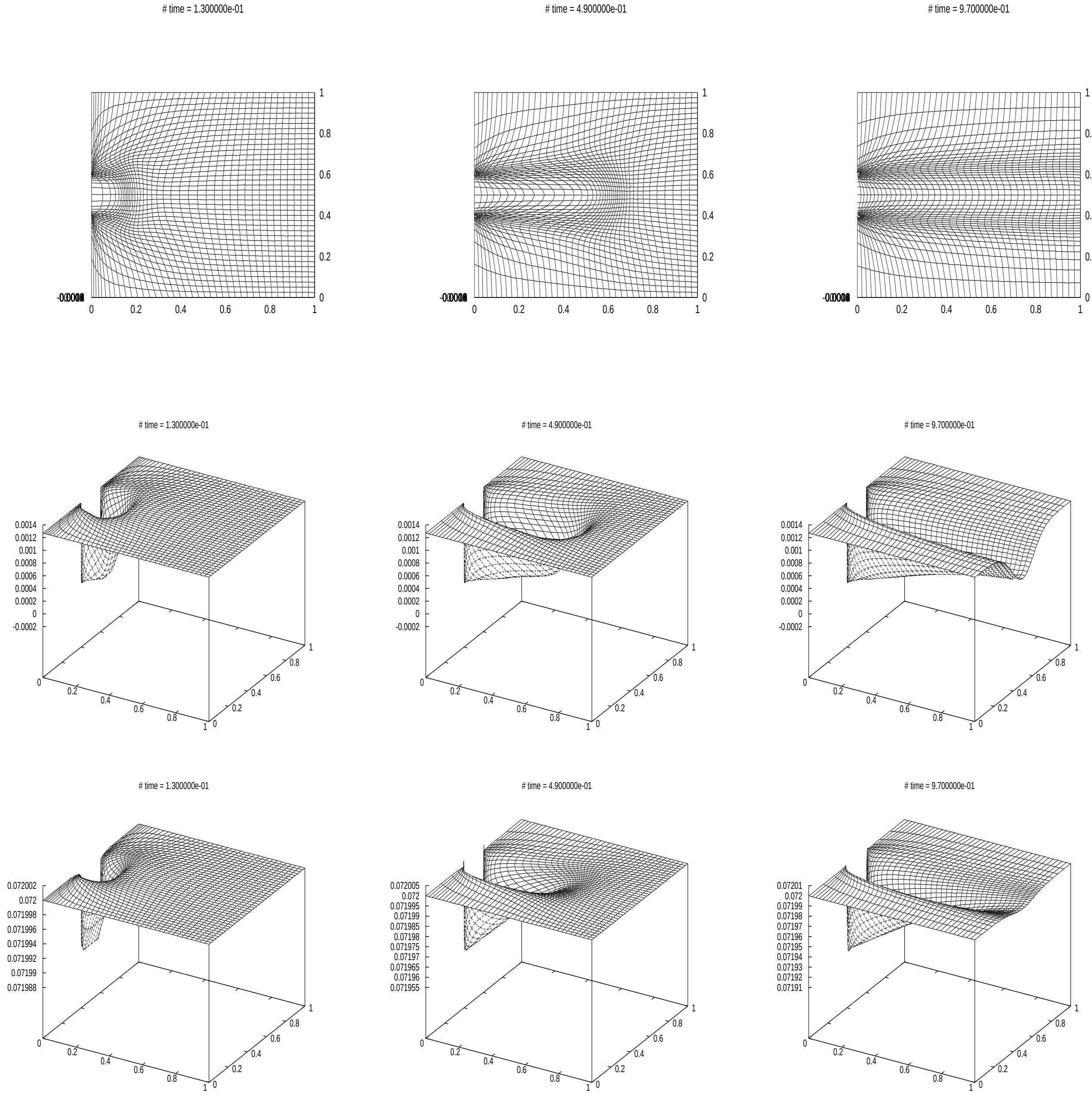


Figure 9.5: Example 9.3.3. Adaptive moving mesh (first row), NAPL (second row), and dissolved NAPL in water (third row) for the NAPL problem.

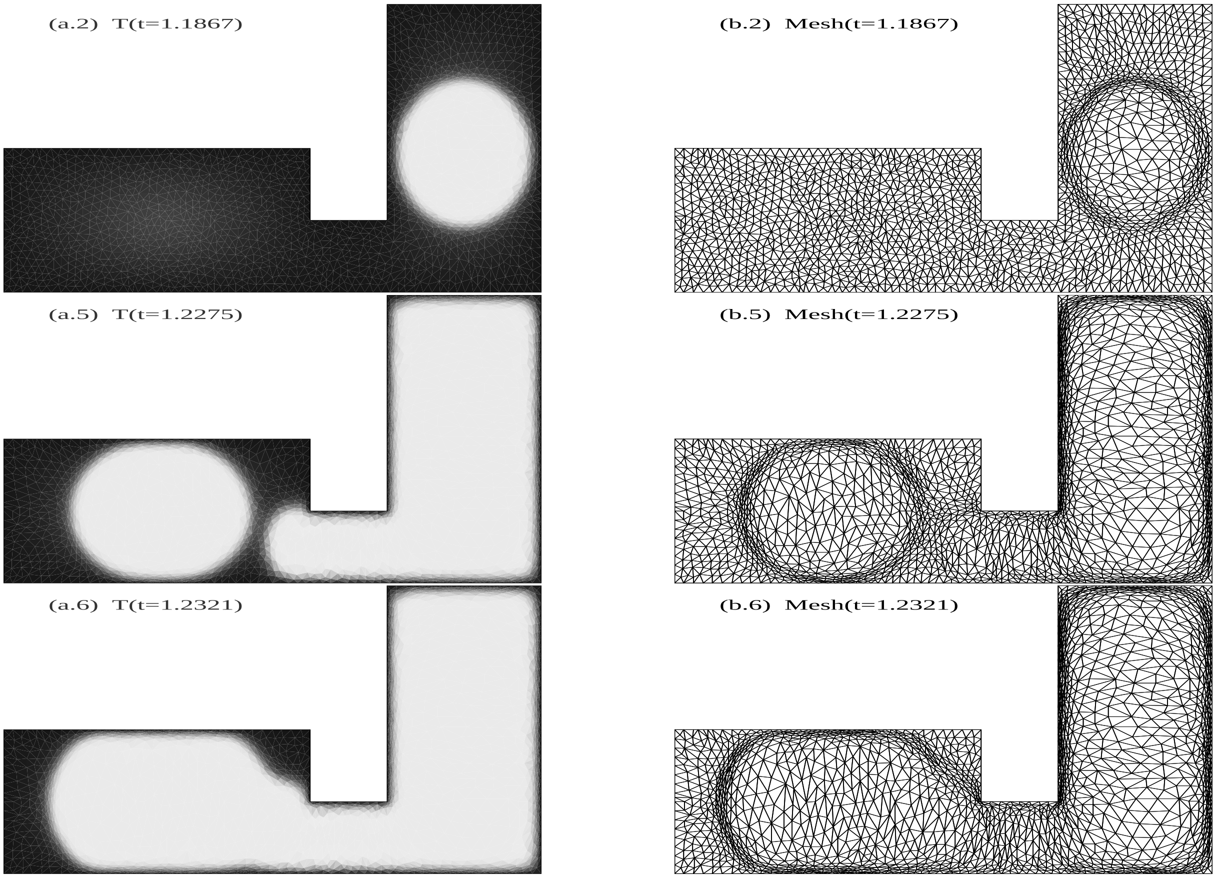


Figure 9.6: Example 9.3.4. The contour plot of the temperature T (where white represents 2.2 and black represents 1) and the moving mesh are shown at various times.

