Scientific Computation

Vladimir D. Liseikin

Grid Generation Methods

Third Edition

Springer

Grid Generation Methods

Scientific Computation

More information about this series at http://www.springer.com/series/718

Vladimir D. Liseikin

Grid Generation Methods

Third Edition

Springer

Vladimir D. Liseikin
Institute of Computational Technologies SB
RAS
Novosibirsk
Russia

and

Novosibirsk State University
Novosibirsk
Russia

ISSN 1434-8322 ISSN 2198-2589 (electronic)
Scientific Computation
ISBN 978-3-319-57845-3 ISBN 978-3-319-57846-0 (eBook)
DOI 10.1007/978-3-319-57846-0

Library of Congress Control Number: 2017939626

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface to the Third Edition

Grid generation codes represent an indispensable tool for solving field problems in applied mathematics, mechanics, physics, and other areas of practical applications. Despite the considerable success achieved in grid generation technologies, development of more efficient and sophisticated algorithms and computer codes for generating grids still remains an important problem. Serious difficulties arise in grid generation in domains with complicated boundary geometries, specifically, with discretely defined boundary segments and in the case when grids have to be adapted to solution singularities, such as boundary and interior layers, shocks, detonation waves, combustion fronts, high-speed jets, and phase transition zones. A promising tool to deal with the numerical problems having such singularities is adaptive grid generation technology. With increasing complexity of the physical problems, there is an increased need for more reliable, robust, and fully automated grid generation codes which enable one to generate suitable meshes in a uniform "black box" mode, without human interaction. The development of such grid systems is a challenging problem in computational physics and applied mathematics.

Grid technology still remains a rapidly advancing field of computational and applied mathematics. New achievements are being added by the creation of more sophisticated techniques, modification of the available methods, and implementation of more subtle tools as well as the results of the classical theories of differential equations, calculus of variations, and Riemannian geometry in the formulation of grid models and analysis of grid properties. Therefore, there is a clear need of students, researchers, and practitioners in the field of applied mathematics and industry for the creation of new books and/or updated editions of the available books which will complement the existing ones, providing a description of current developments relating to grid methods, grid codes, and their applications to the solving of actual problems.

This third edition of the monograph "Grid Generation Methods" is significantly expanded with new material that discusses recent advances in grid generation technology. It includes a description of updated grid generation methods, which were partly presented in the former monograph of the author, as well as new adaptive approaches for structured and unstructured grids and numerical algorithms for their generation. Special attention is paid to a review of those promising approaches and methods which have been developed recently and/or have not been sufficiently covered in other monographs. In particular, this book includes an application for generating grids for immersed boundary methods. It also describes a stretching method adjusted to the numerical solution of singularly perturbed equations having large-scale solution variations, e.g., those modeling high-Reynolds-number flows. A number of functionals related to conformality, orthogonality, energy, and alignment are described. This book includes differential and variational techniques for generating uniform, conformal, and harmonic coordinate transformations on hypersurfaces for the development of a comprehensive approach to the construction of both fixed and adaptive grids in the interior and on the boundary of domains in a unified manner. The monograph is also concerned with the description of all essential grid quality measures such as skewness, curvature, torsion, angle and length values, and conformality. It gives a more detailed and practice-oriented description of control metrics for providing the generation of adaptive, field-aligned, and balanced numerical grids by means of the numerical solution of inverted Beltrami and diffusion equations in the control metrics. Some numerical algorithms are described for generating surface and domain grids. One more new feature of this book is the implementation of adaptive grid technology to the numerical solution of problems in mechanics, physics, fluids, plasmas, and nanotechnologies. Emphasis is placed on mathematical formulations, explanations, and examples of various aspects of grid generation and their applications.

This book will introduce a reader to structured and unstructured grid methods, as well as automated technologies for the generation of adaptive grids for the numerical solution of applied problems with complicated domain segments and complicated solution structures. These technologies are based on advanced algebraic, elliptic, variational, Delaunay, advancing-front, and quad–octree methods, as well as on the methods of finite differences and volumes. The technologies are indispensable for the numerical solution of differential equations, modeling various complex physical processes in energetics, ecology, industry, as well as the medical sphere. Furthermore, this book includes chapters devoted to the implementation of comprehensive grid methods into numerical codes and to the application of the codes to the numerical solution of a range of mechanical, fluid, and plasma-related problems. The new and fast-developing computational tools discussed throughout this book enable a detailed analysis of real-world problems that simply lie beyond the reach of traditional methods.

The major area of attention of this book is grid-mapping techniques. In addition, however, the author has also included an elementary introduction to basic unstructured approaches to mesh generation. A more detailed description of unstructured mesh techniques and corresponding aspects related to parallel processing, mesh quality enhancement, and mesh modification and optimization can be found in the books of the leading experts on these technologies: *Computational Grids: Adaptation and Solution Strategies* by G.F. Carey (1997), *Delaunay Triangulation and Meshing* by P.-L. George and H. Borouchak (1998), *Mesh Generation Application to Finite Elements* by P.J. Frey and P.-L. George (2008), and *Finite Element Mesh Generation* by D.S.H. Lo (2015). These books, though, do not give a detailed introduction to advanced mapping approaches developed in recent years. Thus, the current monograph and these books complement each other, presenting a comprehensive description of all the popular grid generation approaches. As grid generation methodology is well on its way to becoming a formal subject in university curricula, the books mentioned and the current book taken together will provide materials fully sufficient to support a one-year university course related to structured and unstructured mesh technologies.

Since grid technology has a widespread application across nearly all field problems, this new edition of the monograph will be of significant interest to a broad range of readers: teachers, students, researchers, and practitioners in applied mathematics, mechanics, physics, and other areas of application. In addition, it could be used as a textbook for advanced undergraduates or for first-year postgraduate students or as a tutorial for mathematicians, engineers, and scientists who are engaged in the computation of equations in multidimensional domains with complicated boundary geometries.

Chapter 1 of this book provides a general introduction to the subject of grids. It gives an outline of structured, unstructured, hybrid, overlapping, and composite grids. The chapter delineates some of the basic classes of methods, in particular manual or semiautomatic methods, mapping methods, and unstructured methods. The chapter also includes a description of various types of grid topology and touches on certain issues of comprehensive grid codes.

Chapter 2 deals with several mathematical relations that are necessary only for the generation of grids by means of the mapping approach and which are connected with and derived from the metric tensors of coordinate transformations. As an example of an application of these relations, the chapter presents a technique aimed at obtaining conservation-law equations in new fixed or time-dependent coordinates. In the procedures described, the deduction of the expressions for the transformed equations is based only on the formula for the differentiation of the Jacobian of the coordinate transformations.

Very important issues of grid generation, connected with a description of grid quality measures in forms suitable for formulating grid techniques and efficiently analyzing the necessary mesh properties, are discussed in Chap. 3. The definitions of the grid quality measures are based on the metric tensors and on the relations between the metric elements considered in Chap. 2. Special attention is paid to the invariants of the metric tensors, which are the basic elements for the definition of many important grid quality measures. Clear algebraic and geometric interpretations of the invariants are presented.

Chapter 4 describes a stretching method based on the application of special nonuniform stretching coordinates in the regions of large variation of the solution. The use of stretching coordinates is extremely effective for the numerical solution of problems with boundary and interior layers. The chapter acquaints the reader with various types of singularity arising in solutions to equations with a small parameter affecting the higher derivatives. The solutions of these equations undergo large variations in very small boundary and interior zones, called boundary or interior layers, respectively. The chapter gives a detailed description of the qualitative properties of solutions in such layers. Besides the well-known exponential layers, three types of power layer common to bisingular problems having complementary singularities arising from reduced equations are described. Such equations are widespread in applications, for example, in viscous gas dynamics. The specification of the stretching functions is given for each type of basic singularity. The functions are defined in such a way that the singularities are automatically smoothed with respect to the new stretching coordinates. The chapter gives the description of a procedure to generate intermediate coordinate transformations with the stretching functions. The transformations are suitable for smoothing both exponential and power layers. The grids derived with such coordinate transformations are often themselves well adapted to the expected physical features. Therefore, they make it easier to provide dynamic adaptation by taking part of the adaptive burden on themselves.

The simplest and fastest technique of grid generation is the algebraic method of transfinite interpolation described in Chap. 5. Of central importance in transfinite interpolation are the blending functions which provide the matching of the grid lines at the boundary and interior surfaces. Examples of various types of blending functions are reviewed, in particular the functions defined through the basic stretching coordinate transformations for singular layers. These transformations are dependent on a small parameter so that the resulting grid automatically adjusts to the respective physical parameter, e.g., viscosity, Reynolds number, or shell thickness, in practical applications. This chapter also gives a description of a procedure for generating triangular, tetrahedral, or prismatic grids through the method of transfinite interpolation. The chapter ends with a concise presentation of drag and sweeping meshing methods.

Chapter 6 is concerned with grid generation techniques based on the numerical solution of systems of partial differential equations. Generation of grids from these systems of equations is largely based on the numerical solution of elliptic, hyperbolic, and parabolic equations for the coordinates of grid lines which are specified on the boundary segments. The elliptic and parabolic systems reviewed in the chapter provide grid generation within blocks with specified boundary point distributions. These systems are also used to smooth algebraic, hyperbolic, and unstructured grids. A very important role is currently played in grid codes by a system of Poisson equations defined as a sum of Laplace equations and control functions. This system was originally considered by Godunov and Prokopov and further generalized, developed, and implemented for practical applications by Thompson, Thames, Mastin, and others. The chapter describes the properties of the Poisson system and specifies expressions for the control functions required to construct nearly orthogonal coordinates at the boundaries. Hyperbolic systems are useful when an outer boundary is free of specification. The control of the grid spacing in the hyperbolic method is largely performed through the specification of volume distribution functions. Special hyperbolic and elliptic systems are presented for generating orthogonal and nearly orthogonal coordinate lines, in particular those proposed by Ryskin and Leal. The chapter also reviews some parabolic and high-order systems for the generation of structured grids and describes adaptive mesh generation for steady and unsteady simulations.

Chapter 7 reviews the development of variational methods applied to grid generation. Variational grid generation relies on functionals related to grid quality: smoothness, orthogonality, regularity, aspect ratio, adaptivity, etc. By the minimization of a combination of these functionals, a user can define a compromise grid with the desired properties. The chapter discusses a variational approach for generating harmonic maps through the minimization of energy functionals, which was suggested by Dvinsky. Several versions of the functionals from which harmonic maps can be derived are identified.

Methods developed for the generation of grids on curves and surfaces are discussed in Chap. 8. The chapter describes the development and application of hyperbolic, elliptic, and variational techniques for the generation of grids on parametrically defined curves and surfaces. The differential approaches based on the Beltrami equations were proposed by Warsi and Thomas, while the variational methods rely on functionals of surface grid quality measures.

Chapter 9 is devoted to the implementation of inverted Beltrami equations with respect to control metrics for the generation of multidimensional adaptive grids via a mapping approach. The approach is based on mapping from a simple structured or an unstructured grid in the logical domain to a curvilinear grid with the desired properties in the physical domain. The control metrics provide efficient and simply defined conditions for various types of grid adaptation, particularly grid clustering according to given function values and/or gradients, grid alignment with given vector fields, and combinations thereof. The corresponding formulas of control metrics providing these grid properties are demonstrated. Using this approach, both adaptive and fixed grids can be generated in a unified manner, in arbitrary domains.

Numerical algorithms for generating grids by mapping approaches based on the solution of inverted Beltrami and diffusion equations are presented in Chap. 10. Furthermore, the chapter includes numerical methods (finite differences, finite elements, and spectral elements) for finding grid nodes via the solution of the inverted Beltrami equations. Basic approaches in parallelizing the mesh generating process are outlined in Chap. 10.

Chapter 11 describes techniques aimed at controlling grid properties with special control metrics in inverted Beltrami and diffusion equations. The control metrics provide efficient and straightforwardly defined conditions for various types of grid adaptation, particularly grid clustering according to given function values and/or gradients, grid alignment with given vector fields, and combinations thereof.

The subject of unstructured grid generation is discussed in Chap. 12. Unstructured grids may be composed of cells of arbitrary shape, but they are generally composed of triangles and tetrahedrons. Tetrahedral grid methods described in the chapter include Delaunay procedures and the advancing-front method. The Delaunay approach connects neighboring points (of some previously defined set of nodes in the region) to form tetrahedral cells in such a way that the sphere through the vertices of any tetrahedron does not contain any other points. In the advancing-front method, the grid is generated by building cells one at a time, marching from the boundary into the volume by successively connecting new points to points on the front until all the unmeshed space is filled with grid cells. The chapter also outlines the quod-octree approaches.

Chapter 13 is devoted to the implementation of adaptive grid techniques to provide solutions to applied problems. The equations for the evolving grid are incorporated into a single implicit time step in which the grid and the physical solution evolve together. The chapter discusses numerical solutions of three-dimensional diffusion equations with boundary and interior layers on adaptive grids with node clustering in the zones of large solution variations and also touches some applications of adaptive grid technologies to solutions of two-dimensional gas dynamics problems. The application of adaptive grids to calculations of some magnetically confined plasma problems and a tsunami wave run-up on a coastal region is described as well in Chap. 13. This chapter describes the results of numerical modeling of temperature gauged in burning solid fuel via an inserted thermocouple. Subsurface thermocouple sensors are very important technical devices employed to gauge heat fluxes in complicated heat-stressed frameworks in various heat-diffusion mechanisms and in burning solid fuels. Incidental problems related to the accuracy of the sensors' temperature data may appear. The primary source of inaccuracy is the difference in thermal properties of the materials of the thermocouple and of the surrounding medium. High gradients of temperature in heated materials lead, as a rule, to an increased heat transfer from the surface, since the thermal conductivity of metallic thermocouples is higher than that of the surrounding substance. Additional problems may appear due to a variation in the distances to the surface of heat transfer caused by the pyrolysis of the substance. The chapter presents an adaptive grid technology to investigate these problems. Chapter 13 also demonstrates the use of adaptive grids for the numerical simulations of nanopore formations in metals. It also includes a new grid generation approach aimed at solving problems by immersed boundary methods. This approach to grid generation does not involve the initial triangulation of the boundary of a domain, but a global numerical grid is first constructed in a larger domain with mesh refinement near the specified boundary points. Next, boundary and interior cells of the domain under consideration are selected, which comprise its numerical grid. As a result, a grid with thin cells near the boundary is obtained. The thinner these cells, the better the approximation of the domain. The global grid is generated by means of inverted diffusion equations for a spherical metric tensor. This approach is also suitable for domains with boundaries specified by an implicit analytic function.

Acknowledgement. The author is greatly thankful to G.S. Liseikin who prepared the text of the manuscript in LaTeX code. Thanks go as well to G. Lukas for correcting the author's English.

The author is very conscious of helpful suggestions in geometry, algebra, and numerical techniques made by his colleagues, Profs. Borisov, Churkin, Glasser, Kuzminov, Sharafutdinov, and Shvedov.

The author is also greatly obligated to the researchers who responded to his requests and sent files of their papers, namely T.J. Baker, D.A. Field, E. Ivanov, P. Knupp, G. Liao, M.S. Shephard, N.P. Weatherill, and P.P. Zegeling, and to those who prepared illustrations, namely B.S. Azarenok, A.A. Charakhch’yan, V.A. Garanja, A.H. Glasser, S.A. Ivanenko, A.M. Kharitonchick, A.V. Kofanov, l.N. Kudryavtseva, A.S. Lebedev, Yu.V. Likhanova, G.S. Liseikin, D.V. Patrakhin, A.D. Rychkov, and I.A. Vaseva.

Work on this book was supported in part by RFBR through Grant No. 16-0l-00455 and by the Institute of Computational Technologies SB RAS and Novosibirsk State University.

This book was published with the permission of the Academic Councils of the Institute of Computational Technologies of the Siberian Branch of the Russian Academy of Sciences and Novosibirsk State University.

Novosibirsk, Russia Vladimir D. Liseikin
September 2016

Contents

Chapter 1
General Considerations

1.1 Introduction

An indispensable tool of the numerical solution of partial differential equations by finite-element or finite-difference methods on general regions is a grid which represents the physical domain in a discrete form. In fact, the grid is a preprocessing tool or a foundation on which physical, continuous quantities are described by discrete functions and on which the differential equations are approximated by algebraic relations for discrete values that are then numerically analyzed by the application of computational codes. The grid technique also has the capacity, based on an appropriate distribution of the grid points, to enhance the computational efficiency of the numerical solution of complex problems.

The efficiency of a numerical study of a boundary value problem is estimated from the accuracy of the computed solution and from the cost and time of the computation.

The accuracy of the numerical solution in the physical domain depends on both the error of the solution at the grid points and the error of interpolation. Commonly, the error of the numerical computation at the grid points arises from several distinct sources. First, mathematical models do not represent physical phenomena with absolute accuracy. Second, an error arises at the stage of the numerical approximation of the mathematical model. Third, the error is influenced by the size and shape of the grid cells. Fourth, an error is contributed by the computation of the discrete physical quantities satisfying the equations of the numerical approximation. And fifth, an error in the solution is caused by the inaccuracy of the process of interpolation of the discrete solution. Of course, the accurate evaluation of the errors due to their sources remains a formidable task. It is apparent, however, that the quantitative and qualitative properties of the grid play a significant role in controlling the influence of the third and fifth sources of the error on the numerical analysis of physical problems.

Another important characteristic of a numerical algorithm that influences its efficiency is the cost of the operation of obtaining the solution. From this point of view, the process of generating a sophisticated grid may increase the computational costs of the numerical solution and encumber the computer tools with the requirement of

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_1

additional memory. On the other hand, there may be a significant profit in accuracy which allows one to use a smaller number of grid points. Any estimation of the contributions of these opposing factors can help in choosing an appropriate grid. In any case, since grid generation is an important component of numerical modeling, research in this field is aimed at creating techniques which are not too costly but which cause a significant improvement in the accuracy of the solution. The utilization of these techniques provides one with the real opportunities to enhance the efficiency of the numerical solution of complex problems. Thus, grid generation helps to satisfy the constant demand for enhancement of the efficiency of the numerical analysis of practical problems.

The first efforts aimed at the development of grid techniques were undertaken in the 1960s. In the present day, a significant number of advanced methods have been created: algebraic, elliptic, hyperbolic, parabolic, variational, Delaunay, advancing-front, quadtree-octree, etc. The development of these methods has reached a stage in which calculations in fairly complicated domains and on surfaces that arise while analyzing multidimensional problems are possible. Because of its successful development, the field of numerical grid generation has already formed a separate mathematical discipline with its own methodology, approaches, and technology.

At the end of the 1980s there began a new stage in the development of grid generation technique. It is characterized by the creation of comprehensive, multipurpose, three-dimensional grid generation codes which are aimed at providing a uniform environment for the construction of grids in arbitrary multidimensional geometries. Because of the numerous applications in engineering and scientific computations, grid generation has become increasingly recognized as a subject in its own right not only in engineering but also at universities as well.

The current chapter presents a framework for the subject of grid generation. It outlines the most general concepts and techniques, which will be expounded upon in more detail in the following chapters.

1.2 General Concepts Related to Grids

A physical domain, surface, or curve can be considered in a unified manner, as a collection of geometric objects referred to as regular surfaces or physical geometries locally represented as

$$\mathbf{x}(\mathbf{s}) : S^n \to R^{n+k}, \ \mathbf{x} = (x^1, \ldots, x^{n+k}), \ \mathbf{s} = (s^1, \ldots, s^n), \ n \geq 1 \ , \tag{1.1}$$

where S^n is an n-dimensional parametric domain (an interval if $n = 1$), while $\mathbf{x}(\mathbf{s})$ is a smooth vector-valued function of rank n at all points $\mathbf{s} \in S^n$. We shall designate by S^{xn} the regular surface parametrized by (1.1). Note, when $k = 0$ then S^{xn} is a domain $X^n \subset R^n$ which can be identified with the parametric domain S^n.

We assume further throughout this book that we are dealing with an arbitrary geometry S^{xn} locally represented by the parametrization (1.1).

Using the specification (1.1) of the physical geometry S^{xn} allows one to generate grid points first on the parametric domain S^n and then mapping them into S^{xn} through the parametrization $\mathbf{x}(\mathbf{s})$. With such consideration the process of grid generation can be carried out uniformly both for the boundary of a physical geometry and for its interior part. This scheme of grid generation leads to the natural requirement for grid techniques that the grids generated by a specific algorithm in the physical geometry for different parametrizations should be the same, i.e. the grid algorithms should be invariant of parametrizations of S^{xn}.

There are two general notions of a grid in an n-dimensional bounded domain or on a surface. One of these considers the grid as a set of algorithmically specified points of the domain or the surface. The points are called the grid nodes. The second considers the grid as an algorithmically described collection of small standard n-dimensional volumes covering the necessary area of the domain or surface without gaps and overlaps. The volumes are referred to as the grid cells. The cells are bounded curvilinear volumes, whose boundaries are divided into a few segments which are $(n-1)$-dimensional cells. Therefore, the cells can be formulated successively from one dimension to higher dimensions. The boundary points of the one-dimensional cells are called the cell vertices or the grid nodes.

1.2.1 Grid Cells

This subsection discusses some general concepts related to grid cells.

For cells in an n-dimensional domain or surface, there are commonly used n-dimensional volumes of simple standard shapes (see Fig. 1.1 for $n = 1, 2, 3$).

In one dimension, the cell is a closed line or segment, whose boundary is composed of two points referred to as the cell vertices.

A general two-dimensional cell is a two-dimensional simply connected domain, whose boundary is divided into a finite number of one-dimensional cells referred

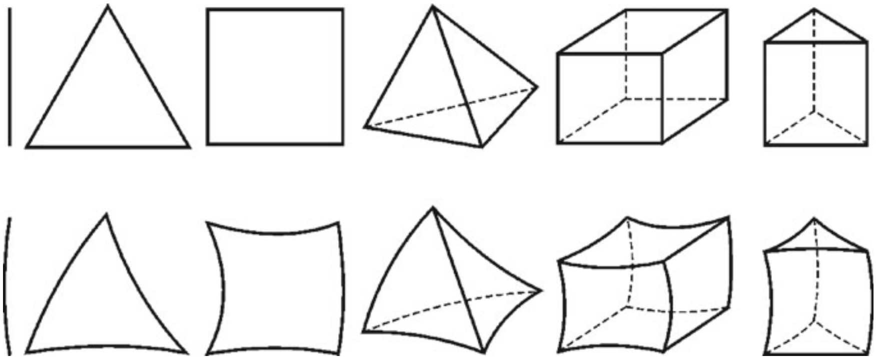


Fig. 1.1 Typical grid cells

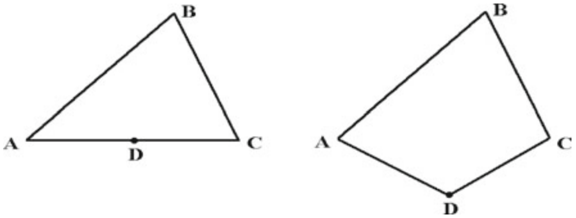


Fig. 1.2 Convex (*left*) and strongly convex (*right*) quadrilateral cells

to as the edges of the cell. Commonly, the cells of two-dimensional domains or surfaces are constructed in the form of linear or nonlinear triangles or quadrilaterals. The boundary of a triangular cell is composed of three segments, while the boundary of a quadrilateral is represented by four segments. These segments are the one-dimensional grid cells.

By a general three-dimensional cell, we mean a simply connected three-dimensional polyhedron, whose boundary is partitioned into a finite number of two-dimensional cells called its faces. In practical applications, three-dimensional cells typically have the shape of linear or curvilinear tetrahedrons or hexahedrons. The boundary of a tetrahedral cell is composed of four triangular cells, while a hexahedron is bounded by six quadrilaterals. Thus, a hexahedral cell has six faces, twelve edges, and eight vertices. Some applications also rely on the three-dimensional cells in the form of prisms and even pyramids as three-dimensional cells. A prism has two triangular and three quadrilateral faces, nine edges, and six vertices, while a pyramid has four triangular and one quadrilateral faces, eight edges, and five vertices.

The selection of the shapes shown in Fig. 1.1 to represent the standard cells is justified, first, by their geometrical simplicity, and second, because the existing procedures for the numerical simulation of physical problems are largely based on approximations of partial differential equations using these elemental volumes. The specific choice of cell shape depends on the geometry and physics of the particular problem and on the method of solution. In particular, tetrahedrons (triangles in two dimensions) are well suited for finite-element methods, while hexahedrons are commonly used for finite-difference techniques.

Convex Cells

A convex n-dimensional cell S is the convex hull of some $n+k,\ k>1$, points $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_{n+k}$ from X^n which do not lie in any $(n-1)$-dimensional plane. Thus, S is composed of all points $\boldsymbol{x} \in R^n$ which are defined through $\boldsymbol{P}_i$ by the equation

$$\boldsymbol{x} = \sum_{i=1}^{n+k} \alpha^i \boldsymbol{P}_i \ ,$$

$$\sum_{i=1}^{n+k} \alpha^i = 1 \ , \qquad 1 \geq \alpha^i \geq 0 \ .$$

We call all those points $\boldsymbol{P}_l$ of the set $\{\boldsymbol{P}_i,\ i = 1, \cdots, n+k\}$ which lie on the boundary of S vertices of the convex cell S.

An m-dimensional face of the convex n-dimensional cell S $(n > m)$ is a convex hull of $m+1$ vertices P_l, which does not contain any other vertices of S.

We call the cell S strongly convex if for all $m < n$ it does not have any two m-dimensional faces which lie in an m-dimensional plane. Figure 1.2 demonstrates the difference between convex (Fig. 1.2, *left-hand*) and strongly convex (Fig. 1.2, *right-hand*) quadrilateral cells. Evidently, if $\boldsymbol{P}$ is an interior point of the strongly convex cell S with the vertices $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_{n+k}$ then there will be at least $n+1$ points in the expansion of $\boldsymbol{P}$

$$\boldsymbol{P} = \sum_{i=1}^{n+k} \alpha^i \boldsymbol{P}_i\ , \qquad \sum_{i=1}^{n+k} \alpha^i = 1\ , \qquad \alpha^i \geq 0\ , \qquad i = 1, \cdots, n+k\ ,$$

with nonzero values of their coefficients α^i.

Simplexes and Simplex Cells

Commonly, the edges and the faces of the cells are linear (Fig. 1.1, above). Linear triangles and tetrahedrons are also referred to as two-dimensional simplexes and three-dimensional simplexes, respectively. The notion of the simplex can be formulated for arbitrary dimensions. Namely, by a k-dimensional simplex in an n-dimensional $(n \geq k)$ geometry S^{xn} we mean a volume of k-dimensional space whose nodes are defined by the equation

$$\boldsymbol{x} = \sum_{i=1}^{k+1} \alpha^i \boldsymbol{x}_i\ ,$$

where $\boldsymbol{x}_i,\ i = 1, \cdots, k+1$, are some specified points which are the verticies of the simplex, and $\alpha^i,\ i = 1, \cdots, k+1$, are real numbers satisfying the relations

$$\sum_{i=1}^{k+1} \alpha^i = 1\ , \qquad \alpha^i \geq 0\ .$$

The points $\boldsymbol{x}_i,\ i = 1, \cdots, k+1$, are subject to the following restriction: they are not all in the same hyperplane, namely, the vectors $\boldsymbol{r}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_1,\ i = 1, \cdots, k$, are independent, i.e. the matrix $\{r_i \cdot r_j\},\ i, j = 1, \cdots k$ is invertible. In this respect, a one-dimensional linear cell is the one-dimensional simplex. The boundary of a k-dimensional simplex is composed of $k+1$ simplexes of $(k-1)$-dimension.

In practical discretizations of domains, convex cells whose boundary faces are simplexes are also applied. Such cells are referred to as simplex cells.

For each n-dimensional simplex cell S, the following relation between the number of faces is valid:

$$\begin{aligned}
&\sum_{i=k}^{n-1}(-1)^i \binom{i+1}{k+1} N_i = (-1)^{n-1} N_k \ , \\
&k = -1, \cdots, n-2 \ , \qquad N_{-1} = 1 \ , \\
&\binom{l}{m} = \frac{l(l-1)\cdots(l-m+1)}{m!} \ , \quad m \geq 1 \ , \quad \binom{l}{0} = 1 \ ,
\end{aligned} \tag{1.2}$$

where N_i, $i = 1, \cdots, n$, is the number of i-dimensional boundary simplexes of S, and N_0 is the number of vertices of S.

Properties of Basic Cells

Some applications consider curvilinear cells as well (Fig. 1.1, below). These grid cells are obtained by deformation of ordinary linear segments, triangles, tetrahedrons, squares, cubes, and prisms.

The major advantage of hexahedral cells (quadrilaterals in two dimensions) is that their faces (or edges) may be aligned with the coordinate surfaces (or curves). In contrast, no coordinates can be aligned with tetrahedral meshes. However, strictly hexahedral meshes may be ineffective near boundaries with sharp corners.

Prismatic cells are generally placed near boundary surfaces which have previously been triangulated. The surface triangular cells serve as faces of prisms, which are grown out from these triangles. Prismatic cells are efficient for treating boundary layers, since they can be constructed with a high aspect ratio in order to resolve the layers, but without small angles, as would be the case for tetrahedral cells.

Triangular cells are the simplest two-dimensional elements and can be produced from quadrilateral cells by constructing interior edges. Analogously, tetrahedral cells are the simplest three-dimensional elements and can be derived from hexahedrons and prisms by constructing interior faces. The strength of triangular and tetrahedral cells is in their applicability to virtually any type of domain configuration. The drawback is that the integration of the physical equations becomes a few times more expensive with these cells in comparison with quadrilateral or hexahedral cells.

The vertices of the cells define grid points which approximate the physical domain. Alternatively, the grid points in the domain may have been generated previously by some other process. In this case, the construction of the grid cells requires special techniques.

1.2.2 Requirements Imposed on Grids

The grid should discretize the physical domain or surface in such a manner that the computation of the physical quantities is carried out as efficiently as desired. The accuracy, which is one of the components of the efficiency of the computation, is influenced by a number of grid factors, such as grid size, grid topology, cell shape and size, and consistency of the grid with the geometry and with the solution. A very general consideration of these grid factors is given in this subsection.

Grid Size and Cell Size

The grid size is indicated by the number of grid nodes, while the cell size implies the maximum value of the lengths of the cell edges. Grid generation requires techniques which possess the intrinsic ability to increase the number of grid nodes. At the same time, the edge lengths of the resulting cells should be reduced in such a manner that they approach zero as the number of nodes tends to infinity.

An instructive example of a grid on the interval [0,1] which does not satisfy the requirement of unlimited reduction of the cell sizes when the number of the nodes is increased is a grid generated by a rule in which the steps are in a geometrical progression:

$$\frac{h_{i+1}}{h_i} = a \ , \qquad a > 0 \ , \qquad a \neq 1 \ , \tag{1.3}$$

where $h_i = x_{i+1} - x_i, \ i = 0, \cdots, N-1$, are the steps of the grid nodes $x_i, \ i = 0, \cdots, N$, with $x_0 = 0, \ x_N = 1$. The grid nodes x_i satisfying (1.3) are computed for arbitrary N by the formula

$$x_i = \frac{a-1}{a^{N-1} - a} \sum_{j=1}^{i} a^j \ , \qquad i = 1, \cdots, N \ ,$$

and consequently we obtain

$$h_i = \frac{a^{i+1}(a-1)}{a^{N-1} - a} \ , \qquad i = 0, \cdots, N-1 \ .$$

Therefore,

$$\lim\nolimits_{N\to\infty} h_0 = 1 - a \qquad \text{if} \qquad 0 < a < 1 \ ,$$

$$\lim\nolimits_{N\to\infty} h_{N-1} = \frac{a-1}{a} \qquad \text{if} \qquad a > 1 \ ,$$

i.e. the left-hand boundary cell of this grid, if $a < 1$, or the right-hand boundary cell, if $a > 1$, does not approach zero even though the number of grid points tends to infinity.

Small cells are necessary to obtain more accurate solutions and to investigate phenomena associated with the physical quantities of small scales, such as boundary and transition layers and turbulence. Also, the opportunity to increase the number of grid points and to reduce the size of the cells enables one to study the convergence rate of a numerical code and to improve the accuracy of the solution by multigrid approaches.

Grid Consistency

By a consistent grid or a consistent discretization, we mean a collection of n-dimensional strongly convex cells satisfying the following condition: if two different cells intersect, then the region of the intersection is a common ($n - 1$-dimensional)

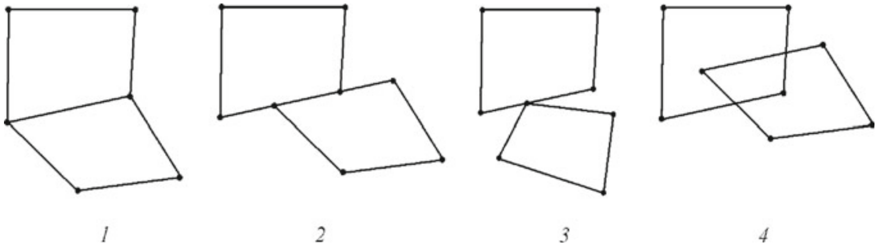


Fig. 1.3 Admitted (*1*) and nonadmitted (*2, 3, 4*) intersections of neighboring quadrilateral cells

face for both cells. This definition does not admit the fragments of two-dimensional discretizations depicted in Fig. 1.3 (*2, 3, 4*). If the union of the cells of the consistent discretization constitutes a simply connected n-dimensional geometry S^{xn}, i.e. a geometry which is homeomorphic to an n-dimensional cube, then, in accordance with the Euler theorem,

$$\sum_{i=0}^{n-1}(-1)^i N_i = 1 + (-1)^{n-1} , \tag{1.4}$$

where $N_i,\ i > 0$, is the number of i-dimensional boundary faces of the domain discretization, while N_0 is the number of boundary vertices. In particular, N_1 is the number of boundary edges. This relation can be used to verify the consistency of a generated grid.

Three-Dimensional Discretization

In three dimensions, we have, for each consistent discretization, the following generalization of the Euler formula (1.4):

$$N_0 - N_1 + N_2 = 2(k - t) , \tag{1.5}$$

where k is the number of simply connected subdomains and t is the number of holes.

For any convex three-dimensional cell, we also have the relations

$$\begin{aligned} N_0 - N_1 + N_2 &= 2 , \\ 4 \le N_0 &\le 2N_2 - 4 , \\ 4 \le N_2 &\le 2N_0 - 4 . \end{aligned} \tag{1.6}$$

Let $N_0(k)$ be the number of those vertices, each of which is common to k edges of a convex cell S. Analogously, by $N_2(k)$ we denote the number of faces, each of which is formed by k edges. Obviously,

$$N_0 = \sum_{k\ge 3} N_0(k) , \qquad N_2 = \sum_{k\ge 3} N_2(k) .$$

Using (1.4), we obtain

$$N_0(3) + N_2(3) = 8 + \sum_{k \geq 4} (k-4)[N_0(k) + N_2(k)] \geq 8 \ . \tag{1.7}$$

If the cell is a simplex cell, then we have, from (1.2), the relations

$$\begin{aligned} &N_0 - N_1 + N_2 = 2 \ , \\ &-2N_1 + 3N_2 = 0 \end{aligned} \tag{1.8}$$

or, in equivalent form,

$$\begin{aligned} N_1 &= 3N_0 - 6 \ , \\ N_2 &= 2N_0 - 4 \ . \end{aligned} \tag{1.9}$$

Also, the following nonlinear inequalities for a simplex cell are valid:

$$\begin{aligned} 2N_1 &< (N_0 - 1)N_0 \ , \\ 3N_2 &< (N_0 - 2)N_1 \ . \end{aligned} \tag{1.10}$$

Discretization by Triangulation

A consistent discretization by simplexes is called a triangulation. Let the number of edges be maximal for a given set of vertices of a two-dimensional triangulation. Let $C(P)$ be the boundary of the hull formed by all vertices of the triangulation and N_C be the number of vertices which lie in $C(P)$. Then, the following relations are valid:

$$\begin{aligned} N_T &= 2(N_V - 1) - N_C \ , \\ N_E &= 3(N_V - 1) - N_C \ , \end{aligned} \tag{1.11}$$

where N_T is the number of triangles, N_E is the number of edges, and N_V is the number of vertices of the triangulation. Thus, the maximal triangulation for the given vertices has a fixed number of triangles and edges.

Grid Organization

There is also a requirement on grids to have some organization of their nodes and cells, which is aimed at facilitating the procedures for formulating and solving the algebraic equations substituted for the differential equations. This organization should identify neighboring points and cells. The grid organization is especially important for that class of finite-difference methods whose procedures for obtaining the algebraic equations consist of substituting differences for derivatives. To a lesser degree, this organization is needed for finite-element methods because of their inherent compatibility with irregular meshes.

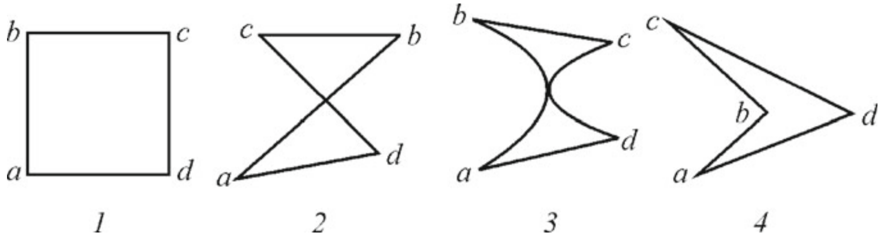


Fig. 1.4 Normal (*1*) and badly deformed (*2-4*) quadrilateral cells

Cell and Grid Deformation

The cell deformation characteristics can be formulated as some measures of the departure of the cell from a reference, represented by the least deformed one (Fig. 1.1, above). Such reference triangular and tetrahedral cells are those with edges of equal length. The least distorted quadrilaterals and hexahedrons are squares and cubes, respectively. The reference prism is evidently the prism with reference linear faces. Cells with low deformity are preferable from the point of view of simplicity and uniformity of the construction of the algebraic equations approximating the differential equations.

Typically, cell deformation is characterized through the edge length, the circumradius, the inradius, the aspect ratio, the angles between the cell edges, and the volume (area in two dimensions) of the cell.

The major requirement for the grid cells is that they must not be folded or degenerate at any points or lines, as demonstrated in Fig. 1.4. Unfolded cells are obtained from reference cells by a one-to-one deformation. Commonly, the value of any grid generation method is judged by its ability to yield grids with unfolded cells in regions with complex geometry.

The grid deformity is also characterized by the rate of the change of the geometrical features of contiguous cells. Grids whose neighboring cells do not change abruptly are referred to as smooth grids.

Consistency with Geometry

The accuracy of the numerical solution of a partial differential equation and of the interpolation of a discrete function is considerably influenced by the degree of compatibility of the mesh with the geometry of the physical domain. First of all, the grid nodes must adequately approximate the original geometry, that is, the distance between any point of the domain and the nearest grid node must not be too large. Moreover, this distance must approach zero when the number of grid nodes tends to infinity. This requirement of adequate geometry approximation by the grid nodes is indispensable for the accurate computation and interpolation of the solution over the whole region.

The second requirement for consistency of the grid with the geometry is concerned with the approximation of the boundary of the physical domain by the grid, i.e. there is to be a sufficient number of nodes which can be considered to be the boundary ones, so that a set of edges (in two dimensions) and cell faces (in three dimensions) formed by these nodes models the boundary efficiently. In this case, the boundary conditions may be applied more easily and accurately. If these points lie on the boundary of the domain, then the grid is referred to as a boundary-fitting or boundary-conforming grid.

Consistency with Solution

It is evident that distribution of the grid points and the form of the grid cells should be dependent on the features of the physical solution. In particular, it is better to generate the cells in the shape of hexahedrons or prisms in boundary layers. Often, the grid points are to be aligned with some preferred directions, e.g. streamlines or vector fields. Furthermore, a nonuniform variation of the solution requires clustering of the grid points in regions of high gradients, so that these areas of the domain have finer resolution. Local grid clustering is needed because the uniform refinement of the entire domain may be very costly for multidimensional computations. It is especially true for problems whose solutions have localized regions of very rapid variation (layers). Without grid clustering in the layers, some important features of the solution can be missed, and the accuracy of the solution can be degraded. Problems with boundary and interior layers occur in many areas of application, for example, in fluid dynamics, combustion, solidification, solid mechanics and wave propagation.

The typical pattern of a solution with large local variation is illustrated by the following univariate monotonic function:

$$u(x) = 1 - \exp(-x/\epsilon) \ , \qquad 0 \le x \le 1 \ ,$$

with a positive parameter ϵ. This function is a solution to the two-point singularly-perturbated boundary value problem

$$\begin{aligned} &\epsilon u'' + u' = 0 \ , \qquad 1 > x > 0 \ , \\ &u(0) = 0 \ , \qquad u(1) = 1 - \exp(-1/\epsilon) \ . \end{aligned}$$

When the parameter ϵ is very small, then $u(x)$ has a boundary layer of rapid variation (Fig. 1.5). Namely, in the interval $[0, \ \epsilon|\ln\epsilon|]$, the function $u(x)$ changes from 0 to $1 - \epsilon$. For example, if $\epsilon = 10^{-5}$, then $\epsilon|\ln\epsilon| = 5 \times 10^{-5} \ln 10 < 2 \times 10^{-4}$. In this small interval, the variation of the function $u(x)$ is $1 - 10^{-5}$.

Let the number of uniform grid points required for the accurate approximation of $u(x)$ on the boundary layer be N_0. Then, the number of uniform grid points on [0, 1] with the same step as in the boundary layer will be

$$N = N_0/\epsilon|\ln\epsilon| \ge 2 \times 10^4 N_0.$$

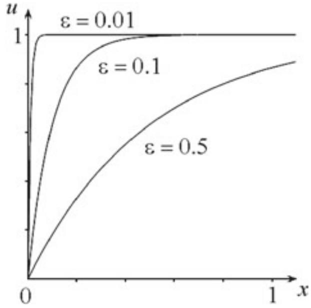


Fig. 1.5 Boundary layer function

However, in order to approximate $u(x)$ with the same accuracy in the interval $[\epsilon|\ln\epsilon|, 1]$, it is not necessary to use more than N_0 points of the uniform grid, since $u(x)$ is monotonic and changes with nearly the same variation in this interval as it does in the boundary layer. Thus, instead of $2 \times 10^4 N_0$, we can restrict the number of grid nodes on the interval [0, 1] to $2N_0$ in order to obtain the same accuracy of interpolation. This spectacular reduction in the number of grid points is obtained at the expense of using a finer grid in the boundary layer only.

This example clearly demonstrates that local grid refinement for problems in which the solution quantities have narrow zones in which the dominant length scales are very small is more promising than the uniform refinement of the entire region, since a significant reduction in the total number of grid nodes and, consequently, in the solution time can be attained. Local refinement becomes indispensable for complex geometries in three dimensions, since otherwise the cost of the numerical solution of a physical problem on the grid will be too high.

The locations of the zones of local refinement are also dependent on the numerical approximation to the physical equations. In particular, the areas of high solution error require more refined grid cells. However, the error is estimated through the derivatives of the solution and the size of the grid cells. Thus, ultimately, the grid point locations are to be defined in accordance with the derivatives of the solution.

Typically the locations where the high resolution is needed are not known beforehand but are found in the process of computation. Consequently, a suitable mesh, tracking the necessary features of the physical quantities as the solution evolves, is required.

A local grid refinement is accomplished in two ways: (a) by moving a fixed number of grid nodes, clustering of them in zones where this is necessary, and coarsening outside of these zones, and (b) by inserting new points in the zones of the domain where they are needed. Local grid refinement in zones of large variation of the solution commonly results in the following improvements:

(1) the solution at the grid points is obtained more accurately;
(2) the solution is interpolated over the whole region more precisely;
(3) oscillations of the solution are eliminated;

(4) larger time steps can be taken in the process of computing solutions of time-dependent problems.

Compatibility with Numerical Methods

In general, numerical methods for solving partial differential equations can be divided into two classes: methods based on direct approximations of the derivatives in the differential equation and methods that approximate the solution of the continuum differential equation by linear combinations of trial functions. Finite-difference methods belong to the first class. This difference in methods has a direct impact on the construction of the numerical grid. For the finite-difference methods, it is desirable to locate the grid points along directions of constant coordinates in the physical region in order to provide a natural approximation of the derivatives: on the other hand, the methods in the second class that approximate the solution with trial functions do not impose such a restriction on the grid, since the approximate derivatives are obtained after substitution of the approximate solution.

1.2.3 Grid Classes

There are two fundamental classes of grid popular in the numerical solution of boundary value problems in multidimensional regions: structured and unstructured. These classes differ in the way in which the grid cells and their nodes are locally organized. In the most general sense, this means that if the local organization of the grid nodes and the form of the grid cells do not depend on their position but are defined by a general rule, the grid is considered to be structured. When the connection of the neighboring grid nodes varies from point to point, the grid is called an unstructured grid or unstructured mesh. As a result, in the structured case, the connectivity of the grid is implicitly taken into account, while the connectivity of unstructured meshes must be explicitly described by an appropriate data structure procedure.

The two fundamental classes of mesh give rise to three additional subdivisions of grid types: block-structured, overset, and hybrid. These kinds of mesh possess to some extent the features of both structured and unstructured grids, thus occupying an intermediate position between the purely structured and unstructured grids.

The grid generation models can be classified into two types according to whether they lead to structured or unstructured grids. This subsection outlines some basic concepts related to structured and unstructured grids.

Structured Grids

The process of structured grid generation on the physical geometry S^{xn} locally represented by a parametrization

$$\mathbf{x}(\mathbf{s}) : S^n \to \mathrm{R}^{n+k}, \ \mathbf{x} = (x^1, \dots, x^{n+k}), \ \mathbf{s} = (s^1, \dots, s^n), \ n \geq 1 \ , \qquad (1.12)$$

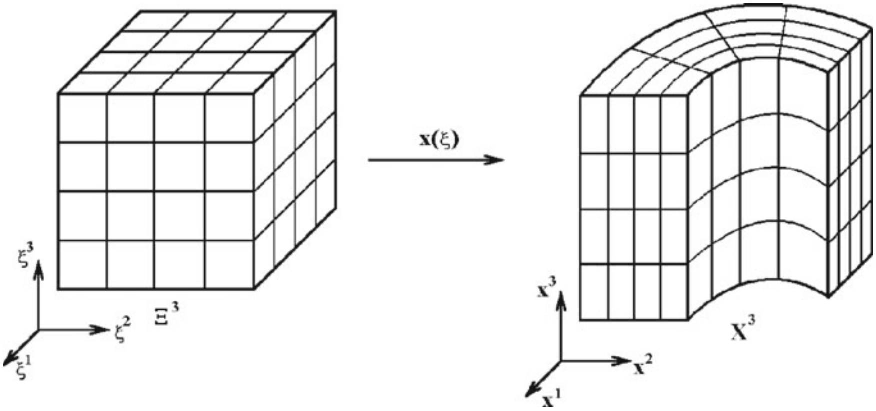


Fig. 1.6 Cylindrical structured grid

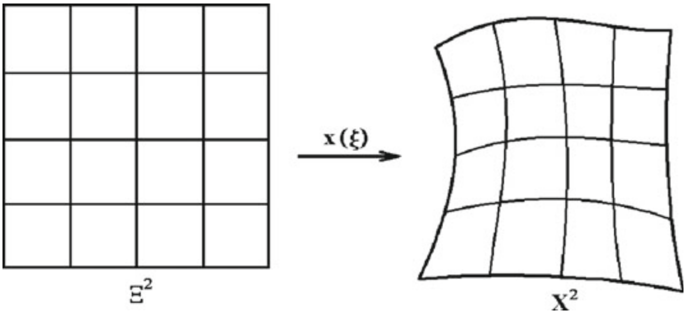


Fig. 1.7 Boundary-conforming quadrilateral grid

is generally carried out by the mapping approach that concludes with finding an intermediate transformation

$$\mathbf{s}(\boldsymbol{\xi}) : \Xi^n \to S^n \,, \quad \boldsymbol{\xi} = (\xi^1, \ldots, \xi^n) \tag{1.13}$$

from a suitable simple computational (logical) domain Ξ^n to the parametric domain S^n. Consequently, the mesh points on S^{xn} are generated as images through

$$\mathbf{x}[\mathbf{s}(\boldsymbol{\xi})] : \Xi^n \to R^{n+k} \tag{1.14}$$

of the nodes of a reference structured grid in Ξ^n (see Figs. 1.6, 1.7 and 1.8).

We will use a designation $\mathbf{x}(\boldsymbol{\xi})$ instead of $\mathbf{x}[\mathbf{s}(\boldsymbol{\xi})]$ in the case in which the physical geometry S^{xn} is a domain X^n and the parametric domain S^n coincides with X^n, while the parametric transformation $\mathbf{x}(\mathbf{s})$ is the identity function $\mathbf{x}(\mathbf{s}) \equiv \mathbf{s}$.

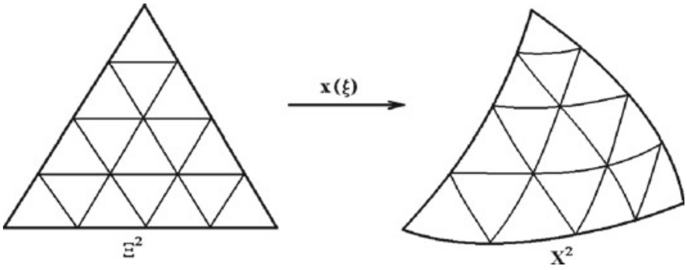


Fig. 1.8 Boundary-conforming triangular grid

The mapping concept was borrowed from examples of grids generated for geometries that are described by analytic coordinate transformations. In particular, two-dimensional transformations have often been defined by analytic functions of a complex variable and by direct shearing. This is the case, for example, for the polar coordinate system for circular regions

$$\boldsymbol{x}(\boldsymbol{\xi}) = \exp(\xi^1)(\cos \xi^2,\ \sin \xi^2)\ , \qquad r_0 \le \xi^1 \le r_1\ , \qquad 0 \le \xi^2 \le 2\pi\ .$$

As an illustrative example of a three-dimensional transformation, the following scaled cylindrical transformation may be considered:

$$\boldsymbol{x}(\boldsymbol{\xi}) : \varXi^3 \to X^3\ , \qquad \boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)\ , \qquad 0 \le \xi^i \le 1\ , \qquad i = 1, 2, 3\ ,$$

described by

$$\begin{aligned} x^1(\boldsymbol{\xi}) &= r\ \cos\theta\ , \\ x^2(\boldsymbol{\xi}) &= r\ \sin\theta\ , \\ x^3(\boldsymbol{\xi}) &= H\xi^3\ , \end{aligned} \tag{1.15}$$

where

$$r = r_0 + (r_1 - r_0)\xi^1\ , \qquad \theta = \theta_0 + (\theta_1 - \theta_0)\xi^2\ , \qquad H > 0\ ,$$

with

$$0 < r_0 < r_1\ , \qquad 0 \le \theta_0 < \theta_1 \le 2\pi\ .$$

If $\theta_1 = 2\pi$ then this function transforms the unit three-dimensional cube into a space bounded by two cylinders of radii r_0 and r_1 and by the two planes $x^3 = 0$ and $x^3 = H$. The reference uniform grid in $\varXi^3$ is defined by the nodes

$$\boldsymbol{\xi}_{ijk} = (ih, jh, kh)\ , \qquad 0 \le i, j, k \le N\ , \qquad h = 1/N\ ,$$

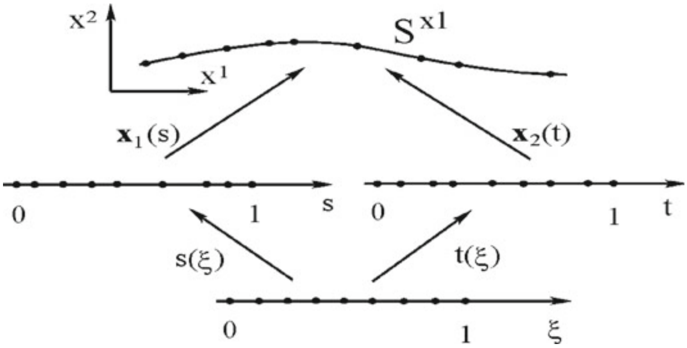


Fig. 1.9 Illustration for independence of grid generation of the choice of parametrizations

where i, j, k and N are positive integers. The cells of this grid are the three-dimensional cubes bounded by the coordinate planes $\xi^1_i = ih$, $\xi^2_j = jh$, and $\xi^3_k = kh$. Correspondingly, the structured grid in the domain X^3 is determined by the nodes

$$\boldsymbol{x}_{ijk} = \boldsymbol{x}(\boldsymbol{\xi}_{ijk}) \ , \qquad 0 \le i, j, k \le N \ .$$

The cells of the grid in X^3 are the curvilinear hexahedrons bounded by the curvilinear coordinate surfaces derived from the parametrization $\boldsymbol{x}(\boldsymbol{\xi})$ (Fig. 1.6).

Independence of Parametrizations of Geometries

If a grid algorithm uses parametrizations of a physical geometry S^{xn} in the process of grid generation, then, inevitably, this algorithm should be independent of the choice of a parametrization. To clarify this, we consider one popular equidistribution approach for generating grids on curves (Fig. 1.9). Let a curve S^{x1} in R^n be specified by two parametrizations

$$\mathbf{x}_1(s) : [0, 1] \to R^n \ , \quad \mathbf{x}_1 = (x^1_1, \dots, x^n_1) \ , \tag{1.16}$$

and

$$\mathbf{x}_2(t) : [0, 1] \to R^n \ , \quad \mathbf{x}_2(t) = \mathbf{x}_1[s(t)] \ , \tag{1.17}$$

where

$$s(t) : [0, 1] \to [0, 1]$$

is a smooth one-to-one function connecting these parametrizations. The popular universal approach, based on the parametrization (1.16), for generating grid nodes on S^{x1} uses a solution of the following two-point boundary value problem:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\xi}\left[\frac{\mathrm{d}s}{\mathrm{d}\xi} w_1(s)\right] = 0 \ , \quad 0 < \xi < 1 \ , \\ &s(0) = 0 \ , \quad s(1) = 1 \ , \end{aligned} \tag{1.18}$$

where $w_1(s) > 0$ is some function called a weight function. If $s(\xi)$ is a solution of this problem, then the grid nodes $\mathbf{x}_i$, $i = 0, 1, \dots, N$, on the curve S^{x1}, obtained by the method, are defined as follows:

$$\mathbf{x}_i = \mathbf{x}_1[s(ih)] \ , \quad i = 0, \dots, N \ , \quad h = 1/N \ .$$

Now, let the parametrization $\mathbf{x}_2(t)$ specified by (1.17) be used in the problem (1.18) with some weight function $w_2(t)$ for the generation of the same grid nodes on S^{x1}. In this case, the function $t_1(\xi)$ for which

$$\mathbf{x}_2(t_1(ih)) = \mathbf{x}_1[s(t_1(ih))] = \mathbf{x}_i = \mathbf{x}_1(s(ih)) \ , \quad i = 0, 1, \dots, n \ ,$$

must coincide with $t[s(\xi)]$, where $s(\xi)$ is the solution of (1.18), while $t(s)$ is the inverse of $s(t)$. Therefore, the function $t_1(\xi)$ is a solution of the boundary value problem

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left[\frac{\mathrm{d}t_1}{\mathrm{d}\xi}\frac{\mathrm{d}s}{\mathrm{d}t}w_1[s(t_1)]\right] = 0 \ , \quad 0 < \xi < 1 \ ,$$

$$t_1(0) = 0 \ , \quad t_1(1) = 1 \ .$$

Since the weight functions $w_1(s)$ and $w_2(t)$ for defining the same grid nodes on S^{x1} by the model (1.18) through the parametrizations (1.16) and (1.17), respectively, are not independent, they should be connected by the following relation:

$$w_2(t) = w_1[s(t)]\frac{\mathrm{d}s}{\mathrm{d}t} \ . \tag{1.19}$$

If this relation is not satisfied, the grid nodes obtained with the help of the solution of (1.18) may vary for different parametrizations of S^{x1}.

It appears that if we take for the weight functions related to the parametrization (1.16) and (1.17) the corresponding functions $w_1(s) = \sqrt{g^s}$ and $w_2(t) = \sqrt{g^t}$, where

$$g^s = \frac{\mathrm{d}\mathbf{x}_1}{\mathrm{d}s} \cdot \frac{\mathrm{d}\mathbf{x}_1}{\mathrm{d}s} \quad \text{and} \quad g^t = \frac{\mathrm{d}\mathbf{x}_2}{\mathrm{d}t} \cdot \frac{\mathrm{d}\mathbf{x}_2}{\mathrm{d}t}$$

is the covariant metric tensor of S^{x1} in the coordinate s and t, respectively, then the Eq. (1.19) holds since there is validity to the obvious equation

$$g^t = g^s\left(\frac{\mathrm{d}s}{\mathrm{d}t}\right)^2 \ .$$

The consideration given for the curvilinear curve also is actual for surfaces. As well as in the one-dimensional case, the application of the elements of the metric tensors of n-dimensional surfaces allows one to formulate grid equations which produce the same grid nodes for different surface parametrizations.

Realization of Grid Requirements

The notion of using an intermediate transformation to generate a mesh is very helpful. The idea is to choose a computational domain Ξ^n with a simpler geometry than that of the physical domain X^n or the parametric domain S^n and then to find a transformation $\boldsymbol{x}(\boldsymbol{\xi})$ between these domains which eliminates the need for a nonuniform mesh when approximating the physical quantities. That is, if the computational area and the transformation are well chosen, the transformed boundary value problem can be accurately represented by a small number of equally spaced mesh points. Emphasis is placed on a small number of points, because any transformed problem (provided only that the transformation is nonsingular) may be accurately approximated with a sufficiently fine, uniform mesh. In practice, there will be a trade-off between the difficulty of finding the transformation and the number of uniformly spaced points required to find the solution to a given accuracy.

The idea of using mappings to generate grids is extremely appropriate for finding the conditions that the grid must satisfy for obtaining accurate solutions of partial differential equations in the physical domain X^n, because these conditions can be readily defined in terms of the transformations. For example, the grid requirements described in Sect. 1.2.2 are readily formulated through the transformation concept.

Since a solution which is a linear function is computed accurately at the grid points and is approximated accurately over the whole region, an attractive possible method for generating structured grids is to find a transformation $\boldsymbol{x}(\boldsymbol{\xi})$ such that the solution is linear in Ξ^n. Though in practice, this requirement for the transformation is not attained even theoretically (except in the case of strongly monotonic univariate functions), it is useful in the sense of an ideal that the developers of structured grid generation techniques should bear in mind. One modification of this requirement which can be practically realized consists of the requirement of a local linearity of the solution in Ξ^n.

The requirements imposed on the grid and the cell size are realized by the construction of a uniform grid in Ξ^n and a smooth function $\boldsymbol{x}(\boldsymbol{\xi})$. The grid cells are not folded if $\boldsymbol{x}(\boldsymbol{\xi})$ is a one-to-one mapping. Consistency with the geometry is satisfied with a transformation $\boldsymbol{x}(\boldsymbol{\xi})$ that maps the boundary of Ξ^n onto the boundary of X^n. Grid concentration in zones of large variation of a function $\boldsymbol{u}(\boldsymbol{x})$ is accomplished with a mapping $\boldsymbol{x}(\boldsymbol{\xi})$ which provides variations of the function $\boldsymbol{u}[\boldsymbol{x}(\boldsymbol{\xi})]$ in the domain Ξ^n that are not large.

Coordinate Grids

Among structured grids, coordinate grids in which the nodes and cell faces are defined by the intersection of lines and surfaces of a coordinate system in X^n (Fig. 1.6 for $n = 3$) are very popular in finite-difference methods. The range of values of this system defines a computation region Ξ^n in which the cells of the uniform grid are rectangular n-dimensional parallelepipeds, and the coordinate values define the function $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n$.

The simplest of such grids are the Cartesian grids obtained by the intersection of the Cartesian coordinates in X^n. The cells of these grids are rectangular parallelepipeds (rectangles in two dimensions). The use of Cartesian coordinates avoids the need to transform the physical equations. However, the nodes of the Cartesian grid do not coincide with the curvilinear boundary, which leads to difficulties in implementing the boundary conditions with second-order accuracy.

Boundary-Conforming Grids

An important subdivision of structured grids is the boundary-fitted or boundary-conforming grids. These grids are obtained from one-to-one transformations $\boldsymbol{x}(\boldsymbol{\xi})$ which map the boundary of the domain Ξ^n onto the boundary of X^n (Figs. 1.7 and 1.8, for $n = 2$).

The most popular of these, for finite-difference methods, have turned out to be the coordinate boundary-fitted grids whose points are formed by intersection of the coordinate lines, while the boundary of X^n is composed of a finite number of coordinate surfaces (lines in two dimensions) $\xi^i = \xi^i_0$. Consequently, in this case, the computation region Ξ^n is a rectangular domain, the boundaries of which are determined by $(n-1)$-dimensional coordinate planes in R^n, and the uniform grid in Ξ^n is the Cartesian grid. Thus, the physical region is represented as a deformation of a rectangular domain and the generated grid as a deformed lattice (Figs. 1.6 and 1.7).

The boundary-conforming grids give a good approximation of the boundary of the region and are therefore suitable for the numerical solution of problems with boundary singularities, such as those with boundary layers in which the solution depends very much on the accuracy of the approximation of the boundary conditions.

The requirements imposed on boundary-conforming grids are naturally satisfied with the coordinate transformations $\boldsymbol{x}(\boldsymbol{\xi})$.

The algorithm for the organization of the nodes of boundary-fitted coordinate grids consists of the trivial identification of neighboring points by incrementing the coordinate indices, while the cells are curvilinear hexahedrons. This kind of grid is very suitable for algorithms with parallelization.

Its design makes it easy to increase or change the number of nodes as required for multigrid methods or in order to estimate the convergence rate and error, and to improve the accuracy of numerical methods for solving boundary value problems.

With boundary-conforming grids, there is no necessity to interpolate the boundary conditions of the problem, and the boundary values of the region can be considered as input data to the algorithm, so automatic codes for grid generation can be designed for a wide class of regions and problems.

In the case of unsteady problems, the most direct way to set up a moving grid is to do it via a coordinate transformation. These grids do not require a complicated data structure, since they are obtained from uniform grids in simple fixed domains, such as rectangular ones where the grid data structure remains intact.

Shape of Computational Domains

The idea of the structured approach is to substitute a complex physical domain X^n with a simpler domain Ξ^n with the help of the parametrization $\boldsymbol{x}(\boldsymbol{\xi})$. The region Ξ^n in (1.12), which is called the computational or logical region, can be either rectangular or of a different shape qualitatively matching the geometry of the physical domain

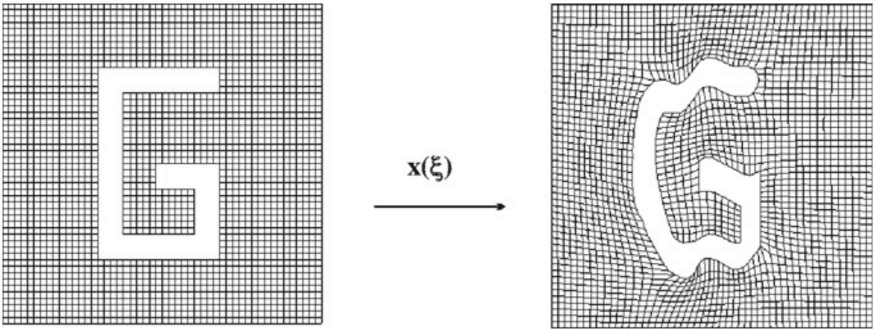


Fig. 1.10 Computational domain adjusted to the physical domain

(Fig. 1.10); in particular, its shape can be triangular for $n = 2$ (Fig. 1.8) or tetrahedral for $n = 3$. Using such parametrizations, a numerical solution of a partial differential equation in a physical region of arbitrary shape can be carried out in a standard computational domain, and codes can be developed that require only changes in the input.

The cells of the uniform grid in the computational domain Ξ^n can be rectangular or of a different shape. Schematic illustrations of two-dimensional triangular and quadrilateral grids are presented in Figs. 1.8 and 1.10, respectively. Note that regions in the form of curvilinear triangles, such as that shown in Fig. 1.8, are more suitable for gridding in the structured approach by triangular cells than by quadrilateral ones.

Unstructured Grids

Many field problems of interest involve very complex geometries that are not easily amenable to the framework of the pure structured grid concept. Structured grids may lack the required flexibility and robustness for handling domains with complicated boundaries, or the grid cells may become too skewed and twisted, thus prohibiting an efficient numerical solution. An unstructured grid concept is considered to be one of the appropriate solutions to the problem of producing grids in regions with complex shapes.

Unstructured grids have irregularly distributed nodes and their cells are not obliged to have any one standard shape. Besides this, the connectivity of neighboring grid cells is not subject to any restrictions; in particular, the cells can overlap or enclose one another. Thus, unstructured grids provide the most flexible tool for the discrete description of a geometry.

These grids are suitable for the discretization of domains with a complicated shape, such as regions around aircraft surfaces or turbomachinery blade rows. They also allow one to apply a natural approach to local adaptation, through either insertion or removal of nodes. Cell refinement in an unstructured system can be accomplished locally by dividing the cells in the appropriate zones into a few smaller cells. Unstructured grids also allow for excessive resolution to be removed by deleting grid cells locally over regions in which the solution does not vary appreciably.

However, the use of unstructured grids complicates the numerical algorithm because of the inherent data management problem, which demands a special program to number and order the nodes, edges, faces, and cells of the grid, with extra memory being required to store information about the connections between the cells of the mesh. One further disadvantage of unstructured grids that causes excessive computational work is associated with increased numbers of cells, cell faces, and edges in comparison with those for hexahedral meshes. For example, a tetrahedral mesh of N points has roughly $6N$ cells, $12N$ faces, and $7N$ edges, while a mesh of hexahedra has roughly N cells, $3N$ faces, and $3N$ edges. Furthermore, moving boundaries or moving internal surfaces of physical domains are difficult to handle with unstructured grids. Besides this, linearized difference scheme operators on unstructured grids are not usually band matrices, which makes it more difficult to use implicit schemes. As a result, the numerical algorithms based on an unstructured grid topology are the most costly in terms of operations per time step and memory per grid point.

Originally, unstructured grids were mainly used in the theory of elasticity and plasticity, and in numerical algorithms based on finite-element methods. However, the field of application of unstructured grids has now expanded considerably and includes three-dimensional fluid dynamics problems with shock waves. Some important aspects of the construction of unstructured grids are considered in Chap. 12.

Block-Structured Grids

In the commonly applied block strategy, the region is divided without holes or overlaps into a few contiguous subdomains, which may be considered as the cells of a coarse, generally unstructured grid (see Fig. 1.11 for a tokamak-related domain). And then, a separate structured grid is generated in each block. The union of these local grids constitutes a mesh referred to as a block-structured or multi-block grid. Grids of this kind can thus be considered as locally structured at the level of an individual block, but globally unstructured when viewed as a collection of blocks. Thus, a common idea in the block-structured grid technique is the use of different structured grids, or coordinate systems, in different regions, allowing the most appropriate grid configuration to be used in each region.

Block-structured grids are considerably more flexible in handling complex geometries than structured grids. Since these grids retain the simple regular connectivity pattern of a structured mesh on a local level, these block-structured grids maintain, in nearly the same manner as structured grids, compatibility with efficient finite-difference or finite-volume algorithms used to solve partial differential equations. However, the generation of block-structured grids may take a fair amount of user interaction and, therefore, requires the implementation of an automation technique to lay out the block topology.

The main reasons for using multi-block grids rather than single-block grids are that

(1) the geometry of the region is complicated, having a multiply connected boundary, cuts, narrow protuberances, cavities, etc.;

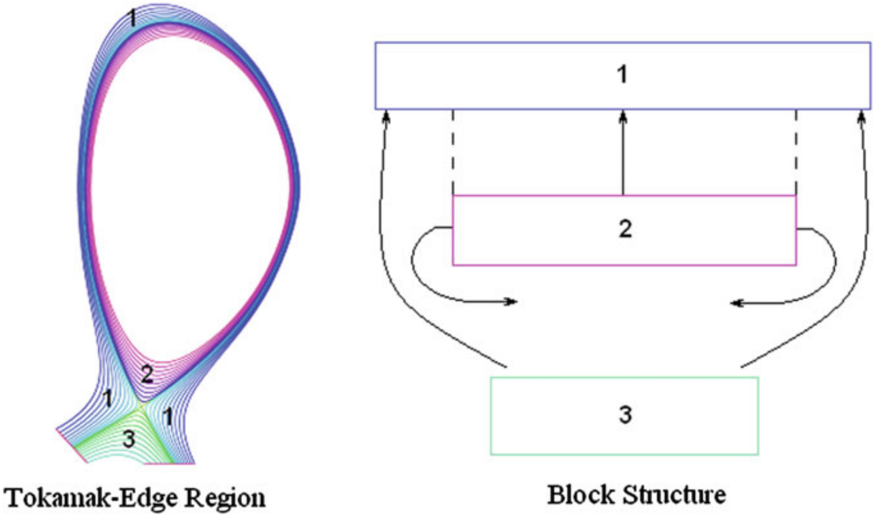


Fig. 1.11 Section of a Tokamak-Edge region (*left*), block-structure (*right*)

(2) the physical problem is heterogeneous relative to some of the physical quantities, so that different mathematical models are required in different zones of the domain to adequately describe the physical phenomena;
(3) the solution to the problem behaves non-uniformly: zones of smooth and rapid variation of different scales may exist.

The blocks of locally structured grids in a three-dimensional region are commonly homeomorphic to a three-dimensional cube, thus having the shape of a curvilinear hexahedron. However, some domains can be more effectively partitioned with the use of cylindrical blocks as well. Cylindrical blocks are commonly applied to the numerical solution of problems in regions with holes and to the calculation of flows past aircraft or aircraft components (wings, fuselages, etc.). For many problems, it is easier to take into account the geometry of the region and the structure of the solution by using cylindrical blocks. Also, the total number of blocks and sections might be smaller than in a case in which only blocks homeomorphic to a cube are used.

Communication of Adjacent Coordinate Lines

The requirement of mutual positioning or “communication” of adjacent grid blocks can also have a considerable influence on the construction of locally structured grids and on the efficiency of the numerical calculations. The coordinate lines defining the grid nodes of two adjacent blocks need not have points in common, and can join smoothly or nonsmoothly (Fig. 1.12). If all adjacent grid blocks join smoothly, interpolation is not required. If the coordinate lines do not join, then during the calculation the solution values at the nodes of one block must be transferred to those of the adjacent block in the neighborhood of their intersection. This is done by interpolation or (in mechanics) using conservation laws.

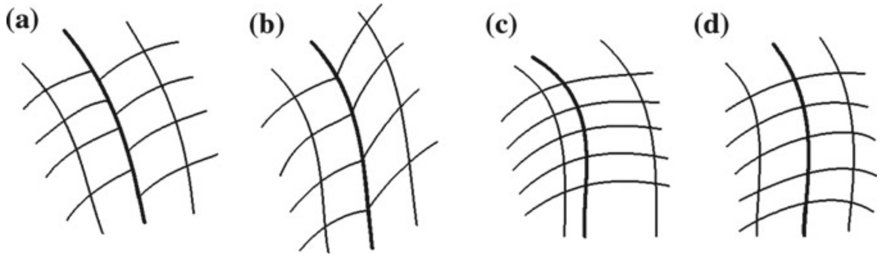


Fig. 1.12 Types of interface between contiguous blocks (**a** discontinuous; **b, c** nonsmooth; **d** smooth)

The types of interaction between adjacent grid blocks are selected on the basis of the features of the physical quantities in the region of their intersection. If the gradient of the physical solution is not high in the vicinity of a boundary between two adjacent blocks and interpolation can, therefore, be performed with high accuracy, the coordinate lines do not need to join. This greatly simplifies the algorithm for constructing the grid in a block. If there are high gradients of the solution near the intersection of two blocks, a smooth matching is usually performed between the coordinate lines of the two blocks. This kind of conformity poses a serious difficulty for structured mesh generation methods. Currently, the problem is overcome through an algebraic technique using Hermitian interpolation, or through elliptic methods involving a choice of control functions. A combination of Laplace and Poisson equations, yielding equations of fourth or even sixth order, is also used for this purpose.

Topology of the Grid

The correct choice of the topology in a block, depending on the geometry of the computational region and the choice of the transformation of the region into the block, has a considerable influence on the quality of the grid. There are two ways of specifying the computational region for a block:

(1) as a complicated polyhedron which maintains the schematic form of the block subdomain (Fig. 1.10);
(2) simply as a solid parallelepiped, triangle, trapezium (parallelogram, tetrahedral, prism) (Figs. 1.14 and 1.15) or a parallelepiped, triangle, trapezium (parallelogram, tetrahedral, prism) with cuts (Fig. 1.13).

With the first approach, the problem of constructing the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is simplified, and this method is often used to generate a single-blocked grid in a complicated domain. The second approach relies on a simplified geometry of the computational domain, but requires sophisticated methods to derive suitable transformations $\boldsymbol{x}(\boldsymbol{\xi})$.

In a block which is homeomorphic to a cylinder with thick walls, the grid topology is determined by the topology of the two-dimensional grids in the transverse sections. In applications, for sections of this kind, which are annular planes or surfaces with

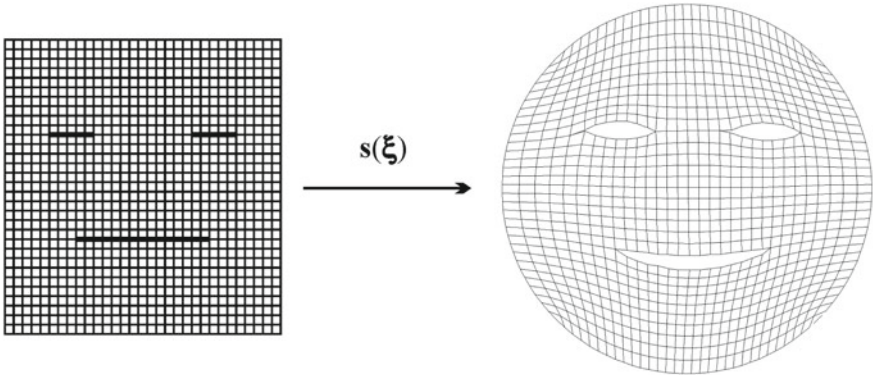


Fig. 1.13 H-type grid

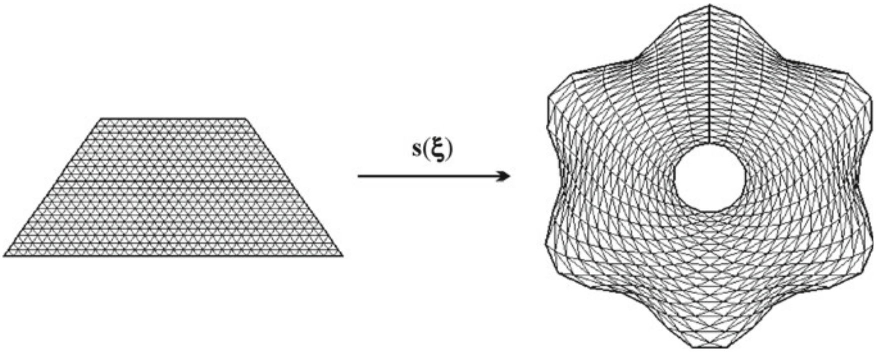


Fig. 1.14 O-type grid

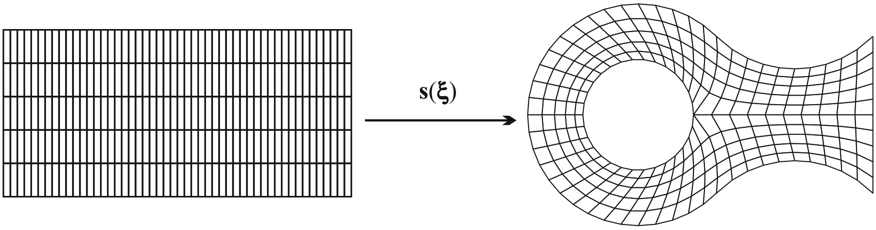


Fig. 1.15 C-type grid

a hole, wide use is made of three basic grid topologies: H, O and C (see Figs. 1.13, 1.14 and 1.15).

In H-type grids, the computational region is a square with an interior cut which is opened by the construction of the coordinate transformation and mapped onto an interior boundary of the region X^2. The outer boundary of the square is mapped onto the exterior of X^2. The interior boundary has two points with singularities where one coordinate line splits. H-type grids are used, for instance, when calculating the flow past thin bodies (aircraft wings, turbine blades, etc.).

In O-type grids, the computational region is a solid square. In this case, the system of coordinates is obtained by bending the square, sticking two opposite sides together and then deforming. The stuck sides determine the cut, called the fictive edge, in the block. An example of an O-type grid is the nodes and cells of a polar system of coordinates. The O-type grid can be constructed without singularities when the boundary of the region is smooth. Grids of this kind are used when calculating the flow past bulky aircraft components (fuselages, gondolas, etc.) and, in combination with H-type grids, for multilayered block structures.

The computational region is also a solid square in a C-type grid, but the mapping onto X^2 involves the identification of some segments of one of its sides and then its deformation. In the C-type grid, the coordinate lines of one family leave the outer boundary, circle the inner boundary and return again to the outer boundary. There is one point on the inner boundary which has the same type of singularity as in the H-type grid. The C-type grids are commonly used in regions with holes and long protuberances.

The O and C-type techniques, in fact, introduce artificial interior cuts in multiply connected regions to generate single block-structured grids. The cuts are used to join the disconnected components of the domain boundary in order to reduce their number. Theoretically, this operation can allow one to generate a single coordinate transformation in a multiply connected domain.

The choice of the grid topology in a block depends on the structure of the solution, the geometry of the domain, and, in the case of continuous or smooth grid-line communication, on the topology of the grid in the adjacent block as well. For complicated domains, such as those near aircraft surfaces or turbines with a large number of blades, it is difficult to choose the grid topology of the blocks, because each component of the system (wing, fuselage, etc.) has its own natural type of grid topology, but these topologies are usually incompatible with each other.

Conditions Imposed on Grids in Blocks

A grid in a block must satisfy the conditions which are required to obtain an acceptable solution. In any specific case, these conditions are determined by features of the computer, the methods of grid generation available, the topology and conditions of interaction of the blocks, the numerical algorithms, and the type of data to be obtained.

One of the main requirements imposed on the grid is its adaptation to the solution. Multidimensional computations are likely to be very costly without the application of adaptive grid techniques. The basic aim of adaptation is to enhance the efficiency of numerical algorithms for solving physical problems through a special nonuniform distribution of grid nodes. The appropriate adaptive displacement of the nodes, depending on the physical solution, can increase the accuracy and rate of convergence and reduce oscillations and the interpolation error.

In addition to adaptation, the construction of locally structured grids often requires the coordinate lines to cross the boundary of the domain or the surface in an orthogonal or nearly orthogonal fashion. The orthogonality at the boundary can greatly simplify the specification of boundary conditions. Also, a more accurate representation of algebraic models of turbulence, the equations of a boundary layer, and parabolic Navier–Stokes equations is possible in this case. If for grids of O and C-type, the coordinate lines are orthogonal to the boundary of each block and its interior cuts, the global block-structured grid will be smooth. It is also desirable for the coordinate lines to be orthogonal or nearly orthogonal inside the blocks. This will improve the convergence of the difference algorithms, and the equations, if written in orthogonal variables, will have a simpler form.

For unsteady gas-dynamics problems, some coordinates in the entire domain or on the boundary are required to have Lagrange or nearly Lagrange properties. With Lagrangian coordinates, the computational region remains fixed in time and simpler expressions for the equations can be obtained in this case.

It is also important that the grid cells do not collapse, the changes in the steps are not too abrupt, the lengths of the cell sides are not very different, and the cells are finer in any domain of high gradient, large error, or slow convergence. Requirements of this kind are taken into account by introducing quantitative and qualitative characteristics of the grid, both with the help of coordinate transformations and by using the sizes of cell edges, faces, angles, and volumes. The characteristics used include the deviation from orthogonality, the Lagrange properties, the values of the transformation Jacobian or cell volume, and the smoothness and adaptivity of the transformation. For cell faces, the deviation from a parallelogram, rectangle, or square, as well as the ratio of the area of the face to its perimeter, is also used.

Overset Grids

Block-structured grids require the partition of the domain into blocks that are restricted so as to abut each other. Overset grids are exempt from this restriction. With the overset concept, the blocks are allowed to overlap, which significantly simplifies the problem of the selection of the blocks covering the physical region. In fact, each block may be a subdomain which is associated only with a single geometry or physical feature. The global grid is obtained as an assembly of structured grids which are generated separately in each block. These structured grids are overset on each other, with data communicated by interpolation in overlapping areas of the blocks (Fig. 1.16).

Hybrid Grids

Hybrid numerical grids are meshes which are obtained by combining both structured and unstructured grids. These meshes are widely used for the numerical analysis of boundary value problems in regions with a complex geometry and with a solution of complicated structure. They are formed by joining structured and unstructured grids on different parts of the region or surface. Commonly, a structured grid is generated about each chosen boundary segment. These structured grids are required not to overlap. The remainder of the domain is filled with the cells of an unstructured grid (Fig. 1.17). This construction is widely applied for the numerical solution of problems with boundary layers.

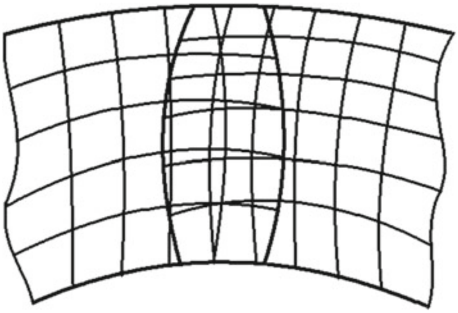


Fig. 1.16 Fragment of an overset grid

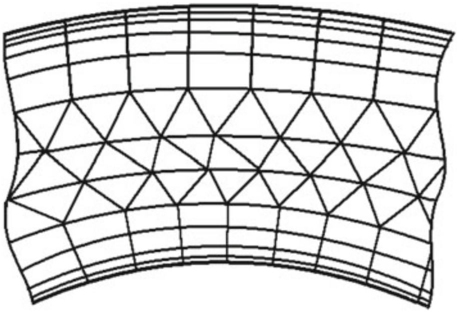


Fig. 1.17 Fragment of a hybrid grid

1.3 Methods for Grid Generation

The unique aspect of grid generation on general domains is that grid generation has a high degree of freedom, i.e. mesh techniques are not obliged to have any specified formulation, so any foundation may be suitable for this purpose if the grid generated is acceptable.

The chief practical difficulty facing grid generation techniques is that of formulating satisfactory techniques which can realize the user's requirements. Grid generation techniques should develop methods that can help in handling problems with multiple variables, each varying over many orders of magnitude. These methods should be capable of generating grids which are locally compressed by large factors when compared with uniform grids.

The methods should incorporate specific control tools, with simple and clear relationships between these control tools and characteristics of the grid such as mesh spacing, skewness, smoothness, and aspect ratio, in order to provide a reliable way to influence the efficiency of the computation. And finally, the methods should be computationally efficient and easy to code.

A number of techniques for grid generation have been developed. Every method has its strengths and its weaknesses. Therefore, there is also the question of how to choose the most efficient method for the solution of any specific problem, taking into

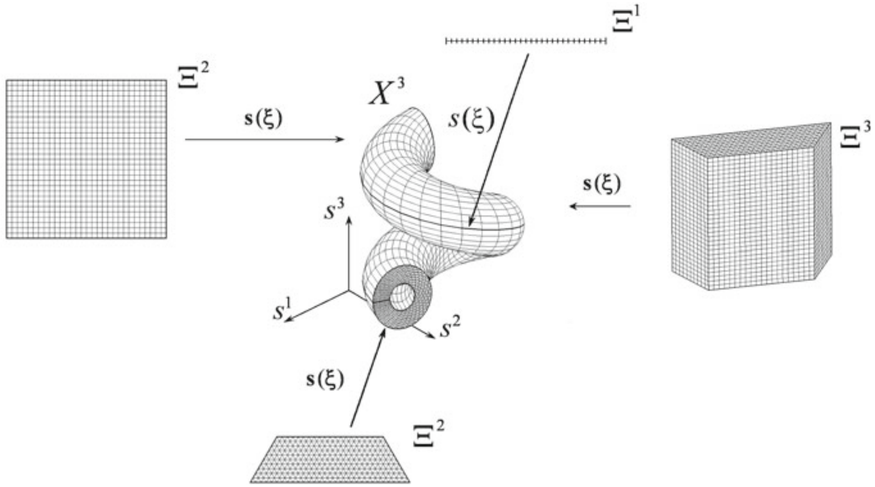


Fig. 1.18 Process for generating prismatic grid

account the geometrical complexity, the computing cost for generating the grid, the grid structure, and other factors.

The goal of the development of these methods is to provide effective and acceptable grid generation systems.

1.3.1 Mapping Methods

The structured grids are typically generated through a mapping approach. In this subsection, the main mapping methods generally used to create structured grids are briefly described. The basic idea common to all structured grid generation methods consists of meshing a canonical (computational) domain Ξ^n of a simple shape and mapping this mesh to a physical domain X^n or to a physical geometry S^{xn}, represented by Eq. (1.12).

The most efficient structured grids are boundary-conforming grids. The generation of these grids can be performed through a number of approaches and techniques. Many of these methods are specifically oriented to the generation of grids for the finite-difference method.

A boundary-fitted coordinate grid in the region X^n is commonly generated first on the boundary of X^n and then successively extended from the boundary to the interior of X^n (Fig. 1.18) This process is analogous to the interpolation of a function from a boundary or to the solution of a differential boundary value problem. On this basis, three basic groups of mapping methods of grid generation have been developed:

(1) algebraic methods, which use various forms of interpolation or special functions;
(2) differential methods, based mainly on the solution of elliptic, parabolic, and hyperbolic equations in a selected transformed region;
(3) variational methods, based on optimization of grid quality properties.

Algebraic Methods

In the algebraic approach, the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$ which extends the boundary mapping $\partial\mathbf{s}(\boldsymbol{\xi}) : \partial\Xi^n \to \partial S^n$ found in the previous step is commonly computed through the formulas of transfinite interpolation. There are two types of transfinite interpolations popular with grid generation: Lagrange and Hermite.

In particular, the three-dimensional Lagrange intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$ between the interior of the unit cube $0 \leq \xi^i \leq 1$, $i = 1, 2, 3$, and the parametric domain S^3 is defined by the following recursive formula:

$$\begin{aligned}
\boldsymbol{F}_1(\boldsymbol{\xi}) &= \alpha_0^1(\xi^1)\boldsymbol{s}(0, \xi^2, \xi^3) + \alpha_1^1(\xi^1)\boldsymbol{s}(1, \xi^2, \xi^3) \ , \\
\boldsymbol{F}_2(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\boldsymbol{\xi}) + \alpha_0^2(\xi^2)\boldsymbol{s}(\xi^1, 0, \xi^3) + \alpha_1^2(\xi^2)\boldsymbol{s}(\xi^1, 1, \xi^3) \ , \\
\boldsymbol{s}(\boldsymbol{\xi}) &= \boldsymbol{F}_2(\boldsymbol{\xi}) + \alpha_0^3(\xi^3)\boldsymbol{s}(\xi^1, \xi^2, 0) + \alpha_1^3(\xi^3)\boldsymbol{s}(\xi^1, \xi^2, 1) \ ,
\end{aligned}$$

where the univariate functions $\alpha_k^i(\xi)$, $i = 1, 2, 3$, $k = 0, 1$, referred to as blending functions, are subject to the relations of consistency

$$\begin{aligned}
\alpha_0^i(0) = 1 \ , &\quad \alpha_0^i(1) = 0 \ , \\
\alpha_1^i(0) = 0 \ , &\quad \alpha_1^i(1) = 1 \ .
\end{aligned}$$

Analogous formulas are held for the Hermite interpolation that matches at the points of the boundary of Ξ^n the values of both the function $\boldsymbol{s}(\boldsymbol{\xi})$ and its first derivatives in the directions normal to the boundary segments. A detailed review of the Lagrange and Hermite techniques for generating algebraic grids is presented in the monograph of Liseikin (1999).

Algebraic methods are simple; they enable the grid to be generated rapidly and the spacing and slope of the coordinate lines to be controlled by the tangential derivatives at the boundary points and blending coefficients in the transfinite interpolation formulas. However, in regions of complicated shape, the coordinate surfaces obtained through algebraic methods can become degenerate or the cells can overlap or cross the boundary. Moreover, they basically preserve the features of the boundary surfaces, in particular, discontinuties. Besides this, the algebraic methods of transfinite interpolation do not guarantee the independence of grid nodes displacement on parametrizations of a physical geometry.

Algebraic approaches are commonly used to generate grids in regions with smooth boundaries that are not highly deformed, or as an initial approximation in order to start the iterative process of an elliptic grid solver.

The construction of intermediate transformations through the formulas of transfinite interpolation was originally formulated by Gordon and Hall (1973) and Gordon and Thiel (1982). The Hermite interpolation was presented by Smith (1982).

The multisurface method which allows for the specification of the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$ at the points of some interior sections of the physical domain was described by Eiseman (1980). The blending functions were implicitly derived from global and/or local interpolants which result from an expression for the tangential derivative spanning between the exterior boundary surfaces. A two-boundary technique was introduced by Smith (1981). It is based on the description of two opposite boundary surfaces, tangential derivatives on the boundary surfaces which are used to compute normal derivatives, and Hermite cubic blending functions.

The construction of some special blending functions aimed at grid clustering at the boundaries of physical geometries was performed by Eriksson (1982), Smith and Eriksson (1987), and Liseikin (1998a, b). A detailed description of various forms of blending functions was presented in monographs by Thompson et al. (1985) and Liseikin (1999).

Differential Methods

For gridding geometries with arbitrary boundaries, differential methods based on the solution of elliptic and parabolic equations are commonly used. Such equations generate smooth grids, allow for full specification of grid nodes on the boundary of a physical geometry, do not propagate boundary singularities into its interior, have less danger of producing cell overlapping, and can be solved efficiently using many well-developed codes. The use of parabolic and elliptic systems enables orthogonal and clustering coordinate lines to be constructed, while, in many cases, the maximum principle, which is typical for these systems, ensures that the intermediate transformations are nondegenerate. Elliptic equations are also used to smooth algebraic or unstructured grids.

Elliptic Equations

Originally, the most popular elliptic equations with differential grid approaches were the generalized Poisson equations formulated with respect to the components $\xi^i(\mathbf{s})$ of the transformation

$$\boldsymbol{\xi}(\boldsymbol{s}) : S^n \to \Xi^n \tag{1.20}$$

that is the inverse of the intermediate transformation (1.13). The equations for generating grids of the physical geometry S^{xn} include coefficients defined by the elements

$$g^{xs}_{ij} = \boldsymbol{x}_{s^i} \cdot \boldsymbol{x}_{s^j} \ , \quad i, j = 1, \dots, n \ ,$$

of the covariant metric tensor of S^{xn} in the parametric coordinates $s^1, \dots, s^n$. A general form of these generalized Poisson equations is as follows:

$$\Delta_B[\xi^i] = P^i(\boldsymbol{s}) \ , \quad i = 1, \dots, n \ . \tag{1.21}$$

Here, $\varDelta_B$ is the operator of Beltrami defined at a function $f(\boldsymbol{s})$ by the formula

$$\varDelta_B[f] = \frac{1}{\sqrt{g^{xs}}} \sum_{j=1}^{n} \frac{\partial}{\partial s^j}\Big(\sqrt{g^{xs}} \sum_{k=1}^{n} g_{sx}^{jk} \frac{\partial f(\mathbf{s})}{\partial s^k}\Big)\ , \tag{1.22}$$

where $g^{xs} = \det\{g_{ij}^{xs}\}$, while g_{sx}^{jk} is the (jk)th element of the contravariant metric tensor of S^{xn} in the parametric coordinates $s^1, \dots, s^n$. The elements g_{sx}^{ij}, $i, j = 1, \dots, n$, comprise a matrix that is the inverse of the matrix formed by the elements g_{ij}^{xs}, $i, j = 1, \dots, n$. The terms $P^i(\boldsymbol{s})$ in (1.21) referred to as source terms or control functions are introduced to control the grid behavior.

A particular form of (1.21) for generating grids in a domain $X^n \subset R^n$, $X^n \equiv S^n$ is given by the Poisson equations

$$\sum_{j=1}^{n} \frac{\partial^2 \xi^i}{\partial x^j \partial x^j} = P^i(\boldsymbol{x})\ , \quad i = 1, \dots, n\ . \tag{1.23}$$

The intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi}) = [s^1(\boldsymbol{\xi}), \dots, s^n(\boldsymbol{\xi})]$ for generating grids on a physical geometry S^{xn} is found from the solution to the Dirichlet boundary value problem for the transformed equations obtained from (1.21) by changing mutually dependent and independent variables. These equations are of the form

$$\sum_{i,j=1}^{n} g_{\xi x}^{ij} \frac{\partial^2 s^k}{\partial \xi^i \partial \xi^j} = \varDelta_B[s^k] - \sum_{i=1}^{n} P^i \frac{\partial s^k}{\partial \xi^i}\ , \quad k = 1, \dots, n\ , \tag{1.24}$$

where $g_{\xi x}^{ij}$ is the (ij)th element of the contravariant metric tensor of S^{xn} in the grid coordinates $\xi^1, \dots, \xi^n$.

Since 1991 a new elliptic approach for controlling grid properties has been developed. By this approach, the task of grid adaptation, instead of source terms $P^i(\mathbf{s})$ in (1.21), is put on monitor metrics in Beltrami equations. Namely, as an elliptic model, the Beltrami equations

$$\varDelta_B[\xi^i] = 0, \quad i = 1, \dots, n\ , \tag{1.25}$$

are used in a monitor metric, where the operator $\varDelta_B$ is of the form (1.22), but the contravariant metric elements are not obliged to be the elements of the physical geometry S^{xn}. The Eq. (1.25) are the Euler-Lagrange equations for the functional of energy. The solutions of these equations are referred to as harmonic transformations.

Hyperbolic Equations

The most well-known hyperbolic equations are the first order partial differential equations of the Cauchy–Riemann type. In practice, two-dimensional hyperbolic equations with respect to the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$ have the following form:

$$A\mathbf{s}_{\xi^1} + B\mathbf{s}_{\xi^2} = \mathbf{f}\ , \tag{1.26}$$

where A and B are some matrices. These equations are simpler then nonlinear elliptic Eq. (1.25) and enable marching methods to be used and an orthogonal system of coordinates to be constructed, while grid adaptation can be performed using the coefficients of the equations. However, methods based on the solution of hyperbolic equations are not always mathematically correct and they are not applicable to regions in which the complete boundary node distribution is strictly specified. Therefore, hyperbolic methods are mainly used for simple regions which have several lateral faces for which no special nodal distribution is required. The marching procedure for the solution of hyperbolic equations allows one to decompose only the boundary geometry in such a way that neighboring boundary grids overlap.

Parabolic Equations

The parabolic grid approach lies between the elliptic and hyperbolic ones.

The two-dimensional parabolic grid generation equation where the marching direction is ξ^2 may be written in the following form:

$$\mathbf{s}_{\xi^2} = A_1\mathbf{s}_{\xi^1\xi^1} - B_1\mathbf{s} + \mathbf{P}\ , \tag{1.27}$$

where A_1 and B_1 are matrix coefficients, and $\mathbf{P}$ is a source vector-valued function that contains the information about the outer boundary configuration. Analogously, the three-dimensional parabolic equations may be written as follows:

$$\mathbf{s}_{\xi^3} = \sum_{i=1}^{2} A_i\mathbf{s}_{\xi^i\xi^i} - B_1\mathbf{s} + \mathbf{P}\ . \tag{1.28}$$

The generation of grids based on a parabolic scheme approximating the inverted Poisson equations was first proposed for two-dimensional grids by Nakamura (1982). A variation of the method of Nakamura was developed by Noack (1985) for use in space-marching solutions to the Euler equations. Extensions of this parabolic technique to generate solution adaptive grids were performed by Edwards (1985) and Noack and Anderson (1990).

Hybrid Grid Generation Scheme

The combination of the hyperbolic and parabolic schemes into a single scheme is attractive because it can use the advantages of both schemes. These advantages are: first, it is a noniterative scheme; second, the orthogonality of the grid near the initial boundary is well controlled; and third, the outer boundary can be prescribed.

A hybrid grid generation scheme in two dimensions for the particular marching direction ξ^2 can be derived by combining hyperbolic and parabolic equations, in particular, as the sum of Eq. (1.26) multiplied by B^{-1} and (1.27) with weights α and $1 - \alpha$, respectively:

$$\alpha(B^{-1}A\mathbf{s}_{\xi^1}+\mathbf{s}_{\xi^2})+(1-\alpha)(\mathbf{s}_{\xi^2}-A_1\mathbf{s}_{\xi^1\xi^1}+B_1\mathbf{s})$$
$$=\alpha B^{-1}\boldsymbol{f}+(1-\alpha)\mathbf{P}\ . \qquad (1.29)$$

The parameter α can be changed as desired to control the proportions of the two methods. If α approaches 1, the system (1.29) becomes the hyperbolic grid system, while if α approaches zero, it becomes the parabolic grid system. In practical applications, α is set to 1, when the grid generation starts from the initial boundary curve $\xi^2=0$, but it gradually decreases and approaches zero when the grid reaches the outer boundary.

An analogous combination can be used to generate three-dimensional grids through a hybrid of parabolic and hyperbolic equations.

A combination of hyperbolic and parabolic schemes that uses the advantages of the two but eliminates the drawbacks of each was proposed by Nakamura and Suzuki (1987).

Variational Methods

Variational methods are widely used to generate grids which are required to satisfy several critical properties, e.g., mesh concentration in areas needing high resolution of the physical solution, mesh alignment to some prescribed vector fields, mesh nondegeneracy, smoothness, uniformity, and near-orthogonality that cannot be realized simultaneously with algebraic or differential techniques. Variational methods take into account the conditions imposed on the grid through the construction of special functionals defined on a set of smooth or discrete transformations. A compromise grid, with properties close to those required, is obtained with the optimum transformation for a combination of these functionals.

The major task of the variational approach to grid generation is to describe all basic measures of the desired grid features in an appropriate functional form and to formulate a combined functional that provides a well-posed minimization problem. These functionals can provide mathematical feedback in an automatic grid procedure.

Commonly, in the calculus of variations, any functional over some admissible set of functions $\boldsymbol{f}: D^n \to R^m$ is defined by the integral

$$I[\boldsymbol{f}]=\int_{D^n} G(\boldsymbol{f})\mathrm{d}V\ , \qquad (1.30)$$

where D^n is a bounded n-dimensional domain, and $G(\boldsymbol{f})$ is some operator specifying, for each vector-valued function $\boldsymbol{f}: D^n \to R^m$, a scalar function $G(\boldsymbol{f}): D^n \to R$. The admissible set is composed of those functions $\boldsymbol{f}$ which satisfy a prescribed boundary condition

$$\boldsymbol{f}\ |_{\partial D^n}=\phi$$

and for which the integral (1.30) is limited.

In the application of the calculus of variations to grid generation, this set of admissible functions is a set of sufficiently smooth invertible coordinate transformations (1.20) between the parametric domain S^n and the computational domain Ξ^n or, vice versa, a set of sufficiently smooth invertible intermediate transformations (1.13) from the computational domain Ξ^n onto the parametric region S^n. The integral (1.30) is defined over the domain S^n or Ξ^n, respectively.

In grid generation applications, the operator G is commonly chosen as a combination of weighted local grid characteristics which are to be optimized. The choice depends, of course, on what is expected from the grid. Some forms of the weight functions and both local and integral grid characteristics were formulated in a monograph of Liseikin (1999) through the transformations (1.13) or (1.20) and their first and second derivatives. Therefore, for the purpose of grid generation, it can be supposed that the most widely acceptable formula for the operator G in (1.30) is one which is derived from some expressions containing the first and second derivatives of the coordinate transformations. Thus, it is generally assumed that the functional (1.30), depending on the transformation $\boldsymbol{\xi}(\boldsymbol{s})$, is of the form

$$I[\boldsymbol{\xi}] = \int_{S^n} G(\mathbf{s}, \boldsymbol{\xi}, \boldsymbol{\xi}_{s^i}, \boldsymbol{\xi}_{s^i s^j})\mathrm{d}\mathbf{s}\ ,$$

where G is a smooth function of its variables.

Analogously, the functional (1.30) formulated over a set of invertible intermediate transformations $\mathbf{s}(\boldsymbol{\xi})$ has the form

$$I[\mathbf{s}] = \int_{\Xi^n} G_1(\boldsymbol{\xi}, \mathbf{s}, \mathbf{s}_{\xi^i}, \mathbf{s}_{\xi^i \xi^j})\mathrm{d}\boldsymbol{\xi}\ .$$

In one popular approach, the functional formulated with respect to the intermediate mapping $\mathbf{s}(\boldsymbol{\xi})$ has the following form:

$$I[\mathbf{s}] = \int_{\Xi^n} (\sqrt{g^{m\xi}} \sum_{i,j=1}^{n} g^{ij}_{\xi m} g^{s\xi}_{ij})\mathrm{d}\boldsymbol{\xi}\ , \tag{1.31}$$

where $g^{ij}_{\xi m}$, $i, j = 1, \dots, n$, are the elements of the contravariant tensor in the logical coordinates $\xi^1, \dots, \xi^n$ of a monitor metric $g^{m\xi}_{ij}$ imposed on Ξ^n, $g^{m\xi} = \det\{g^{m\xi}_{ij}\}$, while $g^{s\xi}_{ij}$ is the covariant Euclidean metric tensor of S^n in the coordinates $\xi^1, \dots, \xi^n$. This functional was proposed for $n = 2$ by Godunov and Prokopov (1967) for generating conformal and quasi-conformal grids in S^2. In their consideration, the elements $g^{m\xi}_{ij}$, $i, j = 1, 2$, of the monitor metric should be dependent on $\boldsymbol{\xi}$ and some, in general vector-valued parameter $\mathbf{r}$. Belinsky et al. (1975) and Godunov et al. (1990) discussed the same two-dimensional functional of the form (1.31) with the following monitor metric introduced in Ξ^2:

$$g^{m\xi}_{ij} = \begin{pmatrix} e^{2p(\boldsymbol{\xi})} & e^{p(\boldsymbol{\xi})+q(\boldsymbol{\xi})}\cos[\alpha(\boldsymbol{\xi}) - \beta(\boldsymbol{\xi})] \\ e^{p(\boldsymbol{\xi})+q(\boldsymbol{\xi})}\cos[\alpha(\boldsymbol{\xi}) - \beta(\boldsymbol{\xi})] & e^{2q(\boldsymbol{\xi})} \end{pmatrix},$$

where the functions $p(\boldsymbol{\xi})$, $q(\boldsymbol{\xi})$, $\alpha(\boldsymbol{\xi})$, and $\beta(\boldsymbol{\xi})$ are subject to the restrictions

$$p(\boldsymbol{\xi}) - q(\boldsymbol{\xi}) = \ln \sqrt{g_{11}^{s\xi}/g_{22}^{s\xi}} \; ,$$

$$\alpha(\boldsymbol{\xi}) - \beta(\boldsymbol{\xi}) = \arccos\Bigl(g_{12}^{s\xi}/\sqrt{g_{11}^{s\xi} g_{22}^{s\xi}}\Bigr) \; .$$

The grid approach based on the minimization of the functional (1.31) for $n = 2$ was also used by Chumakov and Chumakov (1998) for generating quasi-isometric grids by introducing in Ξ^2 a monitor metric borrowed from the metric of a surface of a constant Gauss curvature.

Note the functional (1.31) is twice the energy functional of the function $\mathbf{s}(\boldsymbol{\xi}) : \Xi^n \to S^n$ where Ξ^n is endowed by the monitor metric $g_{ij}^{m\xi}$, while S^n has the Euclidean metric.