Chapter 10

Adaptive moving mesh methods: GCL approach

10.1 Introduction

This chapter is devoted to the description of the GCL method – a velocity-based moving mesh method. The method was developed in [25] based on the Geometric Conservation Law [110], and includes the Lagrange method in computational fluid dynamics and the deformation map method developed by Liao and coworkers [84, 102, 15, 23] as special examples. The GCL method shares many common properties with the Lagrange method, including advantages and disadvantages. The method has not attracted much attention as location-based methods. But it has shown promises for a certain class of problems in the recent work by Baines and his coworkers [10, 119].

10.2 GCL method

The original development of the GCL method [25] was based on the Geometric Conservation Law [110]. The method is derived here in a slightly different way, partially motivated by work of Baines et al. [10, 119].

The focus is on the equidistribution condition (2.4),

$$J\rho = \frac{\sigma}{|\Omega_c|}, \tag{10.1}$$

where J is the Jacobian of the coordinate transformation $x = x(\xi) : \Omega_c \to \Omega$, $\rho = \rho(x,t)$ is a given adaptation function, and $\sigma = \sigma(t) = \int_\Omega \rho(x,t)dx$. Let $K(t)$ be an arbitrary, time dependent sub-domain of Ω and K_c be the corresponding fixed sub-domain in Ω_c under $x = x(\xi)$. Integrating (10.1) over K_c yields

$$\int_{K(t)} \frac{\rho}{\sigma} dx = \frac{|K_c|}{|\Omega_c|}. \tag{10.2}$$

Differentiating this with respect to time, we have

$$\frac{d}{dt}\int_{K(t)} \frac{\rho}{\sigma} dx = 0$$

or

$$\int_{K(t)} \frac{\partial}{\partial t}\left(\frac{\rho}{\sigma}\right) dx + \int_{\partial K(t)} \frac{\rho}{\sigma}\dot{x} \cdot ndS = 0, \tag{10.3}$$

where the mesh speed $\dot{x}$ characterizes the movement of surface $\partial K(t)$ and n is the outward normal to the surface. Using Gauss' theorem, (10.3) can be written as

$$\int_{K(t)} \frac{\partial}{\partial t}\left(\frac{\rho}{\sigma}\right) dx + \int_{K(t)} \nabla \cdot \left(\frac{\rho}{\sigma}\dot{x}\right) dx = 0.$$

It follows from the arbitrariness of $K(t)$ that

$$\frac{\partial}{\partial t}\left(\frac{\rho}{\sigma}\right) + \nabla \cdot \left(\frac{\rho}{\sigma}\dot{x}\right) = 0. \tag{10.4}$$

This is basically the continuity equation in fluid dynamics, with "flow velocity" $\dot{x}$ and "fluid density" ρ/σ. It can also be considered as a condition for determining the divergence of mesh speed $\dot{x}$. Thus, the equidistribution condition (10.1) determines the divergence of $\dot{x}$.

Note that equation (10.4) is insufficient to determine the vector field $\dot{x}$. The motivation for finding supplementary conditions is provided by the Helmholtz decomposition theorem for vectors: A continuous and differentiable vector field can be decomposed into the orthogonal sum of a gradient of a scalar field and the curl of a vector field. Therefore, $\dot{x}$ can be determined by specifying both its divergence through (10.4) and its curl. We require $\dot{x}$ to satisfy

$$\nabla \times w(\dot{x} - v_{ref}) = 0, \tag{10.5}$$

where $w > 0$ is a weight function and v_{ref} is a user-specified reference vector field. Different choices for w and v_{ref} lead to different curl conditions for the vector field $\dot{x}$. Their choices will be discussed in §10.4.

The requirement (10.5) implies that there exists a potential function ϕ such that

$$w(\dot{x} - v_{ref}) = \nabla\phi$$

or

$$\dot{x} = \frac{1}{w}\nabla\phi + v_{ref}. \tag{10.6}$$

Inserting this into (10.4) leads to

$$\nabla \cdot \left(\frac{\rho}{w\sigma}\nabla\phi\right) = -\frac{\partial}{\partial t}\left(\frac{\rho}{\sigma}\right) - \nabla \cdot \left(\frac{\rho}{\sigma}v_{ref}\right) \qquad \text{in } \Omega. \tag{10.7}$$

The boundary condition can be obtained by requiring the mesh points not to move out the domain, i.e., $\dot{x} \cdot n = 0$ where n denotes the outward normal to $\partial\Omega$. Using (10.6), we have the boundary condition

$$\frac{\partial\phi}{\partial n} = -w\, v_{ref} \cdot n \qquad \text{on } \partial\Omega. \tag{10.8}$$

To summarize, the potential function ϕ is determined by the elliptic equation (10.7) supplemented with the Neumann boundary condition (10.8). Once ϕ is known, the mesh location can be obtained by integrating (10.6).

10.3 Relation to the Lagrange method and the deformation map method

The GCL method described in the previous section can be regarded as a generalization of the Lagrange method when choosing the reference vector field v_{ref} to be the flow velocity v_f. Consider a special case where incompressible fluid does not flow in or out of the domain, meaning that $\nabla \cdot v_f = 0$ in Ω and $v_f \cdot n = 0$ on $\partial\Omega$. Taking $\rho =$ constant (no adaption), we have $\phi =$ constant from (10.7) and a pure Lagrange method $\dot{x} = v_f$ results from (10.6).

We now discuss the relation of the GCL method with the deformation map method. The deformation map was introduced by Moser [36, 93] in his study of volume elements of a compact Riemannian manifold to prove the existence of a C^1 diffeomorphism with a specified Jacobian. It has been adopted by Liao and coworkers [84, 102, 15, 23] for generating adaptive moving meshes. The method can be rewritten in the current notation as

$$\begin{aligned} \nabla \cdot \left(\frac{\rho}{\sigma}\dot{x}\right) &= -\frac{\partial}{\partial t}\left(\frac{\rho}{\sigma}\right), \\ \nabla \times \left(\frac{\rho}{\sigma}\dot{x}\right) &= 0. \end{aligned} \tag{10.9}$$

It is easy to see that the deformation map method is a special case of the GCL method with $w = \rho/\sigma$ and $v_{ref} = 0$.