The twice energy functional of the function $\boldsymbol{\xi}(\mathbf{s}) : S^n \to \Xi^n$ between S^n with an imposed monitor metric $g_{ij}^{\mathbf{s}}$ and Ξ^n with the Euclidean metric for generating adaptive grids was considered by Dvinsky (1991) and Liseikin (1991a). This functional has the following form:

$$I[\boldsymbol{\xi}] = \int_{S^n} \Bigl(\sqrt{g^{\mathbf{s}}} \sum_{i,j,k=1}^{n} g_{\mathbf{s}}^{ij} \frac{\partial \xi^k}{\partial s^i} \frac{\partial \xi^k}{\partial s^j}\Bigr) \mathrm{d}\mathbf{s} \; , \tag{1.32}$$

where $g^{\mathbf{s}} = \det\{g_{ij}^{\mathbf{s}}\}$, $g_{\mathbf{s}}^{ij}, i, j = 1, \ldots, n$, are the elements of the contravariant monitor metric tensor introduced in S^n. The Euler-Lagrange equations for the functional (1.32) are equivalent to the Beltrami equations (1.24).

The functionals are used to control and realize various grid properties. This is carried out by combining these functionals with weights in the form

$$I = \sum_i \lambda_i I_i \; , \qquad i = 1, \cdots, k \; . \tag{1.33}$$

Here, $\lambda_i, \ i = 1, \cdots, k$, are specified parameters which determine the individual contribution of each functional I_i to I. The ranges of the parameters λ_i controlling the relative contributions of the functionals can be defined readily when the functionals I_i are dimensionally homogeneous. However, if they are dimensionally inhomogeneous, then the selection of a suitable value for λ_i presents some difficulties. A common rule for selecting the parameters λ_i involves making each component $\lambda_i I_i$ in (1.33) of a similar scale by using a dimensional analysis.

The most common practice in forming the combination (1.33) uses both the functionals of adaptation to the physical solution and the functionals of grid regularization. The first reason for using such a strategy is connected to the fact that the process of adaptation can excessively distort the form of the grid cells. The distortion can be prevented by functionals which impede cell deformation. These functionals are ones which control grid skewness, smoothness, and conformality. The second reason for using the regularization functionals is connected to the natural requirement for the well-posedness of the grid generation process. This requirement is achieved through the utilization of convex functionals in variational grid generators. The convex functionals are represented by energy-type functionals (1.32) producing harmonic maps and by the functionals of conformality.

The various functionals provide broad opportunities to control and realize the required grid properties, though problems still remain; these require more detailed studies of all properties of the functionals. The knowledge of these properties will allow one to utilize the functionals as efficient tools to generate high-quality grids.

1.3.2 Methods for Unstructured Grids

Unstructured grids can be obtained with cells of arbitrary shape, but are generally composed of tetrahedrons (triangles in two dimensions). There are, fundamentally, three approaches to the generation of unstructured grids: quadtree-octree based methods, Delaunay procedures, and advancing-front techniques.

Octree Approach

In the octree approach, the region is first covered by a regular Cartesian grid of cubic cells (squares in two dimensions). Then, the cubes containing segments of the domain surface are recursively subdivided into eight cubes (four squares in two dimensions) until the desired resolution is reached. The cells intersecting the body surfaces are formed into irregular polygonal cells. The grid generated by this octree approach is not considered to be the final one, but serves to simplify the geometry of the final grid, which is commonly composed of tetrahedral (or triangular) cells built from the polygonal cells and the remaining cubes.

The main drawback of the octree approach is the inability to match a prescribed boundary surface grid, so the grid on the surface is not constructed beforehand as desired but rather is derived from the irregular volume cells that intersect the surface. Another drawback of the grid is its rapid variation in cell size near the boundary. In addition, since each surface cell is generated by the intersection of a hexahedron with the boundary, there arise problems in controlling the variation of the surface cell size and shape.

Delaunay Approach

The Delaunay approach connects neighboring points (of some previously specified set of nodes in the region) to form tetrahedral cells in such a way that the circumsphere through the four vertices of a tetrahedral cell does not contain any other point. The points can be generated in two ways: they can be defined at the start by some technique or they can be inserted within the tetrahedra as they are created, starting with very coarse elements connecting boundary points and continuing until the element size criteria are satisfied. In the latter case, a new Delaunay triangulation is constructed at every step, usually using Watson's and Rebay's incremental algorithms.

The major drawback of the Delaunay approach is that it requires the insertion of additional boundary nodes, since the boundary cells may not become the boundary segments of the Delaunay volume cells. Either the Delaunay criterion must be mitigated near the boundaries or boundary points must be added as necessary to avert breakthrough of the boundary.

Advancing-Front Techniques

In these techniques, the grid is generated by building cells progressively one at a time and marching from the boundary into the volume by successively connecting new points to points on the front until all previously unmeshed space is filled with grid cells. Some provision must be made to keep the marching front from intersecting.

Finding suitable vertices for the new cells is a very difficult task in this approach, since significant searches must be made to adjust the new cells to the existing elements. Commonly, the marching directions for the advancing front must take into account the surface normals as well as the adjacent surface points. A particular difficulty of this method occurs in the closing stage of the procedure, when the front folds over itself and the final vertices of the empty space are replaced by tetrahedra. Serious attention must also be paid to the marching step size, depending on the size of the front faces as well as the shape of the unfilled domain that is left.

Unstructured grids, after they have been completed, are generally smoothed by a Laplacian-type or other smoother to enhance their qualitative properties.

A major drawback remaining for unstructured techniques is the increased computational cost of the numerical solution of partial differential equations in comparison with a solution on structured grids.

1.4 Big Codes

A “big grid generation code” is an effective system for generating structured and unstructured grids, as well as hybrid and and overset combinations, in general regions. Such systems are also are referred to as “comprehensive grid generation codes”.

The development of such codes is a considerable problem in its own right. The present comprehensive grid generation codes developed for the solution of multidimensional problems have to incorporate combinations of block-structured, hybrid, and overset grid methods, and they are still rather cumbersome, rely on interactive tools, and take too many man-hours to generate a complicated grid. Efforts to increase the efficiency and productivity of these codes are mainly being conducted in two interconnected research areas.

The first, the “array area”, is concerned with the automation of those routine processes of grid generation which presently require interactive tools and a great deal of human time and effort. Some of these are:

(1) the decomposition of a domain into a set of contiguous or ovelapping blocks consistent with the distinctive features of the domain geometry, the singularities of the physical medium and the sought-for solution, and the computer architecture;

(2) numbering the set of blocks, their faces, and their edges with a connectivity hierarchy and determining the order in which the grids are constructed in the blocks and their boundaries;
(3) choosing the grid topology and the requirements placed on the qualitative and quantitative characteristics of the internal and boundary grids and on their communication between the blocks;
(4) selecting appropriate methods to satisfy the requirements put on the grid in accordance with a particular geometry and solution;
(5) assessment and enhancement of grid quality.

The second, more traditional, "methods area" deals with developing new, more reliable, and more elaborate methods for generating, adapting, and smoothing grids in domains in a unified manner, irrespective of the geometry of the domain or surface and of the qualitative and quantitative characteristics the grids should possess, so that these methods, when incorporated into the comprehensive codes, should ease the bottlenecks of the array area, in particular, by enabling a considerable reduction of the number of blocks required.

There are many demands that are made on the codes. The code must be efficient, expandable, portable, and configurable. It should incorporate state-of-the-art techniques for generating grids. Besides this, the code should include pre- and post-processing tools in order to start from prescribed data of the geometry and end with the final generation of the grid in the proper format for use with the specified partial differential codes. The code should have the ability to be updated through the addition of new features and the removal of obsolete ones.

The overall purpose of the development of these comprehensive grid generation codes is to create a system which enables one to generate grids in a "black box" mode without or with only a slight human interaction. Currently, however, the user has to take an active role and be fully occupied in the grid generation process. The user has to make conclusions about qualitative properties of the grid and undertake corrective measures when necessary. The present codes include significant measures to increase the productivity of such human activity, namely, graphical interactive systems and user-friendly interfaces. Efforts to eliminate the "human component" of the codes are directed towards developing new techniques, in particular, new grid generation methods and automated block decomposition techniques.

1.4.1 Interactive Systems

An important element of the current comprehensive grid generation codes is an interactive system which includes extensive graphical tools that display all elements of the grid generation process and graphical feedback to monitor progress in grid efficiency and to verify, as well as to correct, errors and faults easily. All existing comprehensive codes possess well-developed interactive systems which are used, in particular, to define grid boundaries and surface normals on block faces; to generate multiblock topologies and domain decompositions; to specify connectivity data, grid density, and spatial distribution in the normal direction at the boundaries; and to provide attraction to chosen points or lines. The graphical systems of the codes provide a display of data and domain and surface elements with different colors and markers; a representation of surface grids and their boundaries by specific colors; a visual representation of the qualitative and quantitative properties of the grid in terms of cell skewness, aspect ratio, surface and volume Jacobian checks, estimates of truncation errors, and measures of grid continuity across blocks; and views of surface and block grids at various levels of coarseness.

These capabilities allow for any portion of a multiple-block grid to be displayed in a manner that is quickly discernible to the user. All functions of the generation process are invoked through interactive screens and menus. In an interactive environment, the user can continually examine and correct the surfaces and grids as they are developed.

1.4.2 New Techniques

The present codes are designed in a modular fashion to facilitate both the addition of new techniques in a straightforward manner as they become available and the removal of obsolete ones. Although most of the methods included in the codes have provided appropriate results for specific applications, they lack the desired generality, flexibility, efficiency, automation, and robustness. Efforts to create new techniques are directed towards the automation of domain decomposition, interactive and automated generation of block connectivity, the development of new, more effective methods for generating grids within the blocks, interactive local adaptive adjustment of the control functions in elliptic equations, the optimal specification and modification of the distribution functions in hyperbolic and advancing-front grid generators, interactive local quality enhancement of the grid, and interactive and generally applicable interpolation techniques to transfer data between the separate component grids.

Domain Decomposition

To perform domain zoning well, some expertise is required: the user must have experience with composite zonal grid methods, familiarity with the grid generation capabilities available, knowledge of the behavior of the zonal technique to be used, knowledge of the physical behavior, some expectation of the important physical features of the problem to be solved, and criteria for evaluating the zonings. To perform zoning quickly, the user must have both expertise and interactive, graphical, easy-to-use tools.

However, even with the interactive techniques available, the generation of the block structure is the most difficult and time-consuming task in the grid generation process. Therefore, any automation of domain decomposition is greatly desirable.

The first attempts to overcome the problem of domain decomposition were presented in the 1980s. The proposed approaches laid a foundation for an automated approach to 3-D domain decomposition which relies mainly on observation of how experts perform the task and on a knowledge-based programming approach, typically described by means of examples. The user first represents all components of the domain schematically as rectangular sets of blocks and then the codes develop a schematic block structure.

New Methods

Recent results in the field of grid generation methods have largely been related to the application of harmonic function theory to adaptive grid generation. The suggestion to use harmonic functions for generating adaptive grids was made by Dvinsky (1991). Adaptive grids can be generated by mapping the reference grid into the domain with a coordinate transformation which is inverse to a harmonic vector function (in terms of Riemannian manifolds). Adaptation is performed through a specified adaptive metric in the domain which converts it into a Riemannian manifold. Each harmonic function minimizes some functional of the total energy, and hence it can be found by the numerical solution either of a variational problem or of a boundary value problem for a system of Euler–Lagrange equations.

One version of the harmonic approach, proposed by Liseikin (1991a, 2004), uses a method of generating smooth hypersurface grids. Specifically, the adaptive grid with node clustering in the zones of large values of a vector function is obtained as a projection of a quasiuniform grid from a monitor surface generated as a surface of the function values over the physical space. The vector function can be the physical solution or a combination of its components or derivatives, or it can be any other quantity that suitably monitors the behavior of the solution. A generalization of this approach for generating grids with other properties was performed by Liseikin (2004, 2007). The method developed allows the designer to merge the two tasks of surface grid generation and volume grid generation into one task while developing a comprehensive grid generation code. It also eases the array bottlenecks of the codes by allowing a decrease in the number of blocks required for the decomposition of a complicated region.

1.5 Comments

Detailed descriptions of the most popular structured methods and their theoretical and logical justifications and numerical implementations were given in the monographs by Thompson et al. (1985), Knupp and Steinberg (1993), and Liseikin (1999, 2010). Particular issues concerned with the generation of one-dimensional moving grids for gas-dynamics problems, the stretching technique for the numerical solution of singularly perturbed equations, and nonstationary grid techniques were considered in the books by Alalykin et al. (1970), Liseikin and Petrenko (1989), Liseikin (2001), and Zegeling (1993), respectively.

A considerable number of general structured grid generation methods were reviewed in surveys by Thompson et al. (1982), Thompson (1984a, 1996), Eiseman

(1985), Liseikin (1991b), Thompson and Weatherill (1993), and in the handbook on grid generation edited by Thompson et al. (1999).

Adaptive structured grid methods were first surveyed by Anderson (1983) and Thompson (1984b, 1985). Then, a series of surveys on general adaptive methods was presented by Eiseman (1987), Hawken et al. (1991), Liseikin (1996), and Baker (1997). Adaptive techniques for moving grids were described by Hedstrom and Rodrigue (1982) and Zegeling (1993).

A description of the types of mesh topology and the singular points of the grids around wing-body shapes was carried out by Eriksson (1982).

Methods for unstructured grids were reviewed by Thacker (1980), Ho-Le, (1988), Shephard et al. (1988), Baker (1995, 1997), Field (1995), Carey (1997), George and Borouchaki (1998), Krugljakova et al. (1998), Owen (1998), Frey and George (2008), and Lo (2015). An exhaustive survey of both structured and unstructured techniques has been given by Thompson and Weatherill (1993), and Thompson (1996).

The multiblock strategy for generating grids around complicated shapes was originally proposed by Lee et al. (1980); however, the idea of using different coordinates in different subregions of the domain can be traced back to Thoman and Szewczyk (1969). The overset grid approach was introduced by Atta and Vadyak (1982), Berger and Oliger (1983), Benek et al. (1983), Miki and Takagi (1984), and Benek et al. (1985). The first attempts to overcome the problem of domain decomposition were discussed by Andrews (1988), Georgala and Shaw (1989), Allwright (1989), and Vogel (1990). The concept of blocks with a continuous alignment of grid lines across adjacent block boundaries was described by Weatherill and Forsey (1984) and Thompson (1987b). Thompson (1982) and Eriksson (1983) applied the concept of continuous line slope, while a discontinuity in slope was discussed by Rubbert and Lee (1982). A shape recognition technique based on an analysis of a physical domain and an interactive construction of a computational domain with a similar geometry was proposed by Takahashi and Shimizu (1991) and extended by Chiba et al. (1998). The embedding technique was considered by Albone and Joyce (1990) and Albone (1992). Some of the first applications of block-structured grids to the numerical solution of three-dimensional fluid-flow problems in realistic configurations were demonstrated by Rizk and Ben-Shmuel (1985), Sorenson (1986), Atta et al. (1987), and Belk and Whitefield (1987).

The first comprehensive grid codes were described by Holcomb (1987), Thompson (1987a), Thomas et al. (1990), Widhopf et al. (1990), and Steinbrenner et al. (1990). These codes have stimulated the development of better ones, reviewed by Thompson (1996). This paper also describes the current domain decomposition techniques developed by Shaw and Weatherill (1992), Stewart (1992), Dannenhoffer (1995), Wulf and Akrag (1995), Schonfeld et al. (1995), and Kim and Eberhardt (1995).

Since the time that these pioneering codes appeared, many codes more advanced and sophisticated have been developed by researchers. Exhaustive surveys of the most popular recent codes can be found at the meshing pages http://www.andrew.cmu.edu/user/sowen/softsurv.html and http://www-users.informatik.rwth-aachen.de/~roberts/software.html. In particular, very useful descriptions of mesh generation codes and codes with parallel mesh generation techniques are given in the books of Frey and George (2008), Ivanov (2008), and Lo (2015). An informal survey of software vendors, research labs, and educational institutions that develop grid generation codes was presented by Owen (1998).

References

Alalykin, G. B., Godunov, S. K., Kireyeva, L. L., & Pliner, L. A. (1970). *On solution of one-dimensional problems of gas dynamics in moving grids*. Moscow: Nauka. (Russian).

Albone, C. M. (1992). *Embedded meshes of controllable quality synthesised from elementary geometric features*. AIAA Paper 92-0633.

Albone, C. M., & Joyce, M. G. (1990). Feature-associated mesh embedding for complex configurations. In *AGARD Conference Proceedings 464.13*.

Allwright, S. (1989). Multiblock topology specification and grid generation for complete aircraft configurations. In W. Schmidt (Ed.), *AGARD Conference Proceedings 464, Applications of Mesh Generation to Complex 3-D Configurations, Loen, Norway*. Advisory Group for Aerospace Research and Development, NATO.

Anderson, D. A. (1983). Adaptive grid methods for partial differential equations. In K. N. Ghia & U. Ghia (Eds.), *Advances in grid generation* (pp. 1–15). Houston: ASME.

Andrews, A. E. (1988). Progress and challenges in the application of artificial intelligence to computational fluid dynamics. *AIAA Journal*, *26*, 40–46.

Atta, E. H., & Vadyak, J. (1982). *A grid interfacing zonal algorithm for three-dimensional transonic flows about aircraft configurations*. AIAA Paper 82-1017.

Atta, E. H., Birchelbaw, L., & Hall, K. A. (1987). *A zonal grid generation method for complex configurations*. AIAA Paper 87-0276.

Baker, T. J. (1995). Prospects and expectations for unstructured methods. In *Proceedings of the Surface Modeling, Grid Generation and Related Issues in Computational Fluid Dynamics Workshop. NASA Conference Publication 3291* (pp. 273–287). Cleveland, OH: NASA Lewis Research Center.

Baker, T. J. (1997). Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements in Analysis and Design*, *25*, 243–273.

Belinsky, P. P., Godunov, S. K., Ivanov, Yu V, & Yanenko, I. K. (1975). The use of one class of quasiconformal mappings to generate numerical grids in regions with curvilinear boundaries. *Zh. Vychisl. Maths. Math. Fiz.*, *15*, 1499–1511. (Russian).

Belk, D. M., & Whitefield, D. L. (1987). *Three-dimensional Euler solutions on blocked grids using an implicit two-pass algorithms*. AIAA Paper 87-0450.

Benek, J. A., Steger, J. L., & Dougherty, F. C. (1983). *A flexible grid embedding technique with application to the Euler equations*. AIAA Paper 83-1944.

Benek, J. A., Buning, P. G., & Steger, J. L. (1985). *A 3-D chimera grid embedding technique*. AIAA Paper 85-1523.

Berger, M. J., & Oliger, J. (1983, March). *Adaptive mesh refinement for hyperbolic partial differential equations*. Manuscript NA-83-02, Stanford University.

Carey, G. F. (1997). *Computational grids: Generation, adaptation, and solution strategies*. London: Taylor and Francis.

Chiba, N., Nishigaki, I., Yamashita, Y., Takizawa, C., & Fujishiro, K. (1998). A flexible automatic hexahedral mesh generation by boundary-fit method. *Computer Methods in Applied Mechanics and Engineering*, *161*, 145–154.

Chumakov, G. A., & Chumakov, S. G. (1998). A method for the 2-D quasi-isometric regular grid generation. *Journal of Computational Physics*, *133*, 1–28.

Dannenhoffer, J. F. (1995). Automatic blocking for complex three-dimensional configurations. In *Proceedings of the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics Workshop* (p. 123). Cleveland, OH: NASA Lewis Research Center.

Dvinsky, A. S. (1991). Adaptive grid generation from harmonic maps on Riemannian manifolds. *Journal of Computational Physics*, *95*, 450–476.

Edwards, T. A. (1985). *Noniterative three-dimensional grid generation using parabolic partial differential equations*. AIAA Paper 85-0485.

Eiseman, P. R. (1980). *Geometric methods in computational fluid dynamics*. ICASE Report 80-11 and Von Karman Institute for Fluid Dynamics Lecture Series Notes.

Eiseman, P. R. (1985). Grid generation for fluid mechanics computations. *Annual Review of Fluid Mechanics*, *17*, 487–522.

Eiseman, P. R. (1987). Adaptive grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 321–376.

Eriksson, L. E. (1982). Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation. *AIAA Journal*, *20*, 1313–1320.

Eriksson, L. E. (1983). *Practical three-dimensional mesh generation using transfinite interpolation*. Lecture Series Notes 1983-04. Brussels: von Karman Institute for Fluid Dynamics.

Field, D. A. (1995). The legacy of automatic mesh generation from solid modeling. *Computer Aided Geometric Design*, *12*, 651–673.

Frey, P. J., & George, P.-L. (2008). *Mesh generation application to finite elements*. Verlag: ISTE Ltd and Wiley.

Georgala, J. M. & Shaw, J. A. (1989). A discussion on issues relating to multiblock grid generation. In W. Schmidt (Ed.), *AGARD Conference Proceedings 464, Applications of Mesh Generation to Complex 3-D Configurations, Loen, Norway*. Advisory Group for Aerospace Research and Development, NATO.

George, P. L., & Borouchaki, H. (1998). *Delaunay triangulation and meshing: Application to finite elements*. Paris: Editions Hermes.

Godunov, S. K., & Prokopov, G. P. (1967). Calculation of conformal mappings in the construction of numerical grids. *Journal of Computational Mathematics and Mathematical Physics*, *7*, 1031–1059. (Russian).

Godunov, S. K., Romenskii, E. I., & Chumakov, G. A. (1990). Grid generation in complex domains by means of quasi-conformal mappings. In *Proc. Institute of Mathematics* (Vol. 18, pp. 75–84). Novosibirsk: Nauka (Russian).

Gordon, W. J., & Hall, C. A. (1973). Construction of curvilinear coordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, *7*, 461–477.

Gordon, W. J., & Thiel, L. C. (1982). Transfinite mappings and their application to grid generation. In J. F. Thompson (Ed.), *Numerical grid generation* (pp. 171–192). New York: North-Holland.

Hawken, D. F., Gottlieb, J. J., & Hansen, J. S. (1991). Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. *Journal of Computational Physics*, *95*, 254–302.

Hedstrom, G. W., & Rodrigue, C. M. (1982). Adaptive-grid methods for time-dependent partial differential equations, Lecture notes in mathematics (Vol. 960, pp. 474–484). Berlin: Springer.

Holcomb, J. F. (1987). *Development of a grid generator to support 3-dimensional multizone Navier–Stokes analysis*. AIAA Paper 87-0203.

Ho-Le. (1988). Finite element mesh generation methods: A review and classification. *Computer-Aided Design*, *20*, 27–38.

Ivanov, E. (2008). *Parallel tetrahedral meshing based on a-priori domain decomposition: From scratch to results by utilizing off-the-shelf sequential software*. Saarbrucken: VDM Verlag Dr. Muller.

Kim, B., & Eberhardt, S. D. (1995, May). Automatic multiblock grid generation for high-lift configuration wings. In *Proceedings of the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics Workshop* (p. 671). Cleveland, OH: NASA, Lewis Research Center.

Knupp, P. M., & Steinberg, S. (1993). *Fundamentals of grid generation*. Boca Raton: CRC Press.

Krugljakova, L. V., Neledova, A. V., Tishkin, V. F., & Filatov, A. Yu. (1998). Unstructured adaptive grids for problems of mathematical physics (survey). *Mathematical Modelling*, *10*(3), 93–116. (Russian).

Lee, K. D., Huang, M., Yu, N. J., & Rubbert, P. E. (1980). Grid generation for general three-dimensional confugurations. In R. E. Smith (Ed.), *Proceedings of the NASA Langley Workshop on Numerical Grid Generation Techniques* (p. 355).

Liseikin, V. D. (1991a). On generation of regular grids on n-dimensional surfaces. *Journal of Computational Mathematics and Mathematical Physics*, *31*, 1670–1689. (Russian). [English transl.: USSR Comput. Math. Math. Phys. 31(11) (1991), 47–57].

Liseikin, V. D. (1991b). Techniques for generating three-dimensional grids in aerodynamics (review). *Problems Atomic Sci. Technology Ser. Math. Model. Phys. Process*, *3*, 31–45. (Russian).

Liseikin, V. D. (1996). Construction of structured adaptive grids - a review. *Computational Mathematics and Mathematical Physics*, *36*(1), 1–32.

Liseikin, V. D. (1998a). Algebraic adaptation based on stretching functions. *Russian Journal of Numerical Analysis and Mathematical Modelling*, *13*(4), 307–324.

Liseikin, V. D. (1998b). A method of algebraic adaptation. *Computational Mathematics and Mathematical Physics*, *38*(10), 1624–1640.

Liseikin, V. D. (1999). *Grid generation methods*. Berlin: Springer.

Liseikin, V. D. (2001). *Layer resolving grids and transformations for singular perturbation problems*. Utrecht: VSP.

Liseikin, V. D. (2004). *A computational differential geometry approach to grid generation*. Berlin: Springer.

Liseikin, V. D. (2007). *A computational differential geometry approach to grid generation* (2nd ed.). Berlin: Springer.

Liseikin, V. D. (2010). *Grid generation methods* (2nd ed.). Berlin: Springer.

Liseikin, V. D., & Petrenko, V. E. (1989). *Adaptive Invariant Method for the Numerical Solution of Problems with Boundary and Interior Layers*. Novosibirsk (Russian): Computer Center SD AS USSR.

Lo, S. H. (2015). *Finite element mesh generation*. Boca Raton: CRC Press Taylor and Francis Group.

Miki, K., & Takagi, T. (1984). A domain decomposition and overlapping method for the generation of three-dimensional boundary-fitted coordinate systems. *Journal of Computational Physics*, *53*, 319–330.

Nakamura, S. (1982). Marching grid generation using parabolic partial differential equations. *Applied Mathematics and Computation*, *10*(11), 775–786.

Nakamura, S., & Suzuki, M. (1987). *Noniterative three-dimensional grid generation using a parabolic-hyperbolic hybrid scheme*. AIAA Paper 87-0277.

Noack, R. W. (1985). *Inviscid flow field analysis of maneuvering hypersonic vehicles using the SCM formulation and parabolic grid generation*. AIAA Paper 85-1682.

Noack, R. W., & Anderson, D. A. (1990). Solution adaptive grid generation using parabolic partial differential equations. *AIAA Journal*, *28*(6), 1016–1023.

Owen, S. (1998). A survey of unstructured mesh generation technology. In *7th International Roundtable*

Rizk, Y. M., & Ben-Shmuel, S. (1985). *Computation of the viscous flow around the shuttle orbiter at low supersonic speeds*. AIAA Paper 85-0168.

Rubbert, P. E., & Lee, K. D. (1982). Patched coordinate systems. In J. F. Thompson (Ed.), *Numerical grid generation* (p. 235). New York: North-Holland.

Schonfeld, T., Weinerfelt, P., & Jenssen, C. B. (1995, May). Algorithms for the automatic generation of 2D structured multiblock grids. In *Proceedings of the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics Workshop* (p. 561). Cleveland, OH: NASA, Lewis Research Center.

Shaw, J. A., & Weatherill, N. P. (1992). Automatic topology generation for multiblock grids. *Applied Mathematics and Computation*, *52*, 355–388.

Shephard, M. S., Grice, K. R., Lot, J. A., & Schroeder, W. J. (1988). Trends in automatic three-dimensional mesh generation. *Computers & Structures*, *30*(1/2), 421–429.

Smith, R. E. (1981). *Two-boundary grid generation for the solution of the three-dimensional Navier–Stokes equations*. NASA TM-83123.

Smith, R. E. (1982). Algebraic grid generation. In J. F. Thompson (Ed.), *Numerical grid generation* (pp. 137–170). New York: North-Holland.

Smith, R. E., & Eriksson, L. E. (1987). Algebraic grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 285–300.

Sorenson, R. L. (1986). *Three-dimensional elliptic grid generation about fighter aircraft for zonal finite-difference computations*. AIAA Paper 86-0429.

Steinbrenner, J. P., Chawner, J. R., & Fouts, C. L. (1990). *Multiple block grid generation in the interactive environment*. AIAA Paper 90-1602.

Stewart, M. E. M. (1992). Domain decomposition algorithm applied to multielement airfoil grids. *AIAA Journal*, *30*(6), 1457.

Takahashi, H., & Shimizu, H. (1991). A general purpose automatic mesh generation using shape recognition technique. *Computers in Engineering ASME*, *1*, 519–526.

Thacker, W. C. (1980). A brief review of techniques for generating irregular computational grids. *International Journal for Numerical Methods in Engineering*, *15*(9), 1335–1341.

Thoman, D. C., & Szewczyk, A. A. (1969). Time-dependent viscous flow over a circular cylinder. *Physics of Fluids*, *II*, 76.

Thomas, M. E., Bache, G. E., & Blumenthal, R. F. (1990). *Structured grid generation with PATRAN*. AIAA Paper 90-2244.

Thompson, J. F. (1982). General curvilinear coordinate system. *Applied Mathematics and Computation*, *10*(11), 1–30.

Thompson, J. F. (1984a). Grid generation techniques in computational fluid dynamics. *AIAA Journal*, *22*(11), 1505–1523.

Thompson, J. F. (1984b). *A survey of dynamically-adaptive grids in the numerical solution of partial differential equations*. AIAA Paper 84-1606.

Thompson, J. F. (1985). A survey of dynamically-adaptive grids in the numerical solution of partial differential equations. *Applied Numerical Mathematics*, *1*, 3–27.

Thompson, J. F. (1987a). *A composite grid generation code for general 3-d regions*. AIAA Paper 87-0275.

Thompson, J. F. (1987b). A general three-dimensional elliptic grid generation system on a composite block structure. *Computer Methods in Applied Mechanics and Engineering*, *64*, 377–411.

Thompson, J. F. (1996). A reflection on grid generation in the 90s: Trends, needs influences. In B. K. Soni, J. F. Thompson, J. Hauser, & P. R. Eiseman (Eds.), *Numerical grid generation in CFD* (Vol. 1, pp. 1029–1110). Mississippi State University.

Thompson, J. F., & Weatherill, N. P. (1993). *Aspects of numerical grid generation: Current science and art*. AIAA Paper 93-3539.

Thompson, J. F., Warsi, Z. U. A., & Mastin, C. W. (1982). Boundary-fitted coordinate systems for numerical solution of partial differential equations - a review. *Journal of Computational Physics*, *47*, 1–108.

Thompson, J. F., Warsi, Z. U. A., & Mastin, C. W. (1985). *Numerical grid generation: Foundations and applications*. New York: North-Holland.

Thompson, J. F., Soni, B. K., & Weatherill, N. P. (Eds.). (1999). *Handbook of grid generation*. Boca Raton: CRC Press.

Vogel, A. A. (1990). Automated domain decomposition for computational fluid dynamics. *Computers and Fluids*, *18*(4), 329–346.

Weatherill, N. P., & Forsey, C. R. (1984). *Grid generation and flow calculations for complex aircraft geometries using a multi-block scheme*. AIAA Paper 84-1665.

Widhopf, G. D., Boyd, C. N., Shiba, J. K., Than, P. T., Oliphant, P. H., Huang, S. -C., Swedberg, G. D., & Visich, M. (1990). *RAMBO-4G: An interactive general multiblock grid generation and graphics package for complex multibody CFD applications*. AIAA Paper 90-0328.

Wulf, A., Akrag, V. (1995, May). Tuned grid generation with ICEM CFD. In *Proceedings of the Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamics Workshop* (p. 477). Cleveland, OH: NASA, Lewis Research Center.

Zegeling, P. A. (1993). *Moving-grid methods for time-dependent partial differential equations*. CWI Tract 94, Amsterdam: Centrum voor Wiskund en Informatica.

Chapter 2
Coordinate Transformations

2.1 Introduction

Partial differential equations in the physical domain X^n can be solved on a numerical grid obtained by mapping a reference grid in the logical region Ξ^n into X^n with a coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n$. The mapping approach also gives an alternative way to obtain a numerical solution to a partial differential equation, by solving the transformed equation with respect to the new independent variables ξ^i on the reference grid in the logical domain Ξ^n. Some notions and relations concerning the coordinate transformations yielding grids are discussed in this chapter. These notions and relations are used to represent some conservation-law equations in the new logical coordinates in a convenient form. The relations presented will be used in Chap. 3 to formulate various grid properties.

Conservation-law equations in curvilinear coordinates are typically deduced from the equations in Cartesian coordinates through the classical formulas of tensor calculus, through procedures which include the substitution of tensor derivatives for ordinary derivatives. The formulation and evaluation of the tensor derivatives is rather difficult, and they retain some elements of mystery. However, these derivatives are based on specific transformations of tensors, modeling in the equations some dependent variables, e.g. the components of a fluid velocity vector, which after the transformation have a clear interpretation in terms of the contravariant components of the vector. With this concept, the conservation-law equations are readily written out in this chapter without application to the tensor derivatives, but utilizing instead only some specific transformations of the dependent variables, ordinary derivatives, and one basic identity of coordinate transformations derived from the formula for differentiation of the Jacobian.

For generality, the transformations of the coordinates are mainly considered for arbitrary n-dimensional domains, though in practical applications, the dimension n equals $1, 2, 3$, or 4 for time-dependent transformations of three-dimensional domains. We also apply chiefly a standard vector notation for the coordinates, as variables with indices. Sometimes, however, particularly in figures, the ordinary designation

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_2

for three-dimensional coordinates, namely x, y, z for the physical coordinates and ξ, η, ζ for the logical ones, is used to simplify the presentation.

2.2 General Notions and Relations

This section presents some basic relations between Cartesian and curvilinear coordinates.

2.2.1 Jacobi Matrix

Let

$$\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n , \qquad \boldsymbol{\xi} = (\xi^1, \dots, \xi^n) , \qquad \boldsymbol{x} = (x^1, \dots, x^n) , \tag{2.1}$$

be a smooth invertible coordinate transformation of the physical region $X^n \subset R^n$ from the parametric domain $\Xi^n \subset R^n$. If Ξ^n is a standard logical domain, then, in accordance with Chap. 1, this coordinate transformation can be used to generate a structured grid in X^n. Here and later, R^n presents the Euclidean space with the Cartesian basis $\boldsymbol{e}_1, \dots, \boldsymbol{e}_n$, which represents an orthogonal system of vectors, i.e.

$$\boldsymbol{e}_i \cdot \boldsymbol{e}_j = \begin{cases} 1 & \text{if} \quad i = j , \\ 0 & \text{if} \quad i \neq j . \end{cases}$$

Thus, we have

$$\begin{aligned} \boldsymbol{x} &= x^1 \boldsymbol{e}_1 + \dots + x^n \boldsymbol{e}_n , \\ \boldsymbol{\xi} &= \xi^1 \boldsymbol{e}_1 + \dots + \xi^n \boldsymbol{e}_n . \end{aligned}$$

The values x^i, (ξ^i), $i = 1, \dots, n$, are called the Cartesian coordinates of the vector $\boldsymbol{x}$, $(\boldsymbol{\xi})$ in X^n. Analogously, the values ξ^i, $i = 1, \dots, n$, are called the Cartesian coordinates of the vector $\boldsymbol{\xi}$ in Ξ^n. The coordinate transformation (2.1) defines, in the domain X^n, new coordinates $\xi^1, \dots, \xi^n$, which are called the curvilinear coordinates. The matrix

$$\jmath = \left\{ \frac{\partial x^i}{\partial \xi^j} \right\} , \qquad i, j = 1, \dots, n ,$$

is referred to as the Jacobi matrix, and its Jacobian is designated by J:

$$J = \det \left\{ \frac{\partial x^i}{\partial \xi^j} \right\} , \qquad i, j = 1, \dots, n . \tag{2.2}$$

The inverse transformation to the coordinate mapping $\boldsymbol{x}(\boldsymbol{\xi})$ is denoted by

$$\boldsymbol{\xi}(\boldsymbol{x}) : X^n \to \Xi^n .$$

This transformation can be considered analogously as a mapping introducing a curvilinear coordinate system $x^1, \dots, x^n$ in the domain $\Xi^n \subset R^n$. It is obvious that the inverse to the matrix $\mathcal{J}$ is

$$\mathcal{J}^{-1} = \left\{ \frac{\partial \xi^i}{\partial x^j} \right\} , \qquad i, j = 1, \dots, n ,$$

and consequently

$$\det \left\{ \frac{\partial \xi^i}{\partial x^j} \right\} = \frac{1}{J} , \qquad i, j = 1, \dots, n . \tag{2.3}$$

In the case of two-dimensional space, the elements of the matrices $(\partial x^i / \partial \xi^j)$ and $(\partial \xi^i / \partial x^j)$ are connected by

$$\begin{aligned} \frac{\partial \xi^i}{\partial x^j} &= (-1)^{i+j} \frac{\partial x^{3-j}}{\partial \xi^{3-i}} \bigg/ J , \\ \frac{\partial x^i}{\partial \xi^j} &= (-1)^{i+j} J \frac{\partial \xi^{3-j}}{\partial x^{3-i}} , \qquad i, j = 1, 2 , \end{aligned} \tag{2.4}$$

with fixed indices i and j. Similar relations between the elements of the corresponding three-dimensional matrices have the form

$$\begin{aligned} \frac{\partial \xi^i}{\partial x^j} &= \frac{1}{J} \left(\frac{\partial x^{j+1}}{\partial \xi^{i+1}} \frac{\partial x^{j+2}}{\partial \xi^{i+2}} - \frac{\partial x^{j+1}}{\partial \xi^{i+2}} \frac{\partial x^{j+2}}{\partial \xi^{i+1}} \right) , \\ \frac{\partial x^i}{\partial \xi^j} &= J \left(\frac{\partial \xi^{j+1}}{\partial x^{i+1}} \frac{\partial \xi^{j+2}}{\partial x^{i+2}} - \frac{\partial \xi^{j+1}}{\partial x^{i+2}} \frac{\partial \xi^{j+2}}{\partial x^{i+1}} \right) , \qquad i, j = 1, 2, 3 , \end{aligned} \tag{2.5}$$

where i and j are fixed indices and for each superscript or subscript index, say l, $l + 3$ is equivalent to l. With this condition, the sequence of indices $(l, l + 1, l + 2)$ is the result of a cyclic permutation of $(1, 2, 3)$ and vice versa; the indices of a cyclic sequence (i, j, k) satisfy the relation $j = i + 1, k = i + 2$.

2.2.2 Tangential Vectors

The value of the function $\boldsymbol{x}(\boldsymbol{\xi}) = [x^1(\boldsymbol{\xi}), \dots, x^n(\boldsymbol{\xi})]$ in the Cartesian basis $(\boldsymbol{e}_1, \dots, \boldsymbol{e}_n)$, i.e.

$$\boldsymbol{x}(\boldsymbol{\xi}) = x^1(\boldsymbol{\xi})\boldsymbol{e}_1 + \dots + x^n(\boldsymbol{\xi})\boldsymbol{e}_n ,$$

is a position vector for every $\boldsymbol{\xi} \in \Xi^n$. This vector-valued function $\boldsymbol{x}(\boldsymbol{\xi})$ generates the nodes, edges, faces, etc. of the cells of the coordinate grid in the domain X^n. Each edge of the cell corresponds to a coordinate line ξ^i for some i and is defined by the vector

$$\Delta_i \boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi} + h\boldsymbol{e}^i) - \boldsymbol{x}(\boldsymbol{\xi}) \ ,$$

where h is the step size of the uniform grid in the ξ^i direction in the logical domain Ξ^n. We have

$$\Delta_i \boldsymbol{x} = h\boldsymbol{x}_{\xi^i} + \boldsymbol{t} \ ,$$

where

$$\boldsymbol{x}_{\xi^i} = \left(\frac{\partial x^1}{\partial \xi^i}, \ldots, \frac{\partial x^n}{\partial \xi^i}\right)$$

is the vector tangential to the coordinate curve ξ^i, and $\boldsymbol{t}$ is a residual vector whose length does not exceed the following quantity:

$$\frac{1}{2} \max |\boldsymbol{x}_{\xi^i \xi^i}| h^2 \ .$$

Thus, the cells in the domain X^n whose edges are formed by the vectors $h\boldsymbol{x}_{\xi^i}$, $i = 1, \ldots, n$, are approximately the same as those obtained by mapping the uniform coordinate cells in the computational domain Ξ^n with the transformation $\boldsymbol{x}(\boldsymbol{\xi})$. Consequently, the uniformly contracted n-dimensional parallelepiped spanned by the tangential vectors $\boldsymbol{x}_{\xi^i}$, $i = 1, \ldots, n$, represents, to a high order of accuracy with respect to h, the cell of the coordinate grid at the corresponding point in X^n (see Fig. 2.1 for $n = 2$). In particular, for the length l_i of the ith grid edge, we have

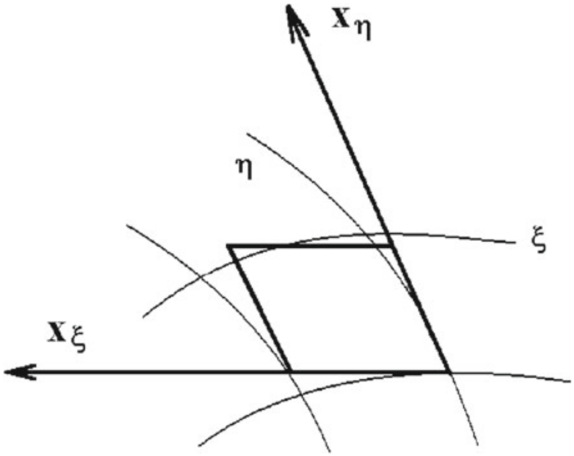


Fig. 2.1 Grid cell and contracted parallelogram

$$l_i = h|\boldsymbol{x}_{\xi^i}| + O(h^2) .$$

The volume V_h (area in two dimensions) of the cell is expressed as follows:

$$V_h = h^n V + O(h^{n+1}) ,$$

where V is the volume of the n-dimensional parallelepiped determined by the tangential vectors $\boldsymbol{x}_{\xi^i}, i = 1, \dots, n$.

The tangential vectors $\boldsymbol{x}_{\xi^i}, i = 1, \dots, n$, are called the base covariant vectors since they comprise a vector basis. The sequence $\boldsymbol{x}_{\xi^1}, \dots, \boldsymbol{x}_{\xi^n}$ of the tangential vectors has a right-handed orientation if the Jacobian of the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is positive. Otherwise, the base vectors $\boldsymbol{x}_{\xi^i}$ have a left-handed orientation.

The operation of the dot product on these vectors produces elements of the covariant metric tensor. These elements generate the coefficients that appear in the transformed governing equations that model the conservation-law equations of mechanics. Besides this, the metric elements play a primary role in studying and formulating various geometric characteristics of the coordinate grid cells.

2.2.3 *Normal Vectors*

For a fixed i, the vector

$$\left(\frac{\partial \xi^i}{\partial x^1}, \dots, \frac{\partial \xi^i}{\partial x^n}\right) ,$$

which is the gradient of $\xi^i(\boldsymbol{x})$ with respect to the Cartesian coordinates $x^1, \dots, x^n$, is denoted by $\nabla \xi^i$. The set of the vectors $\nabla \xi^i,\ i = 1, \dots, n$, is called the set of base contravariant vectors.

Similarly, as the tangential vectors relate to the coordinate curves, the contravariant vectors $\nabla \xi^i,\ i = 1, \dots, n$, are connected with their respective $(n-1)$-dimensional coordinate surfaces (curves in two dimensions). A coordinate surface is defined by the equation $\xi^i = \xi^i_0$; i.e. along the surface, all of the coordinates $\xi^1, \dots, \xi^n$ except ξ^i are allowed to vary. For all of the tangent vectors $\boldsymbol{x}_{\xi^j}$ to the coordinate lines on the surface $\xi^i = \xi^i_0$, we have the obvious identity

$$\boldsymbol{x}_{\xi^j} \cdot \nabla \xi^i = 0 , \qquad i \neq j ,$$

and thus the vector $\nabla \xi^i$ is normal to the coordinate surface $\xi^i = \xi^i_0$. Therefore, the vectors $\nabla \xi^i, i = 1, \dots, n$, are also called the normal base vectors.

Since

$$\boldsymbol{x}_{\xi^i} \cdot \nabla \xi^i = 1$$

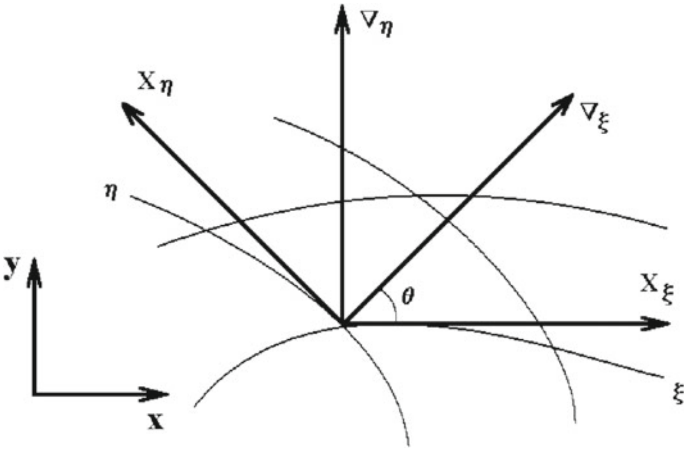


Fig. 2.2 Disposition of the base tangential and normal vectors in two dimensions

for each fixed $i = 1, \ldots, n$, the vectors $\boldsymbol{x}_{\xi^i}$ and $\nabla\xi^i$ intersect each other at an angle θ which is less than $\pi/2$. Now, taking into account the orthogonality of the vector $\nabla\xi^i$ to the surface $\xi^i = \xi^i_0$, we find that these two vectors $\boldsymbol{x}_{\xi^i}$ and $\nabla\xi^i$ are directed to the same side of the $(n-1)$-dimensional coordinate surface (curve in two dimensions). An illustration of this fact in two dimensions is given in Fig. 2.2.

The length of any normal base vector $\nabla\xi^i$ is linked to the distance d_i between the corresponding opposite boundary segments (joined by the vector $\boldsymbol{x}_{\xi^i}$) of the n-dimensional parallelepiped formed by the base tangential vectors, namely,

$$d_i = 1/|\nabla\xi^i| \ , \qquad |\nabla\xi^i| = \sqrt{\nabla\xi^i \cdot \nabla\xi^i} \ .$$

To prove this relation, we recall that the vector $\nabla\xi^i$ is normal to all of the vectors $\boldsymbol{x}_{\xi^j}$, $j \neq i$, and therefore to the boundary segments formed by these $n-1$ vectors. Hence, the unit normal vector $\boldsymbol{n}_i$ to these segments is expressed by

$$\boldsymbol{n}_i = \nabla\xi^i/|\nabla\xi^i| \ .$$

Now, taking into account that

$$d_i = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{n}_i \ ,$$

we readily obtain

$$d_i = \boldsymbol{x}_{\xi^i} \cdot \nabla\xi^i/|\nabla\xi^i| = 1/|\nabla\xi^i| \ .$$

Let l_i denote the distance between a grid point on the coordinate surface $\xi^i = c$ and the nearest point on the neighboring coordinate surface $\xi^i = c + h$; then,

$$l_i = hd_i + O(h^2) = h/|\nabla\xi^i| + O(h^2) \ .$$

This equation shows that the inverse length of the normal vector $\nabla\xi^i$ multiplied by h represents with high accuracy the distance between the corresponding faces of the coordinate cells in the domain X^n.

Note that the volume of the parallelepiped spanned by the tangential vectors equals J, so we find from (2.3) that the volume of the n-dimensional parallelepiped defined by the normal vectors $\nabla\xi^i$, $i = 1, \dots, n$, is equal to $1/J$. Thus, both the base normal vectors $\nabla\xi^i$ and the base tangential vectors $\boldsymbol{x}_{\xi^i}$ have the same right-handed or left-handed orientation.

If the coordinate system $\xi^1, \dots, \xi^n$ is orthogonal, i.e.

$$\boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j} = P(\boldsymbol{x})\delta^i_j \ , \qquad P(\boldsymbol{x}) > 0 \ , \qquad i, j = 1, \dots, n \ ,$$

then for each fixed $i = 1, \dots, n$ the vector $\nabla\xi^i$ is parallel to $\boldsymbol{x}_{\xi^i}$. Here and later, δ^i_j is the Kronecker symbol, i.e.

$$\delta^i_j = 0 \quad \text{if} \quad i \neq j \ , \qquad \delta^i_j = 1 \quad \text{if} \quad i = j \ .$$

2.2.4 Representation of Vectors Through the Base Vectors

If there are n independent base vectors $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ of the Euclidean space R^n, then any vector $\boldsymbol{b}$ with components $b^1, \dots, b^n$ in the Cartesian basis $\boldsymbol{e}_1, \dots, \boldsymbol{e}_n$ is represented through the vectors $\boldsymbol{a}^i$, $i = 1, \dots, n$, by

$$\boldsymbol{b} = a^{ij}(\boldsymbol{b} \cdot \boldsymbol{a}_j)\boldsymbol{a}_i, \qquad i, j = 1, \dots, n \ , \tag{2.6}$$

where a^{ij} are the elements of the matrix (a^{ij}) which is the inverse of the tensor (a_{ij}), $a_{ij} = \boldsymbol{a}_i \cdot \boldsymbol{a}_j$, $i, j = 1, \dots, n$. It is assumed in (2.6) and later that a summation is carried out over repeated indices unless otherwise noted.

The components of the vector $\boldsymbol{b}$ in the natural basis of the tangential vectors $\boldsymbol{x}_{\xi^i}$, $i = 1, \dots, n$, are called contravariant. Let them be denoted by $\overline{b}^i$, $i = 1, \dots, n$. Thus,

$$\boldsymbol{b} = \overline{b}^1 \boldsymbol{x}_{\xi^1} + \cdots + \overline{b}^n \boldsymbol{x}_{\xi^n} \ .$$

Assuming in (2.5) $\boldsymbol{a}_i = \boldsymbol{x}_{\xi^i}$, $i = 1, \dots, n$, we obtain

$$\overline{b}^i = a^{mj}\left(b^k \frac{\partial x^k}{\partial \xi^j}\right)\frac{\partial x^i}{\partial \xi^m} \ , \qquad i, j, k, m = 1, \dots, n \ , \tag{2.7}$$

where $b^1, \dots, b^n$ are the components of the vector $\boldsymbol{b}$ in the Cartesian basis $\boldsymbol{e}_1, \dots, \boldsymbol{e}_n$. Since

$$a_{ij} = \frac{\partial x^k}{\partial \xi^i}\frac{\partial x^k}{\partial \xi^j} \ , \qquad i, j, k = 1, \dots, n \ ,$$

we have

$$a^{ij} = \frac{\partial \xi^i}{\partial x^k}\frac{\partial \xi^j}{\partial x^k}\ , \qquad k = 1, \dots, n\ .$$

Therefore, from (2.7),

$$\overline{b}^i = b^j \frac{\partial \xi^i}{\partial x^j}\ , \qquad i, j = 1, \dots, n\ , \tag{2.8}$$

or, using the dot product notation,

$$\overline{b}^i = \boldsymbol{b} \cdot \nabla \xi^i\ , \qquad i = 1, \dots, n\ . \tag{2.9}$$

Thus, in this case, (2.6) has the form

$$\boldsymbol{b} = (\boldsymbol{b} \cdot \nabla \xi^i)\boldsymbol{x}_{\xi^i}\ , \qquad i = 1, \dots, n\ . \tag{2.10}$$

For example, the normal base vector $\nabla \xi^i$ is expanded through the base tangential vectors $\boldsymbol{x}_{\xi^j},\ j = 1, \dots, n$, by the following formula:

$$\nabla \xi^i = \frac{\partial \xi^i}{\partial x^j}\frac{\partial \xi^k}{\partial x^j}\boldsymbol{x}_{\xi^k}\ , \qquad i, j, k, = 1, \dots, n\ . \tag{2.11}$$

Analogously, a component $\overline{b}_i$ of the vector $\boldsymbol{b}$ in the basis $\nabla \xi^i, i = 1, \dots, n$, is expressed by the formula

$$\overline{b}_i = b^j \frac{\partial x^j}{\partial \xi^i} = \boldsymbol{b} \cdot \boldsymbol{x}_{\xi^i}\ , \qquad i = 1, \dots, n\ , \tag{2.12}$$

and consequently

$$\boldsymbol{b} = \overline{b}_i \nabla \xi^i = (\boldsymbol{b} \cdot \boldsymbol{x}_{\xi^i})\nabla \xi^i, \qquad i = 1, \dots, n\ . \tag{2.13}$$

These components $\overline{b}_i, i = 1, \dots, n$, of the vector $\boldsymbol{b}$ are called covariant. In particular, the base tangential vector $\boldsymbol{x}_{\xi^i}$ is expressed through the base normal vectors $\nabla \xi^j$, $j = 1, \dots, n$, as follows:

$$\boldsymbol{x}_{\xi^i} = \frac{\partial x^j}{\partial \xi^i}\frac{\partial x^j}{\partial \xi^k}\nabla \xi^k\ , \qquad i, j, k = 1, \dots, n\ . \tag{2.14}$$

2.2.5 Metric Tensors

Many grid generation algorithms, in particular those based on the calculus of variations, are typically formulated in terms of fundamental features of coordinate transformations and the corresponding mesh cells. These features are compactly described with the use of the metric notation, which is discussed in this subsection.

Covariant Metric Tensor

The matrix

$$\{g_{ij}\}\ , \qquad i, j = 1, \dots, n\ ,$$

whose elements g_{ij} are the dot products of the pairs of the basic tangential vectors $\boldsymbol{x}_{\xi^i}$,

$$g_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j} = \frac{\partial x^k}{\partial \xi^i}\frac{\partial x^k}{\partial \xi^j}\ , \qquad i, j, k = 1, \dots, n\ , \tag{2.15}$$

is called a covariant metric tensor of the domain X^n in the coordinates $\xi^1, \dots, \xi^n$. Geometrically, each diagonal element g_{ii} of the matrix $\{g_{ij}\}$ is the length of the tangent vector $\boldsymbol{x}_{\xi^i}$ squared:

$$g_{ii} = |\boldsymbol{x}_{\xi^i}|^2\ , \qquad i = 1, \dots, n\ .$$

Also,

$$g_{ij} = |\boldsymbol{x}_{\xi^i}||\boldsymbol{x}_{\xi^j}|\cos\theta = \sqrt{g_{ii}}\sqrt{g_{jj}}\cos\theta\ , \tag{2.16}$$

where θ is the angle between the tangent vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$. In these expressions, for g_{ii} and g_{ij}, the subscripts ii and jj are fixed, i.e. here, the summation over the repeated indices is not carried out.

The matrix $\{g_{ij}\}$ is called the metric tensor because it defines distance measurements with respect to the coordinates $\xi^1, \dots, \xi^n$:

$$\mathrm{d}s = \sqrt{g_{ij}\mathrm{d}\xi^i \mathrm{d}\xi^j}, \qquad i, j = 1, \dots, n\ .$$

Thus, the length s of the curve in X^n prescribed by the parametrization

$$\boldsymbol{x}[\boldsymbol{\xi}(t)] : [a, b] \to X^n$$

is computed by the formula

$$s = \int_a^b \sqrt{g_{ij}\frac{\mathrm{d}\xi^i}{\mathrm{d}t}\frac{\mathrm{d}\xi^j}{\mathrm{d}t}}\mathrm{d}t\ .$$

We designate by g the Jacobian of the covariant matrix $\{g_{ij}\}$. It is evident that

$$\{g_{ij}\} = \jmath\jmath^T \ ,$$

and hence

$$J^2 = g \ .$$

The covariant metric tensor is a symmetric matrix, i.e. $g_{ij} = g_{ji}$. If a coordinate system at a point $\boldsymbol{\xi}$ is orthogonal, then the tensor $\{g_{ij}\}$ has a simple diagonal form at this point. Note that these advantageous properties are in general not possessed by the Jacobi matrix $\{\partial x^i/\partial \xi^j\}$ from which the covariant metric tensor $\{g_{ij}\}$ is defined.

Contravariant Metric Tensor

The contravariant metric tensor of the domain X^n in the coordinates $\xi^1, \ldots, \xi^n$ is the matrix

$$\{g^{ij}\} \ , \qquad i, j = 1, \ldots, n \ ,$$

inverse to $\{g_{ij}\}$, i.e.

$$g_{ij} g^{jk} = \delta_i^k \ , \qquad i, j, k = 1, \ldots, n \ . \tag{2.17}$$

Therefore,

$$\det\{g^{ij}\} = \frac{1}{g} \ .$$

It is easily shown that (2.16) is satisfied if and only if

$$g^{ij} = \nabla \xi^i \cdot \nabla \xi^j = \frac{\partial \xi^i}{\partial x^k} \frac{\partial \xi^j}{\partial x^k} \ , \qquad i, j, k = 1, \ldots, n \ . \tag{2.18}$$

Thus, each diagonal element g^{ii} (where i is fixed) of the matrix $\{g^{ij}\}$ is the square of the length of the vector $\nabla \xi^i$:

$$g^{ii} = |\nabla \xi^i|^2 \ . \tag{2.19}$$

Geometric Interpretation

Now we discuss the geometric meaning of a fixed diagonal element g^{ii}, say g^{11}, of the matrix $\{g^{ij}\}$. Let us consider a three-dimensional coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^3 \to X^3$. Its tangential vectors $\boldsymbol{x}_{\xi^1}, \boldsymbol{x}_{\xi^2}, \boldsymbol{x}_{\xi^3}$ represent geometrically the edges of the parallelepiped formed by these vectors. For the distance d_1 between the opposite faces of the parallelepiped which are defined by the vectors $\boldsymbol{x}_{\xi^2}$ and $\boldsymbol{x}_{\xi^3}$, we have

$$d_1 = \boldsymbol{x}_{\xi^1} \cdot \boldsymbol{n}_1 \ ,$$

where $\boldsymbol{n}_1$ is the unit normal to the plane spanned by the vectors $\boldsymbol{x}_{\xi^2}$ and $\boldsymbol{x}_{\xi^3}$. It is clear that

$$\nabla_{\xi^1} \cdot \boldsymbol{x}_{\xi^j} = 0 \;, \qquad j = 2, 3 \;,$$

and hence the unit normal $\boldsymbol{n}_1$ is parallel to the normal base vector ∇_{ξ^1}. Thus, we obtain

$$\boldsymbol{n}_1 = \nabla_{\xi^1}/|\nabla_{\xi^1}| = \nabla_{\xi^1}/\sqrt{g^{11}} \;.$$

Therefore,

$$d_1 = \nabla_{\xi^1} \cdot \nabla_{\xi^1}/\sqrt{g^{11}} = 1/\sqrt{g^{11}} \;,$$

and consequently

$$g^{11} = 1/(d_1)^2 \;.$$

Analogous relations are valid for g^{22} and g^{33}, i.e. in three dimensions, the diagonal element g^{ii} for a fixed i means the inverse square of the distance d_i between those faces of the parallelepiped which are connected by the vector $\boldsymbol{x}_{\xi^i}$. In two-dimensional space, the element g^{ii} (where i is fixed) is the inverse square of the distance between the edges of the parallelogram defined by the tangential vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$.

The same interpretation of g^{ii} is valid for general multidimensional coordinate transformations:

$$g^{ii} = 1/(d_i)^2, \qquad i = 1, \ldots, n \;, \tag{2.20}$$

where the index i is fixed, and d_i is the distance between those faces of the n-dimensional parallelepiped which are linked by the tangential vector $\boldsymbol{x}_{\xi^i}$.

Relations Between Covariant and Contravariant Elements

Now, in analogy with (2.4) and (2.5), we write out very convenient formulas for natural relations between the contravariant elements g^{ij} and the covariant ones g_{ij} in two and three dimensions.

For $n = 2$,

$$\begin{aligned} g^{ij} &= (-1)^{i+j} \frac{g_{3-i\;3-j}}{g} \;, \\ g_{ij} &= (-1)^{i+j} g g^{3-i\;3-j} \;, \qquad i, j = 1, 2 \;, \end{aligned} \tag{2.21}$$

where the indices i, j on the right-hand side of the relations (2.21) are fixed, i.e. summation over the repeated indices is not carried out here. For $n = 3$, we have

$$\begin{aligned} g^{ij} &= \frac{1}{g}(g_{i+1\;j+1}\, g_{i+2\;j+2} - g_{i+1\;j+2}\, g_{i+2\;j+1}) \;, \\ g_{ij} &= g(g^{i+1\;j+1}\, g^{i+2\;j+2} - g^{i+1\;j+2}\, g^{i+2\;j+1}) \;, \qquad i, j = 1, 2, 3 \;, \end{aligned} \tag{2.22}$$

with the convention that any index, say l, is identified with $l \pm 3$, so, for instance, $g_{45} = g_{12}$.

We also note that, in accordance with the expressions (2.15) and (2.18) for g_{ij} and g^{ij}, respectively, the relations (2.11) and (2.14) between the basic vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{\nabla}\xi^j$ can be written in the form

$$\begin{aligned} \boldsymbol{x}_{\xi^i} &= g_{ik}\boldsymbol{\nabla}\xi^k \ , \\ \boldsymbol{\nabla}\xi^i &= g^{ik}\boldsymbol{x}_{\xi^k} \ , \qquad i,k = 1,\ldots,n \ . \end{aligned} \tag{2.23}$$

So, the first derivatives $\partial x^i/\partial\xi^j$ and $\partial\xi^k/\partial x^m$ of the transformations $\boldsymbol{x}(\boldsymbol{\xi})$ and $\boldsymbol{\xi}(\boldsymbol{x})$, respectively, are connected through the metric elements:

$$\begin{aligned} \frac{\partial x^i}{\partial\xi^j} &= g_{mj}\frac{\partial\xi^m}{\partial x^i} \ , \\ \frac{\partial\xi^i}{\partial x^j} &= g^{mi}\frac{\partial x^j}{\partial\xi^m} \ , \qquad i,j,m = 1,\ldots,n \ . \end{aligned} \tag{2.24}$$

2.2.6 Cross Product

In addition to the dot product, there is another important operation on three-dimensional vectors. This is the cross product, ×, which for any two vectors $\boldsymbol{a} = (a^1, a^2, a^3)$, $\boldsymbol{b} = (b^1, b^2, b^3)$ is expressed as the determinant of a matrix:

$$\boldsymbol{a}\times\boldsymbol{b} = \det\begin{Bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 \\ a^1 & a^2 & a^3 \\ b^1 & b^2 & b^3 \end{Bmatrix} \ , \tag{2.25}$$

where $(\boldsymbol{e}_1,\ \boldsymbol{e}_2,\ \boldsymbol{e}_3)$ is the Cartesian vector basis of the Euclidean space R^3. Thus,

$$\boldsymbol{a}\times\boldsymbol{b} = (a^2b^3 - a^3b^2,\ a^3b^1 - a^1b^3,\ a^1b^2 - a^2b^1) \ ,$$

or, with the previously mentioned convention in three dimensions of the identification of any index j with $j \pm 3$,

$$\boldsymbol{a}\times\boldsymbol{b} = (a^{i+1}b^{i+2} - a^{i+2}b^{i+1})\boldsymbol{e}_i \ , \qquad i = 1,2,3 \ . \tag{2.26}$$

We will now state some facts connected with the cross product operation.

Geometric Meaning

We can readily see that $\boldsymbol{a}\times\boldsymbol{b} = \boldsymbol{0}$ if the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are parallel. Also, from (2.26), we find that $\boldsymbol{a}\cdot(\boldsymbol{a}\times\boldsymbol{b}) = 0$ and $\boldsymbol{b}\cdot(\boldsymbol{a}\times\boldsymbol{b}) = 0$, i.e. the vector $\boldsymbol{a}\times\boldsymbol{b}$ is orthogonal to each of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Thus, if these vectors are not parallel, then

$$\boldsymbol{a}\times\boldsymbol{b} = \alpha|\boldsymbol{a}\times\boldsymbol{b}|\boldsymbol{n} \ , \tag{2.27}$$

where $\alpha = 1$ or $\alpha = -1$ and $\boldsymbol{n}$ is a unit normal vector to the plane determined by the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

Now we show that the length of the vector $\boldsymbol{a} \times \boldsymbol{b}$ equals the area of the parallelogram formed by the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, i.e.

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\theta \ , \tag{2.28}$$

where θ is the angle between the two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. To prove (2.28), we first note that

$$|\boldsymbol{a}|^2|\boldsymbol{b}|^2 \sin^2\theta = |\boldsymbol{a}|^2|\boldsymbol{b}|^2(1-\cos^2\theta) = |\boldsymbol{a}|^2|\boldsymbol{b}|^2 - (\boldsymbol{a}\cdot\boldsymbol{b})^2 \ .$$

We have, furthermore,

$$\begin{aligned}|\boldsymbol{a}|^2|\boldsymbol{b}|^2 - |\boldsymbol{a}\cdot\boldsymbol{b}|^2 &= \left(\sum_{i=1}^{3} a^i a^i\right)\left(\sum_{j=1}^{3} b^j b^j\right) - \left(\sum_{k=1}^{3} a^k b^k\right)^2 \\ &= \sum_{k=1}^{3}[(a^l)^2(b^m)^2 + (a^m)^2(b^l)^2 - 2a^l b^l a^m b^m] \\ &= \sum_{k=1}^{3}(a^l b^m - a^m b^l)^2 \ ,\end{aligned}$$

where (k, l, m) are cyclic, i.e. $l = k+1, \ m = k+2$ with the convention that $j+3$ is equivalent to j for any index j. According to (2.26), the quantity $a^l b^m - a^m b^l$ for the cyclic sequence (k, l, m) is the kth component of the vector $\boldsymbol{a} \times \boldsymbol{b}$, so we find that

$$|\boldsymbol{a}||\boldsymbol{b}| \sin^2\theta = |\boldsymbol{a}|^2|\boldsymbol{b}|^2 - |\boldsymbol{a}\cdot\boldsymbol{b}|^2 = |\boldsymbol{a}\times\boldsymbol{b}|^2 \ , \tag{2.29}$$

which proves (2.28). Thus, we obtain the result that if the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are not parallel, then the vector $\boldsymbol{a} \times \boldsymbol{b}$ is orthogonal to the parallelogram formed by these vectors and its length equals the area of the parallelogram. Therefore, the three vectors $\boldsymbol{a}, \ \boldsymbol{b}$ and $\boldsymbol{a} \times \boldsymbol{b}$ are independent in this case and represent a base vector system in the three-dimensional space R^3. Moreover, the vectors $\boldsymbol{a}, \ \boldsymbol{b}$ and $\boldsymbol{a} \times \boldsymbol{b}$ form a right-handed triad, since $\boldsymbol{a} \times \boldsymbol{b} \neq \boldsymbol{0}$, and consequently the Jacobian of the matrix determined by $\boldsymbol{a}, \ \boldsymbol{b}$, and $\boldsymbol{a} \times \boldsymbol{b}$, is positive; it equals

$$\boldsymbol{a} \times \boldsymbol{b} \cdot \boldsymbol{a} \times \boldsymbol{b} = (\boldsymbol{a} \times \boldsymbol{b})^2 \ .$$

Relation to Volumes

Let $\boldsymbol{c} = (c^1, c^2, c^3)$ be one more vector. The volume V of the parallelepiped whose edges are the vectors $\boldsymbol{a}, \ \boldsymbol{b}$ and $\boldsymbol{c}$ equals the area of the parallelogram formed by the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ multiplied by the modulas of the dot product of the vector $\boldsymbol{c}$ and the unit normal $\boldsymbol{n}$ to the parallelogram. Thus,

$$V = |\boldsymbol{a} \times \boldsymbol{b}||\boldsymbol{n} \cdot \boldsymbol{c}|$$

and from (2.27), we obtain

$$V = |(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c}| \ . \tag{2.30}$$

Taking into account (2.26), we obtain

$$(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c} = c^1(a^2b^3 - a^3b^2) + c^2(a^3b^1 - a^1b^3) + c^3(a^1b^2 - a^2b^1) \ .$$

The right-hand side of this equation is the Jacobian of the matrix whose rows are formed by the vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, i.e.

$$(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c} = \det \begin{Bmatrix} a^1 & a^2 & a^3 \\ b^1 & b^2 & b^3 \\ c^1 & c^2 & c^3 \end{Bmatrix} \ . \tag{2.31}$$

From this equation, we readily obtain

$$(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c} = \boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{c} \times \boldsymbol{a}) \cdot \boldsymbol{b} \ .$$

Thus, the volume of the parallelepiped determined by the vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ equals the Jacobian of the matrix formed by the components of these vectors. In particular, we obtain from (2.2) that the Jacobian of a three-dimensional coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is expressed as follows:

$$J = \boldsymbol{x}_{\xi^1} \cdot (\boldsymbol{x}_{\xi^2} \times \boldsymbol{x}_{\xi^3}) \ . \tag{2.32}$$

Relation to Base Vectors

Applying the operation of the cross product to two base tangential vectors $\boldsymbol{x}_{\xi^l}$ and $\boldsymbol{x}_{\xi^m}$, we find that the vector $\boldsymbol{x}_{\xi^l} \times \boldsymbol{x}_{\xi^m}$ is normal to the coordinate surface $\xi^i = \xi^i_0$ with (i, l, m) cyclic. The base normal vector $\nabla \xi^i$ is also orthogonal to the surface, and therefore it is a scalar multiple of $\boldsymbol{x}_{\xi^l} \times \boldsymbol{x}_{\xi^m}$, i.e.

$$\nabla \xi^i = c(\boldsymbol{x}_{\xi^l} \times \boldsymbol{x}_{\xi^m}) \ .$$

Multiplying this equation for a fixed i by $\boldsymbol{x}_{\xi^i}$, using the operation of the dot product, we obtain, using (2.32),

$$1 = c \ J \ ,$$

and therefore

$$\nabla \xi^i = \frac{1}{J}(\boldsymbol{x}_{\xi^l} \times \boldsymbol{x}_{\xi^m}) \ . \tag{2.33}$$

Thus, the elements of the three-dimensional contravariant metric tensor $\{g^{ij}\}$ are computed through the tangential vectors $\boldsymbol{x}_{\xi^i}$ by the formula

$$g^{ij} = \frac{1}{g}(\boldsymbol{x}_{\xi^{i+1}} \times \boldsymbol{x}_{\xi^{i+2}}) \cdot (\boldsymbol{x}_{\xi^{j+1}} \times \boldsymbol{x}_{\xi^{j+2}}) \ , \qquad i, j = 1, 2, 3 \ .$$

Analogously, every base vector $\boldsymbol{x}_{\xi^i}$, $i = 1, 2, 3$, is expressed by the tensor product of the vectors $\nabla\xi^j$, $j = 1, 2, 3$:

$$\boldsymbol{x}_{\xi^i} = J(\nabla\xi^l \times \nabla\xi^k), \qquad i = 1, 2, 3 \ , \tag{2.34}$$

where $l = i+1$, $k = i+2$, and m is equivalent to $m+3$ for any index m. Accordingly, we have

$$g_{ij} = g(\nabla\xi^{i+1} \times \nabla\xi^{i+2}) \cdot (\nabla\xi^{j+1} \times \nabla\xi^{j+2}) \ , \qquad i, j = 1, 2, 3 \ .$$

Using the relations (2.33) and (2.34) in (2.32), we also obtain

$$\frac{1}{J} = \nabla\xi^1 \cdot \nabla\xi^2 \times \nabla\xi^3 \ . \tag{2.35}$$

Thus, the volume of the parallelepiped formed by the base normal vectors $\nabla\xi^1$, $\nabla\xi^2$, and $\nabla\xi^3$ is the modulus of the inverse of the Jacobian J of the transformation $\boldsymbol{x}(\boldsymbol{\xi})$.

2.3 Relations Concerning Second Derivatives

The elements of the covariant and contravariant metric tensors are defined by the dot products of the base tangential and normal vectors, respectively. These elements are suitable for describing the internal features of the cells such as the lengths of the edges, the areas of the faces, their volumes, and the angles between the edges and the faces. However, as they are derived from the first derivatives of the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$, the direct use of the metric elements is not sufficient for the description of the dynamic features of the grid (e.g. curvature), which reflect changes between adjacent cells. This is because the formulation of these grid features relies not only on the first derivatives but also on the second derivatives of $\boldsymbol{x}(\boldsymbol{\xi})$. Therefore, there is a need to study relations connected with the second derivatives of the coordinate parametrizations.

This section presents some notations and formulas which are concerned with the second derivatives of the components of the coordinate transformations. These notations and relations will be used to describe the curvature and eccentricity of the coordinate lines and to formulate some equations of mechanics in new independent variables.

2.3.1 Christoffel Symbols

The edge of a grid cell in the ξ^i direction can be represented with high accuracy by the base vector $\boldsymbol{x}_{\xi^i}$ contracted by the factor h, which represents the step size of a uniform grid in Ξ^n. Therefore, the local change of the edge in the ξ^j direction is characterized by the derivative of $\boldsymbol{x}_{\xi^i}$ with respect to ξ^j, i.e. by $\boldsymbol{x}_{\xi^i\xi^j}$.

Since the second derivatives may be used to formulate quantitative measures of the grid, we describe these vectors $\boldsymbol{x}_{\xi^i\xi^j}$ through the base tangential and normal vectors using certain three-index quantities known as Christoffel symbols, The Christoffel symbols are commonly used in formulating measures of the mutual interaction of the cells and in formulas for differential equations.

Let us denote by Γ^k_{ij} the kth contravariant component of the vector $\boldsymbol{x}_{\xi^i\xi^j}$ in the base tangential vectors $\boldsymbol{x}_{\xi^k}$, $k = 1, \dots, n$. The superscript k in this designation relates to the base vector $\boldsymbol{x}_{\xi^k}$ and the subscript ij corresponds to the mixed derivative with respect to ξ^i and ξ^j. Thus,

$$\boldsymbol{x}_{\xi^i\xi^j} = \Gamma^k_{ij}\, \boldsymbol{x}_{\xi^k} \ , \qquad i, j, k = 1, \dots, n \ , \tag{2.36}$$

and consequently

$$\frac{\partial^2 x^p}{\partial \xi^j \partial \xi^k} = \Gamma^m_{kj} \frac{\partial x^p}{\partial \xi^m} \ , \qquad j, k, m, p = 1, \dots, n \ . \tag{2.37}$$

In accordance with (2.8), we have

$$\Gamma^i_{kj} = \frac{\partial^2 x^l}{\partial \xi^k \partial \xi^j} \frac{\partial \xi^i}{\partial x^l} \ , \qquad i, j, k, l = 1, \dots, n \ , \tag{2.38}$$

or in vector form,

$$\Gamma^i_{kj} = \boldsymbol{x}_{\xi^k\xi^j} \cdot \nabla \xi^i \ . \tag{2.39}$$

Equation (2.38) is also obtained by multiplying (2.37) by $\partial \xi^i / \partial x^p$ and summing over p.

The quantities Γ^i_{kj} are called the space Christoffel symbols of the second kind and the expression (2.36) is a form of the Gauss relation representing the second derivatives of the position vector $\boldsymbol{x}(\boldsymbol{\xi})$ through the tangential vectors $\boldsymbol{x}_{\xi^i}$.

Analogously, the components of the second derivatives of the position vector $\boldsymbol{x}(\boldsymbol{\xi})$ expanded in the base normal vectors $\nabla \xi^i$, $i = 1, \dots, n$, are referred to as the space Christoffel symbols of the first kind. The mth component of the vector $\boldsymbol{x}_{\xi^k\xi^j}$ in the base vectors $\nabla \xi^i$, $i = 1, \dots, n$, is denoted by $[kj, m]$. Thus, according to (2.12),

$$[kj, m] = \boldsymbol{x}_{\xi^k\xi^j} \cdot \boldsymbol{x}_{\xi^m} = \frac{\partial^2 x^l}{\partial \xi^k \partial \xi^j} \frac{\partial x^l}{\partial \xi^m} \ , \qquad j, k, l, m = 1, \dots, n \ , \tag{2.40}$$

and consequently

$$\boldsymbol{x}_{\xi^k \xi^j} = [kj, m] \nabla \xi^m \ . \tag{2.41}$$

So, in analogy with (2.37), we obtain

$$\frac{\partial^2 x^l}{\partial \xi^j \partial \xi^k} = [kj, m] \frac{\partial \xi^m}{\partial x^i} \ , \qquad i, j, k, m = 1, \dots, n \ . \tag{2.42}$$

Multiplying (2.40) by g^{im} and summing over m, we find that the space Christoffel symbols of the first and second kind are connected by the following relation:

$$\Gamma^i_{kj} = g^{im} [kj, m] \ , \qquad i, j, k, m = 1, \dots, n \ . \tag{2.43}$$

Conversely, from (2.38),

$$[kj, m] = g_{ml} \Gamma^l_{kj}, \qquad j, k, l, m = 1, \dots, n \ . \tag{2.44}$$

The space Christoffel symbols of the first kind $[kj, m]$ can be expressed through the first derivatives of the covariant elements g_{ij} of the metric tensor (g_{ij}) by the following readily verified formula:

$$[kj, m] = \frac{1}{2} \left(\frac{\partial g_{jm}}{\partial \xi^k} + \frac{\partial g_{km}}{\partial \xi^j} - \frac{\partial g_{kj}}{\partial \xi^m} \right) \ , \qquad i, j, k, m = 1, \dots, n \ . \tag{2.45}$$

Thus, taking into account (2.43), we see that the space Christoffel symbols of the second kind Γ^i_{kj} can be written in terms of metric elements and their first derivatives. In particular, in the case of an orthogonal coordinate system ξ^i, we obtain, from (2.43) and (2.45),

$$\Gamma^i_{kj} = \frac{1}{2} g^{ii} \left(\frac{\partial g_{ii}}{\partial \xi^k} + \frac{\partial g_{ii}}{\partial \xi^j} - \frac{\partial g_{kj}}{\partial \xi^i} \right) \ .$$

Here, the index i is fixed, i.e. the summation over i is not carried out.

2.3.2 *Differentiation of the Jacobian*

Of critical importance in obtaining compact conservation-law equations with coefficients derived from the metric elements in new curvilinear coordinates $\xi^1, \dots, \xi^n$ is the formula for differentiation of the Jacobian

$$\frac{\partial J}{\partial \xi^k} \equiv J \frac{\partial^2 x^i}{\partial \xi^k \partial \xi^m} \frac{\partial \xi^m}{\partial x^i} \equiv J \frac{\partial}{\partial x^i} \left(\frac{\partial x^i}{\partial \xi^k} \right) \equiv J \mathrm{div}_x \frac{\partial \boldsymbol{x}}{\partial \xi^k} \ ,$$
$$i, k, m = 1, \dots, n \ . \tag{2.46}$$

In accordance with (2.38), this identity can also be expressed through the space Christoffel symbols of the second kind Γ^i_{kj} by

$$\frac{\partial J}{\partial \xi^k} = J\Gamma^i_{ik} \ , \qquad i, k = 1, \dots, n \ ,$$

with the summation convention over the repeated index i.

In order to prove the identity (2.46), we note that in the case of an arbitrary matrix $\{a_{ij}\}$, the first derivative of its Jacobian with respect to ξ^k is obtained through the process of differentiating the first row (the others are left unchanged), then performing the same operation on the second row, and so on with all of the rows of the matrix. The summation of the Jacobians of the matrices derived in such a manner gives the first derivative of the Jacobian of the original matrix $\{a_{ij}\}$. Thus,

$$\frac{\partial}{\partial \xi^k} \det\{a_{ij}\} = \frac{\partial a_{im}}{\partial \xi^k} G^{im} \ , \qquad i, j, k, m = 1, \dots, n \ , \tag{2.47}$$

where G^{im} is the cofactor of the element a_{im}. For the Jacobi matrix $\{\partial x^i/\partial \xi^j\}$ of the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$, we have

$$G^{im} = J\frac{\partial \xi^m}{\partial x^i}, \qquad i, j = 1, \dots, n \ .$$

Therefore, applying (2.47) to the Jacobi matrix, we obtain (2.46).

2.3.3 Basic Identity

The identity (2.46) implies the extremely important relation

$$\frac{\partial}{\partial \xi^j}\Bigl(J\frac{\partial \xi^j}{\partial x^i}\Bigr) \equiv 0 \ , \qquad i, j = 1, \dots, n \ , \tag{2.48}$$

which leads to specific forms of new dependent variables for conservation-law equations. To prove (2.48), we first note that

$$\frac{\partial^2 \xi^p}{\partial x^k \partial x^j}\frac{\partial x^l}{\partial \xi^p} = -\frac{\partial^2 x^l}{\partial \xi^p \partial \xi^m}\frac{\partial \xi^m}{\partial x^k}\frac{\partial \xi^p}{\partial x^j} \ , \qquad j, k, l, m, p = 1, \dots, n \ .$$

Multiplying this equation by $\partial \xi^i/\partial x^l$ and summing over l, we obtain a formula representing the second derivative $\partial^2 \xi^i/\partial x^k \partial x^m$ of the functions $\xi^i(\boldsymbol{x})$ through the second derivatives $\partial^2 x^m/\partial \xi^l \partial \xi^p$ of the functions $x^m(\boldsymbol{\xi})$, $m = 1, \dots, n$:

$$\frac{\partial^2 \xi^i}{\partial x^k \partial x^m} = -\frac{\partial^2 x^p}{\partial \xi^l \partial \xi^j}\frac{\partial \xi^j}{\partial x^k}\frac{\partial \xi^l}{\partial x^m}\frac{\partial \xi^i}{\partial x^p}\ , \qquad i, j, k, l, m, p = 1, \dots, n\ . \tag{2.49}$$

Now, using this relation and the formula (2.46) for differentiation of the Jacobian in the identity

$$\frac{\partial}{\partial \xi^j}\left(J\frac{\partial \xi^j}{\partial x^i}\right) = \frac{\partial J}{\partial \xi^j}\frac{\partial \xi^j}{\partial x^i} + J\frac{\partial^2 \xi^j}{\partial x^i \partial x^k}\frac{\partial x^k}{\partial \xi^j}\ , \qquad i, j, k = 1, \dots, n\ ,$$

we obtain

$$\begin{aligned}\frac{\partial}{\partial \xi^j}\left(J\frac{\partial \xi^j}{\partial x^i}\right) &= J\frac{\partial^2 x^k}{\partial \xi^p \partial \xi^j}\frac{\partial \xi^p}{\partial x^k}\frac{\partial \xi^j}{\partial x^i} - J\frac{\partial^2 x^p}{\partial \xi^l \partial \xi^m}\frac{\partial \xi^m}{\partial x^i}\frac{\partial \xi^l}{\partial x^k}\frac{\partial \xi^j}{\partial x^p}\frac{\partial x^k}{\partial \xi^j}\\ &= J\frac{\partial^2 x^k}{\partial \xi^p \partial \xi^j}\frac{\partial \xi^p}{\partial x^k}\frac{\partial \xi^j}{\partial x^i} - J\frac{\partial^2 x^p}{\partial \xi^l \partial \xi^m}\frac{\partial \xi^l}{\partial x^p}\frac{\partial \xi^m}{\partial x^i} = 0\ ,\end{aligned}$$

$$i, j, k, l, m, p = 1, \dots, n\ ,$$

i.e. (2.48) has been proved.

The identity (2.48) is obvious when $n = 1$ or $n = 2$. For example, for $n = 2$, we have, from (2.4),

$$J\frac{\partial \xi^j}{\partial x^i} = (-1)^{i+j}\frac{\partial x^{3-i}}{\partial \xi^{3-j}}\ , \qquad i, j = 1, 2\ ,$$

with fixed indices i and j, and therefore

$$\frac{\partial}{\partial \xi^j}\Big(J\frac{\partial \xi^j}{\partial x^i}\Big) = (-1)^{i+1}\Big(\frac{\partial}{\partial \xi^1}\frac{\partial x^{3-i}}{\partial \xi^2} - \frac{\partial}{\partial \xi^2}\frac{\partial x^{3-i}}{\partial \xi^1}\Big) = 0, \qquad i, j = 1, 2\ .$$

An inference from (2.48) for $n = 3$ follows from the differentiation of the cross product of the base tangential vectors $\boldsymbol{r}_{\xi^i}$, $i = 1, 2, 3$. Taking into account (2.26), we readily obtain the following formula for the differentiation of the cross product of two three-dimensional vector-valued functions $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$\frac{\partial}{\partial \xi^i}(\boldsymbol{a} \times \boldsymbol{b}) = \frac{\partial}{\partial \xi^i}\boldsymbol{a} \times \boldsymbol{b} + \boldsymbol{a} \times \frac{\partial}{\partial \xi^i}\boldsymbol{b}\ , \qquad i = 1, 2, 3\ .$$

With this formula, we obtain

$$\sum_{i=1}^{3}\frac{\partial}{\partial \xi^i}(\boldsymbol{x}_{\xi^j} \times \boldsymbol{x}_{\xi^k}) = \sum_{i=1}^{3}\boldsymbol{x}_{\xi^j\xi^i} \times \boldsymbol{x}_{\xi^k} + \sum_{i=1}^{3}\boldsymbol{x}_{\xi^j} \times \boldsymbol{x}_{\xi^k\xi^i}\ , \tag{2.50}$$

where the indices $(i,\ j,\ k)$ are cyclic, i.e. $j = i + 1,\ k = i + 2$, m is equivalent to $m + 3$. For the last summation of the above formula, we obtain

$$\sum_{i=1}^{3} \boldsymbol{x}_{\xi^j} \times \boldsymbol{x}_{\xi^k \xi^i} = \sum_{i=1}^{3} \boldsymbol{x}_{\xi^k} \times \boldsymbol{x}_{\xi^i \xi^j} \ .$$

Therefore, from (2.50),

$$\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} (\boldsymbol{x}_{\xi^j} \times \boldsymbol{x}_{\xi^k}) = 0 \ ,$$

since

$$\boldsymbol{x}_{\xi^i} \times \boldsymbol{x}_{\xi^j \xi^k} = -\boldsymbol{x}_{\xi^j \xi^k} \times \boldsymbol{x}_{\xi^i}$$

and (2.33) implies (2.48) for $n = 3$.

The identity (2.48) can help one to obtain conservative or compact forms of some differential expressions and equations in the curvilinear coordinates $\xi^1, \dots, \xi^n$. For example, for the first derivative of a function $f(\boldsymbol{x})$ with respect to x^i, we obtain, using (2.48),

$$\frac{\partial f}{\partial x^i} = \frac{1}{J} \frac{\partial}{\partial \xi^j} \Big(J \frac{\partial \xi^j}{\partial x^i} f \Big) \ , \qquad j = 1, \dots, n \ . \tag{2.51}$$

For the Laplacian

$$\nabla^2 f = \frac{\partial}{\partial x^j} \frac{\partial f}{\partial x^j} \ , \qquad j = 1, \dots, n \tag{2.52}$$

we have, substituting the quantity $\partial f / \partial x^i$ for f in (2.51),

$$\begin{aligned} \nabla^2 f &= \frac{1}{J} \frac{\partial}{\partial \xi^j} \Big(J \frac{\partial \xi^j}{\partial x^i} \frac{\partial f}{\partial x^i} \Big) = \frac{1}{J} \frac{\partial}{\partial \xi^j} \Big(J \frac{\partial \xi^j}{\partial x^i} \frac{\partial \xi^m}{\partial x^i} \frac{\partial f}{\partial \xi^m} \Big) \\ &= \frac{1}{J} \frac{\partial}{\partial \xi^j} \Big(J g^{mj} \frac{\partial f}{\partial \xi^m} \Big) \ , \qquad i, j, m = 1, \dots, n \ . \end{aligned} \tag{2.53}$$

Therefore, the Poisson equation

$$\nabla^2 f = P \tag{2.54}$$

has the form

$$\frac{1}{J} \frac{\partial}{\partial \xi^j} \Big(J g^{mj} \frac{\partial f}{\partial \xi^m} \Big) = P \ , \qquad j, m = 1, \dots, n \ , \tag{2.55}$$

with respect to the independent variables $\xi^1, \dots, \xi^n$.

2.4 Conservation Laws

This section utilizes the relations described in Sects. 2.2 and 2.3, in particular the identity (2.48), in order to describe some conservation-law equations of mechanics in divergent or compact form in new independent curvilinear coordinates $\xi^1, \dots, \xi^n$. For this purpose, the dependent physical variables are also transformed to new dependent variables using some specific formulas. The essential advantage of the equations described here is that their coefficients are derived from the elements of the covariant metric tensor $\{g_{ij}\}$.

2.4.1 Scalar Conservation Laws

Let $\boldsymbol{A}$ be an n-dimensional vector with components A^i, $i = 1, \dots, n$, in the Cartesian coordinates $x^1, \dots, x^n$. The operator

$$\mathrm{div}_x \boldsymbol{A} = \frac{\partial A^i}{\partial x^i} , \qquad i = 1, \dots, n , \tag{2.56}$$

is commonly used in mechanics for the representation of scalar conservation laws, commonly in the form

$$\mathrm{div}_x \boldsymbol{A} = F .$$

Using (2.48), we easily obtain

$$\mathrm{div}_x \boldsymbol{A} = \frac{1}{J} \frac{\partial}{\partial \xi^j} (J \overline{A}^j) = F , \qquad j = 1, \dots, n , \tag{2.57}$$

where $\overline{A}^j$ is the jth contravariant component of the vector $\boldsymbol{A}$ in the coordinates ξ^i, $i = 1, \dots, n$, i.e. in accordance with (2.8):

$$\overline{A}^j = A^i \frac{\partial \xi^j}{\partial x^i} , \qquad i, j = 1, \dots, n . \tag{2.58}$$

Therefore, a divergent form of the conservation-law equation represented by (2.56) is obtained in the new coordinates when the dependent variables A^i are replaced by new dependent variables $\overline{A}^i$ defined by the rule (2.58). Some examples of scalar conservation-law equations are given below.

Mass Conservation Law

As an example of the application of (2.57), we consider the equation of conservation of mass for steady gas flow

$$\frac{\partial \rho u^i}{\partial x^i} = 0\ , \qquad i = 1, \dots, n\ , \tag{2.59}$$

where ρ is the gas density, and u^i is the ith component of the flow velocity vector $\boldsymbol{u}$ in the Cartesian coordinates $x^1, \dots, x^n$. With the substitution $A^i = \rho u^i$, (2.58) is transformed into the following divergent form with respect to the new dependent variables ρ and $\overline{u}^i$ in the coordinates $\xi^1, \dots, \xi^n$:

$$\frac{\partial}{\partial \xi^i}(J\rho\overline{u}^i) = 0\ , \qquad i = 1, \dots, n\ . \tag{2.60}$$

Here, $\overline{u}^i$ is the ith contravariant component of the flow velocity vector $\boldsymbol{u}$ in the basis $\boldsymbol{x}_{\xi^i},\ i = 1, \dots, n$, i.e.

$$\overline{u}^i = u^j \frac{\partial \xi^i}{\partial x^j}\ , \qquad i, j = 1, \dots, n\ . \tag{2.61}$$

Convection–Diffusion Equation

Another example is the conservation equation for the steady convection–diffusion of a transport variable ϕ, which can be expressed as

$$-\frac{\partial}{\partial x^i}\Big(\epsilon\frac{\partial \phi}{\partial x^i}\Big) + \frac{\partial}{\partial x^i}(\rho\phi u^i) = S\ , \qquad i = 1, \dots, n\ , \tag{2.62}$$

where ρ and ϵ denote the density and diffusion coefficient of the fluid, respectively. Taking

$$A^i = \rho\phi u^i - \epsilon\frac{\partial \phi}{\partial x^i}, \qquad i = 1, \dots, n\ .$$

we obtain, in accordance with the relation (2.58),

$$\overline{A}^j = \rho\phi\overline{u}^j - \epsilon\frac{\partial \phi}{\partial \xi^k}g^{kj}\ , \qquad j, k = 1, \dots, n\ .$$

Therefore, using (2.57), the convection–diffusion equation (2.62) in the curvilinear coordinates $\xi^1, \dots, \xi^n$ is expressed by the divergent form

$$\frac{\partial}{\partial \xi^j}\Big[J\Big(\rho\phi\overline{u}^j - \epsilon g^{kj}\frac{\partial \phi}{\partial \xi^k}\Big)\Big] = JS\ , \qquad j, k = 1, \dots, n\ . \tag{2.63}$$

Laplace Equation

Analogously, the Laplace equation

$$\nabla^2 f = \frac{\partial}{\partial x^j}\frac{\partial f}{\partial x^j} = 0\ , \qquad j = 1, \dots, n\ , \tag{2.64}$$

has the form (2.56) if we take

$$A^i = \frac{\partial f}{\partial x^i} \;, \qquad i = 1, \dots, n \;.$$

Using (2.58), we obtain

$$\overline{A}^j = g^{ij} \frac{\partial f}{\partial \xi^i} \;, \qquad i = 1, \dots, n \;.$$

Therefore, the Laplace equation (2.64) results in

$$\nabla^2 f = \frac{1}{J} \frac{\partial}{\partial \xi^i} \Big(J g^{ij} \frac{\partial f}{\partial \xi^j} \Big) = 0 \;, \qquad i, j = 1, \dots, n \;, \tag{2.65}$$

since (2.57) applies.

2.4.2 Vector Conservation Laws

Many physical problems are also modeled as a system of conservation-law equations in the vector form

$$\frac{\partial A^{ij}}{\partial x^j} = F^i \;, \qquad i, j = 1, \dots, n \;. \tag{2.66}$$

For the representation of the system (2.66) in new coordinates $\xi^1, \dots, \xi^n$ in a form which includes only coefficients derived from the elements of the metric tensor, it is necessary to make a transition from the original expression for A^{ij} to a new one $\overline{A}^{ij}$. One convenient formula for such a transition from the dependent variables A^{ij} to $\overline{A}^{ij}$, $i, j = 1, \dots, n$, is

$$\overline{A}^{ij} = A^{km} \frac{\partial \xi^i}{\partial x^k} \frac{\partial \xi^j}{\partial x^m} \;, \qquad i, j, k, m = 1, \dots, n \;. \tag{2.67}$$

This relation between A^{ij} and $\overline{A}^{km}$ is, in fact, composed of transitions of the kind (2.58) for the rows and columns of the tensor A^{ij}. In tensor analysis, the quantity $\overline{A}^{ij}$ means the (i, j) component of the second-rank contravariant tensor (A^{ij}) in the coordinates $\xi^1, \dots, \xi^n$.

Multiplying (2.67) by $(\partial x^p/\partial \xi^i)(\partial x^l/\partial \xi^j)$ and summing over i and j, we also obtain a formula for the transition from the new dependent variables $\overline{A}^{ij}$ to the original ones A^{ij}:

$$A^{ij} = \overline{A}^{km} \frac{\partial x^i}{\partial \xi^k} \frac{\partial x^j}{\partial \xi^m} \;, \qquad i, j, k, m = 1, \dots, n \;. \tag{2.68}$$

Therefore, we can obtain a system of equations for the new dependent variables $\overline{A}^{ij}$ by replacing the dependent quantities A^{ij} in (2.66) with their expressions (2.68). As a result, we obtain

$$\begin{aligned}\frac{\partial A^{ij}}{\partial x^j} &= \frac{\partial}{\partial x^j}\Big(\overline{A}^{km}\frac{\partial x^i}{\partial \xi^k}\frac{\partial x^j}{\partial \xi^m}\Big)\\ &= \frac{\partial \overline{A}^{km}}{\partial \xi^m}\frac{\partial x^i}{\partial \xi^k} + \overline{A}^{km}\frac{\partial^2 x^i}{\partial \xi^k \partial \xi^m} + \overline{A}^{km}\frac{\partial x^i}{\partial \xi^k}\frac{\partial}{\partial x^j}\Big(\frac{\partial x^j}{\partial \xi^m}\Big) = F^i\ ,\\ &\quad i, j, k, m = 1, \dots, n\ .\end{aligned}$$

The use of the formula (2.46) for differentiation of the Jacobian in the summation in the equation above yields

$$\frac{\partial A^{ij}}{\partial x^j} = \frac{\partial \overline{A}^{km}}{\partial \xi^m}\frac{\partial x^i}{\partial \xi^k} + \overline{A}^{km}\frac{\partial^2 x^i}{\partial \xi^k \partial \xi^m} + \frac{1}{J}\overline{A}^{km}\frac{\partial x^i}{\partial \xi^k}\frac{\partial J}{\partial \xi^m} = F^i\ ,$$

$$i, j, k, l, m = 1, \dots, n\ .$$

Multiplying this system by $\partial \xi^p / \partial x^i$ and summing over i, we obtain, after simple manipulations,

$$\frac{1}{J}\frac{\partial}{\partial \xi^j}(J\overline{A}^{ij}) + \frac{\partial^2 x^l}{\partial \xi^k \partial \xi^j}\frac{\partial \xi^i}{\partial x^l}\overline{A}^{kj} = \overline{F}^i\ , \qquad i, j, k, l = 1, \dots, n\ , \tag{2.69}$$

where

$$\overline{F}^i = F^j\frac{\partial \xi^i}{\partial x^j}\ , \qquad i, j = 1, \dots, n\ ,$$

is the ith contravariant component of the vector $\boldsymbol{F} = (F^1, \dots, F^n)$ in the basis $\boldsymbol{x}_{\xi^1}, \dots, \boldsymbol{x}_{\xi^n}$. The quantities $(\partial^2 x^l / \partial \xi^k \partial \xi^j)(\partial \xi^i / \partial x^l)$ in (2.69) are the space Christoffel symbols of the second kind Γ^i_{kj}. Thus, the system (2.69) has, using the notation Γ^i_{jk}, the form

$$\frac{1}{J}\frac{\partial}{\partial \xi^j}(J\overline{A}^{ij}) + \Gamma^i_{kj}\overline{A}^{kj} = \overline{F}^i\ , \qquad i, j, k = 1, \dots, n\ . \tag{2.70}$$

We see that all coefficients of (2.70) are derived from the metric tensor $\{g_{ij}\}$.

Equations of the form (2.70), in contrast to (2.66), do not have a conservative form. The conservative form of (2.66) in new dependent variables is obtained, in analogy with (2.57), from the system

$$\frac{1}{J}\frac{\partial}{\partial \xi^j}(J\overline{A}^j_i) = F^i\ , \qquad i, j = 1, \dots, n\ , \tag{2.71}$$

where $\overline{A}_i^j$ is the jth component of the vector $\boldsymbol{A}_i = (A^{i1}, \dots, A^{in})$ in the basis $\boldsymbol{x}_{\xi^j},\ j = 1, \dots, n$, i.e.

$$\overline{A}_i^j = A^{ik}\frac{\partial \xi^j}{\partial x^k}\ , \qquad i, j, k = 1, \dots, n\ . \tag{2.72}$$

In fact, (2.71) is the result of the application of (2.57) to the ith line of (2.66). Therefore, in the relations (2.66, 2.71, 2.72), we can assume an arbitrary range for the index i, i.e. the matrix A^{ij} in (2.66) can be a nonsquare matrix with $i = 1, \dots, m,\ j = 1, \dots, n$.

Though the system (2.71) is conservative and more compact than (2.70), it has its drawbacks. In particular, mathematical simulations of fluid flows are generally formulated in the form (2.66) with the tensor A^{ij} represented as

$$A^{ij} = B^{ij} + \rho u^i u^j\ , \qquad i, j = 1, \dots, n\ ,$$

where $u^i,\ i = 1, \dots, n$, are the Cartesian components of the flow velocity. The transformation of the tensor $\rho u^i u^j$ by the rule (2.72),

$$\rho u^i u^k \frac{\partial \xi^j}{\partial x^k} = \rho u^i \overline{u}^j\ , \qquad i, j, k = 1, \dots, n\ ,$$

results in equations with an increased number of dependent variables, namely u^i and $\overline{u}^j$. The substitution of u^i for $\overline{u}^j$ or vice versa leads to equations whose coefficients are derived from the elements $\partial x^i/\partial \xi^j$ of the Jacobi matrix and not from the elements of the metric tensor $\{g_{ij}\}$.

Example

As an example of (2.66), we consider the stationary equation of a compressible gas flow

$$\frac{\partial}{\partial x^j}(\rho u^i u^j) + \frac{\partial p}{\partial x^i} - \frac{\partial}{\partial x^j}\mu\frac{\partial u^i}{\partial x^j} = \rho F^i\ , \qquad i, j = 1, \dots, n\ , \tag{2.73}$$

where u^i is the ith Cartesian component of the vector of the fluid velocity $\boldsymbol{u}$, ρ is the density, p is the pressure and μ is the viscosity. The tensor form of (2.66) is given by

$$A^{ij} = \rho u^i u^j + \delta^i_j p - \mu\frac{\partial u^i}{\partial x^j}\ , \qquad i, j = 1, \dots, n\ .$$

From (2.67), we obtain, in this case,

$$\overline{A}^{ij} = \rho \overline{u}^i \overline{u}^j + g^{ij} p - \mu\frac{\partial u^l}{\partial x^k}\frac{\partial \xi^i}{\partial x^l}\frac{\partial \xi^j}{\partial x^k}\ , \qquad i, j, k, l = 1, \dots, n\ , \tag{2.74}$$

where $\overline{u}^i$ is the ith component of $\boldsymbol{u}$ in the basis $\boldsymbol{x}_{\xi^i}$, i.e. $\overline{u}^i$ is computed from the formula (2.61). It is obvious that

$$u^l = \overline{u}^j \frac{\partial x^l}{\partial \xi^j}\,, \qquad j, l = 1, \ldots, n\,. \tag{2.75}$$

Therefore,

$$\begin{aligned}\frac{\partial u^l}{\partial x^k} &= \frac{\partial}{\partial \xi^m}\Big(\overline{u}^p \frac{\partial x^l}{\partial \xi^p}\Big)\frac{\partial \xi^m}{\partial x^k}\\ &= \frac{\partial \overline{u}^p}{\partial \xi^m}\frac{\partial x^l}{\partial \xi^p}\frac{\partial \xi^m}{\partial x^k} + \overline{u}^p \frac{\partial^2 x^l}{\partial \xi^p \partial \xi^m}\frac{\partial \xi^m}{\partial x^k}\,, \qquad k, l, m, p = 1, \ldots, n\,.\end{aligned}$$

Using this equation, we obtain, for the last term of (2.74),

$$\mu \frac{\partial u^l}{\partial x^k}\frac{\partial \xi^i}{\partial x^l}\frac{\partial \xi^j}{\partial x^k} = \mu g^{mj}\Big(\frac{\partial \overline{u}^i}{\partial \xi^m} + \Gamma^i_{pm}\overline{u}^p\Big)\,, \qquad i, j, m, p = 1, \ldots, n\,,$$

since (2.38) applies. Thus, (2.74) has the form

$$\overline{A}^{ij} = \rho \overline{u}^i \overline{u}^j + g^{ij} p - \mu g^{mj}\Big(\frac{\partial \overline{u}^i}{\partial \xi^m} + \Gamma^i_{pm}\overline{u}^p\Big)\,, \quad i, j, m, p = 1, \ldots, n\,, \tag{2.76}$$

and, applying (2.70), we obtain the following system of stationary equations (2.73) with respect to the new dependent variables $\rho, \overline{u}^i$, and p and the independent variables ξ^i:

$$\begin{aligned}&\frac{1}{J}\frac{\partial}{\partial \xi^j}\Big\{J\Big[\rho \overline{u}^i \overline{u}^j + g^{ij} p - \mu g^{mj}\Big(\frac{\partial \overline{u}^i}{\partial \xi^m} + \Gamma^i_{pm}\overline{u}^p\Big)\Big]\Big\}\\ &+ \Gamma^i_{kj}\Big[\rho \overline{u}^k \overline{u}^j + g^{kj} p - \mu g^{mj}\Big(\frac{\partial \overline{u}^k}{\partial \xi^m} + \Gamma^k_{pm}\overline{u}^p\Big)\Big] = \rho \overline{F}^i\,,\\ &i, j, k, m, p = 1, \ldots, n\,.\end{aligned} \tag{2.77}$$

The application of (2.71)–(2.73) yields the following system of stationary equations:

$$\begin{aligned}&\frac{1}{J}\frac{\partial}{\partial \xi^j}\Big[J\Big(\rho u^i \overline{u}^j + \frac{\partial \xi^j}{\partial x^i} p - \mu \frac{\partial u^i}{\partial \xi^k} g^{kj}\Big)\Big] = \rho F^i\,,\\ &\overline{u}^j = u^k \frac{\partial \xi^j}{\partial x^k}\,, \qquad i, j, k = 1, \ldots, n\,.\end{aligned} \tag{2.78}$$

Now, as an example of the utilization of the Christoffel symbols of the second kind Γ^i_{kj}, we write out the expression for the transformed elements of the tensor

$$\sigma^{ij} = \mu\Big(\frac{\partial u^i}{\partial x^j} + \frac{\partial u^j}{\partial x^i}\Big) , \qquad i, j = 1 \ldots , n , \tag{2.79}$$

in the coordinates $\xi^1, \ldots, \xi^n$, obtained in accordance with the rule (2.67). This tensor is very common and is important in applications simulating deformation in the theory of elasticity and deformation rate in fluid mechanics. Using the notations described above, the tensor $\overline{\sigma}^{ij}$ can be expressed in the coordinates $\xi^1, \ldots, \xi^n$ through the metric elements and the Christoffel symbols of the second kind. For the component $\overline{\sigma}^{ij}$, we have

$$\begin{aligned} \overline{\sigma}^{ij} &= \sigma^{mk} \frac{\partial \xi^i}{\partial x^m} \frac{\partial \xi^j}{\partial x^k} \\ &= \mu\Big(g^{jl}\frac{\partial \overline{u}^i}{\partial \xi^l} + g^{il}\frac{\partial \overline{u}^j}{\partial \xi^l} + (g^{jl}\Gamma^i_{pl} + g^{il}\Gamma^j_{pl})\overline{u}^p\Big) , \\ & i, j, l, p = 1, \ldots , n . \end{aligned} \tag{2.80}$$

This formula is obtained rather easily. For this purpose, one can use the relation (2.75) for the inverse transition from the contravariant components $\overline{u}^i$ to the Cartesian components u^j of the vector $\boldsymbol{u} = (u^1, \ldots, u^n)$ and the formula (2.38). By substituting (2.75) in (2.79), carrying out differentiation by the chain rule, and using the expression (2.38), we obtain (2.80).