Note that both the deformation map method and the GCL method control the Jacobian J of the mapping in the same way, so the cell sizes produced by the two methods are the same. However, the extra freedom given by v_{ref} and w in the GCL method can be used to provide better control of the mesh behavior; cf. Figs. 10.1 and 10.2. For instance, in practical computation it is generally preferable to have an irrotational mesh velocity $\dot{x}$, which can result in less skewed grids. This can be achieved with the GCL method by choosing $w = 1$. But, on the other hand, an irrotational mesh velocity field is generally impossible with the deformation method, since

$$\nabla \times \left(\frac{\rho}{\sigma}\dot{x}\right) = 0$$

implies

$$\nabla \times \dot{x} = -\frac{1}{\rho}\nabla\rho \times \dot{x},$$

and therefore $\nabla \times \dot{x}$ does not vanish in general.

Like the GCL method, the deformation map method satisfies the relation (10.1) for any dimension. Thus, the Jacobian of the mapping stays positive, and the mapping itself is locally nonsingular. This is a very advantageous feature of the deformation method (and the GCL method) since for most methods it is extremely difficult to prove that they produce non-singular coordinate transformations or meshes.

10.4 Choice of w, v_{ref}, and ρ

As mentioned in the previous section, there are two more or less obvious choices for the weight function, $w = \rho/\sigma$ and $w = 1$. The former corresponds to the deformation map method and generally does not result in an irrotational mesh velocity field. The latter results in an irrotational mesh velocity field when $v_{ref} = 0$. Of course, numerous other options are also possible, but their mathematical and/or physical significance is unclear so far.

The choice of the control vector field v_{ref} is very problem dependent. For fluid dynamics problems, a good choice of it can be the flow velocity, since this is likely to reduce the magnitude of the convection term. However, when mesh adaption is allowed (i.e., ρ is not constant), grid movement due to the mesh adaption may increase the convection term and make v_{ref} more difficult to choose. Generally speaking, when physical intuition for choosing v_{ref} is not available, the best option is simply to choose $v_{ref} = 0$.

The choice of the adaptation function ρ should generally be based on the equidistribution principle (2.4) or (10.1). Consequently, it is natural to define it as the square-root of the determinant of the monitor function given in Chapter 6, i.e.,

$$\rho = \sqrt{\det(M)}.$$

It can also be chosen based on other considerations, such as the scaling invariance argument used in [10].

10.5 Numerical examples

Example 10.5.1. The first example is to generate a moving mesh for the given adaptation function

$$\rho(x,y,t) = \begin{cases} 1 + 100(t+0.1)\exp(-50|(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2 - 0.09|) \\ \qquad \text{for } -0.1 < t < 0, \\ 1 + 10\exp(-50|(x-\frac{1}{2}-t)^2 + (y-\frac{1}{2})^2 - 0.09|) \\ \qquad \text{for } t \geq 0. \end{cases} \tag{10.10}$$

This function is defined using two time phases, one from $t = -0.1$ to $t = 0$ and the other for $t > 0$. The purpose of the first phase is to produce an adaptive mesh for $t = 0$, starting from a uniform mesh at $t = -0.1$. For $t > 0$, the function simulates a circular peak which moves right at speed 1 and eventually leaves the domain while maintaining its shape. Numerical results are shown in Figs. 10.1 and 10.2.

Example 10.5.2. This example is to generate a moving mesh for the adaptation function

$$\rho(x,y,t) = \begin{cases} 1 + 50(0.1+t)\exp(-50|(x-\frac{3}{4})^2 + (y-\frac{1}{2})^2 - .01|), \\ \qquad \text{for } -0.1 < t < 0 \\ 1 + 5\exp(-50|(x-\frac{1}{2}-\frac{1}{4}\cos(2\pi t))^2 + (y-\frac{1}{2}-\frac{1}{4}\sin(2\pi t))^2 - .01|), \\ \qquad \text{for } t \geq 0. \end{cases} \tag{10.11}$$

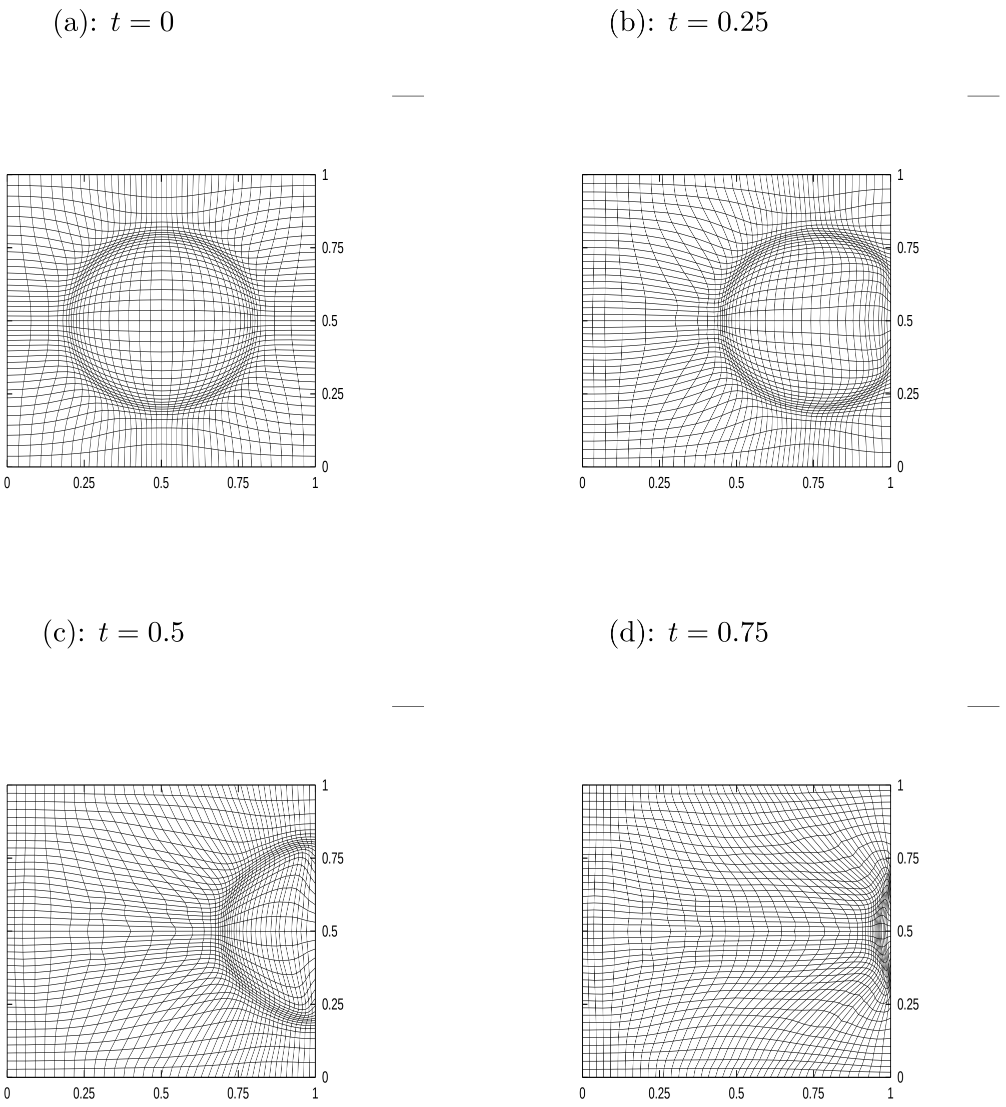
For this monitor function, the largest values of ρ occur around a small circle which rotates about the point $(\frac{1}{2}, \frac{1}{2})$.

This is a very difficult test problem for many moving mesh methods, especially for ones with close similarity to the Lagrange method. With these methods, as the mesh points and concentration follow the small moving circle rotating around the point $(\frac{1}{2}, \frac{1}{2})$, if some of the boundary mesh points stay fixed (as in the current case where the four corner points are fixed), then the mesh becomes more and more skewed and eventually singular. Unfortunately, the present moving mesh based on the GCL also suffers from this difficulty. Although the GCL condition guarantees non-singularity of the Jacobian of the coordinate transformation in the continuous case, it does not prevent the mesh from becoming increasingly skew. Meanwhile, the points of a highly skewed mesh can easily tangle each other numerically. This is illustrated in Fig.10.3.

Example 10.5.3. This example illustrates the effect of the reference vector field v_{ref} on the moving mesh. As discussed in §10.3, the GCL method reduces to a pure Lagrange method when $\rho =$ constant and v_{ref} is divergence free inside the domain and satisfies $v_{ref} \cdot n = 0$ on the boundary. A special example is where the mesh velocity is given by the angular direction v_{ref} for a unit disk Ω. However, the mesh movement is not so obvious if $v_{ref} \cdot n$ is not zero on the boundary and/or some of the boundary mesh points such as corner points stay fixed. To see this, consider the case where $w = 1$, $\rho = 1$, and

$$v_{ref} = 2\pi(-y + \frac{1}{2},\ x - \frac{1}{2}). \tag{10.12}$$

The reference vector field is a divergence free rotation around the center of the domain, with angular speed 1. In Fig. 10.4 the moving mesh is shown at times $t = 0, 0.1, 0.2$. Away from the boundary, the influence of the boundary condition is small and the mesh points move at a speed close to v_{ref}. However, near the boundary, and especially on it, the boundary condition has a much more significant effect, and as a consequence, the mesh points accumulate around the corners (where the corner points are fixed). The mesh stops moving at around $t = 0.2$ because too many points have accumulated near the corners.



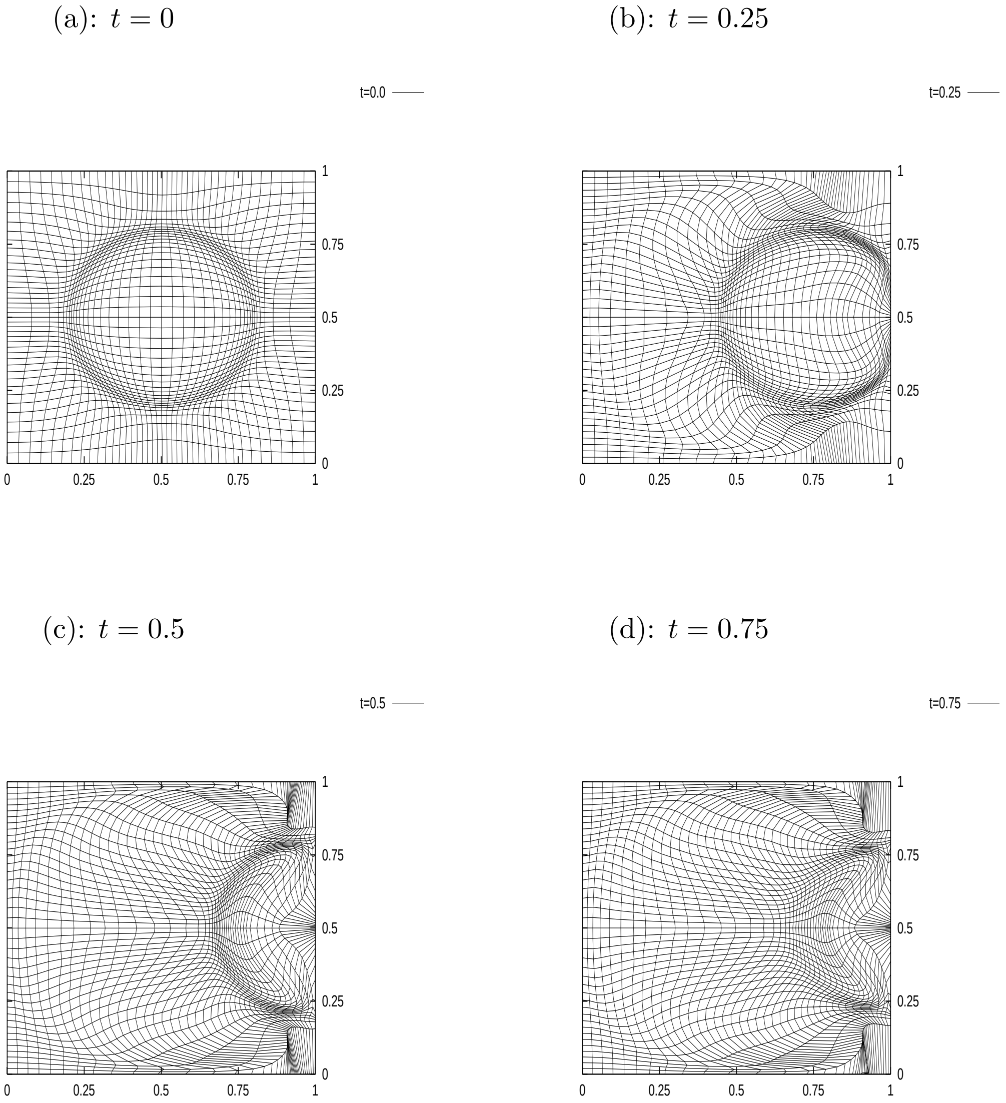
(a): $t = 0$

(b): $t = 0.25$

(c): $t = 0.5$

(d): $t = 0.75$

Figure 10.1: Example 10.5.1. A moving mesh at $t = 0, 0.25, 0.5, 0.75$ is obtained by the GCL method with $w = 1$.