2.5 Time-Dependent Transformations

The numerical solution of time-dependent equations requires the application of moving grids and the corresponding coordinate transformations, which are dependent on time. Commonly, such coordinate transformations are determined in the form of a vector-valued time-dependent function

$$\boldsymbol{x}(t, \boldsymbol{\xi}) : \Xi^n \to X^n_t , \qquad \boldsymbol{\xi} \in \Xi^n , \qquad t \in [0, 1] , \tag{2.81}$$

where the variable t represents the time and X^n_t is an n-dimensional domain whose boundary points change smoothly with respect to t. It is assumed that $\boldsymbol{x}(t, \boldsymbol{\xi})$ is sufficiently smooth with respect to ξ^i and t and, in addition, that it is invertible for all $t \in [0, 1]$. Therefore, there is also the time-dependent inverse transformation

$$\boldsymbol{\xi}(t, \boldsymbol{x}) : X^n_t \to \Xi^n \tag{2.82}$$

for every $t \in [0, 1]$. The introduction of these time-dependent coordinate transformations enables one to compute an unsteady solution on a fixed uniform grid in Ξ^n by the numerical solution of the transformed equations.

2.5.1 Reformulation of Time-Dependent Transformations

Many physical problems are modeled in the form of nonstationary conservation-law equations which include the time derivative. The formulas of Sects. 2.3 and 2.4 can be used directly, by transforming the equations at every value of time t. However, such utilization of the formulas does not influence the temporal derivative, which is transformed simply to the form

$$\frac{\partial}{\partial t}+\frac{\partial \xi^i}{\partial t}\frac{\partial}{\partial \xi^i}\ , \qquad i=1,\dots,n\ ,$$

so that it does not maintain the property of divergency and its coefficients are not derived from the elements of the metric tensor.

Instead, the formulas of Sects. 2.3 and 2.4 can be more successfully applied to time-dependent conservation-law equations if the set of the functions $\boldsymbol{x}(t,\boldsymbol{\xi})$ is expanded to an $(n+1)$-dimensional coordinate transformation in which the temporal parameter t is considered in the same manner as the spatial variables.

To carry out this process, we expand the n-dimensional computational and physical domains in (2.81) to $(n+1)$-dimensional ones, assuming

$$\Xi^{n+1}=I\times\Xi^n\ , \qquad X^{n+1}=\cup_t(t\times X^n_t)\ .$$

Let the points of these domains be designated by $\boldsymbol{\xi}_0=(\xi^0,\xi^1,\dots,\xi^n)$ and $\boldsymbol{x}_0=(x^0,x^1,\dots,x^n)$, respectively. The expanded coordinate transformation is defined as

$$\boldsymbol{x}_0(\boldsymbol{\xi}_0):\Xi^{n+1}\to X^{n+1}\ , \tag{2.83}$$

where $x^0(\boldsymbol{\xi}_0)=\xi^0$ and $x^i(\boldsymbol{\xi}_0)$, $i=1,\dots,n$, which coincides with (2.81) with $\xi^0=t$.

The variables x^0 and ξ^0 in (2.83) represent, in fact, the temporal variable t. For convenience and in order to avoid ambiguity, we shall also designate the variable ξ^0 in Ξ^{n+1} by τ and the variable x^0 in X^{n+1} by t. Thus, $\boldsymbol{x}_0(\boldsymbol{\xi}_0)$ is the $(n+1)$-dimensional coordinate transformation which is identical to $\boldsymbol{x}(\tau,\boldsymbol{\xi})$ at every section $\xi^0=\tau$.

The inverted coordinate transformation

$$\boldsymbol{\xi}_0(\boldsymbol{x}_0):X^{n+1}\to\Xi^{n+1} \tag{2.84}$$

satisfies

$$\xi^0(\boldsymbol{x}_0)=x^0\ , \qquad \xi^i(\boldsymbol{x}_0)=\xi^i(t,\boldsymbol{x})\ , \qquad i=1,\dots,n\ ,$$

where $t=x^0$, $\boldsymbol{x}=(x^1,\dots,x^n)$, and $\xi^i(t,\boldsymbol{x})$ is defined by (2.82). Thus, (2.84) is identical to (2.82) at each section X_t.

2.5.2 Basic Relations

This subsection discusses some relations and, in particular, identities of the kind (2.46) and (2.48) for the time-dependent coordinate transformations (2.81), using for this purpose the $(n+1)$-dimensional vector functions (2.83) and (2.84) introduced above.

Velocity of Grid Movement

The first derivative $\boldsymbol{x}_\tau$, $\boldsymbol{x} = (x^1, x^2, \dots, x^n)$, of the transformation $\boldsymbol{x}(\xi, \tau)$ has a clear physical interpretation as the velocity vector of grid point movement. Let the vector $\boldsymbol{x}_\tau$, in analogy with the flow velocity vector $\boldsymbol{u}$, be designated by $\boldsymbol{w} = (w^1, \dots, w^n)$, i.e. $w^i = x^i_\tau$. The ith component $\overline{w}^i$ of the vector $\boldsymbol{w}^i$ in the tangential bases $\boldsymbol{x}_{\xi^i}$, $i = 1, \dots, n$, is expressed by (2.7) as

$$\overline{w}^i = w^j \frac{\partial \xi^i}{\partial x^j} = \frac{\partial x^j}{\partial \tau} \frac{\partial \xi^i}{\partial x^j} , \qquad i, j = 1, \dots, n .$$

Therefore,

$$\boldsymbol{w} = \overline{w}^i \boldsymbol{x}_{\xi^i} , \qquad i = 1, \dots, n , \tag{2.85}$$

i.e.

$$w^i = \frac{\partial x^i}{\partial \tau} = \overline{w}^j \frac{\partial x^i}{\partial \xi^j} , \qquad i, j = 1, \dots, n .$$

Differentiation with respect to ξ^0 of the composition of $\boldsymbol{x}_0(\boldsymbol{\xi}_0)$ and $\boldsymbol{\xi}_0(\boldsymbol{x}_0)$ yields

$$\frac{\partial \xi^i}{\partial x^0} \frac{\partial x^0}{\partial \xi^0} + \frac{\partial \xi^i}{\partial x^j} \frac{\partial x^j}{\partial \xi^0} = 0 , \qquad i, j = 1, \dots, n .$$

Therefore, we obtain the result

$$\frac{\partial \xi^i}{\partial t} = -\frac{\partial x^j}{\partial \tau} \frac{\partial \xi^i}{\partial x^j} = -\overline{w}^i , \qquad i, j = 1, \dots, n . \tag{2.86}$$

Derivatives of the Jacobian

It is apparent that the Jacobians of the coordinate transformations $\boldsymbol{x}(\tau, \boldsymbol{\xi})$ and $\boldsymbol{x}_0(\boldsymbol{\xi}_0)$ coincide, i.e.

$$\det\left\{\frac{\partial x^i}{\partial \xi^j}\right\} = \det\left\{\frac{\partial x^k}{\partial \xi^l}\right\} = J , \qquad i, j = 0, 1, \dots, n , \qquad k, l = 1, \dots, n .$$

In the notation introduced above, the formula (2.46) for differentiation of the Jacobian of the transformation

$$\boldsymbol{x}_0(\boldsymbol{\xi}_0) : \Xi^{n+1} \to X^{n+1}$$

is expressed by the relation

$$\frac{1}{J}\frac{\partial}{\partial \xi^i}J = \frac{\partial^2 x^k}{\partial \xi^i \partial \xi^m}\frac{\partial \xi^m}{\partial x^k}\ , \qquad i, k, m = 0, 1, \ldots, n\ , \tag{2.87}$$

differing from (2.46) only by the range of the indices. As a result, we obtain from (2.87) for $i = 0$,

$$\frac{1}{J}\frac{\partial}{\partial \tau}J = \frac{\partial}{\partial \xi^m}\Big(\frac{\partial x^k}{\partial \tau}\Big)\frac{\partial \xi^m}{\partial x^k} = \mathrm{div}_x \frac{\partial \boldsymbol{x}}{\partial \tau}\ , \qquad k, m = 0, 1, \ldots, n\ , \tag{2.88}$$

and, taking into account (2.85),

$$\begin{aligned}\frac{1}{J}\frac{\partial}{\partial \tau}J &= \frac{\partial}{\partial \xi^m}\Big(\overline{w}^j \frac{\partial x^k}{\partial \xi^j}\Big)\frac{\partial \xi^m}{\partial x^k} \\ &= \frac{\partial \overline{w}^m}{\partial \xi^m} + \overline{w}^i \frac{\partial^2 x^k}{\partial \xi^j \partial \xi^m}\frac{\partial \xi^m}{\partial x^k}\ , \qquad j, k, m = 1, \ldots, n\ .\end{aligned}$$

Now, taking advantage of the formula for differentiation of the Jacobian (2.46), in the last sum of this equation, we have

$$\frac{1}{J}\frac{\partial}{\partial \tau}J = \frac{\partial \overline{w}^m}{\partial \xi^m} + \frac{1}{J}\overline{w}^j \frac{\partial J}{\partial \xi^j}\ , \qquad j, m = 1, \ldots, n\ ,$$

and consequently

$$\frac{1}{J}\frac{\partial}{\partial \tau}J = \frac{1}{J}\frac{\partial}{\partial \xi^j}(J\overline{w}^j)\ , \qquad j = 1, \ldots, n\ . \tag{2.89}$$

Basic Identity

Analogously, the system of identities (2.48) has the following form:

$$\frac{\partial}{\partial \xi^j}\left(J\frac{\partial \xi^j}{\partial x^i}\right) = 0\ , \qquad i, j = 0, 1, \ldots, n\ . \tag{2.90}$$

Therefore, for $i = 0$, we obtain

$$\frac{\partial}{\partial \tau}(J) + \frac{\partial}{\partial \xi^j}\left(J\frac{\partial \xi^j}{\partial t}\right) = 0\ , \qquad j = 1, \ldots, n\ , \tag{2.91}$$

and, taking into account (2.86),

$$\frac{\partial}{\partial \tau}J - \frac{\partial}{\partial \xi^j}(J\overline{w}^j) = 0\ , \qquad j = 1, \ldots, n\ , \tag{2.92}$$

which corresponds to (2.89). For $i > 0$, the identity (2.90) coincides with (2.48), i.e.

$$\frac{\partial}{\partial \xi^j}\left(J\frac{\partial \xi^j}{\partial x^i}\right) = 0\ , \qquad i, j = 1, \dots, n\ .$$

As a result of (2.91), we obtain, in analogy with (2.51),

$$\frac{\partial f}{\partial t} = \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(J\frac{\partial \xi^j}{\partial t}f\right) = \frac{1}{J}\left(\frac{\partial}{\partial \tau}(Jf) - \frac{\partial}{\partial \xi^k}(J\overline{w}^k f)\right)\ ,$$
$$j = 0, 1, \dots, n\ , \qquad k = 1, \dots, n\ . \tag{2.93}$$

2.5.3 Equations in the Form of Scalar Conservation Laws

Many time-dependent equations can be expressed in the form of a scalar conservation law in the Cartesian coordinates $t, x^1, \dots, x^n$:

$$\frac{\partial A^0}{\partial t} + \frac{\partial A^i}{\partial x^i} = F\ , \qquad i = 1, \dots, n\ . \tag{2.94}$$

Using (2.90), in analogy with (2.57), this equation is transformed in the coordinates $\xi^0, \xi^1, \dots, \xi^n,\ \xi^0 = \tau$ to

$$\frac{1}{J}\Big(\frac{\partial}{\partial \xi^j}(J\overline{A}_0^j)\Big) = F\ , \qquad j = 0, 1, \dots, n\ , \tag{2.95}$$

where by $\overline{A}_0^j$ we denote the jth contravariant component of the $(n+1)$-dimensional vector $\boldsymbol{A}_0 = (A^0, A^1, \dots, A^n)$ in the basis $\partial \boldsymbol{x}_0/\partial \xi^i,\ i = 0, 1, \dots, n$, i.e.

$$\overline{A}_0^j = A^i\frac{\partial \xi^j}{\partial x^i}\ , \qquad i, j = 0, 1, \dots, n\ . \tag{2.96}$$

We can express each component $\overline{A}_0^j,\ j = 1, \dots, n$, of the vector $\boldsymbol{A}_0$ through the components $\overline{A}^i$ and $\overline{w}^k,\ i, k = 1, \dots, n$, of the n-dimensional spatial vectors $\boldsymbol{A} = (A^1, \dots, A^n)$ and $\boldsymbol{w} = (w^1, \dots, w^n)$ in the coordinates $\xi^l,\ l = 1, \dots, n$, where $\boldsymbol{A}$ is a vector obtained by projecting the vector $\boldsymbol{A}_0$ into the space R^n, i.e. $P(A^0, A^1, \dots, A^n) = (A^1, \dots, A^n)$. Namely,

$$\overline{A}_0^j = A^0\frac{\partial \xi^j}{\partial t} + A^i\frac{\partial \xi^j}{\partial x^i} = \overline{A}^j - A^0\overline{w}^j\ , \qquad i, j = 1, \dots, n\ ,$$

using (2.86). Furthermore, we have

$$\overline{A}_0^0 = A^k \frac{\partial \xi^0}{\partial x^k} = A^0 \ , \qquad k = 0, 1, \dots, n \ .$$

Therefore, (2.95) implies a conservation law in the variables $\tau, \xi^1, \dots, \xi^n$ in the conservative form

$$\frac{1}{J}\left(\frac{\partial}{\partial \tau}(JA^0) + \frac{\partial}{\partial \xi^j}[J(\overline{A}^j - A^0 \overline{w}^j)]\right) = F \ , \qquad j = 1, \dots, n \ . \qquad (2.97)$$

Examples of Scalar Conservation-Law Equations

As an illustration of the formula (2.97), we write out some time-dependent scalar conservation-law equations presented first in the form (2.94).

Parabolic Equation

For the parabolic equation

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial x^j}\frac{\partial f}{\partial x^j} \ , \qquad j = 1, \dots, n \ , \qquad (2.98)$$

we obtain from (2.97), with $A^0 = f$ and $A^i = \partial f / \partial x^i$, $i = 1, \dots n$,

$$\frac{\partial J f}{\partial \tau} = \frac{\partial}{\partial \xi^j}\left[J\left(g^{jk}\frac{\partial f}{\partial \xi^k} + f\overline{w}^j\right)\right] , \qquad j, k = 1, \dots n \ . \qquad (2.99)$$

Mass Conservation Law

The scalar mass conservation law for unsteady compressible gas flow

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u^i}{\partial x^i} = F \ , \qquad i = 1, 2, 3 \ , \qquad (2.100)$$

is expressed in the new coordinates as

$$\frac{\partial J\rho}{\partial t} + \frac{\partial J\rho(\overline{u}^j - \overline{w}^j)}{\partial \xi^j} = JF \ , \qquad j = 1, 2, 3 \ . \qquad (2.101)$$

Convection–Diffusion Equation

The unsteady convection–diffusion conservation equation

$$\frac{\partial}{\partial t}(\rho\phi) + \frac{\partial}{\partial x^i}(\rho\phi u^i) - \frac{\partial}{\partial x^i}\left(\epsilon\frac{\partial \phi}{\partial x^i}\right) = S \ , \qquad i = 1, \dots, n \ , \qquad (2.102)$$

has the form in the coordinates $\tau, \xi^1, \dots, \xi^n$

$$\frac{\partial}{\partial\tau}(J\rho\phi)+\frac{\partial}{\partial\xi^j}\Big(J\rho\phi(\overline{u}^j-\overline{w}^j)-Jg^{kj}\epsilon\frac{\partial\phi}{\partial\xi^k}\Big)=JS\ ,\ \ j,k=1,\dots,n\ . \tag{2.103}$$

Energy Conservation Law

Analogously, the energy conservation law

$$\frac{\partial}{\partial t}\rho(e+u^2/2)+\frac{\partial}{\partial x^j}\rho u^j(e+u^2/2+\overline{p}/\rho)=\rho F_j u_j\ ,\quad j=1,2,3\ , \tag{2.104}$$

where

$$e=e(\rho,p)\ ,\qquad \overline{p}=p-\gamma\frac{\partial u_i}{\partial x^i}\ ,\qquad i=1,2,3\ ,$$

$$u^2=\sum_{i=1}^{3}(u_i)^2\ ,$$

is transformed in accordance with (2.97) to

$$\frac{\partial}{\partial\tau}\Big[J\rho\Big(e+\frac{1}{2}g_{mk}\overline{u}^m\overline{u}^k\Big)\Big]+\frac{\partial}{\partial\xi^j}\Big[J\rho\Big(e+\frac{1}{2}g_{mk}\overline{u}^m\overline{u}^k\Big)(\overline{u}^j-\overline{w}^j)+J\overline{pu}^j\Big]$$

$$=J\rho g_{mk}\overline{f}^m\overline{u}^k\ ,\qquad j,m,k=1,2,3\ , \tag{2.105}$$

where, taking into account (2.57),

$$\overline{p}=p-\frac{\gamma}{J}\frac{\partial}{\partial\xi^i}(J\overline{u}^i)\ ,\qquad i=1,2,3\ .$$

Linear Wave Equation

The linear wave equation

$$u_{tt}=c^2\nabla^2 u \tag{2.106}$$

arises in many areas such as fluid dynamics, elasticity, acoustics, and magnetohydrodynamics. If the coefficient c is constant, then (2.106) has a divergent form (2.94) with

$$A^0=u_t\ ,\qquad A^i=-c^2\frac{\partial u}{\partial x^i}\ ,\qquad i=1,\dots,n\ ,$$

or, in the coordinates $\tau,\xi^1,\dots,\xi^n$,

$$A^0=u_\tau-\overline{w}^i\frac{\partial u}{\partial\xi^i}\ ,\qquad A^i=-c^2\frac{\partial u}{\partial\xi^k}\frac{\partial\xi^k}{\partial x^i}\ ,\qquad i,k=1,\dots,n\ .$$

Therefore, the divergent form (2.95) of (2.106) in the coordinates $\tau, \xi^1, \dots, \xi^n$ has the form

$$\frac{\partial}{\partial \tau}\left[J\left(u_\tau - \overline{w}^i \frac{\partial u}{\partial \xi^i}\right)\right] + \frac{\partial}{\partial \xi^j}\left[J\left(u_\tau \overline{w}^j + (c^2 g^{mj} - \overline{w}^i \overline{w}^j)\frac{\partial u}{\partial \xi^i}\right)\right] = 0 \,. \tag{2.107}$$

Another representation of the linear wave equation (2.106) in the coordinates $\tau, \xi^1, \dots, \xi^n$ comes from the formula (2.65) for the Laplace operator and the description of the temporal derivative (2.93). Taking advantage of (2.93), we obtain

$$\begin{aligned} u_{tt} &= \frac{1}{J}\Big[\frac{\partial}{\partial \tau}\Big(J\frac{\partial u}{\partial t}\Big) - \frac{\partial}{\partial \xi^k}\Big(J\overline{w}^k \frac{\partial u}{\partial t}\Big)\Big] \\ &= \frac{1}{J}\frac{\partial}{\partial \tau}\Big[J\Big(u_\tau - \overline{w}^i \frac{\partial u}{\partial \xi^i}\Big)\Big] - \frac{1}{J}\frac{\partial}{\partial \xi^k}\Big[J\overline{w}^k\Big(u_\tau - \overline{w}^i \frac{\partial u}{\partial \xi^i}\Big)\Big] \,, \quad i, k = 1, \dots, n \,. \end{aligned}$$

This equation and (2.65) allow one to derive the following form of (2.106) in the coordinates $\tau, \xi^1, \dots, \xi^n$:

$$\begin{aligned} \frac{\partial}{\partial \tau}\Big[J\Big(u_\tau - \overline{w}^i \frac{\partial u}{\partial \xi^i}\Big)\Big] &= \frac{\partial}{\partial \xi^k}\Big[J\overline{w}^k\Big(u_\tau - \overline{w}^j \frac{\partial u}{\partial \xi^j}\Big)\Big] \\ &+ c^2 \frac{\partial}{\partial \xi^k}\Big(J g^{kj}\frac{\partial u}{\partial \xi^j}\Big) \,, \quad i, j, k = 1, \dots, n \,, \end{aligned} \tag{2.108}$$

which coincides with (2.107) if c^2 is a constant.

Lagrangian Coordinates

One of the most popular systems of coordinates in fluid dynamics is the Lagrangian system. A coordinate ξ^i is Lagrangian if the both the ith component of the flow velocity vector $\boldsymbol{u}$ and the grid velocity $\boldsymbol{w}$ in the tangent basis $\boldsymbol{x}_{\xi^j}$, $j = 1, \dots, m$, coincide, i.e.

$$\overline{u}^i - \overline{w}^i = 0 \,. \tag{2.109}$$

The examples of gas-dynamics equations described above, which include the terms $\overline{w}^i$, allow one to obtain the equations in Lagrange coordinates by substituting $\overline{u}^i$ for $\overline{w}^i$ in the written-out equations in accordance with the relation (2.109). In such a manner, we obtain the equation of mass conservation, for example, in the Lagrangian coordinates $\xi^1, \dots, \xi^n$, as

$$\frac{\partial J\rho}{\partial \tau} = JF \tag{2.110}$$

from (2.101). Analogously, the convection–diffusion equation (2.103) and the energy conservation law (2.105) have the forms in Lagrangian coordinates ξ^i

$$\frac{\partial}{\partial \tau}(J\rho\phi) + \frac{\partial}{\partial \xi^j}\Big(J g^{kj}\epsilon \frac{\partial \phi}{\partial \xi^k}\Big) = JS \,, \qquad j, k = 1, \dots, n \,,$$

and

$$\frac{\partial}{\partial \tau}\Big[J\rho\Big(e+\frac{1}{2}g_{mk}\overline{u}^m\overline{u}^k\Big)\Big]+\frac{\partial}{\partial \xi^j}\Big(J\overline{pu}^j\Big)=J\rho g_{mk}\overline{f}^m\overline{u}^k\ , \qquad j,m,k=1,2,3\ ,$$

respectively.

In the same manner, the equations can be written in the Euler–Lagrange form. where some coordinates are Lagrangian while the rest are Cartesian coordinates.

2.5.4 Equations in the Form of Vector Conservation Laws

Now we consider a formula for a vector conservation law with time-dependent physical magnitudes A^{ij}

$$\frac{\partial}{\partial x^j}A^{ij}=F^i\ , \qquad i,j=0,1,\dots,n\ , \tag{2.111}$$

where the independent variable x^0 represents the time variable t, i.e. $x^0=t$. Let the new independent variables $\xi^0,\xi^1,\dots,\xi^n$ be obtained by means of (2.83). Along with (2.70), which expresses the vector conservation law (2.66) in the coordinates $\xi^1,\dots,\xi^n$, we find that the transformation (2.111) has the form of the following system of equations for the new dependent quantities $\overline{A}_0^{ij}$, $i,j=0,1,\dots,n$, with respect to the independent variables $\xi^0,\xi^1,\dots,\xi^n$, $\xi^0=\tau$:

$$\frac{1}{J}\frac{\partial}{\partial \xi^j}(J\overline{A}_0^{ij})+\overline{\Gamma}^i_{kj}\overline{A}_0^{kj}=\overline{F}_0^i\ , \qquad i,j=0,1,\dots,n\ , \tag{2.112}$$

where

$$\begin{aligned}\overline{A}_0^{ij}&=A^{mn}\frac{\partial \xi^i}{\partial x^m}\frac{\partial \xi^j}{\partial x^n}\ ,\ i,j,m,n=0,1,\dots,n\ ,\\ \overline{\Gamma}^i_{kj}&=\frac{\partial^2 x^l}{\partial \xi^k\partial \xi^j}\frac{\partial \xi^i}{\partial x^l}\ , \quad i,j,k,l=0,1,\dots,n\ ,\\ \overline{F}_0^i&=F^j\frac{\partial \xi^i}{\partial x^j}\ , \qquad i,j=0,1,\dots,n\ .\end{aligned}$$

As in the case of the scalar conservation law, we represent all of the terms of (2.112) through A^{00} and the spatial components:

$$
\begin{aligned}
\overline{A}^{ij} &= A^{km}\frac{\partial \xi^i}{\partial x^k}\frac{\partial \xi^j}{\partial x^m}\ , && i,j,m,n = 1,\dots,n\ ,\\
\Gamma^i_{kj} &= \frac{\partial^2 x^l}{\partial \xi^k \partial \xi^j}\frac{\partial \xi^i}{\partial x^l}\ , && i,j,k,l = 1,\dots,n\ ,\\
\overline{F}^i &= F^j\frac{\partial \xi^i}{\partial x^j}\ , && i,j = 1,\dots,n\ ,\\
\overline{w}^i &= -\frac{\partial \xi^i}{\partial t} = \frac{\partial x^j}{\partial \tau}\frac{\partial \xi^i}{\partial x^j}\ , && i,j = 1,\dots,n\ .
\end{aligned}
$$

For $\overline{A}^{ij}_0$, we obtain

$$
\begin{aligned}
\overline{A}^{00}_0 &= A^{00}\ ,\\
\overline{A}^{0i}_0 &= A^{00}\frac{\partial \xi^i}{\partial t} + A^{0m}\frac{\partial \xi^i}{\partial x^m} = \overline{A}^{0i} - A^{00}\overline{w}^i\ , && i = 1,\dots,n\ ,\\
\overline{A}^{i0}_0 &= \overline{A}^{i0} - A^{00}\overline{w}^i\ , && i = 1,\dots,n\ ,\\
\overline{A}^{ij}_0 &= A^{00}\overline{w}^i\overline{w}^j - \overline{A}^{0j}\overline{w}^i - \overline{A}^{i0}\overline{w}^j + \overline{A}^{ij}\ , && i,j = 1,\dots,n\ .
\end{aligned}
$$

Analogously, for $\overline{\Gamma}^i_{kj}$ we obtain

$$
\begin{aligned}
\overline{\Gamma}^0_{kj} &= 0\ , && k,j = 0,1,\dots,n\ ,\\
\overline{\Gamma}^i_{00} &= \frac{\partial \overline{w}^i}{\partial t} + \overline{w}^l\overline{w}^m\Gamma^i_{lm}\ , && i,l,m = 1,\dots,n\ ,\\
\overline{\Gamma}^i_{j0} &= \Gamma^i_{0j} = \frac{\partial \overline{w}^i}{\partial \xi^j} + \overline{w}^l\Gamma^i_{jl}\ , && i,j,l = 1,\dots,n\ ,\\
\overline{\Gamma}^i_{jk} &= \Gamma^i_{jk}\ , && i,j,k = 1,\dots,n\ ,
\end{aligned}
$$

and for $\overline{F}^i_0$,

$$
\begin{aligned}
\overline{F}^0_0 &= F^0\ ,\\
\overline{F}^i_0 &= \overline{F}^i - A^0\overline{w}^i\ , && i = 1,\dots,n\ .
\end{aligned}
$$

Using these expression in (2.112), we obtain a system of equations for the vector conservation law in the coordinates $\tau, \xi^1, \dots, \xi^n$ with an explicit expression for the components of the speed of the grid movement:

$$
\begin{aligned}
&\frac{\partial}{\partial \tau}(JA^{00}) + \frac{\partial}{\partial \xi^j}[J(\overline{A}^{0i} - A^{00}\overline{w}^j)] = JF^0\ ,\\
&\frac{\partial}{\partial \tau}J(\overline{A}^{i0} - A^{00}\overline{w}^j) + \frac{\partial}{\partial \xi^j}J(\overline{A}^{ij} + A^{00}\overline{w}^i\overline{w}^j - \overline{A}^{0j}\overline{w}^j - \overline{A}^{i0}\overline{w}^i)\\
&+ JA^{00}\Big(\frac{\partial \overline{w}^i}{\partial \tau} + \overline{w}^l\frac{\partial \overline{w}^i}{\partial \xi^l} + \overline{w}^l\overline{w}^j\Gamma^i_{lj}\Big)\\
&+ J(\overline{A}^{j0} + \overline{A}^{0j} - 2A^{00}\overline{w}^j)\Big(\frac{\partial \overline{w}^i}{\partial \xi^j} + \overline{w}^l\Gamma^i_{jl}\Big)
\end{aligned}
$$

$$\begin{aligned}&+J(\overline{A}^{lj}+A^{00}\overline{w}^l\overline{w}^j-\overline{A}^{0j}\overline{w}^l-\overline{A}^{0l}\overline{w}^j)\Gamma^i_{lj}\\&=J(\overline{F}^i-F^0\overline{w}^i)\ ,\qquad i,j,l=1,\dots,n\ .\end{aligned}\tag{2.113}$$

Another representation of (2.111) in new coordinates can be derived in the form of (2.97) by applying (2.97) to each line of the system (2.111). As a result, we obtain

$$\frac{1}{J}\left\{\frac{\partial}{\partial\tau}(JA^{i0})+\frac{\partial}{\partial\xi^j}\left[J\left(A^{ik}\frac{\partial\xi^j}{\partial x^k}-A^{i0}\overline{w}^j\right)\right]\right\}=F\ ,\quad j,k=1,\dots,n\ .\tag{2.114}$$

Recall that this approach is not restricted to a square form of the system (2.111), i.e. the ranges for the indices i and j can be different.

As an illustration of these equations for a vector conservation law in the curvilinear coordinates $\tau,\xi^1,\dots,\xi^n$, we write out a joint system for the conservation of mass and momentum, which in the coordinates t,x^1,x^2,x^3 has the following form:

$$\begin{aligned}&\frac{\partial\rho}{\partial t}+\frac{\partial}{\partial x^i}\rho u^i=0\ ,\qquad i=1,2,3\ ,\\&\frac{\partial\rho u^i}{\partial t}+\frac{\partial}{\partial x^j}(\rho u^iu^j+\overline{p}\delta^i_j)=\rho f^i\ ,\qquad i,j=1,2,3\ ,\end{aligned}\tag{2.115}$$

where

$$\overline{p}=p-\gamma\frac{\partial u^i}{\partial x^i}\ ,\quad \delta^i_j=0\quad\text{if}\quad i\neq j\quad\text{and}\quad\delta^i_j=1\quad\text{if}\quad i=j\ .$$

This system is represented in the form (2.111) with

$$\begin{aligned}A^{00}&=\rho\ ,\\A^{0i}&=A^{i0}=\rho u^i,\qquad i=1,2,3\ ,\\A^{ij}&=\rho u^iu^j+\delta^{ij}p,\qquad i,j=1,2,3\ ,\end{aligned}$$

i.e.

$$(A^{ij})=\begin{pmatrix}\rho&\rho u^1&\rho u^2&\rho u^3\\\rho u^1&\rho u^1u^1+\overline{p}&\rho u^1u^2&\rho u^1u^3\\\rho u^2&\rho u^2u^1&\rho u^2u^2+\overline{p}&\rho u^2u^3\\\rho u^3&\rho u^3u^1&\rho u^3u^2&\rho u^3u^3+\overline{p}\end{pmatrix}\ ,\qquad i,j=0,1,2,3\tag{2.116}$$

and

$$\begin{aligned}F^0&=0\ ,\\F^i&=\rho f^i,\qquad i=1,2,3\ .\end{aligned}$$

For the coordinate system $\tau, \xi^1, \ldots, \xi^n$, we obtain

$$\begin{aligned}
&\overline{A}^{00} = \rho \ , \\
&\overline{A}^{0i} = \overline{A}^{i0} = \rho(\overline{u}^i - \overline{w}^i) \ , \qquad i = 1, 2, 3 \ , \\
&\overline{A}^{ij} = \rho(\overline{u}^i - \overline{w}^i)(\overline{u}^j - \overline{w}^j) + \overline{p}g^{ij}, \qquad i, j = 1, 2, 3 \ .
\end{aligned}$$

Substituting these expressions in (2.113), we obtain a system of equations for the mass and momentum conservation laws in the coordinates $\tau, \xi^1, \ldots, \xi^n$:

$$\begin{aligned}
&\frac{\partial}{\partial \tau}(J\rho) + \frac{\partial}{\partial \xi^j}[J\rho(\overline{u}^j - \overline{w}^j)] = 0 \ , \qquad i = 1, 2, 3 \ , \\
&\frac{\partial}{\partial \tau}[J\rho(\overline{u}^i - \overline{w}^i)] + \frac{\partial}{\partial \xi^j}[J\rho(\overline{u}^i - \overline{w}^i)(\overline{u}^j - \overline{w}^j) + J\overline{p}g^{ij}] \\
&+ J\rho\frac{\partial \overline{w}^i}{\partial \tau} + J\rho(2\overline{u}^j - \overline{w}^j)\frac{\partial \overline{w}^i}{\partial \xi^j} + J(\rho\overline{u}^l\overline{u}^j + \overline{p}g^{lj})\Gamma^i_{lj} \\
&= J\rho\overline{f}^i \ , \qquad i, j, l = 1, 2, 3 \ .
\end{aligned} \tag{2.117}$$

If the coordinates ξ^i are the Lagrangian ones, i.e. $\overline{u}^i = \overline{w}^i$, then we obtain, from (2.117),

$$\begin{aligned}
&\frac{\partial}{\partial \tau}(J\rho) = 0 \ , \\
&J\rho\frac{\partial \overline{u}^i}{\partial \tau} + \frac{\partial}{\partial \xi^j}(J\overline{p}g^{ij}) + J\rho\overline{u}^j\frac{\partial \overline{u}^i}{\partial \xi^j} + J(\rho\overline{u}^l\overline{u}^j + \overline{p}g^{lj})\Gamma^i_{lj} = J\rho\overline{f}^i \ , \\
&i, j, l = 1, 2, 3 \ .
\end{aligned} \tag{2.118}$$

Note that the first equation of the system (2.117) coincides with (2.101) if $F = 0$; this was obtained as the scalar mass conservation law.

In the same manner, we can obtain an expression for the general Navier–Stokes equations of mass and momentum conservation by inserting the tensor $\{\sigma^{ij}\}$ described by (2.79) in the system (2.115) and the tensor $\{\overline{\sigma}^{ij}\}$ represented by (2.80) in the system (2.117).

A divergent form of (2.115) in arbitrary coordinates $\tau, \xi^1, \ldots, \xi^n$ is obtained by applying (2.114). With this, we obtain the system

$$\begin{aligned}
&\frac{\partial}{\partial \tau}(J\rho) + \frac{\partial}{\partial \xi^j}[J\rho(\overline{u}^j - \overline{w}^j)] = 0 \ , \\
&\frac{\partial}{\partial \tau}(J\rho\overline{u}^i) + \frac{\partial}{\partial \xi^j}\left[J\left(\rho u^i(\overline{u}^j - \overline{w}^j) + \overline{p}\frac{\partial \xi^j}{\partial x^i}\right)\right] = JF^i \ , \\
&\overline{u}^j = u^k\frac{\partial \xi^j}{\partial x^k} \ , \qquad i, j, k = 1, 2, 3 \ ,
\end{aligned} \tag{2.119}$$

and, in the Lagrangian coordinates,

$$\begin{aligned}
&\frac{\partial}{\partial \tau}(J\rho) = 0 \, , \\
&\frac{\partial}{\partial \tau}(J\rho \overline{u}^i) + \frac{\partial}{\partial \xi^j}\left(J\overline{p}\frac{\partial \xi^j}{\partial x^i}\right) = JF^i \, , \qquad i, j = 1, 2, 3 \, .
\end{aligned} \tag{2.120}$$

2.6 Comments

Many of the basic formulations of vector calculus and tensor analysis may be found in the books by Kochin (1951), Sokolnikoff (1964) and Gurtin (1981).

The formulation of general metric and tensor concepts specifically aimed at grid generation was originally performed by Eiseman (1980) and Warsi (1981).

Very important applications of the most general tensor relations to the formulation of unsteady equations in curvilinear coordinates in a strong conservative form were presented by Vinokur (1974). A strong conservation-law form of unsteady Euler equations was also described by Viviand (1974).

A derivation of various forms of the Navier–Stokes equations in general moving coordinates was described by Ogawa and Ishiguto (1987).

References

Eiseman, P. R. (1980). *Geometric methods in computational fluid dynamics*. ICASE Report 80-11 and Von Karman Institute for Fluid Dynamics Lecture Series Notes.

Gurtin, M. E. (1981). *An introduction to continuum mechanics*. New York: Academic Press.

Kochin, N. E. (1951). *Vector calculus and principles of tensor calculus*. Moscow (Russian): Nauka.

Ogawa, S., & Ishiguto, T. (1987). A method for computing flow fields around moving bodies. *Journal of Computational Physics*, *69*, 49–68.

Sokolnikoff, I. S. (1964). *Tensor analysis*. New York: Wiley.

Vinokur, M. (1974). Conservation equation of gasdynamics in curvilinear coordinate systems. *Journal of Computational Physics*, *14*(2), 105–125.

Viviand, H. (1974). Conservative forms of gas dynamic equations. *La Recherche Aerospatiale*, *1*, 65–68.

Warsi, Z. U. A. (1981). *Tensors and differential geometry applied to analytic and numerical coordinate generation*. MSSU-EIRS-81-1, Aerospace Engineering, Mississippi State University.

Chapter 3
Grid Quality Measures

3.1 Introduction

It is very important to develop grid generation techniques which sense grid quality features and possess means to eliminate the deficiencies of the grids. These requirements give rise to the problem of selecting and adequately formulating the necessary grid quality measures and finding out how they affect the solution error and the solution efficiency, in order to control the performance of the numerical analysis of physical problems with grids. Commonly, these quality measures encompass grid skewness, stretching, torsion, cell aspect ratio, cell volume, departure from conformality, cell deformation and various related constructions (centroids, circumcenters, circumcircles, incircles, etc.).

In this chapter, we utilize the notions and relations discussed in Sects. 2.2 and 2.3 to describe some qualitative and quantitative characteristics of structured grids. The structured grid concept allows one to define the grid characteristics through coordinate transformations as features of the coordinate curves, coordinate surfaces, coordinate volumes, etc. In general, these features are determined through the elements of the metric tensors and their derivatives. In particular, some grid properties can be described in terms of the invariants of the covariant metric tensor.

The chapter starts with an introduction to the elementary theory of curves and surfaces, necessary for the description of the quality measures of the coordinate curves and coordinate surfaces. It also includes a discussion of the metric invariants. Various grid characteristics are then formulated through quantities which measure the features of the coordinate curves, surfaces, and transformations.

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_3

3.2 Curve Geometry

Commonly, the curves lying in the n-dimensional space R^n are represented by smooth nondegenerate parametrizations

$$\boldsymbol{x}(\varphi) : [a, b] \to R^n, \qquad \boldsymbol{x}(\varphi) = [x^1(\varphi), \cdots, x^n(\varphi)] . \tag{3.1}$$

In our considerations, we will use the designation S^{x1} for the curve with the parametrization $\boldsymbol{x}(\varphi)$. In this chapter, we discuss the important measures of the local curve quality known as curvature and torsion. These measures are derived by some manipulations of basic curve vectors using the operations of dot and cross products.

3.2.1 Basic Curve Vectors

Tangent Vector

The first derivative of the parametrization $\boldsymbol{x}(\varphi)$ in (3.1) is a tangential vector

$$\boldsymbol{x}_\varphi = (x^1_\varphi, \ldots, x^n_\varphi)$$

to the curve S^{x1}. The quantity

$$g^{x\varphi} = \boldsymbol{x}_\varphi \cdot \boldsymbol{x}_\varphi = x^i_\varphi x^i_\varphi , \qquad i = 1, \ldots, n ,$$

is the metric tensor of the curve and its square root is the length of the tangent vector $\boldsymbol{x}_\varphi$. Thus, the length l of the curve S^{x1} is computed from the integral

$$l = \int_a^b \sqrt{g^{x\varphi}} \mathrm{d}\varphi .$$

The most important notions related to curves are connected with the arc length parameter s defined by the equation

$$s(\varphi) = \int_0^\varphi \sqrt{g^{x\varphi}} \mathrm{d}\varphi . \tag{3.2}$$

The vector $\mathrm{d}\boldsymbol{x}[\varphi(s)]/\mathrm{d}s$, where $\varphi(s)$ is the inverse of $s(\varphi)$, is a tangent vector designated by $\boldsymbol{t}$. From (3.2), we obtain

$$\boldsymbol{t} = \frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{x}[\varphi(s)] = \frac{\mathrm{d}\varphi}{\mathrm{d}s}\boldsymbol{x}_\varphi = \frac{1}{\sqrt{g^{x\varphi}}}\boldsymbol{x}_\varphi .$$

Therefore, $\boldsymbol{t}$ is the unit tangent vector and, after differentiating the relation $\boldsymbol{t}\cdot\boldsymbol{t}=1$, we find that the derivative $\boldsymbol{t}_s$ is orthogonal to $\boldsymbol{t}$. The vector $\boldsymbol{t}_s$ is called the curvature vector and denoted by $\boldsymbol{k}$. Let $\boldsymbol{n}$ be a unit vector that is parallel to $\boldsymbol{t}_s$; there then exists a scalar k, such that

$$\boldsymbol{t}_s=\boldsymbol{k}=k\boldsymbol{n}, \qquad k=(\boldsymbol{t}_s\cdot\boldsymbol{t}_s)^{1/2}=\alpha|\boldsymbol{k}| \ , \tag{3.3}$$

where $\alpha=1$ or $\alpha=-1$.

The magnitude k is called the curvature, while the quantity $\rho=1/k$ is called the radius of curvature of the curve.

Using the identity $\boldsymbol{x}_\varphi=\sqrt{g^{x\varphi}}\boldsymbol{t}$, we obtain, from (3.3),

$$\boldsymbol{x}_{\varphi\varphi}=\frac{1}{\sqrt{g^{x\varphi}}}(\boldsymbol{x}_{\varphi\varphi}\cdot\boldsymbol{x}_\varphi)\boldsymbol{t}+g^{x\varphi}k\boldsymbol{n} \ . \tag{3.4}$$

The identity (3.4) is an analog of the Gauss relations (2.36). This identity shows that the vector $\boldsymbol{x}_{\varphi\varphi}$ lies in the $\boldsymbol{t}$–$\boldsymbol{n}$ plane.

Curves in Three-Dimensional Space

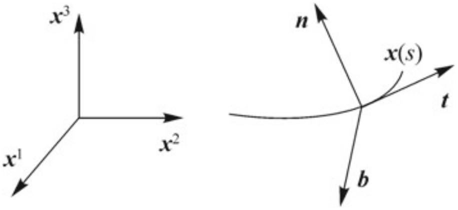
In three dimensions, we can apply the operation of the cross product to the basic tangential and normal vectors. The vector $\boldsymbol{b}=\boldsymbol{t}\times\boldsymbol{n}$ is a unit vector which is orthogonal to both $\boldsymbol{t}$ and $\boldsymbol{n}$. It is called the binormal vector. From (3.4), we find that $\boldsymbol{b}$ is orthogonal to $\boldsymbol{x}_{\varphi\varphi}$.

The three vectors $(\boldsymbol{t},\ \boldsymbol{n},\ \boldsymbol{b})$ form a right-handed triad (Fig. 3.1). Note that if the curve lies in a plane, then the vectors $\boldsymbol{t}$ and $\boldsymbol{n}$ lie in the plane as well and $\boldsymbol{b}$ is a constant unit vector normal to the plane.

The vectors $\boldsymbol{t},\ \boldsymbol{n}$, and $\boldsymbol{b}$ are connected by the Serret–Frenet equations

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{t}}{\mathrm{d}s}&=k\boldsymbol{n} \ ,\\ \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s}&=-k\boldsymbol{t}+\tau\boldsymbol{b} \ ,\\ \frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s}&=-\tau\boldsymbol{n} \ ,\end{aligned} \tag{3.5}$$

Fig. 3.1 Base curve vectors



where the coefficient τ is called the torsion of the curve. The first equation of the system (3.5) is taken from (3.3). The second and third equations are readily obtained from the formula (2.6) by replacing the $\boldsymbol{b}$ in (2.6) with the vectors on the left-hand side of (3.5), while the vectors $\boldsymbol{t},\ \boldsymbol{n}$, and $\boldsymbol{b}$ substitute for $\boldsymbol{e}_1,\ \boldsymbol{e}_2$, and $\boldsymbol{e}_3$, respectively. The vectors $\boldsymbol{t},\ \boldsymbol{n}$, and $\boldsymbol{b}$ constitute an orthonormal basis, i.e.

$$a_{ij} = a^{ij} = \delta^i_j \ , \qquad i, j = 1, 2, 3 \ ,$$

where, in accordance with Sect. 2.2.4, $a_{ij} = \boldsymbol{e}_i \cdot \boldsymbol{e}_j$, and the tensor $\{a^{ij}\}$ is the inverse of the tensor $\{a_{ij}\}$. Now, using (2.6), we obtain

$$\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} = \left(\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \cdot \boldsymbol{t}\right)\boldsymbol{t} + \left(\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \cdot \boldsymbol{n}\right)\boldsymbol{n} + \left(\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \cdot \boldsymbol{b}\right)\boldsymbol{b} = -k\boldsymbol{t} + \left(\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \cdot \boldsymbol{b}\right)\boldsymbol{b} \ ,$$

since $\boldsymbol{n}_s \cdot \boldsymbol{t} = -\boldsymbol{n} \cdot \boldsymbol{t}_s,\ \boldsymbol{n}_s \cdot \boldsymbol{n} = 0$. Thus, we obtain the second equation of (3.5) with $\tau = \boldsymbol{n}_s \cdot \boldsymbol{b}$. Analogously, we obtain the last equation of (3.5) by expanding the vector $\boldsymbol{b}_s$ through $\boldsymbol{t},\ \boldsymbol{n}$, and $\boldsymbol{b}$ using the relation (2.6):

$$\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s} = \left(\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s} \cdot \boldsymbol{t}\right)\boldsymbol{t} + \left(\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s} \cdot \boldsymbol{n}\right)\boldsymbol{n} + \left(\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s} \cdot \boldsymbol{b}\right)\boldsymbol{b} = -\left(\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \cdot \boldsymbol{b}\right)\boldsymbol{n} = -\tau\boldsymbol{n} \ ,$$

as $\boldsymbol{b}_s \cdot \boldsymbol{t} = -\boldsymbol{b} \cdot \boldsymbol{t}_s = 0,\ \boldsymbol{b}_s \cdot \boldsymbol{b} = 0$.

3.2.2 Curvature

A very important characteristic of a curve which is related to grid generation is the curvature k. This quantity is used as a measure of coordinate line bending.

One way to compute the curvature is to multiply (3.3) by $\boldsymbol{n}$ using the dot product operation. As

$$\frac{\mathrm{d}\boldsymbol{t}}{\mathrm{d}s} = \frac{1}{\sqrt{g^{x\varphi}}}\frac{\mathrm{d}}{\mathrm{d}\varphi}\left(\frac{1}{\sqrt{g^{x\varphi}}}\boldsymbol{x}_\varphi\right) = \frac{1}{g^{x\varphi}}\boldsymbol{x}_{\varphi\varphi} - \frac{1}{(g^{x\varphi})^2}(\boldsymbol{x}_\varphi \cdot \boldsymbol{x}_{\varphi\varphi})\boldsymbol{x}_\varphi \ ,$$

from (3.2), (3.3), the result is

$$k = \frac{1}{g^{x\varphi}}\boldsymbol{x}_{\varphi\varphi} \cdot \boldsymbol{n} \ . \tag{3.6}$$

The vector $\boldsymbol{n}$ is independent of the curve parametrization, and therefore we find from (3.4), (3.6) that k is an invariant of parametrizations of the curve.

In two dimensions,

$$\boldsymbol{n} = \frac{1}{\sqrt{g^{x\varphi}}}(-x^2_\varphi, x^1_\varphi) \ ,$$

therefore, in this case, we obtain, from (3.6),

$$k^2 = \frac{(x_\varphi y_{\varphi\varphi} - y_\varphi x_{\varphi\varphi})^2}{[(x_\varphi)^2 + (y_\varphi)^2]^3} \tag{3.7}$$

with the convention $x = x^1,\ y = x^2$. In particular, when the curve in R^2 is defined by a function $u = u(x)$, we obtain from (3.7), assuming in (3.1) $\boldsymbol{x}(\varphi) = [\varphi, u(\varphi)]$, $\varphi = x$,

$$k^2 = (u_{xx})^2/[1 + (u_x)^2]^3 \ .$$

In the case of three-dimensional space, the curvature k can also be computed from the relation obtained by multiplying (3.4) by $\boldsymbol{x}_\varphi$ with the cross product operation:

$$\boldsymbol{x}_\varphi \times \boldsymbol{x}_{\varphi\varphi} = g^{x\varphi} k(\boldsymbol{x}_\varphi \times \boldsymbol{n}) = (g^{x\varphi})^{3/2} k \boldsymbol{b} \ .$$

Thus, we obtain

$$k^2 = \frac{|\boldsymbol{x}_\varphi \times \boldsymbol{x}_{\varphi\varphi}|^2}{(g^{x\varphi})^3} \tag{3.8}$$

and consequently, from (2.26),

$$k^2 = \frac{(x^1_\varphi x^2_{\varphi\varphi} - x^2_\varphi x^1_{\varphi\varphi})^2 + (x^2_\varphi x^3_{\varphi\varphi} - x^3_\varphi x^2_{\varphi\varphi})^2 + (x^3_\varphi x^1_{\varphi\varphi} - x^1_\varphi x^3_{\varphi\varphi})^2}{[(x^1_\varphi)^2 + (x^2_\varphi)^2 + (x^3_\varphi)^2]^3} \ .$$

3.2.3 Torsion

Another important quality measure of curves in three-dimensional space is the torsion τ. This quantity is suitable for measuring the rate of twisting of the lines of coordinate grids.