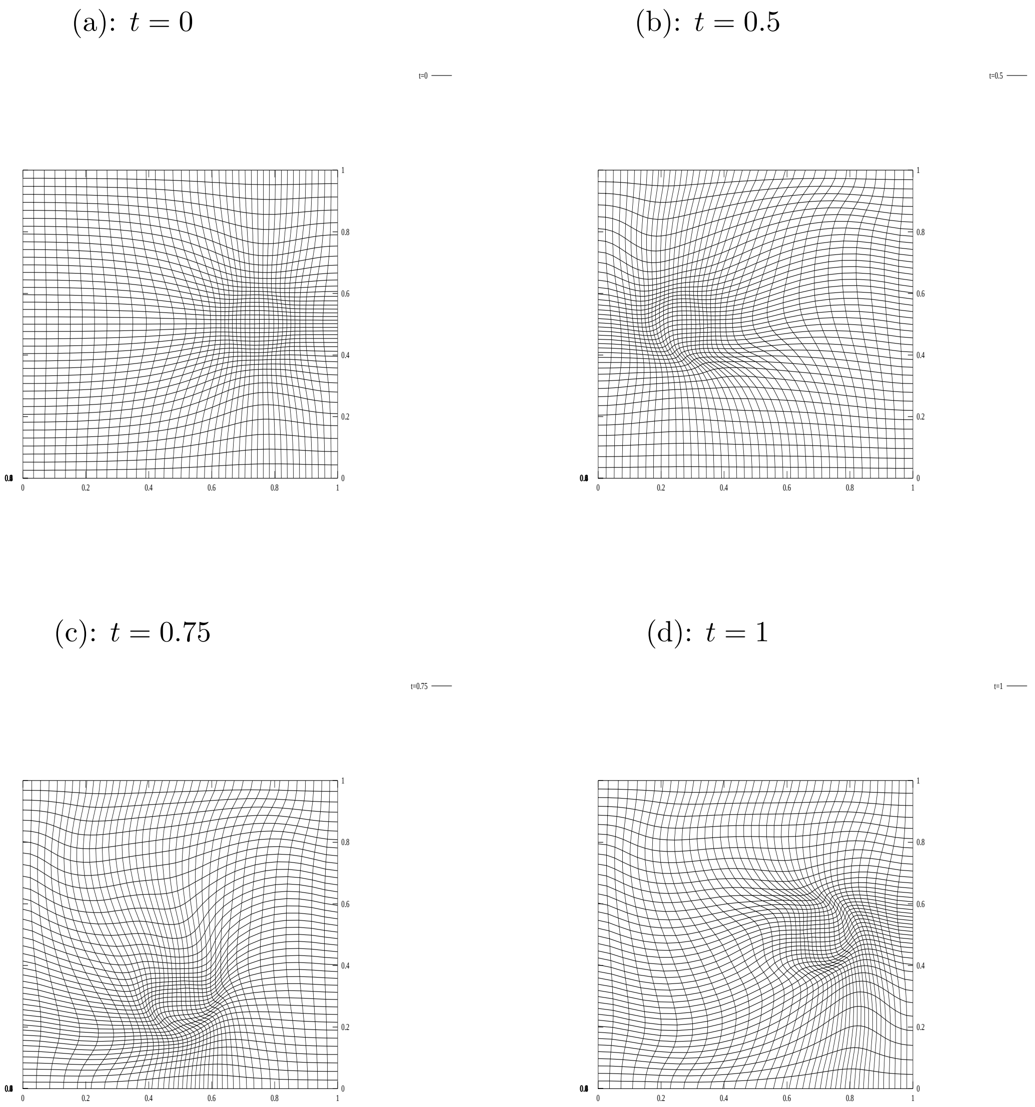
(a): $t = 0$

(b): $t = 0.25$

(c): $t = 0.5$

(d): $t = 0.75$

Figure 10.2: Example 10.5.1. A moving mesh at $t = 0, 0.25, 0.5, 0.75$ is obtained by the GCL method with $w = \rho/\sigma$.



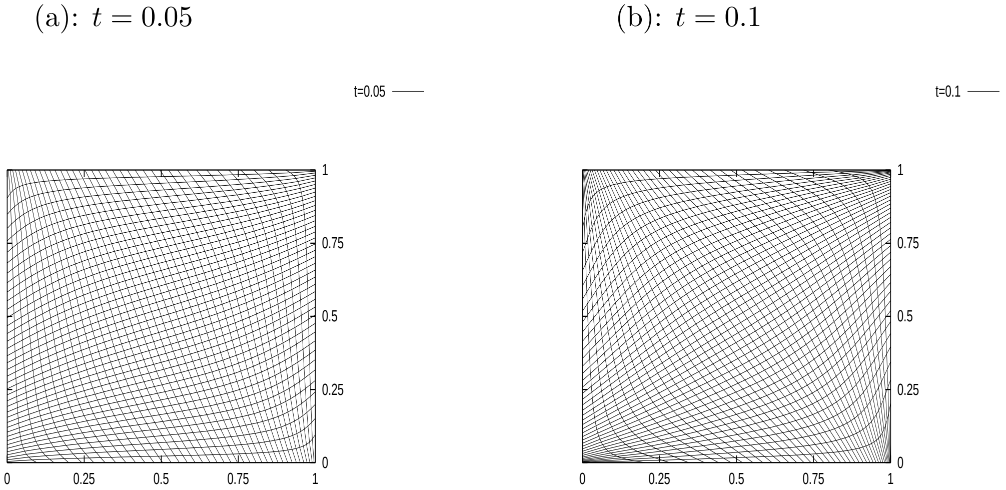
(a): $t = 0$

(b): $t = 0.5$

(c): $t = 0.75$

(d): $t = 1$

Figure 10.3: Example 10.5.2. A moving mesh at $t = 0, 0.5, 0.75, 1$ is obtained by the GCL method with $w = 1$.


(a): $t = 0.05$

(b): $t = 0.1$

Figure 10.4: Example 10.5.3. A moving mesh at $t = 0.05, 0.1$ is obtained by the GCL method with $w = 1$.

Chapter 11

Conclusions and comments

Mesh adaptation has become an indispensable tool for use in the numerical solution of partial differential equations. A proper use of mesh adaptation can often significantly improve the efficiency of numerical simulation and enhance computational resolution. To take a full benefit from mesh adaptation, not only the size but also the shape and orientation of mesh elements should be allowed to change to adapt to the behavior of the physical solution. This is especially true when the solution of the problem under consideration exhibits an anisotropic feature that it changes more significantly in one direction than the others.

Mesh adaptation lies in the ability to control the size, shape, and orientation of mesh elements throughout the physical domain. As shown in Chapter 2, this can be done in two steps. The first step is to define a monitor function or a metric tensor which specifies the element size, shape, and orientation on the physical domain. The second is to develop an algorithm to generate a mesh that satisfies the alignment and equdistribution conditions (2.3) and (2.4) (or (2.10) and (2.11) in the terminology of finite elements) for the monitor function defined in the first step.

The monitor function has been defined in Chapter 6 based on isotropic and anisotropic estimates for interpolation error and using mesh quality measures developed in Chapter 6. The metric tensor used for unstructured mesh generation has been defined in Chapter 7.

On the practical side, anisotropic unstructured meshes can be generated as isotropic ones in the metric specified by a metric tensor. The feasibility of this approach as well as the advantages of using an anisotropic mesh have been demonstrated in Chapter 7.

A variational adaption method has been studied in Chapter 8 based on the alignment and equdistribution conditions (2.3) and (2.4). Two moving mesh methods have been discussed in Chapters 9 and 10. They are also based on these conditions but utilize very different meshing strategies. The MMPDE method employs an adaptation functional to define the coordinate transformation needed for mesh generation, and is location-based. On the other hand, the GCL method, for which the equidistribution condition (2.4) is satisfied exactly, is a generalization of the Lagrange method in computational fluid dynamics and targets the mesh speed directly.