In order to figure out the value of τ for a curve in R^3, represented by (3.1) for $n = 3$, we use the last relation in (3.5), which yields

$$\tau = -\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s} \cdot \boldsymbol{n} \ .$$

As $\boldsymbol{b} = \boldsymbol{t} \times \boldsymbol{n}$, we obtain

$$\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}s} = \frac{\mathrm{d}\boldsymbol{t}}{\mathrm{d}s} \times \boldsymbol{n} + \boldsymbol{t} \times \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} = \boldsymbol{t} \times \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \ ,$$

since $\mathrm{d}\boldsymbol{t}/\mathrm{d}s = k\boldsymbol{n}$. Thus,

$$\tau = \left(-\boldsymbol{t} \times \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s}\right) \cdot \boldsymbol{n} \ . \tag{3.9}$$

From (3.2), (3.3), we have the following obvious relations for the basic vectors $\boldsymbol{t}$ and $\boldsymbol{n}$ in terms of the parametrization $\boldsymbol{x}(\varphi)$ and its derivatives:

$$\begin{aligned}
\boldsymbol{t} &= \frac{1}{\sqrt{g^{x\varphi}}}\boldsymbol{x}_\varphi \ , \\
\boldsymbol{n} &= \frac{1}{k}\frac{\mathrm{d}\boldsymbol{t}}{\mathrm{d}s} = \frac{1}{k}\left(\frac{1}{g^{x\varphi}}\boldsymbol{x}_{\varphi\varphi} - \frac{\boldsymbol{x}_\varphi \cdot \boldsymbol{x}_{\varphi\varphi}}{(g^{x\varphi})^2}\boldsymbol{x}_\varphi\right) \ , \\
\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} &= \frac{1}{k}\left(\frac{1}{(g^{x\varphi})^{3/2}}\boldsymbol{x}_{\varphi\varphi\varphi} - 2\frac{\boldsymbol{x}_\varphi \cdot \boldsymbol{x}_{\varphi\varphi}}{(g^{x\varphi})^2}\boldsymbol{x}_{\varphi\varphi}\right. \\
&\quad \left. - \frac{\mathrm{d}}{\mathrm{d}\varphi}\left(\frac{\boldsymbol{x}_\varphi \cdot \boldsymbol{x}_{\varphi\varphi}}{(g^{x\varphi})^2}\right)\boldsymbol{x}_\varphi - \frac{1}{k}\frac{\mathrm{d}k}{\mathrm{d}s}\boldsymbol{n}\right) \ .
\end{aligned} \tag{3.10}$$

Thus,

$$\boldsymbol{t} \times \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} = \frac{1}{k(g^{x\varphi})^2}\boldsymbol{x}_\varphi \times \boldsymbol{x}_{\varphi\varphi\varphi} - 2\frac{\boldsymbol{x}_\varphi \cdot \boldsymbol{x}_{\varphi\varphi}}{k(g^{x\varphi})^{5/2}}\boldsymbol{x}_\varphi \times \boldsymbol{x}_{\varphi\varphi} - \frac{1}{k^2\sqrt{g^{x\varphi}}}\frac{\mathrm{d}k}{\mathrm{d}s}\boldsymbol{x}_\varphi \times \boldsymbol{n} \ .$$

As $(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{a} = (\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{b} = 0$ for arbitrary vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we obtain, from (3.9, 3.10),

$$\tau = -\frac{1}{k^2(g^{x\varphi})^3}(\boldsymbol{x}_\varphi \times \boldsymbol{x}_{\varphi\varphi\varphi}) \cdot \boldsymbol{x}_{\varphi\varphi} = \frac{1}{k^2(g^{x\varphi})^3}(\boldsymbol{x}_\varphi \times \boldsymbol{x}_{\varphi\varphi}) \cdot \boldsymbol{x}_{\varphi\varphi\varphi} \ . \tag{3.11}$$

And using (2.31), we also obtain

$$\tau = \frac{1}{k^2(g^{x\varphi})^3}\det\left\{\begin{matrix} x^1_\varphi & x^2_\varphi & x^3_\varphi \\ x^1_{\varphi\varphi} & x^2_{\varphi\varphi} & x^3_{\varphi\varphi} \\ x^1_{\varphi\varphi\varphi} & x^2_{\varphi\varphi\varphi} & x^3_{\varphi\varphi\varphi} \end{matrix}\right\}.$$

3.3 Surface Geometry

In general, a surface in the three-dimensional space R^3 is assumed to be locally represented by some parametric two-dimensional domain S^2 and a parametrization

$$\boldsymbol{x}(\boldsymbol{s}) : S^2 \to R^3 \ , \quad \boldsymbol{x}(\boldsymbol{s}) = [x^1(\boldsymbol{s}), x^2(\boldsymbol{s}), x^3(\boldsymbol{s})] \ , \quad \boldsymbol{s} = (s^1, s^2) \ , \tag{3.12}$$

where $\boldsymbol{x}(\boldsymbol{s})$ is a smooth nondegenerate vector function. We use the designation S^{x2} for the surface with the parametrization $\boldsymbol{x}(\boldsymbol{s})$. In analogy with domains, the transformation $\boldsymbol{x}(\boldsymbol{s})$ defines the curvilinear coordinate system $s^1,\ s^2$ on the surface, as well as the respective base vectors and metric tensors.

For the purpose of adaptive grid generation, the so-called monitor surfaces are very important. These surfaces are defined by the values of some vector-valued function $\boldsymbol{u}(\boldsymbol{s})$, referred to as the height function, over the domain S^2. The natural form (3.12) of the parametrization of the monitor surface formed with a scalar height function $u(\boldsymbol{x})$ is represented by the formula

$$\boldsymbol{x}(\boldsymbol{s}) = [s^1, s^2, u(s^1, s^2)] \ . \tag{3.13}$$

3.3.1 Surface Base Vectors

A surface in R^3, represented by (3.12), has three base vectors: two tangents (one to each coordinate curve) and a normal. The two tangential vectors to the coordinates s^1 and s^2 represented by $\boldsymbol{x}(\boldsymbol{s})$ are, respectively,

$$\boldsymbol{x}_{s^i} = \frac{\partial \boldsymbol{x}}{\partial s^i} = \left(\frac{\partial x^1}{\partial s^i}\ ,\ \frac{\partial x^2}{\partial s^i}\ ,\ \frac{\partial x^3}{\partial s^i} \right)\ , \qquad i = 1, 2\ .$$

The unit normal vector to the surface S^{x2} is defined through the cross product of the tangent vectors $\boldsymbol{x}_{s^1}$ and $\boldsymbol{x}_{s^2}$:

$$\boldsymbol{n} = \frac{1}{|\boldsymbol{x}_{s^1} \times \boldsymbol{x}_{s^2}|} (\boldsymbol{x}_{s^1} \times \boldsymbol{x}_{s^2})\ .$$

Since $(\boldsymbol{x}_{s^1} \times \boldsymbol{x}_{s^2}) \cdot \boldsymbol{n} > 0$, the base surface vectors $\boldsymbol{x}_{s^1},\ \boldsymbol{x}_{s^2}$, and $\boldsymbol{n}$ comprise a right-handed triad (Fig. 3.2). In accordance with (2.26) and (2.27), the unit normal $\boldsymbol{n}$ can also be expressed as

$$\boldsymbol{n} = \frac{1}{\sqrt{g^{rs}}} \left(\frac{\partial x^{l+1}}{\partial s^1} \frac{\partial x^{l+2}}{\partial s^2} - \frac{\partial x^{l+2}}{\partial s^1} \frac{\partial x^{l+1}}{\partial s^2} \right) \boldsymbol{e}_l\ , \qquad l = 1, 2, 3\ , \tag{3.14}$$

where $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ is the Cartesian basis of R^3. Recall that this formula implies the identification convention for indices in three dimensions, where k is equivalent to $k \pm 3$. If the surface S^{x2} is a monitor surface represented by a height function $u(\boldsymbol{s})$, then we obtain, from (3.14),

$$\boldsymbol{n} = \frac{1}{\sqrt{1 + (u_{s^1})^2 + (u_{s^2})^2}} \left(-\frac{\partial u}{\partial s^1},\ -\frac{\partial u}{\partial s^2},\ 1 \right)\ .$$

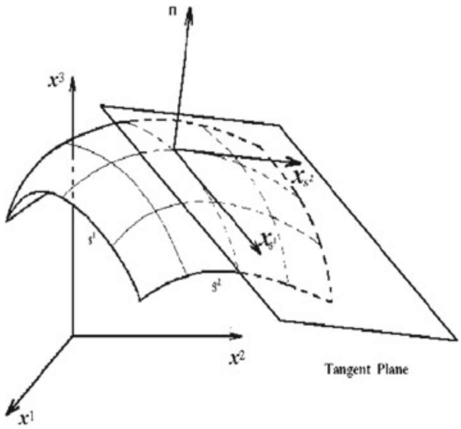


Fig. 3.2 Surface base vectors

In another particular case, when the surface points are found from the equation $f(\boldsymbol{x}) = c$, we obtain $\nabla f \cdot \boldsymbol{x}_{s^i} = 0,\ i = 1, 2$, and therefore

$$\boldsymbol{n} = l\nabla f\ , \qquad |l| = 1/|\nabla f|\ .$$

3.3.2 Metric Tensors

The surface metric tensors, like the domain metric tensors, are defined through the operation of the dot product on the vectors tangential to the coordinate lines.

Covariant Metric Tensor

We designate the covariant metric tensor of the surface S^{x2}, represented by (3.12) in the coordinates $s^1,\ s^2$ as G^{xs}, i.e.

$$G^{xs} = \{g_{ij}^{xs}\}, \qquad i, j = 1, 2\ ,$$

where

$$g_{ij}^{xs} = \boldsymbol{x}_{s^i} \cdot \boldsymbol{x}_{s^j}\ , \qquad i, j = 1, 2\ . \tag{3.15}$$

In particular, when a surface is defined by the values of some scalar function $u(\boldsymbol{s})$ over the domain S^2 then, from (3.13),

$$g_{ij}^{xs} = \delta_i^j + \frac{\partial u}{\partial s^i}\frac{\partial u}{\partial s^j}\ , \qquad i, j = 1, 2\ .$$

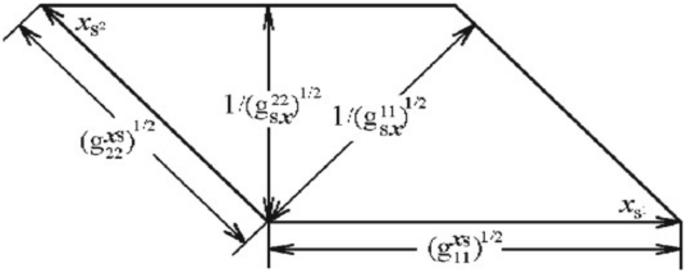


Fig. 3.3 Geometric meaning of the metric elements

The quantity $\sqrt{g_{ii}^{xs}}$ in (3.15) for a fixed i has the geometrical meaning of the length of the tangent vector $\boldsymbol{x}_{s^i}$ to the coordinate curve s^i (see Fig. 3.3).

The differential quadratic form

$$g_{ij}^{xs} \mathrm{d}s^i \mathrm{d}s^j \ , \qquad i, j = 1, 2 \ ,$$

relating to the line elements in space, is called the first fundamental form of the surface. It represents the value of the square of the length of an elementary displacement $\mathrm{d}\boldsymbol{x}$ on the surface.

Let the Jacobian of G^{xs} be designated by g^{xs}. Since

$$g^{xs} = |\boldsymbol{x}_{s^1}|^2 |\boldsymbol{x}_{s^2}|^2 (1 - \cos^2 \theta) = (|\boldsymbol{x}_{s^1}| \cdot |\boldsymbol{x}_{s^2}| \sin \theta)^2 = (\boldsymbol{x}_{s^1} \times \boldsymbol{x}_{s^2})^2 \ ,$$

where θ is the angle between $\boldsymbol{x}_{s^1}$ and $\boldsymbol{x}_{s^2}$, we find that the quantity g^{xs} is the area squared of the parallelogram formed by the vectors $\boldsymbol{x}_{s^1}$ and $\boldsymbol{x}_{s^2}$. Therefore, the area of the surface S^{x2} is computed from the formula

$$S = \int_{S^2} \sqrt{g^{xs}} \mathrm{d}\boldsymbol{s} \ .$$

Contravariant Metric Tensor

Consequently, the contravariant metric tensor of the surface S^{x2}, represented by (3.12), in the coordinates $s^1, \ s^2$ is the matrix designated as G_{sx}, and consequently

$$G_{sx} = \{g_{sx}^{ij}\} \ , \qquad i, j = 1, 2.$$

The tensors G^{xs} and G_{sx} are inverse to each other, i.e.

$$g_{ij}^{xs} \ g_{sx}^{jk} = \delta_k^i \ , \qquad i, j, k = 1, 2 \ .$$

Thus, in analogy with (2.21), we obtain

$$\begin{aligned} g^{ij}_{sx} &= (-1)^{i+j} g^{xs}_{3-i\ 3-j}/g^{xs} \ , \\ g^{xs}_{ij} &= (-1)^{i+j} g^{xs} g^{3-i\ 3-j}_{sx} \ , \qquad i, j = 1, 2 \ , \end{aligned} \tag{3.16}$$

with fixed indices i and j. The diagonal elements g^{11}_{sx} and g^{22}_{sx} of the contravariant metric tensor G_{sx} are connected with the natural geometric quantities of the parallelogram defined by the tangent vectors $\boldsymbol{x}_{s^1}$ and $\boldsymbol{x}_{s^2}$ (see Fig. 3.3). Namely, taking into account the relation $g^{xs} = g^{xs}_{11}/g^{22}_{sx}$, we find that $\sqrt{g^{22}_{sx}}$ is the inverse of the value of the distance between the parallel edges of the parallelogram formed by the vector $\boldsymbol{x}_{s^1}$. Analogously, $\sqrt{g^{11}_{sx}}$ is the inverse of the distance between the other pair of parallelogram edges, i.e. those formed by $\boldsymbol{x}_{s^2}$.

3.3.3 Second Fundamental Form

The coefficients of the second fundamental form

$$b_{ij} \mathrm{d}s^i \mathrm{d}s^j \ , \qquad i, j = 1, 2 \ ,$$

of the surface S^{x2} are defined by the dot products of the second derivatives of the vector function $\boldsymbol{x}(\boldsymbol{s})$ and the unit normal vector $\boldsymbol{n}$ to the surface at the point $\boldsymbol{s}$ under consideration:

$$b_{ij} = \boldsymbol{x}_{s^i s^j} \cdot \boldsymbol{n} \ , \qquad i, j = 1, 2 \ . \tag{3.17}$$

Thus, from (3.14), (3.17), we obtain for b_{ij}, $i, j = 1, 2$,

$$b_{ij} = \frac{1}{\sqrt{g^{xs}}} \left[\frac{\partial^2 x^l}{\partial s^i \partial s^j} \left(\frac{\partial x^{l+1}}{\partial s^1} \frac{\partial x^{l+2}}{\partial s^2} - \frac{\partial x^{l+2}}{\partial s^1} \frac{\partial x^{l+1}}{\partial s^2} \right) \right] \ , \quad l = 1, 2, 3 \ , \tag{3.18}$$

with the identification convention for the superscripts that k is equivalent to $k \pm 3$. Correspondingly, for the monitor surface with the height function $u(\boldsymbol{s})$, we obtain

$$b_{ij} = \frac{1}{\sqrt{1 + (u_{s^1})^2 + (u_{s^2})^2}} u_{s^i s^j} \ , \qquad i, j = 1, 2 \ .$$

The tensor $\{b_{ij}\}$ reflects the local warping of the surface, namely its deviation from the tangent plane at the point under consideration. In particular, if $\{b_{ij}\} \equiv 0$ at all points of S^2, then the surface is a plane.

3.3.4 Surface Curvatures

Principal Curvatures

Let a curve on the surface be defined by the intersection of a plane containing the normal $\boldsymbol{n}$ with the surface. It is obvious that either $\boldsymbol{n}$ or $-\boldsymbol{n}$ is also the curve normal vector. Taking into account (3.6), we obtain for the curvature of this curve

$$k = \frac{b_{ij}\mathrm{d}s^i\mathrm{d}s^j}{g_{ij}^{xs}\mathrm{d}s^i\mathrm{d}s^j}\ , \qquad i, j = 1, 2\ . \tag{3.19}$$

Here, $(\mathrm{d}s^1,\ \mathrm{d}s^2)$ is the direction of the curve, i.e. $\mathrm{d}s^i = c(\mathrm{d}s^i/\mathrm{d}\varphi)$, where $\boldsymbol{s}(\varphi)$ is a curve parametrization. The two extreme quantities K_{I} and K_{II} of the values of k are called the principal curvatures of the surface at the point under consideration. In order to compute the principal curvatures, we consider the following relation for the value of the curvature:

$$(b_{ij} - kg_{ij}^{xs})\mathrm{d}s^i\mathrm{d}s^j = 0\ , \qquad i, j = 1, 2\ , \tag{3.20}$$

which follows from (3.19). In order to find the maximum and minimum values of k, the usual method of equating to zero the derivative with respect to $\mathrm{d}s^i$ is applied. Thus, the components of the $(\mathrm{d}s^1,\ \mathrm{d}s^2)$ direction giving an extreme value of k are subject to the restriction

$$(b_{ij} - kg_{ij}^{xs})\mathrm{d}s^j = 0\ , \qquad i, j = 1, 2\ ,$$

which, in fact, is the eigenvalue problem for curvature. One finds the eigenvalues k by setting the determinant of this equation equal to zero, obtaining thereby the secular equation for k:

$$\det(b_{ij} - kg_{ij}^{xs}) = 0\ , \qquad i, j = 1, 2\ .$$

This equation, written out in full, is a quadratic equation

$$k^2 - g_{sx}^{ij}b_{ij}k + [b_{11}b_{22} - (b_{12})^2]/g^{xs} = 0\ ,$$

with two roots, which are the maximum and minimum values K_{I} and K_{II} of the curvature k:

$$K_{\mathrm{I,II}} = \frac{1}{2}g_{sx}^{ij}b_{ij} \pm \sqrt{\frac{1}{4}(g_{sx}^{ij}b_{ij})^2 - \frac{1}{g^{xs}}[b_{11}b_{22} - (b_{12})^2]}\ . \tag{3.21}$$

Gaussian Curvature

The determinant of the tensor $\{K^i_j\}$ represents the Gaussian curvature of the surface (3.12)

$$K_{\mathrm{G}} = \det\{K^i_j\} = \frac{1}{g^{xs}}[b_{11}b_{22} - (b_{12})^2] \ . \tag{3.22}$$

Taking into account (3.21), we readily see that the Gaussian curvature is the product of the two principal curvatures K_{I} and K_{II}, i.e.

$$K_{\mathrm{G}} = K_{\mathrm{I}}\ K_{\mathrm{II}} \ .$$

In terms of the height function $u(\boldsymbol{s})$ representing the monitor surface S^{x2}, we have

$$K_{\mathrm{G}} = \frac{u_{s^1s^1}u_{s^2s^2} - (u_{s^1s^2})^2}{[1 + (u_{s^1})^2 + (u_{s^2})^2]^2} \ .$$

A surface point is called elliptic if $K_{\mathrm{G}} > 0$, i.e. both K_{I} and K_{II} are both negative or both positive at the point of consideration. A saddle or hyperbolic point has principal curvatures of opposite sign, and therefore has negative Gaussian curvature. A parabolic point has one principal curvature vanishing and, consequently, a vanishing Gaussian curvature. This classification of points is prompted by the form of the curve which is obtained by the intersection of the surface with a slightly offset tangent plane. For an elliptic point, the curve is an ellipse; for a saddle point, it is a hyperbola. It is a pair of lines (degenerate conic) at a parabolic point, and it vanishes at a planar point, where both principal curvatures are zero.

Mean Curvature

One half of the sum of the principal curvatures is referred to as the mean surface curvature. Taking advantage of (3.21), the mean curvature, designated by K_{m}, is defined through the coefficients of the second fundamental form and elements of the contravariant metric tensor by

$$K_{\mathrm{m}} = \frac{1}{2}(K_{\mathrm{I}} + K_{\mathrm{II}}) = \frac{1}{2}g^{ij}_{sx}b_{ij} \ , \qquad i, j = 1, 2 \ . \tag{3.23}$$

In the case of the monitor surface represented by the function $u(s^1, s^2)$, we obtain

$$K_{\mathrm{m}} = \frac{u_{s^1s^1}[1 + (u_{s^2})^2] + u_{s^2s^2}[1 + (u_{s^1})^2] - 2u_{s^1}u_{s^2}u_{s^1s^2}}{2[1 + (u_{s^1})^2 + (u_{s^2})^2]^{3/2}} \ .$$

Now we consider the tensor

$$\{K^i_j\} \equiv \{g^{ik}_{sx}b_{kj}\} \ , \qquad i, j, k = 1, 2 \ .$$

As a reminder, the repeated index k means summation over it. It is easy to see that $\{K^i_j\}$ is a mixed tensor contravariant with respect to the upper index and covariant with respect to the lower one. From (3.23), we find that the mean curvature is defined as the trace of the tensor, namely,

$$2K_{\mathrm{m}} = \mathrm{tr}\{K^i_j\}\ , \qquad i, j = 1, 2\ . \tag{3.24}$$

A surface whose mean curvature is zero, i.e. $K_{\mathrm{I}} = -K_{\mathrm{II}}$, possesses the following unique property. Namely, if a surface bounded by a specified contour has a minimum area, then its mean curvature is zero. Conversely, of all the surfaces bounded by a curve whose length is sufficiently small, the minimum area is possessed by the surface whose mean curvature is zero.

It is easily shown that both the mean and the Gaussian curvatures are invariant of surface parametrizations.

3.3.5 Curvatures of Discrete Surfaces

From a computational standpoint, the discrete objects are attractive because they have been designed from the ground up with data-structures and algorithms in mind. From a mathematical standpoint, they present a great challenge: the discrete objects should have properties which are analogues of the properties of continuous objects. One important property of curves and surfaces is their curvature, which plays a significant role in many application areas. In the continuous formulations, there are remarkable theorems dealing with curvatures; a key requirement for a discrete curve or surface with discrete curvatures is that they satisfy analogous theorems.

Relying on the results presented in the paper of Sullivan (see Pinkall and Polhier 1993), we consider here some formulations of the curvatures of discrete surfaces, meaning triangulated polyhedral surfaces. Often, the most useful formulations are those which are based on integral relations for curvature, like the Gauss–Bonnet theorem or the force balance equation for mean curvature.

We assume that all cells meeting at a grid node $\boldsymbol{P}$ of the discrete surface under consideration are triangles. Such a triangulation at the node $\boldsymbol{P}$ can be obtained from arbitrary polyhedron triangulations by connecting the nodes adjacent to $\boldsymbol{P}$ of each of the two neighboring edges emanating from $\boldsymbol{P}$.

Gauss Curvature

Gauss curvature K_{G} at a grid vertex $\boldsymbol{P}$ must be subject to the following relation

$$\int\int_D K_{\mathrm{G}} dA := \sum_{\boldsymbol{P}\in D} K_{\boldsymbol{P}}\ , \quad \text{with} \quad K_{\boldsymbol{P}} = 2\pi - \sum_i \theta_i\ , \tag{3.25}$$

where the angles θ_i are the interior angles at $\boldsymbol{P}$ of the triangles meeting there, and $K_{\boldsymbol{P}}$ is often known as the angle defect at $\boldsymbol{P}$.

From this relation, one formula for discrete Gauss curvature $K_{\mathrm{G}}^{d}(\boldsymbol{P})$ at the node $\boldsymbol{P}$ may be defined by

$$K_{\mathrm{G}}^{d}(\boldsymbol{P}) = \frac{1}{A(\mathrm{star}\boldsymbol{P})}(2\pi - \sum_{i} \theta_i) \ , \tag{3.26}$$

where star$\boldsymbol{P}$ is a designation for the union of all triangles containing the vertex $\boldsymbol{P}$, and $A(\mathrm{star}\boldsymbol{P})$ is the area of star$\boldsymbol{P}$.

One more intrinsic characterization of Gauss curvature K_{G} is obtained by comparing the circumferences C_ε of (intrinsic) ε-balls around $\boldsymbol{P}$ to the value $2\pi\varepsilon$. We have

$$\frac{C_\varepsilon}{2\pi\varepsilon} = 1 - \frac{\varepsilon^2}{6} K_{\mathrm{G}}(\boldsymbol{P}) + O(\varepsilon^3) \ . \tag{3.27}$$

From this formula, discrete Gauss curvature $K_{\mathrm{G}}^{d}(\boldsymbol{P})$ at the node $\boldsymbol{P}$ may be defined by

$$K_{\mathrm{G}}^{d}(\boldsymbol{P}) = \frac{6}{\varepsilon^2}\left(1 - \frac{S_\varepsilon}{2\pi\varepsilon}\right) \ , \tag{3.28}$$

where S_ε is the length of the curve obtained by intersecting an ε-ball with star$\boldsymbol{P}$, and ε is a small number.

These formulations of the discrete Gauss curvature depend significantly on the choice of which pairs of cone points are connected by triangle edges (see Bobenko and Springborn 2005).

Mean Curvature

Suppose that the vertices adjacent to $\boldsymbol{P}$, in cyclic order, are $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_k$. Then, the discrete vector mean curvature $\boldsymbol{K}_{\mathrm{m}}^{d}(\boldsymbol{P})$ can be expressed explicitly in terms of these neighbors by the following formula:

$$\boldsymbol{K}_{\mathrm{m}}^{d}(\boldsymbol{P}) = \frac{1}{2}\sum_{i}(\cot\alpha_i + \cot\beta_i)(\boldsymbol{P} - \boldsymbol{P}_i) \ , \tag{3.29}$$

where α_i and β_i are the angles opposite the edge $\boldsymbol{P}\boldsymbol{P}_i$ in the two incident triangles (see Pinkall and Polhier 1993 and Sullivan 2008).

Alternatively, we have

$$\boldsymbol{K}_{\mathrm{m}}(\boldsymbol{P}) = \Delta_B[\boldsymbol{x}](\boldsymbol{P}) \ ,$$

where $\boldsymbol{x}$ is the position vector, and Δ_B is the Beltrami operator. Computing the Beltrami operator numerically at each grid point of a discrete surface gives the value of $\boldsymbol{K}_{\mathrm{m}}^{d}(\boldsymbol{P})$.

3.4 Metric-Tensor Invariants

The coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n$ of a physical n-dimensional domain X^n applied to generate grids through mapping approaches can be locally interpreted as some deformation of a uniform cell in the computational domain Ξ^n into the corresponding cell in the domain X^n. The local deformation of any cell is approximated by a linear transformation represented by the Jacobi matrix $\{\partial x^i/\partial \xi^j\}$. This deformation is not changed if any orthogonal transformation is applied to the cell in X^n. The deformation is also preserved if the orientation of the computational domain Ξ^n is changed. Therefore, it is logical to formulate the features of the coordinate grid cells in terms of the invariants of the orthogonal transformations of the covariant metric tensor $\{g_{ij}\}$, in the coordinates $\xi_1, \dots, \xi_n$, i.e.

$$g_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j} \ , \qquad i, j = 1, \dots, n \ . \tag{3.30}$$

3.4.1 *Algebraic Expressions for the Invariants*

According to the theory of matrices, a symmetric nondegenerate $(n \times n)$ matrix $\{a_{ij}\}$ has n independent invariants $I_i, \ i = 1, \dots, n$, of its orthogonal transformations. The ith invariant I_i is defined by summing all of the principal minors of order i of the matrix. Recall that the principal minors of a square matrix are the determinants of the square submatrices of the matrix. Thus, for example,

$$\begin{aligned} I_1 &= \sum_{i=1}^{n} a_{ii} = \mathrm{tr}\{a_{ij}\} \ , \\ I_{n-1} &= \sum_{i=1}^{n} \text{cofactor } a_{ii} = \det\{a_{ij}\} \sum_{i=1}^{n} a^{ii} = \det\{a_{ij}\}\, \mathrm{tr}\{a^{ij}\} \ , \\ I_n &= \det\{a_{ij}\} \ , \end{aligned} \tag{3.31}$$

where the matrix $\{a^{ij}\}$ is the inverse of $\{a_{ij}\}$.

When we use, for $\{a_{ij}\}$, the covariant metric tensor $\{g_{ij}\}$, $g_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j}$, of a domain X^n, then, taking advantage of (3.31), the invariants I_1 and I_2 in two dimensions are expressed as

$$\begin{aligned} I_1 &= g_{11} + g_{22} \ , \\ I_2 &= g_{11}\, g_{22} - (g_{12})^2 = g = J^2 \ , \end{aligned} \tag{3.32}$$

where $J = \det\{\partial x^i/\partial \xi^j\}$. The invariants of the three-dimensional metric tensor $\{g_{ij}\}$ are expressed as follows:

$$\begin{aligned} I_1 &= g_{11} + g_{22} + g_{33} \ , \\ I_2 &= g(g^{11} + g^{22} + g^{33}) \\ I_3 &= \det\{g_{ij}\} = g \ , \qquad i, j = 1, 2, 3 \ , \end{aligned} \tag{3.33}$$

where $g^{ij} = \nabla\xi^i \cdot \nabla\xi^j$. Analogously, the invariants of the surface metric tensor G^{xs}, represented in the coordinates $s^1, \ s^2$ by (3.12), are written out as

$$\begin{aligned} I_1 &= g^{xs}_{11} + g^{xs}_{22} \\ I_2 &= g^{xs} \ . \end{aligned} \tag{3.34}$$

The notion of an invariant can be helpful to identity conformal coordinate transformations. For example, in two dimensions, we know that a conformal mapping $\boldsymbol{x}(\boldsymbol{\xi})$ satisfies the Cauchy–Riemann equations

$$\frac{\partial x^1}{\partial \xi^1} = \frac{\partial x^2}{\partial \xi^2} \ , \qquad \frac{\partial x^1}{\partial \xi^2} = -\frac{\partial x^2}{\partial \xi^1} \ .$$

Therefore, a zero value of the quantity

$$Q = \left(\frac{\partial x^1}{\partial \xi^1} - \frac{\partial x^2}{\partial \xi^2}\right)^2 + \left(\frac{\partial x^1}{\partial \xi^2} + \frac{\partial x^2}{\partial \xi^1}\right)^2$$

is an indication of the conformality of $\boldsymbol{x}(\boldsymbol{\xi})$. We obtain

$$Q = g_{11} + g_{22} - 2J = I_1 - 2\sqrt{I_2} \ ,$$

using (3.32). Thus, the two-dimensional coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal if only if the invariants I_1 and I_2 satisfy the restriction $I_1/\sqrt{I_2} = 2$. In Sect. 3.7.7, it will be shown that an analogous relation is valid for an arbitrary dimension $n \geq 2$.

We also can see that the mean and Gaussian curvatures described by (3.24) and (3.22), respectively, are defined through the invariants of the tensor $\{K^i_j\}$, namely,

$$K_{\mathrm{m}} = \frac{1}{2} I_1 \ , \qquad K_{\mathrm{G}} = I_2 \ .$$

3.4.2 Geometric Interpretation

The invariants of the covariant metric tensor $\{g_{ij}\}$ can also be described in terms of some geometric characteristics of the n-dimensional parallelepiped (parallelogram in two dimensions) determined by the tangent vectors $\boldsymbol{x}_{\xi^i}$, thus giving a relationship between the cell characteristics of coordinate grids and the invariants. For example, we see from (3.32), (3.34) in two dimensions that the invariant I_1 equals the sum squares of the parallelogram edge lengths, while I_2 is equal to the parallelogram area squared. In three-dimensional space, we find, from (3.33), that I_1 equals the sum of the squares of the lengths of the base vectors $\boldsymbol{x}_{\xi^i},\ i = 1, 2, 3$, which are the edges of the parallelepiped. The invariant I_2 is the sum of the squares of the areas of the faces of the parallelepiped, while the invariant I_3 is its volume squared.

These geometric interpretations can be extended to arbitrary dimensions by the following consideration. Every principal minor of order m is the determinant of an m-dimensional square matrix A^m obtained from the covariant tensor $\{g_{ij}\}$ by crossing out $n - m$ rows and columns that intersect pairwise on the diagonal. Therefore, the elements of A^m are the dot products of m particular vectors of the base tangential vectors $\boldsymbol{x}_{\xi^i},\ i = 1, \ldots, n$. Thus, geometrically, the determinant of A^m equals the square of the m-dimensional volume of the m-dimensional parallelepiped constructed by the vectors of the basic set $\boldsymbol{x}_{\xi^i},\ i = 1, \ldots, n$, whose dot products form the matrix A^m. Therefore, $I_i,\ i = 1, \ldots, n$, is geometrically the sum of the squares of the i-dimensional volumes of the i-dimensional sides of the n-dimensional parallelepiped spanned by the base vectors $\boldsymbol{x}_{\xi^i},\ i = 1, \ldots, n$.

We note that the invariants do not describe all of the geometric features of the grid cells. In the two-dimensional case, the invariants I_1 and I_2 given by (3.32) can be the same for parallelepipeds that are not similar. For example, if we take a transformation $\boldsymbol{x}(\boldsymbol{\xi})$ whose tangential vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$ define a rectangle with sides of different lengths a and b, then we obtain

$$I_1 = a^2 + b^2 \ , \qquad I_2 = (ab)^2 \ .$$

However, as demonstrated in Fig. 3.4, the same invariants are produced by a transformation $\boldsymbol{x}(\boldsymbol{\xi})$, whose tangent vectors yield a rhombus with a side length l equal to $\sqrt{(a^2 + b^2)/2}$ and an angle θ defined by

$$\theta = \arcsin \frac{2ab}{a^2 + b^2} \ ,$$

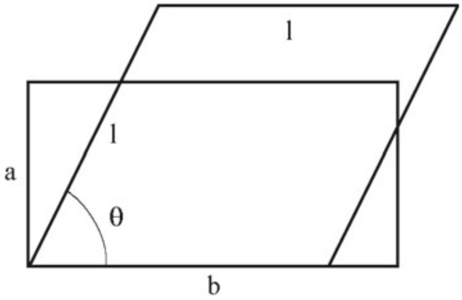


Fig. 3.4 Quadrilaterals with the same invariants

since

$$I_1 = 2l^2 = a^2 + b^2 ,$$
$$I_2 = l^2 \sin^2 \theta = (ab)^2 .$$

Thus, knowledge of the values of the invariants I1 and I2 alone is not sufficient to distinguish the rectangle from the rhombus. However, the value of the quantity $I_1/\sqrt{I_2}$ imposes restriction on the maximal angle between the parallelogram edges and on the maximum cell aspect ratio. These bounds will be evaluated in Sect. 3.7.7. In particular, if $I_1 = 2\sqrt{I_2}$, then we can definitely state that the parallelogram is a square.

3.5 Characteristics of Grid Lines

This section describes some characteristics of curvilinear coordinate lines in domains specified by the parametrization $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n$. These characteristics can be used for the evaluation of the grid properties and for the formulation of grid generation techniques through the calculus of variations.

All considerations in this section are concerned with a selected coordinate line ξ^i for a specified i, and therefore summation is not carried out over the repeated index i here.

3.5.1 *Sum of Squares of Cell Edge Lengths*

The length l_i of any cell edge along the coordinate curve ξ^i is expressed through the element g_{ii} of the covariant metric tensor $\{g_{ij}\}$:

$$l_i \approx \sqrt{g_{ii}}h .$$

The sum of the squares of the cell edge lengths equals $Q_l h^2$, where

$$Q_l = \sum_{j=1}^{n} g_{jj} = \mathrm{tr}\,\{g_{ij}\} . \tag{3.35}$$

The quantity Q_l is one of the important characteristics of the grid cell. This characteristic is the first invariant I_1 of the tensor matrix $\{g_{ij}\}$.

3.5.2 Eccentricity

The ratio between two adjacent grid steps along any coordinate curve ξ^i is a quantity which characterizes the change of the length of the cell edge in the ξ^i direction. This quantity is designated as ϵ^i, and at the point $\boldsymbol{\xi}$, it is expressed as follows:

$$\epsilon^i \approx \frac{|\boldsymbol{x}_{\xi^i}(\boldsymbol{\xi} + h\boldsymbol{e}^i)|}{|\boldsymbol{x}_{\xi^i}(\boldsymbol{\xi})|} .$$

We also find that

$$\epsilon^i \approx \frac{\sqrt{g_{ii}(\boldsymbol{\xi} + h\boldsymbol{e}^i)} - \sqrt{g_{ii}(\boldsymbol{\xi})}}{\sqrt{g_{ii}(\boldsymbol{\xi})}} + 1 \approx h\frac{1}{\sqrt{g_{ii}}}\frac{\partial}{\partial \xi^i}\sqrt{g_{ii}} + 1 ,$$

for a fixed i, since $|\boldsymbol{x}_{\xi^i}| = \sqrt{g_{ii}}$. The quantity

$$Q^i_\epsilon = \left(\frac{1}{\sqrt{g_{ii}}}\frac{\partial}{\partial \xi^i}\sqrt{g_{ii}}\right)^2 = \left(\frac{\partial}{\partial \xi^i}\ln\sqrt{g_{ii}}\right)^2 , \quad i \text{ fixed} \tag{3.36}$$

obtained from the expression for ϵ^i is a measure of the relative eccentricity. When $Q_\epsilon = 0$, then the length of the cell edge does not change in the ξ^i direction. With the Christoffel symbol notation (2.40), we also obtain

$$Q^i_\epsilon = \left(\frac{1}{g_{ii}}\frac{\partial \boldsymbol{x}}{\partial \xi^i} \cdot \frac{\partial^2 \boldsymbol{x}}{\partial \xi^i \partial \xi^i}\right)^2 = \left(\frac{1}{g_{ii}}[ii, i]\right)^2 , \quad i \text{ fixed} . \tag{3.37}$$

3.5.3 Curvature

The relative eccentricity Q^i_ϵ describes the change of the length of the cell edge along the coordinate curve ξ^i, however, it fails to describe the change in its direction. The quantity which characterizes this grid quality is derived through a curvature vector.

In accordance with (3.3), the curvature vector $\boldsymbol{k}_i$ of the coordinate line ξ^i for a fixed i is defined by the relation $\boldsymbol{k}_i = \boldsymbol{x}_{ss}$, where s is the arc length parametrization of the coordinate line ξ^i, i.e. the variable s is defined by the transformation $s(\xi^i)$ satisfying the equation

$$\frac{\mathrm{d}s}{\mathrm{d}\xi^i} = \sqrt{g_{ii}} , \quad i \text{ fixed} .$$

Therefore,

$$\frac{\partial}{\partial s} = \frac{1}{\sqrt{g_{ii}}}\frac{\partial}{\partial \xi^i} , \quad i \text{ fixed}$$

and consequently

$$\boldsymbol{k}_i = \frac{1}{g_{ii}}\boldsymbol{x}_{\xi^i\xi^i} - \frac{\boldsymbol{x}_{\xi^i}}{(g_{ii})^2}\boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^i\xi^i} \ , \quad i \text{ fixed} \ . \tag{3.38}$$

Local Straightness of the Coordinate Line

Equation (3.38) shows that if the curvature vector $\boldsymbol{k}_i$ equals zero ($\boldsymbol{k}_i = \boldsymbol{0}$), then the vector $\boldsymbol{x}_{\xi^i\xi^i}$ is parallel to the vector $\boldsymbol{x}_{\xi^i}$, i.e. the tangential vector does not change its direction. Therefore, the coordinate line ξ^i is locally straight at a point of zero curvature. From (3.38), we obtain, in this case,

$$\boldsymbol{x}_{\xi^i\xi^i} = \frac{(\boldsymbol{x}_{\xi^i\xi^i} \cdot \boldsymbol{x}_{\xi^i})}{g_{ii}}\boldsymbol{x}_{\xi^i} \ , \quad i \text{ fixed} \ .$$

Using the Gauss relations (2.36), we also obtain

$$\boldsymbol{x}_{\xi^i\xi^i} = \Gamma^l_{ii}\boldsymbol{x}_{\xi^l} \ , \qquad l = 1, \dots, n \ , \quad i \text{ fixed} \ .$$

Comparing these two expansions of $\boldsymbol{x}_{\xi^i\xi^i}$, we see that the vector $\boldsymbol{x}_{\xi^i\xi^i}$ is parallel to $\boldsymbol{x}_{\xi^i}$ if

$$\Gamma^l_{ii} = 0 \qquad \text{for all } l \neq i \ , \quad i \text{ fixed} \ . \tag{3.39}$$

The relation (3.39) is a criterion of local straightness of the coordinate curve ξ^i. A measure of the deviation of the curve ξ^i from a straight line may, therefore, be determined as

$$Q^i_{\text{st}} = d_{lm}\Gamma^l_{ii}\Gamma^m_{ii} \ , \qquad l, m \neq i \ , \quad i \text{ fixed} \ , \tag{3.40}$$

where d_{lm} is a positive $(n-1) \times (n-1)$ tensor.

Expansion of the Curvature Vector in the Normal Vectors

We know that the curvature vector $\boldsymbol{k}_i$ is orthogonal to the unit tangential vector $\boldsymbol{x}_s$. On the other hand, the normal base vectors $\nabla\xi^j$, $j \neq i$, are also orthogonal to the tangent vector $\boldsymbol{x}_{\xi^i}$ and therefore to $\boldsymbol{x}_s$. Thus, the curvature vector $\boldsymbol{k}_i$ of the coordinate curve ξ^i can be expanded in the $n-1$ normal vectors $\nabla\xi^j$, $j \neq i$. In order to find such an expansion, we first recall that in accordance with (2.41),

$$\boldsymbol{x}_{\xi^i\xi^i} = [ii, m]\nabla\xi^m \ , \qquad m = 1, \dots, n \ , \ , \quad i \text{ fixed}$$

with summation over m, where

$$[ii, m] = \boldsymbol{x}_{\xi^i\xi^i} \cdot \boldsymbol{x}_{\xi^m} = \frac{\partial g_{im}}{\partial \xi^i} - \frac{1}{2}\frac{\partial g_{ii}}{\partial \xi^m} \ , \quad i \text{ fixed} \ ,$$

from (2.45). Furthermore, from (2.23),

$$\boldsymbol{x}_{\xi^i} = g_{im}\nabla\xi^m, \qquad m = 1, \dots, n .$$

Therefore, the relation (3.38) is equivalent to

$$\begin{aligned}\boldsymbol{k}_i &= \frac{1}{g_{ii}}\left([ii,m]\nabla\xi^m - \frac{1}{g_{ii}}[ii,i]\right) g_{im}\nabla\xi^m \\ &= \frac{1}{(g_{ii})^2}(g_{ii}[ii,l] - g_{il}[ii,i])\nabla\xi^l \ , \\ &\quad m = 1, \dots, n \ , \quad l = 1, \dots, n \ , \quad l \neq i \ , \quad i \ \text{ fixed} \ . \end{aligned} \tag{3.41}$$

This equation represents the curvature vector $\boldsymbol{k}_i$ through the $n-1$ normal base vectors $\nabla\xi^l$, $l \neq i$.

In particular, in two dimensions, the relation (3.41) for $i = 1$ becomes

$$\boldsymbol{k}_1 = \frac{1}{(g_{11})^2}(g_{11}[11,2] - g_{12}[11,1])\nabla\xi^2 \ . \tag{3.42}$$

And, from (2.21),

$$\boldsymbol{k}_1 = \frac{g}{(g_{11})^2}(g^{22}[11,2] + g^{21}[11,1])\nabla\xi^2 \ .$$

Therefore, using (2.43), we obtain

$$\boldsymbol{k}_1 = \frac{g}{(g_{11})^2}\Gamma^2_{11}\nabla\xi^2 \ . \tag{3.43}$$

Analogously, the curvature vector $\boldsymbol{k}_2$ along the coordinate ξ^2 is expressed as follows:

$$\boldsymbol{k}_2 = \frac{g}{(g_{22})^2}\Gamma^1_{22}\nabla\xi^1 \ . \tag{3.44}$$

In the same way, the curvature vector of the coordinate curves in the case of three-dimensional space R^3 is computed. For example, in accordance with (3.41), the vector $\boldsymbol{k}_1$ can be expanded in the normal vectors $\nabla\xi^2$ and $\nabla\xi^3$ as

$$\begin{aligned}\boldsymbol{k}_1 = \frac{1}{(g_{11})^2}\{&(g_{11}[11,2] - g_{12}[11,1])\nabla\xi^2 \\ &+(g_{11}[11,3] - g_{13}[11,1])\nabla\xi^3\} \ . \end{aligned} \tag{3.45}$$

Measure of Coordinate Line Curvature

The length of the vector $\boldsymbol{k}_i$ is the modulus of the curvature and denoted by $|k_i|$. Thus, for the curvature k_1 of the coordinate line ξ^1 in the two-dimensional domain X^2, we obtain, from (3.43),

$$|k_1| = \frac{g\sqrt{g^{22}}}{(g_{11})^2}|\Gamma^2_{11}| = \frac{g\sqrt{g^{22}}}{(g_{11})^2}\left|\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}\frac{\partial\xi^2}{\partial x^1} + \frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1}\frac{\partial\xi^2}{\partial x^2}\right| . \tag{3.46}$$

Taking into account the two-dimensional relation (2.4)

$$\frac{\partial\xi^i}{\partial x^j} = (-1)^{i+j}\frac{1}{J}\frac{\partial x^{3-j}}{\partial\xi^{3-i}} , \qquad i, j = 1, 2 , \qquad J = \sqrt{g} ,$$

with i, j fixed, we find that

$$\Gamma^2_{11} = \frac{1}{J}\left(\frac{\partial x^1}{\partial\xi^1}\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1} - \frac{\partial x^2}{\partial\xi^1}\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}\right).$$

Therefore, for the curvature of the coordinate ξ^1, we also obtain, from (2.21) and (3.46),

$$|k_1| = \frac{1}{(g_{11})^{3/2}}\left|\frac{\partial x^1}{\partial\xi^1}\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1} - \frac{\partial x^2}{\partial\xi^1}\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}\right| . \tag{3.47}$$

Analogously, using the relation (3.44), we get for the curvature of the coordinate curve ξ^2

$$|k_2| = \frac{1}{(g_{22})^{3/2}}\left|\frac{\partial x^2}{\partial\xi^2}\frac{\partial^2 x^1}{\partial\xi^2\partial\xi^2} - \frac{\partial x^1}{\partial\xi^2}\frac{\partial^2 x^2}{\partial\xi^2\partial\xi^2}\right| . \tag{3.48}$$

In the case of three-dimensional space, the curvature measure of the coordinate line ξ^i is computed from the relation (3.8):

$$|k_i| = \frac{1}{\sqrt{(g_{ii})^3}}|\boldsymbol{x}_{\xi^i} \times \boldsymbol{x}_{\xi^i\xi^i}| , \qquad i = 1, 2, 3 , \quad i \text{ fixed} . \tag{3.49}$$

The curvature representation can provide various measures of the curvature of the coordinate line ξ^i. The simplest measure may be described in the common manner as the square of the curvature

$$Q^i_k = (k_i)^2 . \tag{3.50}$$

In analogy with (3.40), the quantity Q^i_k is also a measure of the departure of the coordinate line ξ^i from a straight line.

3.5.4 Measure of Coordinate Line Torsion

The square of the torsion is another measure of a coordinate line ξ^i lying in three-dimensional space. This measure is computed in accordance with (3.11) from the relation

$$\begin{aligned} Q^i_\tau &= \frac{1}{(k_i)^4 (g_{ii})^6} [(\boldsymbol{x}_{\xi^i} \times \boldsymbol{x}_{\xi^i \xi^i}) \cdot \boldsymbol{x}_{\xi^i \xi^i \xi^i}]^2 \\ &= \frac{1}{(k_i)^4 (g_{ii})^6} \det{}^2 \begin{pmatrix} \boldsymbol{x}_{\xi^i} \\ \boldsymbol{x}_{\xi^i \xi^i} \\ \boldsymbol{x}_{\xi^i \xi^i \xi^i} \end{pmatrix} , \quad i \text{ fixed} . \end{aligned} \tag{3.51}$$

The condition $Q^i_\tau \equiv 0$ means that the coordinate line ξ^i lies in a plane. Thus, the quantity Q^i_τ is a measure of the departure of the coordinate line ξ^i from a plane line.

3.6 Characteristics of Faces of Three-Dimensional Cells

A coordinate grid in a three-dimensional domain X^3 is composed of three-dimensional curvilinear hexahedral cells which are images of elementary cubes obtained through a coordinate transformation

$$\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^3 \to X^3.$$

The boundary of each cell is segmented into six curvilinear quadrilaterals, through which some characteristics of the cell can be defined. This section describes some important quality measures of the faces of three-dimensional coordinate cells.

3.6.1 Cell Face Skewness

The skewness of a cell face is described through the angle between the two tangent vectors defining the cell face. Let the cell face lie in the surface $\xi^l = \text{const}$; the tangent vectors of the surface are then the vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$, $i = l+1$, $j = l+2$, with the identification convention for the index m that m is equivalent to $m \pm 3$. One of the cell face skewness characteristics can be determined as the square of the cosine of the angle between the vectors. Thus, for a fixed l,

$$Q^l_{\text{sk},1} = \cos^2\theta = \frac{(g_{ij})^2}{g_{ii} g_{jj}} , \qquad i = l+1 , \qquad j = l+2 . \tag{3.52}$$

Another expression for the cell face skewness is specified by the cotangent squared of the angle θ:

$$Q^l_{\mathrm{sk},2} = \cot^2\theta = \frac{(g_{ij})^2}{g_{ii}g_{jj} - (g_{ij})^2} \ , \qquad i = l + 1 \ , \qquad j = l + 2 \ . \tag{3.53}$$

Taking into account the relations (2.29) and (2.33), this can also be written in the form

$$Q^l_{\mathrm{sk},2} = \frac{(g_{ij})^2}{(\boldsymbol{x}_{\xi^i} \times \boldsymbol{x}_{\xi^j})^2} = \frac{(g_{ij})^2}{gg^{ll}} \ , \qquad i = l + 1 \ , \qquad j = l + 2 \ .$$

Since $(g_{ij})^2 = g_{ii}g_{jj}(1 - \sin^2\theta)$, we also obtain

$$Q^l_{\mathrm{sk},2} = \frac{g_{ii}g_{jj}}{(\boldsymbol{x}_{\xi^i} \times \boldsymbol{x}_{\xi^j})^2} - 1 = \frac{g_{ii}g_{jj}}{gg^{ll}} \ , \qquad i = l + 1 \ , \qquad j = l + 2 \ .$$

The quantities for the grid face skewness introduced above equal zero when the edges of the cell face are orthogonal. Therefore, these quantities characterize the departure of the cell face from a rectangle. One more characteristic of the cell face nonorthogonality is defined as square of the dot product of the vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$:

$$Q^l_{\mathrm{o},1} = (g_{ij})^2 \ , \qquad i = l + 1 \ , \qquad j = l + 2 \ . \tag{3.54}$$

3.6.2 Face Aspect-Ratio

A measure of the aspect-ratio of the cell face formed by the tangent vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$ is defined through the diagonal elements g_{ii} and g_{jj} of the covariant metric tensor $\{g_{km}\},\ k, m = 1, 2, 3$. One form of this measure is given by the expression

$$Q^l_{\mathrm{as}} = \frac{g_{ii}}{g_{jj}} + \frac{g_{jj}}{g_{ii}} = \frac{(g_{ii} + g_{jj})^2}{g_{ii}g_{jj}} - 2 \ , \tag{3.55}$$

where $i = l + 1,\ j = l + 2$, and $m + 3$ is equivalent to $\pm m$. We have the inequality $Q^l_{\mathrm{as}} \geq 2$, which is an equality if and only if $g_{ii} = g_{jj}$, i.e. the parallelogram formed by the vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$ is a rhombus. Thus, (3.55) is a measure of the departure of the cell from a rhombus.

3.6.3 Cell Face Area Squared

The square of the area of the face of the basic parallelepiped formed by the two tangential vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$ is expressed as follows:

$$Q^l_{\mathrm{ar}} = |\boldsymbol{x}_{\xi^i}|^2 |\boldsymbol{x}_{\xi^j}|^2 \sin^2 \theta = g_{ii} g_{jj} - (g_{ij})^2 \ , \quad i = l+1 \ , \quad j = l+2 \ , \tag{3.56}$$

where θ is the angle of intersection of the vectors and i and j are chosen to satisfy the condition $l \neq i$ and $l \neq j$. Taking advantage of (2.29) and (2.33), we see that

$$Q^l_{\mathrm{ar}} = |\boldsymbol{x}_{\xi^i} \times \boldsymbol{x}_{\xi^j}|^2 = g|\nabla \boldsymbol{\xi}^l|^2 = g g^{ll} \ , \qquad l \text{ fixed} \ . \tag{3.57}$$

As the square of the area of the coordinate cell face which corresponds to the parallelogram defined by the vectors $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^j}$ equals $h^2 Q_{\mathrm{ar}} + O(h^3)$, the quantity Q^l_{ar} can be applied to characterize the area of the cell face.