Bibliography

[1] R. A. Adams. *Sobolev Spaces.* Academic Press, New York, 1975.

[2] Mark Ainsworth and J. Tinsley Oden. *A posteriori error estimation in finite element analysis.* Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000.

[3] D. Ait-Ali-Yahia, G. Baruzzi, W. G. Habashi, M. Fortin, J. Dompierre, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-indepedent, mesh-independent and solver-independent CFD. Part II: structured grids. *Int. J. Numer. Meth. Fluids*, 39:657 – 673, 2002.

[4] T. Apel. *Anisotropic finite elements: local estimates and applications.* B. G. Teubner Stuttgart, Leipzig, 1999. Series of Advances in Numerical Mathematics.

[5] T. Apel, M. Berzins, P. K. Jimack, G. Kunert, A. Plaks, I. Tsukerman, and M. Walkley. Mesh shape and anisotropic elements: theory and practice. In J. R. Whiteman, editor, *The Mathematics of Finite Elements and Applications X*, pages 367 – 376, Oxford, 2000. Elsevier. MAFELAP 1999 (Uxbridge).

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

[7] T. Apel, S. Nicaise, and J. Schoberl. Crouzeix-raviart type finite elements on anisotropic meshes. *Numer. Math.*, 89:193 – 223, 2001.

[8] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM J. Numer. Anal.*, 13:214 – 226, 1976.

[9] M. J. Baines. *Moving Finite Elements.* Oxford University Press, 1994.

[10] M. J. Baines, M. E. Hubbard, and P. K. Jimack. A moving mesh finite element algorithm for fluid flow problems with moving boundaries. *Internat. J. Numer. Methods Fluids*, 47:1077 – 1083, 2005. 8th ICFD Conference on Numerical Methods for Fluid Dynamics. Part 2.

[11] R. E. Bank and R. K. Smith. Mesh smoothing using a posteriori error estimates. *SIAM J. Numer. Anal.*, 34:979 – 997, 1997.

[12] R. E. Barnhill and J. A. Gregory. Interpolation remainder theory from taylor expansions on triangles. *Numer. Math.*, 25:401 – 408, 1976.

[13] G. Beckett, J. A. Mackenzie, and M. L. Robertson. A moving mesh finite element method for the solution of two-dimensional Stefan problems. *J. Comput. Phys.*, 168:500 – 518, 2001.

[14] M. Berzins. A solution-based triangular and tetrahedral mesh quality indicator. *SIAM J. Sci. Comput.*, 19:2051 – 2060, 1998.

[15] P. Bochev, G. Liao, and G. d. Pena. Analysis and computation of adaptive moving grids by deformation. *Numer. Meth. PDEs*, 12:489 – 506, 1996.

[16] H. Borouchaki, P. L. George, P. Hecht, P. Laug, and E. Saletl. Delaunay mesh generation governed by metric specification: Part I. algorithms. *Finite Elemment in Analysis and Design*, 25:61 – 83, 1997.

[17] H. Borouchaki, P. L. George, and B. Mohammadi. Delaunay mesh generation governed by metric specification: Part II. applications. *Finite Elemment in Analysis and Design*, 25:85 – 109, 1997.

[18] F. J. Bossen and P. S. Heckbert. A pliant method for anisotropic mesh generation. In *Proceedings, 5th International Meshing Roundtable*, pages 63 – 74, Sandia National Laboratories, Albuquerque, NM, 1996. Sandia Report 96-2301.

[19] J. U. Brackbill. An adaptive grid with directional control. *J. Comput. Phys.*, 108:38 – 50, 1993.

[20] J. U. Brackbill and J. S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.*, 46:342 – 368, 1982.

[21] C. J. Budd, W. Huang, and R. D. Russell. Moving mesh methods for problems with blow-up. *SIAM J. Sci. Comput.*, 17:305 – 327, 1996.

[22] H. G. Burchard. Splines (with optimal knots) are better. *Appl. Anal.*, 3:309 – 319, 1974.

[23] X. Cai, D. Fleitas, B. Jiang, and G. Liao. Adaptive grid generation based on the least-squares finite-element method. *Comput. Math. Appl.*, 48:1077 – 1085, 2004.

[24] W. Cao, W. Huang, and R. D. Russell. An r-adaptive finite element method based upon moving mesh pdes. *J. Comp. Phys.*, 149:221 – 244, 1999.

[25] W. Cao, W. Huang, and R. D. Russell. A moving mesh method based on the geometric conservation law. *SIAM J. Sci. Comput.*, 24:118 – 142, 2002.

[26] W. Cao, W. Huang, and R. D. Russell. Approaches for generating moving adaptive meshes: location versus velocity. *Appl. Numer. Math.*, 47:121 – 138, 2003.

[27] G. Carey. *Computational grids: Generation, adaptation, and solution strategies.* Taylor & Francis, Washington, DC, 1997.

[28] N. Carlson and K. Miller. Design and application of a gradient-weighted moving finite element code, Part I, in 1-D. *SIAM J. Sci. Comput.*, 19:728 – 765, 1998.

[29] N. Carlson and K. Miller. Design and application of a gradient-weighted moving finite element code. Part II in 2-D. *SIAM J. Sci. Comput.*, 19:766 – 798, 1998.

[30] J. R. Cash. Diagonally implicit runge-kutta formulae with error estimates. *J. Inst. Math. Appl.*, 24:293 – 301, 1979.

[31] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *Int. J. Numer. Meth. Fluids*, 25:475 – 491, 1997.

[32] H. D. Ceniceros and T. Y. Hou. An efficient dynamically adaptive mesh for potentially singular solutions. *J. Comput. Phys.*, 172:609 – 639, 2001.

[33] L. Chen, P. Sun, and J. C. Xu. Optimal anisotropic meshes for minimizing interpolation errors in l^p-norm. Technical Report AM265, Department of Mathematics, The Pennsylvania State University, 2004.

[34] Long Chen and Jin-chao Xu. Optimal Delaunay triangulations. *J. Comput. Math.*, 22:299 – 308, 2004.

[35] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1978.

[36] B. Dacorogna and J. Moser. On a partial differential equation involving the Jacobian determinant. *Ann. Inst. Henri Poincare Analyse non lineaire*, 7:1 – 26, 1990.

[37] E. F. D'Azevedo. Optimal triangular mesh generation by coordinate transformation. *SIAM J. Sci. Stat. Comput.*, 12:755 – 786, 1991.

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

[39] C. de Boor. Good approximation by splines with variables knots ii. In G. A. Watson, editor, *Lecture Notes in Mathematics 363*, pages 12 – 20, Berlin, 1974. Springer-Verlag. Conference on the Numerical Solution of Differential Equations, Dundee, Scotland, 1973.

[40] Yana Di, Ruo Li, Tao Tang, and Pingwen Zhang. Moving mesh finite element methods for the incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 26:1036 – 1056, 2005.

[41] M. Dobrowolski, S. Greaf, and C. Pflaum. On a posteriori error estimators in the finite element method on anisotropic meshes. *Elec. Trans. Numer. Anal.*, 8:36 – 45, 1999.