3.6.4 Cell Face Warping

Measures of the cell face warping are obtained through the curvatures of the coordinate surface on which the face lies. Let this be the coordinate surface $\xi^3 = \xi^3_0$. Then, a natural parametrization $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^2 \to R^3$, $\boldsymbol{\xi} = (\xi^1, \xi^2)$ of the surface is represented by $\boldsymbol{x}(\xi^1, \xi^2, \xi^3_0)$.

Mean Curvature of the Coordinate Surface

Twice the mean curvature of the coordinate surface is defined through the formula (3.23) or (3.24) as

$$2K_{3,m} = g^{ij}_{\xi x} b_{ij}, \qquad i, j = 1, 2 \ , \tag{3.58}$$

where $b_{ij} = \boldsymbol{x}_{\xi^i \xi^j} \cdot \boldsymbol{n}$. It is obvious that the contravariant metric tensor $\{g^{ij}_{\xi x}\}$ of the surface $\xi^3 = \xi^3_0$ in the coordinates ξ^1, ξ^2 is the inverse of the 2×2 matrix $\{g^{x\xi}_{ij}\}$ whose elements are the elements of the volume metric tensor $\{g_{ij}\}$ with the indices $i, j = 1, 2$, i.e.

$$g^{x\xi}_{ij} = g_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j} \ , \qquad i, j = 1, 2.$$

Therefore, using (3.16) and (2.33), we have

$$g^{ij}_{\xi x} = (-1)^{i+j} g_{3-i\ 3-j} / (\boldsymbol{x}_{\xi^1} \times \boldsymbol{x}_{\xi^2})^2 = \frac{(-1)^{i+j} g^{33}}{g} g_{3-i\ 3-j} \ , \qquad i, j = 1, 2 \ ,$$

without summation over i or j. Also, it is clear that

$$\boldsymbol{n} = \frac{1}{\sqrt{g^{33}}} \nabla \xi^3 ,$$

and consequently the coefficients of the second fundamental form of the coordinate surface $\xi^3 = \xi_0^3$ are expressed as follows:

$$b_{ij} = \frac{1}{\sqrt{g^{33}}} \boldsymbol{x}_{\xi^i \xi^j} \cdot \nabla \xi^3 = \frac{1}{\sqrt{g^{33}}} \Gamma_{ij}^3 .$$

Thus, (3.58) results in

$$2K_{3,m} = \frac{(-1)^{i+j}\sqrt{g^{33}}}{g} g_{3-i\ 3-j} \Gamma_{ij}^3 , \qquad i, j = 1, 2 .$$

Analogously, we obtain a general formula for the coefficients of the second fundamental form of the coordinate surface $\xi^l = \xi_0^l,\ l = 1, 2, 3$:

$$b_{ij} = \frac{1}{\sqrt{g^{ll}}} \Gamma_{l+i\ l+j}^l , \qquad i, j = 1, 2 , \tag{3.59}$$

with l fixed and where m is equivalent to $m \pm 3$. Thus, twice the mean curvature of the coordinate surface $\xi^l = \xi_0^l,\ l = 1, 2$, is expressed by

$$2K_{l,m} = \frac{(-1)^{i+j}\sqrt{g^{ll}}}{g} g_{l-i\ l-j} \Gamma_{l+i\ l+j}^l , \qquad i, j = 1, 2 , \tag{3.60}$$

with l fixed.

Gaussian Curvature of the Coordinate Surface

Taking advantage of (3.22) and (3.59), the Gaussian curvature of the coordinate surface $\xi^l = \xi_0^l$ can be expressed as follows:

$$K_{l,\mathrm{G}} = \frac{\sqrt{g^{ll}}}{g} [\Gamma_{l+1\ l+1}^l \Gamma_{l+2\ l+2}^l - (\Gamma_{l+1\ l+2}^l)^2] , \tag{3.61}$$

with the index l fixed.

Measures of Face Warping

The quantities which measure the warping of the face of a three-dimensional cell are obtained through the coefficients of the second fundamental form or through the mean and Gaussian curvatures of a coordinate surface containing the face. Let this be the surface $\xi^l = \xi_0^l$. Then, taking advantage of (3.60) and (3.61), the measures may be expressed as follows:

$$Q^l_{\mathrm{w},1} = (K_{l,m})^2 = \frac{g^{ll}}{g^2}[(-1)^{i+j} g_{l-i\ l-j} \Gamma^l_{l+i\ l+j}]^2 \ ,$$
$$Q^l_{\mathrm{w},2} = (K_{l,g})^2 = \frac{g^{ll}}{g^2}[\Gamma^l_{l+1\ l+1}\Gamma^l_{l+2\ l+2} - (\Gamma^l_{l+1\ l+2})^2] \ , \tag{3.62}$$

with l fixed.

Equation (3.59) for the second fundamental form of the surface $\xi^l = \xi^l_0$ also gives an expression for the third measure of the cell face warpness:

$$Q^l_{\mathrm{w},3} = \sum_{i,j=1}^{2} (b_{ij})^2 = \frac{1}{g^{ll}} \sum_{i,j=1}^{2} (\Gamma^l_{l+i\ l+j})^2 \ , \quad l \text{ fixed} \ . \tag{3.63}$$

3.7 Characteristics of Grid Cells

Cell features are described by the cell volume (area in two dimensions) and by the characteristics of the cell edges and faces.

3.7.1 Cell Aspect-Ratio

A measure of the aspect-ratio of a three-dimensional cell is formulated through the measures of the aspect-ratio of its faces described by (3.55). The simplest formulation is provided by summing these measures, which results in

$$Q_{\mathrm{as}} = \sum_{l=1}^{3} Q^l_{\mathrm{as}} \ . \tag{3.64}$$

3.7.2 Square of Cell Volume

The characteristic related to the square of the cell volume is

$$Q_V = g = \det\{g_{ij}\} = I_n \ . \tag{3.65}$$

In three dimensions, we also obtain, from (2.32),

$$Q_V = [\boldsymbol{x}_{\xi^1} \cdot (\boldsymbol{x}_{\xi^2} \times \boldsymbol{x}_{\xi^3})]^2 \ .$$

3.7.3 Cell Area Squared

We denote by Q_{ar} the sum of the quantities Q^{ij}_{ar}, $i \neq j$, from (3.57). These quantities are the area characteristics of the faces of a three-dimensional cell; thus, in accordance with (3.33), the magnitude Q_{ar} coincides with the invariant I_2 :

$$Q_{\text{ar}} = \sum_{i=1}^{3} g g^{ii} = I_2 \ . \tag{3.66}$$

3.7.4 Cell Skewness

One way to describe the cell skewness characteristics of three-dimensional grids utilizes the angles between the tangential vectors in the forms of the corresponding expressions (3.52) and (3.53) introduced for the formulation of the face skewness. For example, summation of these quantities gives the following expressions for the cell skewness measures:

$$\begin{aligned} Q_{\text{sk},1} &= \frac{(g_{12})^2}{g_{11}g_{22}} + \frac{(g_{23})^2}{g_{22}g_{33}} + \frac{(g_{13})^2}{g_{11}g_{33}} \\ Q_{\text{sk},2} &= \frac{(g_{12})^2}{g_{11}g_{22} - (g_{12})^2} + \frac{(g_{13})^2}{g_{11}g_{33} - (g_{13})^2} + \frac{(g_{23})^2}{g_{22}g_{33} - (g_{23})^2} \\ &= \frac{1}{g}\left(\frac{(g_{12})^2}{g^{33}} + \frac{(g_{13})^2}{g^{22}} + \frac{(g_{23})^2}{g^{11}}\right) . \end{aligned} \tag{3.67}$$

Here, $Q_{\text{sk},1}$ is the sum of the squares of the cosines of the angles between the edges of the cell, while $Q_{\text{sk},2}$ is the sum of the squares of the cotangents of the angles.

Other quantities for expressing the three-dimensional cell skewness can be defined through the angles between the normals to the coordinate surfaces. Any normal to the coordinate surface $\xi^i = \xi^i_0$ is parallel to the normal vector $\nabla \xi^i$. Therefore, the cell skewness can be derived through the angles between the base normal vectors $\nabla \xi^i$. The quantity

$$\frac{(\nabla \xi^i \cdot \nabla \xi^j)^2}{g^{ii}g^{jj}} = \frac{(g^{ij})^2}{g^{ii}g^{jj}} \ , \qquad i, j \text{ fixed}$$

is the cosine squared of the angle between the respective faces of the coordinate cell. This characteristic is a dimensionless magnitude. The sum of such quantities is the third characteristic of the three-dimensional cell skewness:

$$Q_{\text{sk},3} = \frac{(g^{12})^2}{g^{11}g^{22}} + \frac{(g^{13})^2}{g^{11}g^{33}} + \frac{(g^{23})^2}{g^{22}g^{33}} \ . \tag{3.68}$$

Another dimensionless quantity which characterizes the mutual skewness of two faces of the cell is the cotangent squared of the angle between the normal vectors $\nabla\xi^i$ and $\nabla\xi^j$:

$$\frac{(\nabla\xi^i\cdot\nabla\xi^j)^2}{|\nabla\xi^i\times\nabla\xi^j|^2}=\frac{g(g^{ij})^2}{g_{kk}}=\frac{(g^{ij})^2}{g^{ii}g^{jj}-(g^{ij})^2}\ ,$$

where (i,j,k) are cyclic and fixed. The summation of this over k defines the fourth grid skewness characteristic

$$\begin{aligned}Q_{\mathrm{sk},4}&=\frac{(g^{12})^2}{g^{11}g^{22}-(g^{12})^2}+\frac{(g^{13})^2}{g^{11}g^{33}-(g^{13})^2}+\frac{(g^{23})^2}{g^{22}g^{33}-(g^{23})^2}\\&=g\left(\frac{(g^{12})^2}{g_{33}}+\frac{(g^{13})^2}{g_{22}}+\frac{(g^{23})^2}{g_{11}}\right)\ .\end{aligned}\tag{3.69}$$

Note that the three-dimensional cell skewness quantities $Q_{\mathrm{sk},1}$ and $Q_{\mathrm{sk},3}$ can be readily extended to arbitrary dimensions $n\geq 2$.

3.7.5 Characteristics of Nonorthogonality

The quantities $Q_{\mathrm{sk},i},\ i=1,2,3,4$, from (3.67)–(3.69) reach their minimum values equal to zero only when the three-dimensional transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is orthogonal at the respective point, and vice-versa. Therefore, these quantities, which provide the possibility to detect orthogonal grids, may be considered as some measures of grid nonorthogonality.

Other quantities characterizing the departure of a three-dimensional grid from an orthogonal one are as follows:

$$\begin{aligned}Q_{\mathrm{o},1}&=\frac{g_{11}g_{22}g_{33}}{g}\ ,\\Q_{\mathrm{o},2}&=g\,(g^{11}g^{22}g^{33})\ .\end{aligned}\tag{3.70}$$

Obviously, these quantities $Q_{\mathrm{o},1}$ and $Q_{\mathrm{o},2}$ are dimensionless and reach their minimum equal to 1 if and only if the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is orthogonal.

The sum of the squares of the nondiagonal elements of the covariant metric tensor $\{g_{ij}\}$ yields another characteristic of cell nonorthogonality,

$$Q_{\mathrm{o},3}=(g_{12})^2+(g_{13})^2+(g_{23})^2\ .\tag{3.71}$$

An analogous formulation is given through the elements of the contravariant metric tensor,

$$Q_{\mathrm{o},4}=(g^{12})^2+(g^{13})^2+(g^{23})^2\ .\tag{3.72}$$

Note that, in contrast to $Q_{o,1}$ and $Q_{o,2}$, the quantities $Q_{o,3}$ and $Q_{o,4}$ are dimensionally heterogeneous.

3.7.6 Grid Density

The invariants of the tensor $\{g_{ij}\}$ can be useful for specifying some characteristics of grid quality. For example, one important characteristic describing the concentration of grid nodes can be derived from the ratio I_{n-1}/I_n.

In order to show this, we first note that in accordance with the geometrical interpretation of the invariants given in Sect. 3.4.2, we can write

$$\frac{I_{n-1}}{I_n} = \sum_{m=1}^{n} \left(V_m^{n-1}\right)^2 \Big/ \left(V^n\right)^2 , \tag{3.73}$$

where V_m^{n-1} is the space of the boundary segment $\xi^m = \text{const}$ of the basic parallelepiped defined by the tangential vectors $\boldsymbol{x}_{\xi^i},\ i = 1, \ldots, n$.

It is evident that

$$V^n = d_m V_m^{n-1}, \qquad m = 1, \ldots, n ,$$

where d_m is the distance between the vertex of the tangential vector $\boldsymbol{x}_{\xi^m}$ and the $(n-1)$-dimensional plane P^{n-1} spanned by the vectors $\boldsymbol{x}_{\xi^i},\ i \neq m$. Hence, from (3.73),

$$\frac{I_{n-1}}{I_n} = \sum_{m=1}^{n} (1/d^m)^2 . \tag{3.74}$$

Now let us consider two grid surfaces $\xi^m = c$ and $\xi^m = c+h$ obtained by mapping a uniform rectangular grid with a step size h in the computational domain Ξ^n onto X^n. Let us denote by l_m the distance between a node on the coordinate surface $\xi^m = c$ and the nearest node on the surface $\xi^m = c + h$ (Fig. 3.5). We have

$$l_m = d_m h + O(h)^2$$

and therefore, from (3.74),

$$\frac{I_{n-1}}{I_n} = \sum_{m=1}^{n} (h/l_m)^2 + O(h) .$$

The quantity $(h/l_m)^2$ increases if the grid nodes cluster in the direction normal to the surface $\xi^m = c$. Therefore, this quantity can be considered as some measure of the grid concentration in the normal direction and, consequently, the magnitude $1/d_m$

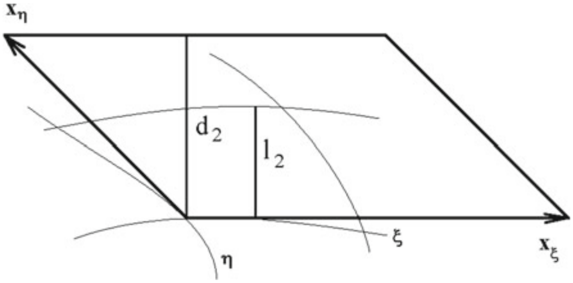


Fig. 3.5 Illustration of invariants

means the density of the grid concentration in the $\nabla\xi^m$ direction. In particular, we readily see that $1/d_m = \sqrt{g^{mm}}$, with m fixed. Thus, the expression (3.74) defines a measure of the grid density in all directions. We denote this quantity by Q_{cn}, where the subscript "cn" represents "concentration". Note that, in accordance with (3.31), this measure can be expressed as follows:

$$Q_{\text{cn}} = \frac{I_{n-1}}{I_n} = g^{11} + \cdots + g^{nn} \ . \tag{3.75}$$

3.7.7 *Characteristics of Deviation from Conformality*

Conformal coordinate transformations are distinguished by the fact that the Jacobi matrix J is orthonormal, and consequently the metric tensor $\{g_{ij}\}$ is a multiple of the unit matrix:

$$\{g_{ij}\} = g(\boldsymbol{\xi})I = g(\boldsymbol{\xi})\{\delta^i_j\} \ , \qquad i, j = 1, \ldots, n \ .$$

The cells of the coordinate grid derived from the conformal mapping $\boldsymbol{x}(\boldsymbol{\xi})$ are close to n-dimensional cubes (squares in two dimensions). Grids with such cells are attractive from the computational point of view. Therefore, it is desirable to define simple grid quantities which can allow one to detect grids whose cells are close to n-dimensional cubes. It is clear that the condition of conformality can be described by the system

$$\begin{aligned} &g_{ij} = 0 \ , \qquad i \neq j \ , \\ &g_{11} = g_{22} = \cdots = g_{nn} \ . \end{aligned}$$

These relations give rise to a natural quantity

$$Q = \sum_{i \neq j} (g_{ij})^2 + \sum_{i=2}^{n} (g_{ii} - g_{11})^2 \ ,$$

which is zero if and only if the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal. So, this quantity can help one to detect when the grid is conformal. However, the above formula is too cumbersome, as well as being dimensionally heterogeneous. More compact expressions for the analysis of the conformality or nonconformality of grid cells and for the formulation of algorithms to construct nearly conformal grids are obtained through the use of the metric-tensor invariants.

Two-Dimensional Space

The departure from conformality of the two-dimensional transformation $\boldsymbol{x}(\xi) : \Xi^2 \to X^2$ is expressed by the quantity

$$Q_{\mathrm{cf},1} = \frac{I_1}{\sqrt{I_2}} = \frac{|\boldsymbol{x}_{\xi^1}|^2 + |\boldsymbol{x}_{\xi^2}|^2}{|\boldsymbol{x}_{\xi^1}||\boldsymbol{r}_{\xi^2}||\sin\theta|} = \frac{g_{11} + g_{22}}{\sqrt{g_{11}}\sqrt{g_{22}}|\sin\theta|}\ , \tag{3.76}$$

where θ is the angle between the tangent vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$. Since

$$Q_{\mathrm{cf},1} \geq \frac{g_{11} + g_{22}}{\sqrt{g_{11}g_{22}}}\ ,$$

it is clear that the value of $I_1/\sqrt{I_2}$ exceeds 2. The minimum value 2 is achieved only if $g_{11} = g_{22}$ and $\theta = \pi/2$, i.e. when the parallelogram with sides defined by the vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$ is a square. Thus, the characteristic $Q_{\mathrm{cf},1}$ allows one to state with certainty when the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal at a point $\boldsymbol{\xi}$, namely when $Q_{\mathrm{cf},1}(\boldsymbol{\xi}) = 2$. Therefore, in the two-dimensional case, the quantity

$$Q_{\mathrm{cf},1} - 2 = I_1/\sqrt{I_2} - 2$$

reflects some measure of the deviation of the cell from a square. We see that the quantity $Q_{\mathrm{cf},1}$ given by (3.76) is dimensionally homogeneous.

Through the quantity $Q_{\mathrm{cf},1}$, we can also estimate the bounds of the aspect ratio of the two-dimensional cell and the angle between the edges of this cell.

Evaluation of the Cell Angles

First, we obtain an estimate of the angle between the cell edges. From (3.76), we have

$$\sin^2\theta = \frac{(F^2+1)^2}{F^2}/Q^2_{\mathrm{cf},1}\ , \tag{3.77}$$

where $F^2 = g_{11}/g_{22}$. As $(F^2+1)^2/F^2 \geq 4$, we have from (3.77) that

$$\sin^2\theta \geq 4/Q^2_{\mathrm{cf},1} \tag{3.78}$$

and, accordingly, we obtain the following estimate for the angle θ:

$$\pi - \arcsin(2/Q_{\mathrm{cf},1}) \geq \theta \geq \arcsin(2/Q_{\mathrm{cf},1}) \,. \tag{3.79}$$

From (3.77), we find that the minimum value $4/Q^2_{\mathrm{cf},1}$ of $\sin^2\theta$ for a fixed value of $Q_{\mathrm{cf},1}$ is achieved when $F = 1$, i.e. when the parallelogram is the rhombus.

Although it is desirable to generate orthogonal grids, a departure from orthogonality is practically inevitable when grid adaptation is performed. Commonly, this departure is required to be restricted to 45°. Beyond this range, the contribution of the grid skewness to the truncation error may become unacceptable. The inequality (3.79) shows that this barrier of 45° is not broken if $Q_{\mathrm{cf},1} \leq 2\sqrt{2}$.

Evaluation of the Cell Aspect Ratio

Now we estimate the quantity $F = \sqrt{g_{11}/g_{22}}$. The quantity F, called the cell aspect-ratio, is the ratio of the lengths of the edges of the cell. By computing F from (3.77), we obtain

$$F = \frac{\alpha}{2} - 1 \pm \sqrt{\frac{\alpha^2}{4} - \alpha} \,, \qquad \alpha = Q^2_{\mathrm{cf},1}\sin^2\theta \,. \tag{3.80}$$

Equation (3.80) gives two values of the cell aspect-ratio,

$$F_1 = \frac{\alpha}{2} - 1 + \sqrt{\frac{\alpha^2}{4} - \alpha} \quad \text{and} \quad F_2 = \frac{\alpha}{2} - 1 - \sqrt{\frac{\alpha^2}{4} - \alpha} \,,$$

satisfying the relation $F_1F_2 = 1$. We find that

$$F_1 = \max(\sqrt{g_{11}/g_{22}},\ \sqrt{g_{22}/g_{11}})$$

and

$$F_2 = \min(\sqrt{g_{11}/g_{22}},\ \sqrt{g_{22}/g_{11}}) \,.$$

Thus,

$$\frac{\alpha}{2} - 1 - \sqrt{\frac{\alpha^2}{4} - \alpha} \leq F_i \leq \frac{\alpha}{2} - 1 + \sqrt{\frac{\alpha^2}{4} - \alpha} \,, \qquad i = 1, 2 \,, \tag{3.81}$$

and consequently

$$2 \leq F_i + 1/F_i \leq \alpha - 2 \,, \qquad i = 1, 2 \,. \tag{3.82}$$

As $Q^2_{\mathrm{cf},1} \geq \alpha \geq 4$, from (3.78), we also obtain, from (3.81) and (3.82), the following upper and lower estimates of the aspect ratios $F_i,\ i = 1, 2$, which depend only on the quantity $Q_{\mathrm{cf},1}$:

$$\frac{Q_{\mathrm{cf},1}^2}{2} - 1 - Q_{\mathrm{cf},1}\sqrt{\frac{Q_{\mathrm{cf},1}^2}{4} - 1} \le F_i \le \frac{Q_{\mathrm{cf},1}^2}{2} - 1 + Q_{\mathrm{cf},1}\sqrt{\frac{Q_{\mathrm{cf},1}^2}{4} - 1}\ , \tag{3.83}$$

and

$$2 \le F_i + 1/F_i \le Q_{\mathrm{cf},1}^2 - 2\ , \qquad i = 1, 2\ . \tag{3.84}$$

The maximum value of F_i for a given value of $Q_{\mathrm{cf},1}$ is realized when $\sin^2\theta = 1$, i.e. the parallelogram is a rectangle.

Three-Dimensional Space

In three-dimensional space, the deviation from conformality can be described by the dimensionless magnitude

$$Q_{\mathrm{cf},1} = (g)^{1/3}(g^{11} + g^{22} + g^{33})\ , \tag{3.85}$$

which, in accordance with (3.33), is expressed by means of the invariants I_2 and I_3 as follows:

$$Q_{\mathrm{cf},1} = I_2/(I_3)^{2/3}\ . \tag{3.86}$$

The value of (3.86) reaches its minimum only if

$$g^{11} = g^{22} = g^{33} \qquad \text{and} \qquad g^{-1} = g^{11}g^{22}g^{33}\ , \tag{3.87}$$

i.e. when the parallelogram defined by the basic normal vectors $\nabla\xi^i$ is a cube. To prove this fact, we note that

$$\frac{1}{g} \le g^{11}g^{22}g^{33}\ .$$

Therefore, from (3.85),

$$Q_{\mathrm{cf},1} \ge \frac{g^{11} + g^{22} + g^{33}}{\sqrt[3]{g^{11}g^{22}g^{33}}}$$

and, taking into account the general inequality for arbitrary positive numbers $a_1, \ldots, a_n$

$$\frac{1}{n}\sum_{i=1}^{n} a_i \ge \sqrt[n]{\prod_{i=1}^{n} a_i}\ ,$$

we find that $Q_{\mathrm{cf},1} \ge 3$. Obviously, $Q_{\mathrm{cf},1} = 3$ when the relations (3.87) are satified. From (2.35),

$$\frac{1}{g} = |\nabla\xi^1 \cdot \nabla\xi^2 \times \nabla\xi^3|^2$$

and therefore (3.87) is satisfied only when the normal vectors $\nabla\xi^i$, $i = 1, 2, 3$, are orthogonal to each other and have the same length. But then this is valid for the base tangential vectors $\boldsymbol{x}_{\xi^i}$, $i = 1, 2, 3$, as well. Thus, (3.87) is satisfied only when the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal.

In the same manner as in the two-dimensional case, one can derive bounds on the angles of the parallelepiped and on the ratio of the lengths of its edges that depend on the quantity $Q_{\mathrm{cf},1}$.

Generalization to Arbitrary Dimensions

Analogously, in the n-dimensional case, a local measure of the deviation of the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ from a conformal one is expressed by the quantity $Q_{\mathrm{cf},1} - n$, where

$$Q_{\mathrm{cf},1} = I_{n-1}/(I_n)^{1-1/n} = g^{1/n}(g^{11} + \cdots + g^{nn}) \,. \tag{3.88}$$

The quantity $Q_{\mathrm{cf},1}$ equals n if and only if the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal.

Another local characteristic of the deviation from conformality is described by the quantity $Q_{\mathrm{cf},2} - n$, where

$$Q_{\mathrm{cf},2} = I_1/(I_n)^{1/n} \,. \tag{3.89}$$

As for $Q_{\mathrm{cf},1}$, we can show that $Q_{\mathrm{cf},2} \geq n$ and that $Q_{\mathrm{cf},2} = n$ if the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal at the point under consideration. Note also that $Q_{\mathrm{cf},1} = Q_{\mathrm{cf},2}$ in two dimensions.

3.7.8 Grid Eccentricity

One grid eccentricity characteristic is defined by summing the squares of the coordinate-line eccentricities (3.36). Thus, the quantity

$$Q_{\epsilon,1} = \sum_{i=1}^{n} \left(\frac{\partial}{\partial \xi^i} \ln \sqrt{g_{ii}} \right)^2 \tag{3.90}$$

is a measure of the change of the lengths of all of the grid cell edges.

A similar characteristic of eccentricity can be formulated through the terms g^{ii}, namely

$$Q_{\epsilon,2} = \sum_{i=1}^{n} \left(\frac{\partial}{\partial x^i} \ln \sqrt{g^{ii}} \right)^2 \,. \tag{3.91}$$

3.7.9 Measures of Grid Warping and Grid Torsion

In the same way as for grid eccentricity, we may formulate measures of three-dimensional grid warping by summing the surface-coordinate characteristics (3.62) and (3.63). As a result, we obtain

$$\begin{aligned} Q_{\mathrm{w},1} &= \frac{1}{g^2}\sum_{l=1}^{3} g^{ll}\left((-1)^{i+j} g_{l-i\ l-j}\Gamma^{l}_{l+i\ l+j}\right)^2 \ , \\ Q_{\mathrm{w},2} &= \frac{1}{g^2}\sum_{l=1}^{3} g^{ll}\left[\Gamma^{l}_{l+1\ l+1}\Gamma^{l}_{l+2\ l+2} - \left(\Gamma^{l}_{l+1\ l+1}\right)^2\right] \ , \\ Q_{\mathrm{w},3} &= \sum_{l=1}^{3}\sum_{i,j=1}^{2}\frac{1}{g^{ll}}\left(\Gamma^{l}_{l+i\ l+j}\right)^2 \ . \end{aligned} \tag{3.92}$$

The measure of grid torsion is formulated by summing the torsion measures (3.51) of the coordinate lines $\xi^i,\ i = 1, 2, 3$:

$$Q_\tau = \sum_{i=1}^{3} Q^i_\tau \ . \tag{3.93}$$

3.7.10 Quality Measures of Simplexes

The quantities which are applied to measure the quality of triangles and tetrahedrons are the following:

(1) the maximum edge length H,
(2) the minimum edge length h,
(3) the circum-radius R,
(4) the inradius r.

There are four deformation measures that allow one to characterize the quality of triangular and tetrahedral cells:

$$Q_{\mathrm{d},1} = \frac{H}{r}\ , \qquad Q_{\mathrm{d},2} = \frac{R}{H}\ , \qquad Q_{\mathrm{d},3} = \frac{H}{h}\ , \qquad Q_{\mathrm{d},4} = \frac{R}{r}\ .$$

The uniformity condition for a cell is satisfied when $Q_{\mathrm{d},1} = O(1)$ or $Q_{\mathrm{d},4} = O(1)$.

Examples of poorly shaped cells are shown in Fig. 3.6. Cases a and c correspond to needle-shaped cells. Figure 3.6d shows a wedge-shaped cell, while Fig. 3.6b, e show sliver-shaped cells.

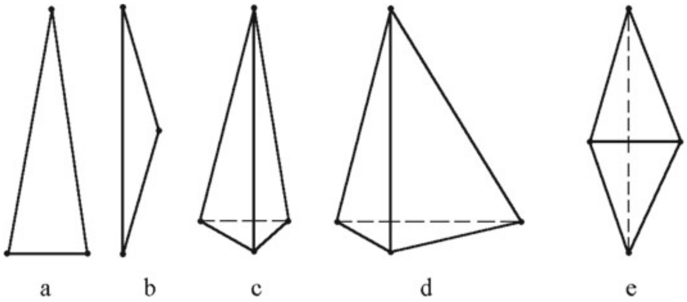


Fig. 3.6 Examples of poorly shaped triangles (**a**, **b**) and tetrahedrons (**c**, **d**, **e**)

The cell is excessively deformed if $Q_{\mathrm{d},1} \gg 1$. In this case, the cell has either a very acute or a very obtuse angle. The former case corresponds to $Q_{\mathrm{d},2} = O(1)$, $Q_{\mathrm{d},3} \gg 1$ (Fig. 3.6a, c, d), while the latter corresponds to $Q_{\mathrm{d},4} \gg 1$, $Q_{\mathrm{d},3} = O(1)$ (Fig. 3.6b, e). The condition $Q_{\mathrm{d},2} = O(1)$ precludes obtuse angles.

3.8 Comments

Various aspects of mesh quality were surveyed by Knupp (2001, 2007). The introduction of metric-tensor invariants to describe some of the qualitative properties of grids was originally proposed by Jacquotte (1987). The grid measures in terms of the invariants and their relations described in this chapter were obtained by the author.

Prokopov (1989) introduced the dimensionless characteristics of two-dimensional cells.

Triangular elements were extensively analyzed by Field (2000).

Some questions concerned with the assessment of the contribution of the grid quality properties to the accuracy of solutions obtained using the grid were discussed by Kerlic and Klopfer (1982), Mastin (1982), Lee and Tsuei (1992), and Huang and Prosperetti (1994).

Discrete length, area, and orthogonality grid measures using averages and deviations were formulated by Steinberg and Roache (1992).

Babuŝka and Aziz (1976) have shown that the minimum-angle condition in a planar triangulation is too restrictive and can be replaced by a condition that limits the maximum allowable angle. Also, the influence of grid quality measures on solution accuracy was discussed by Knupp (2007) and Shewchuck (2002).

Measures to quantify the shape of triangles and tetrahedrons were introduced by Field (1986), Baker (1989), Cougny et al. (1990), and Dannenlongue and Tanguy (1991).

A brief overview of tetrahedron quality measures with a comparison of the fidelity of these measures to a set of distortion sensitivity tests, as well as a comparison of the computational expense of such measures, was given by Parthasarathy et al. (1993).

An overview of several element quality metrics was given by Field (2000). Chen et al. (2003) extended the angle-based quality metric originally defined by Lee and Lo (1994) for use in the optimization of meshes consisting of triangles and quadrilaterals. They also extended the formulation of cell unfolding by adding a barrier part to their quality functional (2003).

A special tetrahedron shape measure was given by Liu and Joe (1994a). It is based on eigenvalues of the metric tensor for the transformation between a tetrahedron and a regular reference tetrahedron. The geometric explanation of this measure is that it characterizes the shape of the inscribed ellipsoid. Another shape regularity quality of a triangle was given Bank and Xu (1996) and Bank and Smith (1997). They showed that the quality has circular level sets, when considered a function of the location of one vertex of a triangle with the other two vertices fixed. Three tetrahedron measures – the minimum solid angle, the radius ratio, and the mean ratio – and their relations were discussed by Liu and Joe (1994b).

An algorithm for construction of solution-adapted triangular meshes within an optimization framework was considered by Buscaglia and Dari (1997). Here, the optimized quality measure is a product of "shape" quality and a function of mesh size.

A local cell quality measure as a function of Jacobian matrix and combined element-shape and size-control metrics for different cell types was analyzed by Garanzha (2000) and Branets and Carey (2005).

Dompierre et al. (2005) analyzed and generalized several simplex shape measures documented in the literature and used them for mesh adaptation and mesh optimization. Conclusions were drawn on the choice of simplex shape measures to control mesh optimization.

References

Babuska, I., & Aziz, A. K. (1976). On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, *13*(2), 214–226.

Baker, T. J. (1989). Automatic mesh generation for complex three-dimensional region using a constrained Delaunay triangulation. *Engineering with Computers*, *5*, 161–175.

Bank, R. E., & Smith, R. K. (1997). Mesh smoothing using a posteriori error estimates. *SIAM Journal on Numerical Analysis*, *34*, 979–997.

Bank, R. E., & Xu, J. (1996). An algorithm for coarsening unstructured meshes. *Numerische Mathematik*, *73*, 1–36.

Bobenko, A. I., & Springborn, B. A. (2005). A discrete Laplace-Beltrami operator for simplicial surfaces, preprint. arXiv:math.DG/0503219.

Branets, L., & Carey, G. F. (2005). A local cell quality metric and variational grid smoothing algorithm. *Engineering with Computers*, *21*, 19–28.

Buscaglia, G. C., & Dari, E. A. (1997). Anisotropic mesh optimization and its application in adaptivity. *International Journal for Numerical Methods in Engineering*, *40*, 4119–4136.

Chen, Z., Tristano, J. R., & Kwok, W. (2003). Combined Laplacian and optimization-based smoothing for quadratic mixed surface meshes. In *Proceedings of the 12th International Meshing Roundtable* (pp. 361–370).

Cougny, H. L., Shephard, M. S., & Georges, M. K. (1990). *Explicit node point smoothing within the octree mesh generator*. Report 10-1990, SCOREC, RPI, Troy, NY.

Dannenlongue, H. H., & Tanguy, P. A. (1991). Three-dimensional adaptive finite element computations and applications to non-newtonian flows. *International Journal for Numerical Methods in Fluids*, *13*, 145–165.

Dompierre, J., Vallet, M.-G., Labbe, P., & Guibault, F. (2005). An analysis of simplex shape measures for anisotropic meshes. *Computer Methods in Applied Mechanics and Engineering*, *194*, 4895–4914.

Field, D. A. (1986). Implementing Watson's algorithm in three dimensions. In *Proceedings of the 2nd Annual Symposium on Computational Geometry* (pp. 246–259).

Field, D. A. (2000). Qualitative measures for initial meshes. *International Journal for Numerical Methods in Engineering*, *47*, 887–906.

Garanzha, V. A. (2000). Barrier variational generation of quasi-isimetric grids. *Computational Mathematics and Mathematical Physics*, *40*, 1617–1637.

Huang, H., & Prosperetti, A. (1994). Effect of grid orthogonality on the solution accuracy of the two-dimensional convection diffusion equation. *Numerical Heat Transfer, Part B*, *26*, 1–20.

Jacquotte, O.-P. (1987). A mechanical model for a new grid generation method in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, *66*, 323–338.

Kerlic, C. D., & Klopfer, G. H. (1982). Assessing the quality of curvilinear meshes by decomposing the Jacobian Matrix. *Applied Mathematics and Computation*, *10*(11), 787.

Knupp, P. M. (2001). Algebraic mesh quality metrics. *SIAM Journal on Scientific Computing*, *23*, 193–218.

Knupp, P. M. (2007). *Remarks on mesh quality*. Reno, NV: AIAA.

Lee, C. K., & Lo, S. H. (1994). A new scheme for the generation of a graded quadrilateral mesh. *Computers and Structures*, *52*, 847–857.

Lee, D. T., & Tsuei, Y. (1992). A formula for estimation of truncation errors of convection terms in a curvilinear coordinate system. *Journal of Computational Physics*, *98*, 90–100.

Liu, A., & Joe, B. (1994a). On the shape of tetrahedra from bisection. *Mathematics of Computation*, *63*, 141–154.

Liu, A., & Joe, B. (1994b). Relationship between tetrahedron shape measures. *BIT*, *34*, 268–287.

Mastin, C. W. (1982). Error induced by coordinate systems. In J. F. Thompson (Ed.), *Numerical grid generation* (pp. 31–40). New York: North-Holland.

Parthasarathy, V. N., Graichen, C., & Hathaway, A. F. (1993). A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design*, *15*, 255–261.

Pinkall, U., & Polhier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, *2*(1), 15–36.

Prokopov, G. P. (1989). Systematic comparison of algorithms and programs for constructing regular two-dimensional grids. *Problems Atomic Sci. Technol. Ser. Math. Model. Phys. Process.*, *3*, 98–107 (Russian).

Shewchuck, J. R. (2002). What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures. (Preprint). University of California at Berkeley, CA 94720.

Steinberg, S., & Roache, P. J. (1992). Variational curve and surface grid generation. *Journal of Computational Physics*, *100*, 163–178.

Sullivan, J. M. (2008). Curvatures of smooth and discrete surfaces. In A. I. Bobenko, P. Schroder, J. M. Sullivan & G. M. Zeigler (Eds.), Discrete differential geometry. Oberwolfach seminars (Vol. 38, pp. 175–188). Basel: Birkhauser.

Chapter 4
Stretching Method

4.1 Introduction

In many practical problems, there may exist narrow zones in the physical domains where the dependent quantities undergo large variations. These zones include shock waves in compressible flows, shear layers in laminar and turbulent flows, expansion fans, contact surfaces, slipstreams, phase-change interfaces, and boundary and interior layers, which, when interacting, can present significant difficulties in the numerical treatment. The need for a detailed description of the physical solutions to such problems requires the development of adaptive methods whose adaptivity is judged by their ability to provide a suitable concentration of grid nodes in these regions in comparison with the distribution of the nodes in the rest of the domain. Analytical and numerical investigations have demonstrated that the adaptive grid technique has a significant potential to enhance the accuracy and efficiency of computational algorithms. This is especially true for the calculation of multidimensional and unstable problems with boundary and interior layers for which the derivatives of the solution are large. Adaptivity can eliminate oscillations associated with inadequate resolution of large gradients more effectively, reduce the undesirable numerical viscosity, damp out instabilities, and considerably curtail the number of grid nodes needed to yield an acceptable solution to a problem relative to the number of nodes of a uniform grid. The interpolation of functions by discrete values is also more accurately performed over the whole region when the grid nodes are clustered in the zones of large derivatives of the functions.

Adaptive grids are, therefore, an important subject to study because of their potential for improving the accuracy and efficiency of the numerical solution of boundary value problems modeling various complex physical phenomena, and serious efforts have been undertaken to develop and enhance the methods of adaptive grid generation and to incorporate these methods into the numerical algorithms for solving field problems.

However, the problem of the development of robust adaptive grid methods is a serious challenge, since conflicting demands are imposed on the methods; in particular,

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_4

they should provide adequate resolution of the solution quantities in regions of high gradients, while also limiting the total number of points and excessive deformation of grid cells.

This chapter is concerned with the specification of the intermediate transformations and of the stretching mappings for the generation of adaptive grids with node clustering in the areas of solution singularities.

The stretching approach for generating adaptive grids is applied widely for the numerical solution of partial differential equations with singularities. Its major advantage is the rapidity of grid generation and direct control of grid spacing, while the main disadvantage is the necessity to explicitly select the zones where the stretching is needed. Of central importance in the method are intermediate transformations constructed on the basis of some standard stretching functions which provide the required spacing between the coordinate lines in selected zones.

For this purpose, some basic univariate, nonuniform coordinate transformations are described. These transformations can smooth the singularities arising in boundary value problems whose solutions undergo large variations in narrow zones. The grids generated through the use of such functions, each of which transforms an individual coordinate, appear to be well adapted to the expected physical features.

The basic functions incorporated into the method allow the grid to adjust automatically to solution singularities arising from the physical parameters, e.g. viscosity, high Reynolds number, or shell thickness, while a practical problem is solved. Such automation is one of the requirements imposed on comprehensive grid codes. The grids obtained through such methods enable users to obtain numerical solutions of singularly perturbed equations which converge uniformly to the exact solution with respect to the parameter. They also provide uniform interpolation of the numerical solution over the entire region, including boundary and interior layers.

A stretching method utilizing the standard stretching functions supplies one with a very simple means to cluster the nodes of the computational grid within the regions of steep gradients without an increase in the total number of grid nodes. This grid concentration improves the spatial resolution in the regions of large variation, thus enhancing the accuracy of the algorithms applied to the numerical solution of partial differential equations.

The stretching mappings can also be used successively to derive the blending functions in algebraic methods of transfinite interpolation. The algebraic techniques are usually contained in large, multipurpose grid generation codes in combination with more sophisticated elliptic and parabolic methods, in which their major task is to provide an initial grid which serves to start the iterative process of the grid generators. The blending functions implemented through the stretching mappings ease the process of the generation of the elliptic and parabolic grids by taking part of the solution adjustment on themselves.

4.2 Formulation of the Method

The stretching method is one of the simplest and fastest approaches applied to generate nonuniform adaptive structured grids. As a preliminary step, it requires the introduction of some specified curvilinear coordinates in the physical region X^n. The coordinates are chosen by a parametrization

$$\boldsymbol{x}(\boldsymbol{q}) : Q^n \to X^n \ , \qquad \boldsymbol{q} = (q^1, \dots, q^n) \ , \qquad \boldsymbol{x} = (x^1, \dots, x^n) \ ,$$

from a domain $Q^n \subset R^n$ with a system of Cartesian coordinates $q^i, \ i = 1, \dots, n$. This system is selected in such a way that it includes the coordinates along which the grid nodes are to be redistributed by the stretching technique. Then, in the zones where the nodes are to be concentrated, every required variable q^i is replaced by some stretching variable ξ^i using a specified separate univariate transformation $\xi^i(q^i)$. To provide stretching of the coordinate q^i, the function $\xi^i(q^i)$ must have a large first derivative with respect to q^i. The inverse transformation $q^i(\xi^i)$, having, in contrast, a small first derivative with respect to ξ^i, is a contraction transformation in these zones. A smooth or continuous expansion of these separate local contraction functions $q^i(\xi^i)$ to produce a new coordinate system $\xi^1, \dots, \xi^n$ in the whole region Q^n provides an intermediate transformation

$$\boldsymbol{q}(\boldsymbol{\xi}) : \Xi^n \to Q^n$$

from some parametric domain $\Xi^n \subset R^n$. The composition $\boldsymbol{x}[\boldsymbol{q}(\boldsymbol{\xi})]$ defines a coordinate transformation which yields a numerical grid with nodal clustering in the required parts of the domain X^n.

Analogously, the transformation $\boldsymbol{q}(\boldsymbol{\xi})$ can be obtained as the inverse to a mapping

$$\boldsymbol{\xi}(\boldsymbol{q}) : Q^n \to \Xi^n \ ,$$

which is an expansion of the local stretching functions $\xi^i(q^i)$ over the whole domain Q^n.

Without losing generality, we assume that the domain Q^n, called the intermediate domain, as well as the domain Ξ^n, called the logical or computational domain, is the unit n-dimensional cube. So, the coordinate transformation from the unit logical cube Ξ^n onto the simply connected bounded physical region X^n is defined as the composition of two transformations: $\boldsymbol{q}(\boldsymbol{\xi})$ from Ξ^n onto Q^n and $\boldsymbol{x}(\boldsymbol{q})$ from Q^n onto X^n (Fig. 4.1), i.e.

$$\boldsymbol{x}(\boldsymbol{q})(\boldsymbol{\xi}) : \Xi^n \to Q^n \to X^n .$$

The splitting of the sought coordinate transformation $\boldsymbol{x}(\boldsymbol{q}(\boldsymbol{\xi}))$ into the two transformations $\boldsymbol{q}(\boldsymbol{\xi})$ and $\boldsymbol{x}(\boldsymbol{q})$ enables one to divide the task of grid generation into two steps: one performed by the intermediate transformation $\boldsymbol{q}(\boldsymbol{\xi}) : \Xi^n \to Q^n$, obtained with the help of some specified contraction functions $q^i(\xi^i)$ and responsible for the control of the grid, and another one performed by the parametric mapping

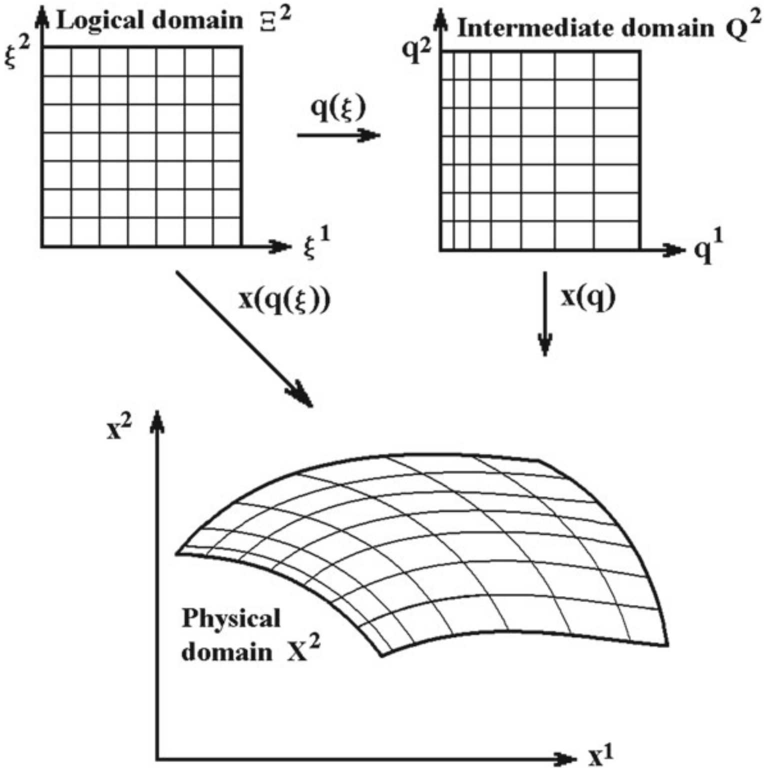


Fig. 4.1 Illustration of the stretching method

$\boldsymbol{x}(\boldsymbol{q}) : Q^n \rightarrow X^n$, which is concerned with the specification of the coordinates requiring stretching in some zones of the physical domain X^n. These two steps can be considered as separate and distinct operations and, as such, can be developed in an independent and modular way.

The intermediate coordinate transformations $\boldsymbol{q}(\boldsymbol{\xi})$ are designed to be between the standard unit cubes Ξ^n and Q^n, which are fixed regardless of the physical domain X^n and the physical solution. Therefore, there is an opportunity to create a kind of database of reference functions which can be used as elements to construct comprehensive intermediate transformations $\boldsymbol{q}(\boldsymbol{\xi})$.

4.3 Theoretical Foundation

The construction of the basic intermediate transformations $\boldsymbol{q}(\boldsymbol{\xi}) : \Xi^n \rightarrow Q^n$ starts with the definition of the basic univariate functions $q^i(\xi^i)$, which are, as was mentioned, the inverses of the basic univariate stretching functions $\xi^i(q^i)$. The functions

$q^i(\xi^i)$ should be suitable for providing adequate grid clustering in the necessary zones through their implementation in formulas for the intermediate transformations.

The form of the univariate transformation $\xi^i(q^i)$, which stretches the coordinate q^i in the zones of large derivatives, depends on the qualitative behavior of the solution. Thus, for effective use of the stretching method in the numerical solution of multidimensional problems, one needs both to select the directions q^i in the region X^n along which the solution has large derivatives and to have some information on its structure along these particular directions. Information about the qualitative solution structure is obtained from a theoretical analysis of simpler model equations, in particular, ordinary differential equations which simulate the qualitative features of the solutions, or it can be obtained from a preliminary numerical calculation for similar problems on coarse grids.

One set of stretching functions can be formed by local nonuniform mappings applied to the numerical solution of equations with a small parameter ϵ affecting the higher derivatives. Equations with a small parameter ϵ before the higher derivatives are widespread in practical applications. For example, such equations can model flows with small viscosity or high Reynolds number, describe problems of elasticity for which the small parameter represents the shell thickness, or simulate flows of liquid in regions having orifices with a small diameter. These problems have narrow boundary and interior layers where the derivatives of the solutions with respect to the coordinates orthogonal to the layers reach very large magnitudes when the parameter ϵ is small. In the center of such a layer, the mth derivative may have values of order $\epsilon^{-\alpha m}, \quad \alpha > 0$.

Problems with a small parameter affecting the higher derivatives have been studied thoroughly through analytical and numerical methods. At present, there is a lot of analytical information related to the qualitative features of the solutions to these problems in the layers, which can be efficiently applied to the development of well-behaved numerical methods, in particular, to the generation of grids with nodal clustering in narrow zones of large variation of the physical quantities.

The analysis of these problems has revealed new forms of local stretching functions in addition to the well-known ones aimed chiefly at the treatment of exponential-type layers. These new stretching functions are very suitable for coping not only with exponential-type layers, but with power-type and mixed layers as well, which are common in practical applications. A stretching technique based on the new functions provides efficient concentration of coordinate lines, and consequently efficient concentration of grid cells in boundary and interior layers.

In this section some theoretical facts concerning qualitative features of solutions in boundary and interior layers are outlined. These facts serve to justify the forms of the basic stretching functions applied to generate grids through the stretching method.

4.3.1 Model Problems

The stretching method has efficient application to the numerical solution of ordinary differential equations and multidimensional problems with boundary and interior layers, where the solutions may have large variations along the coordinate lines intersecting the layers. For the numerical calculation of a two-dimensional viscous gas flow, for instance, the stretching method can be used successfully to generate grids with nonuniform clustering in the region of the boundary layer, where the longitudinal component u of the velocity $\boldsymbol{u} = (u, v)$ has its highest gradient near a solid boundary, in the direction x orthogonal to the boundary, in the case of a laminar flow. Some information about the qualitative behavior of the tangential velocity u in the direction x can be gained from the study of a model two-point boundary value semilinear problem with a small parameter ϵ:

$$\begin{aligned} &\epsilon u'' + a(x)u' = f(x, u) \ , \qquad 0 < x < 1 \ , \\ &u(0) = u_0 \ , \qquad u(1) = u_1 \ . \end{aligned} \tag{4.1}$$

The model Eq. (4.1) is derived from the steady equation for the tangential velocity component u of the Navier–Stokes system. With respect to the independent transverse and longitudinal variables x and y, it can be written in the form

$$\frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) - \rho v \frac{\partial u}{\partial x} = g\left[u, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial v}{\partial y}, \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial y}\right), \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial x}\right)\right] \ . \tag{4.2}$$

The model two-point boundary value problem (4.1) is obtained from this equation with the assumptions that the dynamic viscosity μ is a constant, the longitudinal coordinate y is a parameter, and the right-hand side is uniformly bounded with respect to μ. Therefore, $a(x)$ in (4.1) corresponds to $-\rho v$, ϵ to μ, and $f(x, u)$ to the right-hand side of (4.2).

The boundary-layer behavior of the solution $u(x, \epsilon)$ of the two-point boundary value problem (4.1) for $f_u(x, u) > 0$ is of three types (exponential and two power ones), depending on the values of $a(x_0)$ and $a'(x_0)$ (if $a(x_0) = 0$) $x_0 = 0$ or $x_0 = 1$, and is characterized by estimates for the derivatives of $u(x, \epsilon)$ with respect to x. The solution to (4.1) may also have interior power layers in the vicinity of interior points x_0 at which $a(x_0) = 0$.