[42] V. Dolejsi. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Computing and Visualisation in Science*, 1:165 – 178, 1998.

[43] J. Dompierre, M.-G. Vallet, Y. Bourgault, M. Fortin, and W. G. Habashi. Anisotropic mesh adaptation: towards user-indepedent, mesh-independent and solver-independent CFD. Part III: unstructured meshes. *Int. J. Numer. Meth. Fluids*, 39:675 – 702, 2002.

[44] E. A. Dorfi and L. O'c Drury. Simple adaptive grids for 1-d initial value problems. *J. Comput. Phys.*, 69:175 – 195, 1987.

[45] A. S. Dvinsky. Adaptive grid generation from harmonic maps on riemannian manifolds. *J. Comput. Phys.*, 95:450 – 476, 1991.

[46] J. E. Castillo (ed.). *Mathematical aspects of numerical grid generation.* SIAM, Philadelphia, 1991. Frontiers in Applied Mathematics vol 8.

[47] P. R. Eiseman. Grid generation for fluid mechnics computation. *Ann. Rev. Fluid Mech.*, 17:487 – 522, 1985.

[48] P. R. Eiseman. Adaptive grid generation. *Comput. Meth. Appl. Mech. Engr.*, 64:321 – 376, 1987.

[49] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numer. Math.*, 89:641 – 667, 2001.

[50] R. V. Garimella and M. S. Shephard. Boundary layer meshing for viscous flows in complex domain. In *Proceedings, 7th International Meshing Roundtable*, pages 107 – 118, Sandia National Laboratories, Albuquerque, NM, 1998.

[51] P. L. George. Automatic mesh generation and finite element computation. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of Numerical Analysis, Vol. IV.* Elsevier Science B.V., 1996.

[52] P. L. George and F. Hecht. Nonisotropic grids. In J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors, *Handbook of Grid Generation*, pages 20–1 – 20–29, Boca Raton, 1999. CRC Press.

[53] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-indepedent, mesh-independent and solver-independent CFD. Part I: general principles. *Int. J. Numer. Meth. Fluids*, 32:725 – 744, 2000.

[54] R. Hagmeeijer. Grid adaption based on modified anisotropic diffusion equations formulated in the parametric domain. *J. Comput. Phys.*, 115:169 – 183, 1994.

[55] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities.* Cambridge University Press, Cambridge, 1934.

[56] D. F. Hawken, J. J. Gottlieb, and J. S. Hansen. Review of some adaptive node-movement techniques in finite element and finite difference solutions of pdes. *J. Comput. Phys.*, 95:254 – 302, 1991.

[57] F. Hecht. Bidimensional anisotropic mesh generator. Technical report, INRIA, Rocquencourt, 1997. Source code: http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm.

[58] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *J. Comput. Phys.*, 171:753 – 775, 2001.

[59] W. Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174:903 – 924, 2001.

[60] W. Huang. Convergence analysis of finite element solution of one-dimensional singularly perturbed differential equations on equidistributing meshes. *International Journal of Numerical Analysis & Modeling*, 2:57 – 74, 2005.

[61] W. Huang. Measuring mesh qualities and application to variational mesh adaptation. *SIAM J. Sci. Comput.*, 26:1643 – 1666, 2005.

[62] W. Huang. Metric tensors for anisotropic mesh generation. *J. Comput. Phys.*, 204:633 – 665, 2005.

[63] W. Huang, Y. Ren, and R. D. Russell. Moving mesh methods based on moving mesh partial differential equations. *J. Comput. Phys.*, 113:279 – 290, 1994.

[64] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations (mmpdes) based upon the equidistribution principle. *SIAM J. Numer. Anal.*, 31:709 – 730, 1994.

[65] W. Huang and R. D. Russell. Analysis of moving mesh partial differential equations with spatial smoothing. *SIAM J. Numer. Anal.*, 34:1106 – 1126, 1997.

[66] W. Huang and R. D. Russell. A high dimensional moving mesh strategy. *Appl. Numer. Math.*, 26:63 – 76, 1997.

[67] W. Huang and R. D. Russell. Moving mesh strategy based upon a gradient flow equation for two dimensional problems. *SIAM J. Sci. Comput.*, 20:998 – 1015, 1999.

[68] W. Huang and W. Sun. Variational mesh adaptation II: error estimates and monitor functions. *J. Comput. Phys.*, 184:619 – 648, 2003.

[69] W. Huang and X. Zhan. Adaptive moving mesh modeling for two dimensional groundwater flow and transport. In *Recent advances in adaptive computation (Hangzhou, 2004)*, volume ??? of *Contemp. Math.*, pages ??? – ??? Amer. Math. Soc., Providence, RI, ???

[70] P. Jamet. Estimations de l'erreur pour des elements finis droits preque degeneres. *RAIRO Anal. Numer.*, 10:43 – 60, 1976.

[71] P. Knupp. Mesh generation using vector-fields. *J. Comput. Phys.*, 119:142 – 148, 1995.

[72] P. Knupp and S. Steinberg. *Fundamentals of Grid Generation.* CRC Press, Boca Raton, 1994.

[73] P. M. Knupp. Jacobian-weighted elliptic grid generation. *SIAM J. Sci. Comput.*, 17:1475 – 1490, 1996.

[74] P. M. Knupp and N. Robidoux. A framework for variational grid generation: conditioning the jacobian matrix with matrix norms. *SIAM J. Sci. Comput.*, 21:2029 – 2047, 2000.

[75] H. Kober. On the arithmetic and geometric means and on hölder's inequality. *Proc. Amer. Math. Soc.*, 9:452 – 459, 1958.

[76] M. Krizek. On the maximum angle condition for linear tetrahedral elements. *SIAM J. Numer. Anal.*, 29:513 – 520, 1992.

[77] G. Kunert. A posteriori error estimation for anisotropic tetrahedral and triangular finite element meshes. Technical report, TU Chemnitz, 1999. Ph. D. Thesis.

[78] G. Kunert. An a posteriori residual error estimator for the finite element method on anisotropic tetrahedral meshes. *Numer. Math.*, 86:471 – 490, 2000.

[79] G. Kunert. A local problem error estimator for anisotropic tetrahedral finite element meshes. *SIAM J. Numer. Anal.*, 39:668 – 689, 2001.

[80] G. Kunert and R. Verfeurth. Edge residuals dominate a posteriori error estimates for linear finite element methods on anisotropic triangular and tetrahedral meshes. *Numer. Math.*, 86:283 – 303, 2000.

[81] J. Lang, W. Cao, W. Huang, and R. D. Russell. A two–dimensional moving finite element method with local refinement based on a posteriori error estimates. *Appl. Numer. Math.*, 46:75 – 94, 2003.

[82] R. Li, T. Tang, and P. W. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *J. Comput. Phys.*, 170:562 – 588, 2001.

[83] R. Li, T. Tang, and P. W. Zhang. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *J. Comput. Phys.*, 177:365 – 393, 2002.

[84] G. J. Liao and D. Anderson. A new approach to grid generation. *Appl. Anal.*, 44:285 – 298, 1992.

[85] V. D. Liseikin. *Grid Generation Methods.* Springer, Berlin, 1999.

[86] A. Liu and B. Joe. Relationship between tetrahedron quality measures. *BIT*, 34:268 – 287, 1994.

[87] J. A. Mackenzie and M. L. Robertson. A moving mesh method for the solution of the one-dimensional phase-field equations. *J. Comput. Phys.*, 181:526 – 544, 2002.

[88] C. T. Miller, S. N. Gleyzer, and P. T. Imhoff. Numerical modeling of napl dissolution fingering in porous media. In H. M. Selim and L. Ma, editors, *Physical nonequilibrium in soils modeling and application*, Chelsea, Michigan, 1998. Ann Arbor Press.

[89] K. Miller. Moving finite elements II. *SIAM J. Numer. Anal.*, 18:1033 – 1057, 1981.

[90] K. Miller and R. N. Miller. Moving finite elements I. *SIAM J. Numer. Anal.*, 18:1019 – 1032, 1981.

[91] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in higher dimensions. *SIAM J. Comput.*, 29:1334 – 1370, 2000.

[92] P. K. Moore and J. E. Flaherty. Adaptive local overlapping grid methods for parabolic system in two space dimensions. *J. Comput. Phys.*, 98:54 – 63, 1992.

[93] J. Moser. On the volume elements of a manifold. *Trans. AMS*, 120:286 – 294, 1965.

[94] S. Owen. Meshing software survey. Technical report. http://www.andrew.cmu.edu/user/sowen/softsurv.html.

[95] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comput. Phys.*, 72:449 – 466, 1997.

[96] L. R. Petzold. Observations on an adaptive moving grid method for one-dimensional systems for partial differential equations. *Appl. Numer. Math.*, 3:347 – 360, 1987.

[97] M. Picasso. An anisotropic error indicator based on zienkiewicz-zhu error estimator: Application to elliptic and parabolic problems. *SIAM J. Sci. Comput.*, 24:1328 – 1355, 2003.

[98] J. Remacle, X. Li, M. S. Shephard, and J. E. Flaherty. Anisotropic adaptive simulation of transient flows using discontinuous galerkin methods. *Int. J. Numer. Meth. Engr.*, page (accepted), 2003.

[99] Yu. G. Reshetnyak. *Space mappings with bounded distortion.* American Mathematical Society, Providence, Rhode Island, 1989. Translation of Mathematical Monographs, Volume 73.

[100] S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM J. Numer. Anal.*, 29:257 – 270, 1992.

[101] Robert Schneiders. Mesh generation & grid generation on the web. Technical report. http://www-users.informatik.rwth-aachen.de/ roberts/meshgeneration.html.

[102] B. Semper and G. Liao. A moving grid finite-element method using grid deformation. *Numer. Methods in PDEs*, 11:603 – 615, 1995.

[103] N. A. Shenk. Uniform error estimates for certain narrow lagrangian finite elements. *Math. Comput.*, 63:105 – 119, 1994.

[104] K. G. Siebert. An a posteriori error estimator for anisotropic refinement. *Numer. Math.*, 73:373 – 398, 1996.

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

[106] S. Steinberg and P. J. Roache. Variational grid generation. *Numer. Meth. P.D.E.*, 2:71 – 96, 1986.

[107] J. M. Stockie, J. A. Mackenzie, and R. D. Russell. A moving mesh method for one dimensional hyperbolic conservation law. *SIAM J. Sci. Comput.*, 22:1791 – 1813, 2000.

[108] J. L. Synge. *The Hypercircle in Mathematical Physics.* Cambridge University Press, Cambridge, 1957.

[109] Huazhong Tang and Tao Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 41:487 – 515 (electronic), 2003.

[110] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA J.*, 17:1030 – 1037, 1979.

[111] J. F. Thompson, editor. *THe use of solution adaptive grids in solving partial differential equations*, Amsterdam, 1983. North-Holland.

[112] J. F. Thompson. A survey of dynamically-adaptive grids in the numerical solution of partial differential equations. *Appl. Numer. Math.*, 1:3 – 27, 1985.

[113] J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors. *Handbook of grid generation*, Boca Raton and London, 1999. CRC Press.

[114] J. F. Thompson, Z. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundations and Applications.* North-Holland, New York, 1985.

[115] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. Boundary-fitted coordinate systems for numerical solution of partial differential equations - a review. *J. Comput. Phys.*, 47:1 – 108, 1982.

[116] J. F. Thompson and N. P. Weatherill. Structured and unstructured grid generation. *Critical Reviews Biomed. Eng.*, 20:73 – 120, 1992.

[117] Weiqing Wang and Xiao-Ping Wang. An iterative grid redistribution method for singular problems in multiple dimensions. *J. Comput. Phys.*, 159:246 – 273, 2000.

[118] A. J. Wathen and M. J. Baines. ??? *IMA J. Numer. Anal.*, 5:161 – ???, 1985.

[119] B. V. Wells, M. J. Baines, and P. Glaister. Generation of arbitrary Lagrangian-Eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations. *Internat. J. Numer. Methods Fluids*, 47:1375 – 1381, 2005. 8th ICFD Conference on Numerical Methods for Fluid Dynamics. Part 2.

[120] A. Winslow. Numerical solution of the quasi-linear poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.*, 1:149 – 172, 1967.

[121] A. M. Winslow. Adaptive mesh zoning by the equipotential method. Technical Report UCID-19062, Lawrence Livemore Laboratory, 1981.

[122] S. Yamakawa and K. Shimada. High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. In *Proceedings, 9th International Meshing Roundtable*, pages 263 – 273, Sandia National Laboratories, Albuquerque, NM, 2000. Sandia Report 2000-2207.

[123] N. N. Yanenko, E. A. Kroshko, V. V. Lisejkin, V. M. Fomin, V. P. Shapeev, and Y. A. Shitov. Methods for the construction of moving grids for problems of fluid dynamics with big deformations. pages 454 – , Berlin, 1976. Springer. Lecture Notes in Physics 59.

[124] P. A. Zegeling. *Moving-grid methods for time-dependent partial differential equations.* CWI Tract 94, Stichting Mathematisch Centrum, Centrum voor Wiskunde en Informatica, Amsterdam, 1993.

[125] P. A. Zegeling and H. P. Kok. Adaptive moving mesh computations for reaction-diffusion systems. *J. Comput. Appl. Math.*, 168:519 – 528, 2004.

[126] Z. Zhang and A. Naga. A new finite element gradient recovery method: Superconvergence property. *SIAM J. Sci. Comput.*, (in press).

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

[128] O. C. Zienkiewicz and J. Z. Zhu. The superconvergence patch recovery and a posteriori error estimates. Part 2: Error esimates and adaptivity. *Int. J. Numer. Methods Engrg.*, 33:1365 – 1382, 1992.

[129] M. Zlámal. On the finite element method. *Numer. Math.*, 12:394 – 409, 1968.