Another model equation to investigate the behavior of the solutions in boundary and interior layers is obtained from a problem simulating the shock wave structure of steady heat-conducting gas flow:

$$\begin{aligned}
&\frac{\mathrm{d}\rho u}{\mathrm{d}x} = 0\ ,\\
&\rho u \frac{\mathrm{d}u}{\mathrm{d}x} + \frac{\mathrm{d}p}{\mathrm{d}x} - \epsilon \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = 0\ ,\\
&\rho u \frac{\mathrm{d}e}{\mathrm{d}x} + p \frac{\mathrm{d}u}{\mathrm{d}x} - \epsilon \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 - \frac{\mathrm{d}}{\mathrm{d}x}\left(\chi \frac{\mathrm{d}T}{\mathrm{d}x}\right) = 0\ , \qquad 0 < x < 1\ ,\\
&(\rho, u, e)(0) = (\rho_0, u_0, e_0)\ , \qquad (\rho, u, e)(1) = (\rho_1, u_1, e_1)\ ,
\end{aligned} \tag{4.3}$$

where ρ is the density, u the velocity, p the pressure, T the temperature, e the energy, ϵ the coefficient of viscosity of the gas, and χ the coefficient of thermal conductivity.

In the case

$$\begin{aligned}
e &= c_v T\ ,\\
p &= (\nu - 1)\rho e\ ,
\end{aligned}$$

we obtain, from the system (4.3),

$$\begin{aligned}
&-\epsilon u'' + c[u + (\nu - 1)e/u]' = 0\ , \qquad 0 < x < 1\ ,\\
&u(0) = u_0\ , \qquad u(1) = u_1\ ,
\end{aligned} \tag{4.4}$$

$$\begin{aligned}
&-(\epsilon_1 e')' + c\left(e - \frac{u^2}{2} + \frac{c_2}{c} u\right)' = 0\ , \qquad 0 < x < 1\ ,\\
&e(0) = e_0\ , \qquad e(1) = e_1\ ,
\end{aligned} \tag{4.5}$$

where

$$c = \rho_0 u_0\ , \qquad \epsilon_1 = \chi / c_v\ ,$$

$$c_2 = \{-\epsilon u' + c[u + (\nu - 1)e/u]\}|_{x=0}\ .$$

The functions $u(x)$ and $e(x)$ are monotonic in the layer of their rapid variation. Hence, the dependent variables u and e are connected by some relations

$$e = E(u)\ , \qquad u = U(e)\ .$$

Therefore, the problem (4.4) can be presented as a two-point boundary value problem of a very simple, standard, autonomous quasilinear form

$$\begin{aligned}
&-\epsilon u'' + a(u)u' = 0\ , \qquad 0 < x < 1\ ,\\
&u(0) = u_0\ , \qquad u(1) = u_1\ ,
\end{aligned} \tag{4.6}$$

which represents a model problem to study the qualitative features of solutions with singularities in interior layers. An analogous expression is valid for the problem (4.5) if ϵ_1 is a constant. The solutions to problem (4.6) may have boundary and interior layers of both exponential, power, and mixed types.

One more model suitable for investigating the qualitative features of solutions in layers is the boundary value problem of a gas flow near a round hole with a small radius $r = \epsilon$ (Polubarinova-Kochina 1977) or a corresponding problem of electron motion (Zamaraev et al. 1985). The behavior of the solution to these problems in the vicinity of the boundary layer is simulated qualitatively by a semilinear two-point boundary value problem

$$\begin{aligned} &(\epsilon + x)^p u'' + a(x)u' = f(x, u) , \qquad p > 0 , \qquad 0 < x < 1 , \\ &u(0) = u_0 , \qquad u(1) = u_1 . \end{aligned} \tag{4.7}$$

A popular model of a singularly perturbed problem is presented by the stationary Burger's two-point boundary value problem

$$\begin{aligned} &-\epsilon u'' + uu' + f(x, u) = 0 , \qquad 0 < x < 1 , \\ &u(0, \epsilon) = A_0 , \qquad u(1, \epsilon) = A_1 . \end{aligned} \tag{4.8}$$

The qualitative analysis shows that the solution can have only exponential boundary and interior layers. The position of the interior layer is computed exactly and is determined by the boundary values A_0 and A_1 and solutions to the reduced problem.

A more general problem than (4.8) is represented by the following singularly perturbed quasilinear boundary value problem:

$$\begin{aligned} &-\epsilon u'' + g(x, u)u' + f(x, u) = 0 , \qquad 0 < x < 1 , \\ &u(0, \epsilon) = A_0 , \qquad u(1, \epsilon) = A_1 . \end{aligned} \tag{4.9}$$

Physically, problem (4.9) may be considered to be a simulation of a one-dimensional, steady-state, reaction-diffusion-convection process. In this connection, ϵ is a measure of the diffusivity, g is a measure of convection, and f is a measure of the effect of reaction and of sources in the medium. The second interpretation of (4.9) refers to a model nonlinear mass-spring system, or, more generally, simply as a formulation of Newton's Second Law of Motion. In this connection, x is time, ϵ is the mass of the object, g is a measure of the damping effect of the medium, and f is a nonlinear restoring force.

The problems (4.1), (4.6)–(4.9) are amenable to analytical study. Though they represent highly idealized cases, they nevertheless give a rather profound understanding of the variety and complexity of the singularities arising in practical applications. The study of these two-point boundary value problems has provided solid knowledge about the possible qualitative features of solutions in boundary and interior layers.

The next considerations of this section are concerned with some results related to the qualitative behavior of the solutions to the problems (4.1), (4.6)–(4.9). The results mainly apply to estimates of the derivatives of the solution appropriate to specifying the stretching functions. Any analytical proof of the facts outlined below is beyond the scope of this book, but it can be found in the book "Layer Resolving Grids and Transformations for Singular Perturbation Problems" written by Liseikin

(2001). However, we note that the principal technique used to analyze the asymptotic behavior of the solutions and to provide estimates of the solutions and of their derivatives employs the theory of differential inequalities developed by Nagumo (1937). For the Dirichlet problem

$$\begin{aligned} & u'' = f(x, u, u') \ , \qquad 0 < x < 1 \ , \\ & u(0) = u_0 \ , \qquad u(1) = u_1 \ , \end{aligned} \tag{4.10}$$

where f is a continuous function of the arguments x, u, u', the Nagumo inequality theory states that if there exist continuous, twice differentiable functions $\alpha(x)$ and $\beta(x)$ with the properties

$$\begin{aligned} & \alpha(x) \leq \beta(x) \ , && 0 \leq x \leq 1 \ , \\ & \alpha(0) \leq u(0) \leq \beta(0) \ , && \alpha(1) \leq u(1) \leq \beta(1) \ , \\ & \alpha'' \geq f(x, \alpha, \alpha') \ , && 0 < x < 1 \ , \\ & \beta'' \leq f(x, \beta, \beta'') \ , && 0 < x < 1 \ . \end{aligned}$$

then the problem (4.10) with the condition $f(x, u, z) = O(z^2)$ has a solution $u(x)$ and

$$\alpha(x) \leq u(x) \leq \beta(x) \ .$$

The functions $\alpha(x)$ and $\beta(x)$ are called the bounding functions. Estimates of the solutions to the problems (4.1), (4.6)–(4.9) and of their derivatives are obtained by selecting the appropriate bounding functions $\alpha(x)$ and $\beta(x)$.

4.3.2 *Basic Majorants*

This subsection presents some estimates of the first and higher derivatives of the solutions to the problems (4.1), (4.6)–(4.9). The solutions to these problems may have highly localized zones of rapid variation. The first derivative of the solutions in these zones may reach a magnitude of $\epsilon^{-\alpha}$, $1 \geq \alpha > 0$, and therefore it tends to infinity when ϵ approaches zero. Outside the layers, the derivative is estimated by a constant M independent of the parameter ϵ. These estimates are used to define an optimum coordinate ξ with a transformation $x(\xi)$.

Relation Between Optimal Univariate Transformations and Majorants of the First Derivative

The optimum univariate transformation $x(\xi)$ for a monotonic univariate function $u(x)$ would be one for which $u[x(\xi)]$ varied linearly with respect to ξ, since this would result in zero truncation errors for any approximation. However, if the function $u(x)$ is not monotonic and is found from a solution to a particular problem, this formulation of the optimal transformation is too good to be realized in practice. With regard to the problems (4.1), (4.6)–(4.9) with the small parameter ϵ, the optimum transformation

$x(\xi)$ would be one that eliminated the first order singularity of the solution $u(x,\epsilon)$ of these problems, i.e. in particular, one in which the first derivative with respect to ξ of the transformation $u[x(\xi),\epsilon]$ was limited by a constant M independent of ϵ. Such a transformation $x(\xi)$ eliminates the first order singularities of $u(x,\epsilon)$, and as a result, the function $u[x(\xi),\epsilon]$ does not have large variations, and therefore the transformed problem with respect to this independent variable ξ may be efficiently solved on the uniform grid

$$\xi_i = ih \ , \qquad i = 1, \dots, N \ , \qquad h = 1/N \ .$$

The univariate transformations eliminating the first order singularities inherent in the solutions to the problems (4.1), (4.6)–(4.9) depend inevitably on the small parameter ϵ. Nevertheless, for simplicity, we use the notation $x(\xi)$ for such functions.

Let the ranges of the variables x and ξ be normalized, say

$$0 \le x \le 1 \ , \qquad 0 \le \xi \le 1 \ .$$

Then, the optimum transformation $x(\xi)$ exists if the derivative of the function $u(x,\epsilon)$ is bounded by a strictly positive function $\psi(x,\epsilon)$ whose total integral is limited by a constant M independent of ϵ, i.e.

$$\left|\frac{\mathrm{d}u}{\mathrm{d}x}\right| \le \psi(x,\epsilon) \ ,$$

with

$$\int_0^1 \psi(x,\epsilon)\mathrm{d}x \le M \ . \tag{4.11}$$

Equation (4.11) means that $u(x,\epsilon)$ is a function with a uniformly limited total variation on the interval [0, 1], i.e.

$$\int_0^1 \left|\frac{\mathrm{d}u}{\mathrm{d}x}\right| \mathrm{d}x \le M \ .$$

The required function $x(\xi)$ eliminating the singularity of the first order of $u(x,\epsilon)$ is obtained, for example, as the inverse of the solution $\xi(x)$ of the initial-value problem

$$\begin{aligned} &\frac{\mathrm{d}\xi}{\mathrm{d}x} = c\psi(x,\epsilon) \ , \qquad x > 0 \ , \\ &\xi(0) = 0 \ , \end{aligned} \tag{4.12}$$

where c is a scaling constant providing the condition $\xi(1) = 1$. After integrating (4.12), we obtain

$$c = 1 \Big/ \int_0^1 \psi(x,\epsilon)\mathrm{d}x$$

and hence we have

$$\left|\frac{\mathrm{d}u}{\mathrm{d}\xi}\right| = \left|\frac{\mathrm{d}u}{\mathrm{d}x}\right|\frac{\mathrm{d}x}{\mathrm{d}\xi} \le \int_0^1 \psi(x,\epsilon)\mathrm{d}x \le M \ . \tag{4.13}$$

So, the function $u[x(\xi),\epsilon]$ does not have layers of rapid variation in the interval [0, 1] of the independent variable ξ.

The problem (4.12) for generating adaptive grids is referred to as the equidistribution principle.

Original Formulation

The original one-dimensional formulation of the equidistribution principle (4.12) for generating grid steps was proposed by Boor (1974) for the purpose of obtaining more accurate interpolation of functions by splines. The principle was formulated as a rule for determining the grid nodes $x_i,\ i = 1,\dots,N$, in the interval $[a,b]$ in accordance with the relation

$$\begin{aligned} &\int_{x_i}^{x_{i+1}} w(x)\mathrm{d}x = c \ , \qquad i = 0,1,\dots,N-1 \ , \\ &x_0 = a \ , \qquad x_N = b \ , \end{aligned} \tag{4.14}$$

where $w(x)$ is a certain positive quantity called either a monitor function or a weight function. A discrete form of (4.14) may be represented as

$$\begin{aligned} &h_i w_i = c \ , \qquad i = 0,\dots,N-1 \ , \\ &x_0 = a \ , \qquad x_N = b \ , \qquad h_i = x_{i+1} - x_i \ , \end{aligned} \tag{4.15}$$

where $w_i = w(x_i'),\ x_i' \in h_i,\ i = 0,\dots,N-1$. The constant c in (4.15) is determined from the condition $x_N = b$, i.e. after summing (4.14):

$$c = \frac{1}{N}\int_a^b w(x)\mathrm{d}x \ .$$

The grid steps satisfying (4.15) will be small where the weight w is large, and vice versa. Thus, the weight function provides information about the desired grid clustering.

This formulation of the equidistribution principle for generating one-dimensional grids has been used with success to generate one-dimensional adaptive grids for the numerical solution of both stationary and nonstationary problems with singularities. Commonly, the solution to the problem of interest is first found on an initial background grid, then the weight function w is computed at the points of this background grid and interpolated over the entire interval $[a,b]$, and afterwards, grid points are either moved or added to satisfy (4.14). The process is then repeated until a desired convergence tolerance is achieved.

Optimally Distributed Grid

The idea of the equidistribution principle is to equidistribute the solution error by placing more grid nodes where the error is large, so as to gain high accuracy overall with a fixed number of grid points. The grid which minimizes the error of the numerical solution to a differential problem is an optimally distributed grid with the nodes optimally refined in the areas of large solution error. Thus, if a measure of the error $\boldsymbol{e}(x)$ is estimated for the grid interval h_i by the relation

$$\| \boldsymbol{e}(x) \| = (h_i)^p Q(x) , \qquad \boldsymbol{e}(x) = \boldsymbol{u}(x) - \boldsymbol{u}^h(x) ,$$

where $\boldsymbol{u}(x)$ is the exact solution to the physical problem and $\boldsymbol{u}^h(x)$ is the numerical solution, it is quite natural, for obtaining the optimally distributed grid, to define the weight function through a measure of the error by means of the following relation:

$$w(x) = [Q(x)]^{1/p} .$$

The error $\boldsymbol{e}(x)$ may be found with high accuracy as a solution to the boundary value problem

$$L(\boldsymbol{e}) = \boldsymbol{T} , \qquad \boldsymbol{e}|_{\partial X^n} = 0 , \tag{4.16}$$

where L is the operator of linearization of the governing equations for the physical boundary value problem, while $\boldsymbol{T}$ is the approximation error of the problem. The derivation of (4.16) is carried out analogously to that of stability equations.

Although an asymptotically accurate solution error $\boldsymbol{e}(x)$ can in principle be obtained from the equation for variation (4.16), in practice, this task is very difficult and expensive, even for a one-dimensional problem, to which (4.16) is applicable.

A more promising way seems to lie in the generation of the grid in accordance with a uniform distribution of some norm of $\boldsymbol{T}(x)$, i.e. by defining the monitor function $w(x)$ through a measure of the approximation error $\boldsymbol{T}(x)$. In this case,

$$w(x) = [Q_1(x)]^{1/p}$$

if

$$\| \boldsymbol{T}(x) \| = (h_i)^p Q_1(x) ,$$

and $\boldsymbol{T}(x)$ can be readily expressed through the solution derivatives. This assumption has some meaning, since the relation

$$\| \boldsymbol{e}(x) \| \sim c(x) \| \boldsymbol{T}(x) \|$$

is generally valid for the numerical solution of boundary value problems. However, the utilization of a weight function defined through the error of approximation $\boldsymbol{T}(x)$ to generate adaptive grids for the numerical solution of singularly perturbed equations may result in too large a grid spacing in the areas which lie outside the boundary and interior layers.

As an example, we consider the following two-point boundary value problem of the type (4.1) with a boundary layer:

$$\begin{aligned} &\epsilon u'' + u' - u = f(x) , \qquad 0 < x < 1 , \\ &u(0) = a , \qquad u(1) = b , \end{aligned} \tag{4.17}$$

where $1 \geq \epsilon > 0$ is a small parameter. An approximation of (4.17) using the upwind differencing

$$\begin{aligned} &\frac{2\epsilon}{h_i + h_{i-1}} \left(\frac{u_{i+1} - u_i}{h_i} - \frac{u_i - u_{i-1}}{h_{i-1}} \right) + \frac{u_{i+1} - u_i}{h_i} - u_i = f(x_i) , \\ &0 < i < N , \qquad u_0 = a , \qquad u_N = b , \end{aligned} \tag{4.18}$$

on a nonuniform grid x_i, $i = 1, \dots, N$, results in the approximation error

$$T \sim c \left\{ \epsilon \left[h_i \frac{\mathrm{d}^3 u}{\mathrm{d}x^3}(x_i) + h_{i-1} \frac{\mathrm{d}^3 u}{\mathrm{d}x^3}(x_{i-1}) \right] + h_i \frac{\mathrm{d}^2 u}{\mathrm{d}x^2}(x_i) \right\} . \tag{4.19}$$

Let $v(x)$ be a solution to the following initial-value problem associated with (4.17):

$$\begin{aligned} &v' - v = f(x) , \qquad x < 1 , \\ &v(1) = b . \end{aligned}$$

If $|v(0) - a| > m$, where m is a positive constant independent of ϵ, then the solution $u(x)$ of (4.17) is a function of the exponential boundary-layer type (see Fig. 4.2) satisfying the inequality

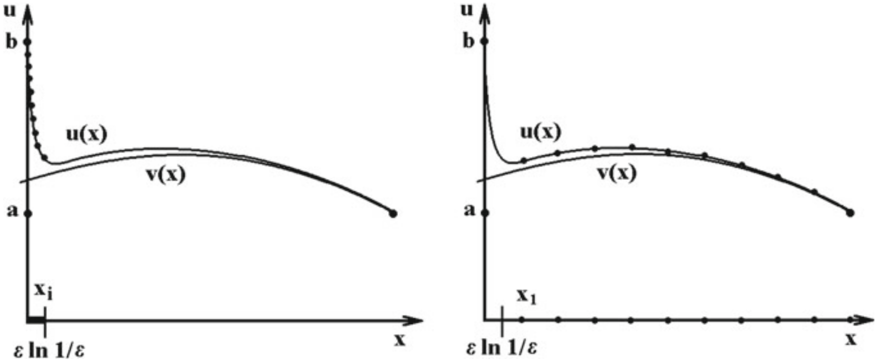


Fig. 4.2 Node placement generated by equidistribution of the approximation error (*left*) and of the solution error computed at the grid points (*right*)

$$|u(x) - v(x)| \leq M[\exp(-x/\epsilon) + \epsilon] , \qquad 0 \leq x \leq 1 ,$$

and its kth derivative in the interval [0, 1] is estimated by

$$|u^{(k)}(x)| \leq M[\epsilon^{-k} \exp(-x/\epsilon) + 1] ,$$

where M is a constant independent of ϵ. It is important to note that in the boundary layer interval $[0, k\epsilon \ln(1/\epsilon)]$ the kth derivative of $u(x)$ is bounded from both sides by the estimate

$$M_1[\epsilon^{-k} \exp(-x/\epsilon) + 1] \leq |u^{(k)}(x)| \leq M_2[\epsilon^{-k} \exp(-x/\epsilon) + 1] , \tag{4.20}$$

where M_1, M_2 are positive constants which do not depend on ϵ. Thus, in accordance with (4.20), we obtain, from (4.19), the following estimate for $T(x)$ in the boundary layer interval:

$$M_3 h_i[\epsilon^{-2} \exp(-x/\epsilon) + 1] \leq T(x) \leq M_4 h_i[\epsilon^{-2} \exp(-x/\epsilon) + 1] ,$$

with $M_i > 0$ and independent of ϵ. From this inequality, it seems natural to choose the weight $w(x)$, for the purpose of obtaining a uniform distribution of $T(x)$ with respect to the parameter ϵ, as

$$w(x) = \epsilon^{-2} \exp(-x/\epsilon) + 1 , \qquad 0 \leq x \leq 1 .$$

However, by computing (4.12) with this weight function $w(x)$, we obtain

$$\xi(x) = \frac{1 - \exp(-x/\epsilon) + \epsilon x}{1 - \exp(-1/\epsilon) + \epsilon} . \tag{4.21}$$

If $x_0 = \epsilon \ln(1/\epsilon)$, (4.21) yields

$$\xi(x_0) = \frac{1 - \epsilon + \epsilon^2 \ln(1/\epsilon)}{1 - \exp(-1/\epsilon) + \epsilon} \sim 1 , \qquad \text{if} \qquad 0 < \epsilon \ll 1 ,$$

i.e. nearly all of the computational interval [0, 1] (and consequently nearly the entire set of grid points) is mapped by $x(\xi)$ into the boundary layer. As a result, the area outside the boundary layer is not provided with a sufficient number of grid nodes (Fig. 4.2, left-hand).

Note that this phenomenon does not occur if the scale of the layer is less then 1, i.e. when the first derivative of $u(x)$ satisfies the estimate

$$|u'(x)| \leq M\epsilon^{-\alpha} , \qquad 0 \leq x \leq 1 , \qquad \alpha < 1 ,$$

with M independent of ϵ. This case occurs, for example, when the first derivative u' in (4.17) is either deleted or multiplied by a positive function vanishing at the point $x = 0$.

If the generation of the grid for the solution of (4.17) is determined by the condition of a uniform distribution of the solution error $e = u(x_i) - u_i(x_i)$, then both the boundary layer and the rest of the interval [0, 1] are provided with a sufficient number of grid points. Indeed, the following estimate of e is valid:

$$|e_i| \leq Mh_i[\epsilon^{-1}\exp(-x/\epsilon) + 1] ,$$

and consequently, assuming

$$w(x) = \epsilon^{-1}\exp(-x/\epsilon) + 1 , \qquad 0 \leq x \leq 1 ,$$

we obtain, solving (4.12),

$$\xi(x) = \frac{1 + x - \exp(-x/\epsilon)}{2 - \exp(-1/\epsilon)} , \qquad 0 \leq x \leq 1 .$$

This expression for $\xi(x)$ yields, for $x_0 = \epsilon\ln(1/\epsilon)$,

$$\xi(x_0) = \frac{1 + x_0 - \epsilon}{2 - \exp(-1/\epsilon)} \sim \frac{1}{2} , \qquad \text{if} \quad 0 < \epsilon \ll 1 .$$

Thus, in this case, unlike the previous one, nearly $N/2$ grid points will be placed in the boundary layer and the remaining $N/2$ nodes will be distributed outside the layer. The proportion of grid points placed in the boundary layer can be controlled by a constant c if we propose

$$w(x) = \epsilon^{-1}\exp(-x/\epsilon) + c ,$$

which leads to the placement of approximately $cN/(1 + c)$ nodes in the boundary layer.

Let us consider another phenomenon connected with the solution of (4.17) by the scheme (4.18). Since $|u^i| \leq M$, where u^i is the solution to (4.18) and M is independent of ϵ, we have, from (4.18),

$$|e_i| \leq M \max_{i \geq 2} \left\{ \frac{\epsilon}{(h_1)^2} , \quad h_i[\exp(-x_i/\epsilon) + 1] \right\} , \qquad i = 0, 1, \ldots, N - 1 .$$

Therefore, if $\epsilon \leq h(h_1)^2$, we obtain a uniform estimate of the solution error

$$|e_i| \leq Mh , \qquad i = 0, \ldots, N - 1 .$$

In this case (Fig. 4.2, right-hand), i.e. when all grid nodes lie outside the boundary layer, we find that the solution of the associated initial-value problem is solved more accurately than in the case where a part of the nodes is put in the boundary layer. As a result, the solution to a problem of the type (4.17) may be more accurate at the points of a uniform grid than at the points of a grid with node clustering in the boundary layer when the parameter ϵ is sufficiently small, namely $\epsilon \leq 1/N^3$. However, this solution is not interpolated uniformly over the entire interval [0, 1]. This consideration shows that a measure of the solution error computed only at the grid points cannot always be used as a successful driving mechanism for the adaptive distribution of grid points.

One more disadvantage of generating grids in accordance with a uniform distribution of the error $\boldsymbol{T}(x)$ is the fact that for highly accurate approximations of the governing equations, the expression for $\boldsymbol{T}(x)$ includes terms dependent on high-order derivatives of the solution, which may cause much numerical noise and instability.

Thus, though it is quite natural to incorporate directly the error measurements $\parallel \boldsymbol{e}(x) \parallel$ or $\parallel \boldsymbol{T}(x) \parallel$ into a formulation of the monitor functions, the computation of the optimally distributed grids defined by these measures may be an expensive and unsuccessful procedure and, in fact, relies on exact knowledge of the physical solution. So, the requirement for efficiency of the algorithms leads the practitioners to specify the weight functions in more simple forms, applying for this purpose only lower-order derivatives of the solution. Generally, the largest numerical errors are found in regions of rapid variation of the lower derivatives of the solution, in particular, the first derivative. Therefore, even the first derivatives of the solution can often be used to derive the weight functions.

Two-Point Boundary Value Problem

The initial-value problem (4.12) can also be replaced by an equivalent linear two-point boundary value problem for an ordinary equation of the second order,

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}/\psi(x,\epsilon)\right)=0\,, \qquad 0<x<1\,,\\&\xi(0)=0\,, \qquad \xi(1)=1\,,\end{aligned} \tag{4.22}$$

or by a nonlinear problem for an equation with ξ as the independent variable and x as the dependent variable,

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\psi(x,\epsilon)\right)=0\,, \qquad 0<\xi<1\,,\\&x(0)=0\,, \qquad x(1)=1\,.\end{aligned} \tag{4.23}$$

So, the singular functions $u(x,\epsilon)$ whose total variation is limited on the interval [0, 1] by a constant M independent of the parameter ϵ can be transformed to the function $u[x(\xi),\epsilon]$ with a uniformly limited first derivative with respect to ξ on the interval [0, 1].

The one-dimensional grid derived through a transformation $x(\xi)$ which satisfies the relation (4.23) is optimal in the above sense. Taking into account (4.13), which shows that the first derivative of the function $u[x(\xi), \epsilon]$ with respect to ξ is uniformly bounded, we find that the variation of the function $u[x(\xi), \epsilon]$ on the neighboring points x_{i+1} and x_i of the grid derived by the transformation $x(\xi)$, where

$$x_i = x(ih) \ , \qquad h = 1/N \ , \qquad i = 0, \ldots, N \ ,$$

is uniformly limited as well, i.e.

$$|u_{i+1} - u_i| \leq Mh \ , \qquad u_i = u(x_i, \epsilon) \ , \qquad i = 0, \ldots, N-1 \ ,$$

where M is independent of ϵ. Therefore, the values of u_i can be uniformly interpolated over the whole interval [0, 1] by a piecewise function $P(x)$ which uniformly approximates $u(x, \epsilon)$ over the entire interval [0, 1]:

$$|u(x, \xi) - P(x)| \leq Mh \ , \qquad 0 \leq x \leq 1 \ .$$

Analytical results guarantee the existence of a majorant $\psi(x, \epsilon)$ satisfying the condition (4.11) for the solutions to the problems (4.1), (4.7) if the function $f(x, u)$ satisfies the condition of strong ellipticity, i.e.

$$f_u(x, u) \geq m > 0 \ ,$$

which also ensures uniqueness of the solution. Note, for example, that the problem

$$\begin{aligned} &\epsilon u'' = -u = 0 \ , \qquad 0 < x < 1 \ , \\ &u(0) = 0 \ , \qquad u(1) = 1 \ , \end{aligned}$$

does not satisfy the above condition of strong ellipticity and, as a result, the total variation of its solution is not uniformly limited with respect to ϵ.

The solution $u(x, \epsilon)$ to the problem (4.6) is always a monotonic function, and therefore its total variation equals $|u_1 - u_0|$.

Now we present four positive basic singular functions $\psi_i(x, \epsilon)$, with a uniformly limited total variation, whose combinations bound the first derivative of the solutions to the problems (4.1), (4.6)–(4.9) in the boundary layers.

Exponential Functions

The most popular function used to demonstrate a boundary singularity near $x = 0$ is the exponential function

$$u(x, \epsilon) = \exp(-bx/\epsilon^{\alpha}) \ , \qquad 0 \leq x \leq 1 \ ,$$

$\alpha > 0,\ b > 0$, whose first derivative yields an expression for the basic majorant $M\psi_1(x, b, \epsilon)$, where

$$\psi_1(x, b, \epsilon) = \epsilon^{-\alpha} \exp(-bx/\epsilon^{\alpha}) \ , \tag{4.24}$$

satisfying the condition (4.11).

An exponential singularity in the form (4.24) of the solution $u(x, \epsilon)$ to problem (4.1) can occur only near the boundary point $x = 0$ with $\alpha = 1$ when $a(0) \geq m > 0$, and with $\alpha = 1/2$ when $a(0) = 0,\ a'(0) = 0$. Similarly, near the boundary point $x = 1$, the exponential singularity in the form

$$u(x, \epsilon) = \exp(-b(1 - x)/\epsilon^{\alpha}) \ , \quad 0 \leq x \leq 1 \ ,$$

can occur with $\alpha = 1$ when $-a(0) \geq m > 0$, and with $\alpha = 1/2$ when $a(1) = 0,\ a'(1) = 0$. The condition $a(0) > 0$ in the gas flow simulation (4.2) corresponds to $v(0) < 0$, which means the physical situation of the gas being sucked through a side wall, while the condition $a(0) = 0$ corresponds to gas adhesion to the side wall.

The first derivative of the solution to (4.7) is also estimated by the majorant $M\psi_1(x, b, \xi)$, with $\alpha = p$, when $p > 1$.

Power Singularities

The most common condition for viscous gas flows is that of adhesion of the gas to a solid wall, which, for two-dimensional flows (4.2), is expressed mathematically by the equation

$$u(0) = 0 \ , \qquad v(0) = 0 \ ,$$

corresponding to $a(0) = 0$ in (4.1). In this case, the nature of the boundary layer singularity of the solution to the problem (4.1) depends on the sign of the first derivative of $a(x)$ at the point $x = 0$. The relations

$$a(0) = 0 \ , \quad a'(0) < 0 \ , \quad \text{or} \ \ a'(0) > 0 \ ,$$

physically express attraction or repulsion of the gas to or from the wall, respectively. Therefore, the singularities of the gas flow are directly connected to the direction of the transverse velocity near the solid wall.

For $a(0) = 0,\ a'(0) < 0$, the first derivative of the solution $u(x, \epsilon)$ to the problem (4.1) in the vicinity of the boundary point $x = 0$ when $f_u(x, u) \geq m > 0$ is estimated by the majorant $M\psi_2(x, b, \epsilon)$, where

$$\psi_2(x, b, \epsilon) = \epsilon^{\alpha b}/(\epsilon^{\alpha} + x)^{b+1} \ , \qquad b > 0 \ , \qquad \alpha = 1/2 \ . \tag{4.25}$$

The function $M\psi_2(x, b, \epsilon)$ with $\alpha = 1$ also estimates the first derivative of the solution to the problem (4.7) in the vicinity of the boundary point $x = 0$ when $p = 1$ and $a(0) < 1$.

Another power function near the boundary $x = 0$ is expressed by the majorant $M\psi_3(x, b, \epsilon)$:

$$\psi_3(x, b, \epsilon) = (\epsilon^\alpha + x)^{b-1} \ , \quad 1 > b > 0 \ , \quad \alpha > 0 \ . \tag{4.26}$$

The combination of this function with the majorant $M\psi_2(x, b, \epsilon)$, both for $\alpha = 1/2$, estimates the first derivative of the solution to the problem (4.1) in the vicinity of $x = 0$ when $a(0) = 0$, $a'(0) > 0$. So, a viscous flow in the direction of the repulsion of the gas from the wall may have a combined boundary layer. The function $M\psi_3(x, b, \epsilon)$ with $\alpha = 1$ also bounds in the vicinity of the boundary point $x = 0$ the first derivative of the solution to the problem (4.7) when $p = 1$, $a(0) > 1$.

Logarithmic Function

One more important majorant function satisfying (4.11) appears in an estimate of the first derivative of the solution to the problem (4.7) with $p = 1$, $a(0) = 1$. Qualitatively, the solution $u(x, \epsilon)$ in the boundary layer is described in this case by a logarithmic function

$$c(x)\frac{\ln(\epsilon^\alpha + x)}{\ln \epsilon^\alpha}$$

with $\alpha = 1$. The first derivative of $u(x, \epsilon)$ is estimated in the boundary layer by the basic majorant $M\psi_4(x, \epsilon)$ with $\alpha = 1$, where

$$\psi_4(x, \epsilon) = \frac{1}{(\epsilon^\alpha + x)|\ln \epsilon|} \ . \tag{4.27}$$

Relations Among Basic Majorants

For the majorants $\psi_i(x, b, \epsilon)$, $i = 1, 2, 3$, and $\psi_4(x, \epsilon)$, the following relations, expressed by inequalities, apply:

$$\begin{aligned}
&\psi_i(x, b_1, \epsilon) \le M\psi_i(x, b_2, \epsilon) \ , \qquad b_1 \ge b_2 > 0 \ , \qquad i = 1, 2, 3 \ , \\
&\psi_1(x, b, \epsilon) \le M\psi_2(x, d, \epsilon) \qquad \text{for arbitrary } d > 0 \ , \\
&\psi_i(x, b, \epsilon) \le M|\ln \epsilon^\alpha|\psi_4(x, \epsilon) \ , \qquad b > 0 \ , \qquad i = 1, 2 \ .
\end{aligned} \tag{4.28}$$

These relations are readily proved. For example, the confirmation of the second inequality follows from

$$\begin{aligned}
\frac{\psi_1(x, b, \epsilon)}{\psi_2(x, d, \epsilon)} &= \frac{(\epsilon^\alpha + x)^{d+1}}{\epsilon^{\alpha(1+d)}} \exp(-bx/\epsilon^\alpha) \\
&\le 2^{d+1} \exp(-bx/\epsilon^\alpha) + \left(\frac{x}{\epsilon^\alpha}\right)^{d+1} \exp(-bx/\epsilon^\alpha) \ .
\end{aligned}$$

As

$$x^n \exp(-cx) \le M \ , \qquad n > 0 \ , \qquad c > 0 \ , \qquad 0 \le x < \infty \ ,$$

where the constant M is dependent only on n and c, we find that

$$\frac{\psi_1(x,b,\epsilon)}{\psi_2(x,d,\epsilon)} \le M,$$

i.e. the second inequality of (4.28) is proved.

Interior Layers

The solutions to the problems (4.1) and (4.6) also have interior layers with large variations. Moreover, the problem (4.6) qualitatively models the wave tracks and shocks of many gas dynamic flows with layers of sharp variation away from the boundaries. The problems (4.1), (4.6) are remarkable in that the first derivatives of their solutions in the interior layers can be estimated by combinations of the same majorants of the types $\psi_i(x,b,\epsilon),\ i=1,2,3$, however, in which the independent variable x is replaced by $|x-x_0|$, namely,

$$\psi_i(|x-x_0|,b,\xi)\ , \qquad i=1,2,3\ ,$$

where x_0 is the center of the layer, i.e. x_0 is the point of the fastest local variation. Thus, the estimates near the boundary point $x=0$ serve also to estimate the first derivative in the interior layer near the point $0<x_0<1$, as well in the boundary layer near the point $x_0=1$. For example, for the derivative of the solution $u(x,\xi)$ to the problem (4.1), we have

$$\left|\frac{\mathrm{d}u}{\mathrm{d}x}(x,\epsilon)\right| \le M\psi_3(|x-x_0|,b,\xi)\ , \qquad |x-x_0|\le m\ ,$$

for same $m>0$, where x_0 is defined by the condition

$$a(x_0)=0\ , \qquad a'(x_0)\ge f_u(u,x)\ .$$

The location of the center of the interior layer of the solution to the problem (4.6) is dependent on the properties of the function

$$b(u)=\int_{u_0}^{u} a(\eta)\mathrm{d}\eta.$$

An interior layer of the solution exists if

$$b(u_0)=b(u_1)\ , \qquad b(u)>b(u_0)\ , \tag{4.29}$$

and its center point x_0 is defined by the first nonzero coefficients of the Taylor expansions of the function $b(u)$ in the vicinity of the points u_0 and u_1. For example, if in addition to the condition (4.29), the condition

$$b'(u_i)=a_i\ne 0\ , \qquad i=0,1,$$

is satisfied, then

$$x_0 = a_1/(a_1 - a_0) \ , \qquad i = 0, 1 \ ,$$

and

$$|u'(x)| \leq M\psi_1(|x - x_0|, b, \epsilon) \ , \qquad 0 \leq x \leq 1 \ ,$$

where the constant b is defined by a_0 and a_1.

An instructive example is that of an interior layer of the solution to the problem (4.6) that moves unlimitedly to the boundary as the parameter ϵ approaches zero. Such a layer is realized in a solution to the problem (4.6) if the function $b(u)$ satisfies the condition (4.29) and

$$a(u_0) \neq 0 \ , \qquad \frac{\mathrm{d}^k}{\mathrm{d}u^k} b(u_1) = 0 \ , \qquad k \leq p, \ \ p \geq 1 \ ,$$

$$\frac{\mathrm{d}^{p+1}}{\mathrm{d}u^{p+1}} b(u_1) \neq 0 \ .$$

For the derivative of the solution $u(x, \epsilon)$, in this case, we have

$$|u'(x, \epsilon)| \leq \begin{cases} M[1 + \epsilon^{-1}\{\exp[a_0(x - x_0)/\epsilon]\}] \ , \ 0 \leq x \leq x_0 \ , \\ M[1 + \epsilon^{b}(\epsilon + x - x_0)^{(-1-b)}] \ , \qquad x_0 \leq x \leq 1 \ , \end{cases} \tag{4.30}$$

where

$$b = 1/p \ , \qquad x_0 = (1 + b)\frac{\epsilon}{a_0} \ln \epsilon^{-1} \ .$$

So, the derivative in the left-hand part of the layer is estimated by the majorant $M\psi_1(x - x_0, a_0, \epsilon)$ and in the right-hand part it is bounded by $M\psi_2(x - x_0, b, \epsilon)$. Therefore, the stretching of the variable x should be different in the left- and right-hand parts of the boundary layer. As the center point x_0 approaches the boundary unlimitedly as $\epsilon \to 0$, the solution with an interior layer tends to the solution of the reduced problem ($\epsilon = 0$) with a boundary layer (Fig. 4.3). Thus, this example shows a drawback of the analysis of the locations of layers by means of reduced problems.

Estimates of the Higher Derivatives

The basic majorants of the higher derivatives of the solutions to the problems (4.1), (4.6)–(4.9) in the layers have the form of the derivatives of the majorants (4.24)–(4.27). Namely, they are expressed by the following functions $\psi_i^n(x, b, \epsilon)$, $i = 1, 2, 3$, and $\psi_4^n(x, \epsilon)$:

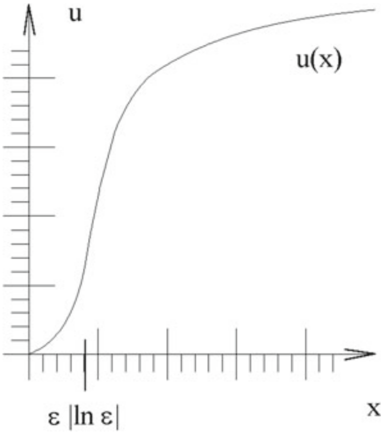


Fig. 4.3 Function with a mixed interior layer approaching the boundary unlimitedly

$$\begin{aligned}
\psi_1^n(x, b, \epsilon) &= \epsilon^{-\alpha n} \exp(-b|x - x_0|/\epsilon^{\alpha}) \ ,\\
\psi_2^n(x, b, \epsilon) &= \epsilon^{\alpha b}/(\epsilon^{\alpha} + |x - x_0|)^{b+n} \ ,\\
\psi_3^n(x, b, \epsilon) &= (\epsilon^{\alpha} + |x - x_0|)^{b-n} \ ,\\
\psi_4^n(x, \epsilon) &= \frac{1}{(\epsilon^{\alpha} + |x - x_0|)^n |\ln \epsilon|} \ ,
\end{aligned} \tag{4.31}$$

where x_0 is the point at the center of the layer. Here, the constants α and b may differ for different functions.

The general estimate of the nth derivative of $u(x, \epsilon)$ in the vicinity of the point x_0 has the form

$$\left|\frac{\mathrm{d}^n u}{\mathrm{d}x^n}\right| \leq M[1 + \psi^n(x, \epsilon)] \ , \qquad |x - x_0| \leq m \ , \tag{4.32}$$

where $\psi^n(x, \epsilon)$ is a combination of the functions described by the formulas (4.31), i. e.

$$\begin{aligned}
&\psi^n(x, \epsilon) = c_1\psi_1^n(x, b_1, \epsilon) + c_2\psi_2^n(x, b_2, \epsilon) + c_3\psi_3^n(x, b_3, \epsilon) + c_4\psi_4^n(x, \epsilon) \ ,\\
&b_i > 0 \ , \quad i = 1, 2, 3 \ , \quad c_i \geq 0 \ , \quad i = 1, 2, 3, 4 \ .
\end{aligned} \tag{4.33}$$

The distinctive feature of these estimates (4.32) is that they guarantee that the local transformation $x(\xi)$ obtained from equations of the forms (4.12), (4.22), (4.23) for

$$\psi(x, \epsilon) = 1 + m_1(\psi^n(x, \epsilon))^{1/n} \ , \quad m_1 > 0 \ , \tag{4.34}$$

is suitable for smoothing the higher-order singularities, i.e. the following estimate is valid:

$$\left|\frac{\mathrm{d}^n}{\mathrm{d}\xi^n} u[x(\xi), \epsilon]\right| \leq M \ , \qquad |x - x_0| \leq m \ . \tag{4.35}$$

As

$$\psi^n(x,\epsilon) \le M_1\overline{\psi}(x,\epsilon)\ ,$$

where

$$\overline{\psi}(x,\epsilon) = d_1\psi_1^1(x, b_1/n, \epsilon) + d_2\psi_2^1(x, b_2/n, \epsilon) + d_3\psi_3^1(x, b_3/n, \epsilon) + d_4(\psi_4^n(x,\epsilon))^{1/n}\ ,$$

$d_i \ge 0,\ i = 1, 2, 3, 4,$ we can also assume in (4.34)

$$\psi(x,\epsilon) = 1 + m_2\overline{\psi}(x,\epsilon)\ , \quad m_2 > 0\ .$$

Thus, the majorant $\overline{\psi}(x,\epsilon)$ is a combination of the basic functions (4.24)–(4.27), where x is substituted with $\mid x - x_0 \mid$, which estimate the first derivative in layers.

Invariants of Equations

It is apparent that the boundary layers of the singularly perturbed solutions do not vanish when the coordinate x is replaced by a coordinate q with a one-to-one smooth transformation $x(q)$ of the interval [0, 1] if $x(q)$ is independent of ϵ. And it is apparent that (4.1) has some invariants under such transformations which determine the qualitative behavior of the solutions to the equations. For example, (4.1) in the new independent variable q and dependent variable $u_1(q,\epsilon) = u[x(q),\epsilon]$ has the form

$$\begin{aligned}&\epsilon u_1'' + a_1(q)u_1' = f_1(q, u_1)\ , \qquad 0 < q < 1\ ,\\ &a_1(q) = x'a[x(q)] + \epsilon x''/(x')^2\ ,\\ &f_1(q_1, u_1) = (x')^2 f[x(a), u_1]\ .\end{aligned}$$

So, the invariants are

(1) the sign of the coefficient $a_1(q)$ of the first derivative for $q = 0,\ \epsilon = 0$,
(2) the expression

$$a_1'(0)/f_{u_1}[0, u_1(0)]$$

for $\epsilon = 0$ when $a_1(0) = 0$.

As was mentioned above, the estimate of the first derivative of the solution to this problem is defined through these two invariants.

The invariant defining the structure of the solution to the problem (4.7) in the boundary layer is also the value of the coefficient $a(x)$ for $x = 0$.

The qualitative behavior of the solution to the problem (4.6) is determined by the values of the derivatives of the function $b(u) = \int_{u_0}^{u} a(u)\mathrm{d}u$ at the points u_0 and u_1. These values are the invariants of transformations $v = f(u)$ of the dependent variable u.

The preceding remarks show the importance of the study of the invariants of equations and their connection with the qualitative features of the solutions in the layers.

4.4 Basic Intermediate Transformations

This section gives a detailed description of the basic univariate stretching functions and, consequently, the contraction functions, which are applied to construct the intermediate transformations that provide grid clustering in boundary and interior layers.

4.4.1 Basic Local Stretching Functions

The derivatives of the solution of a singularly perturbed equation are large in the center of a layer and decrease towards its boundary. Outside the layers, the derivatives are estimated by a constant M independent of the small parameter ϵ, while within a layer, the derivatives of any singular solution with respect to the coordinate x transverse to the layer can be bounded by the derivatives of one or a combination of the basic functions $\psi_i^n(x, b, \epsilon),\ i = 1, 2, 3$, and $\psi_4^n(x, \epsilon)$ defined by (4.31). These basic functions generate four basic univariate transformations $\varphi_i(x, \epsilon)$ which stretch the layers. The introduction of these functions to stretch the coordinate transverse to a layer nonuniformly allows one to build a new local coordinate system with respect to which the solution has no layers with large derivatives.

Local coordinate transformations $\varphi_i(x, \epsilon)$ which nonuniformly stretch the coordinate lines within the boundary layers have already been utilized to generate grids for the numerical solution of some singularly perturbed problems. Analytical and numerical analyses have demonstrated that the grids generated in the layers by these coordinate transformations allow one to obtain a numerical solution to a singularly perturbed problem which converges uniformly with respect to the small parameter to the exact solution. Also, the solution can be interpolated uniformly over the entire region, including the layers. Therefore, the incorporation of stretching functions into formulas for intermediate transformations is a promising way to develop grid techniques.

The four standard, local, stretching, coordinate transformations denoted by $\varphi_i(x, \epsilon),\ i = 1, 2, 3, 4$, where x is a scalar-valued independent variable interpreted here as a coordinate orthogonal to a layer and ϵ is a small parameter, have been designed only to stretch the boundary layer at the point $x = 0$. These functions are defined by integrating the basic majorants (4.24)–(4.27). In reality, these local stretching transformations are boundary layer functions which describe the qualitative behavior of the physical solutions across the boundary layers. The functions which stretch the interior layers are derived from these basic transformations by the procedures described in Sect. 4.4.4.

The boundary layer functions corresponding to the majorants (4.24)–(4.27) are computed by solving an initial-value problem of the type (4.12):

$$\begin{aligned}
&\frac{\mathrm{d}\varphi}{\mathrm{d}x} = c\psi(x, \epsilon) , \qquad x > 0 , \\
&\varphi(0) = 0 ,
\end{aligned}$$

where $\psi(x,\epsilon)$ is the majorant of the function $(u^{(n)}(x,\epsilon))^{1/n}$ first derivative. For convenience, the local stretching functions are written in a form that satisfies the following conditions:

$$\varphi(0,\epsilon)=0\;, \qquad \frac{\mathrm{d}}{\mathrm{d}x}\varphi(x,\epsilon)>0\;.$$

The first function is the well-known exponential mapping

$$\varphi_1(x,\epsilon)=\frac{1-\exp(-bx/\epsilon^{\alpha})}{c}\;, \qquad \alpha>0\;, \qquad b>0\;, \qquad c>0\;. \tag{4.36}$$

The next two local stretching mappings are power functions,

$$\varphi_2(x,\epsilon)=\frac{1-[\epsilon^{\alpha}/(\epsilon^{\alpha}+x)]^{b}}{c}\;, \qquad \alpha>0\;, \qquad b>0\;, \qquad c>0\;, \tag{4.37}$$

and

$$\varphi_3(x,\epsilon)=\frac{(\epsilon^{\alpha}+x)^{b}-\epsilon^{\alpha b}}{c}\;, \qquad \alpha>0\;, \qquad 1>b>0\;, \qquad c>0\;. \tag{4.38}$$

The fourth local stretching function is a logarithmic map

$$\varphi_4(x,\epsilon)=\frac{\ln(1+x\epsilon^{-\alpha})}{c\ln(1+\epsilon^{-\alpha})}\;, \qquad \alpha>0\;, \qquad c>0\;. \tag{4.39}$$

The numbers α, b, and c in these expressions for the stretching functions $\varphi_i(x,\epsilon)$ are positive constants. The number α shows the scale of a layer. It is easily computed analytically. For example, for problems of viscous flows, $\alpha=1/2$ in a boundary layer and $\alpha=1$ in a shock wave. The constant c serves to control the length of the interval of the new stretching coordinate φ that is transformed into the layer. The constant b controls the type of stretching nonuniformity and the width of the layer. The parameter ϵ provides the major contribution to determining the slopes of the stretching functions in the vicinity of the point $x=0$.

The stretching functions $\varphi_i(x,\epsilon),\; i=1,2,3,4$, for $\epsilon^{\alpha}=1/30$ are shown in Fig. 4.4. The symbols $+$, $\times$, $\Diamond$, and $\Box$ identify the functions $\varphi_1(x,\epsilon)$, $\varphi_2(x,\epsilon)$, $\varphi_3(x,\epsilon)$, and $\varphi_4(x,\epsilon)$, respectively. The constant c is selected to satisfy the restriction $\varphi_i(1,\epsilon)=1,\; i=1,2,3,4$.

Width of Boundary Layers

The interval where any function $\varphi_i(x,\epsilon)$ provides a stretching of the coordinate x coincides with the interval where the first derivative with respect to x of this function $\varphi_i(x,\epsilon)$ is large. The first derivatives of the basic stretching transformations $\varphi_i(x,\epsilon),\; i=1,2,3,4$, are

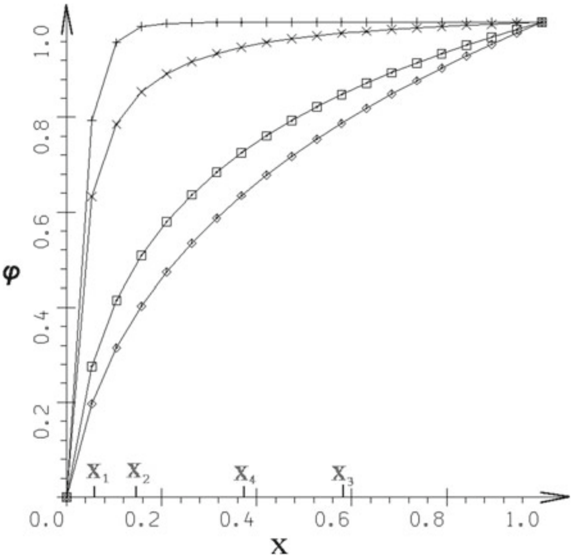


Fig. 4.4 Basic local stretching functions

$$\begin{aligned}
\frac{\mathrm{d}\varphi_1}{\mathrm{d}x}(x,\epsilon) &= \frac{b\epsilon^{-\alpha}}{c}\exp(-bx/\epsilon^{\alpha})\ , \qquad \alpha>0\ , \quad b>0\ ,\\
\frac{\mathrm{d}\varphi_2}{\mathrm{d}x}(x,\epsilon) &= \frac{b\epsilon^{\alpha b}}{c(\epsilon^{\alpha}+x)^{b+1}}\ , \qquad \alpha>0\ , \quad b>0\ ,\\
\frac{\mathrm{d}\varphi_3}{\mathrm{d}x}(x,\epsilon) &= \frac{b}{c}(\epsilon^{\alpha}+x)^{b-1}\ , \qquad \alpha>0\ , \quad 1>b>0\ ,\\
\frac{\mathrm{d}\varphi_4}{\mathrm{d}x}(x,\epsilon) &= \frac{1}{c\ln(1+\epsilon^{-\alpha})(\epsilon^{\alpha}+x)}\ , \qquad \alpha>0\ . \qquad (4.40)
\end{aligned}$$

For the first derivative $\mathrm{d}\varphi_i(x,\epsilon)/\mathrm{d}x$ of the stretching functions $\varphi_i(x,\epsilon)$, $i = 1, 2, 4$, one can readily obtain the following relations:

$$\begin{aligned}
&\frac{\mathrm{d}\varphi_1}{\mathrm{d}x}(x,\epsilon) \le M\frac{\mathrm{d}\varphi_2}{\mathrm{d}x}(x,\epsilon)\ , \qquad 0\le x\le 1\ ,\\
&\frac{\mathrm{d}}{\mathrm{d}x}\varphi_2(x,\epsilon) \le M\frac{\mathrm{d}}{\mathrm{d}x}\varphi_4(x,\epsilon^{p})\ , \qquad p>1\ , \qquad 0\le x\le 1\ ,
\end{aligned}$$

where the constant M does not depend on ϵ. Therefore, the stretching transformation $\varphi_2(x,\epsilon)$ can be used to eliminate both exponential and power layers, while the mapping $\varphi_4(x,\epsilon)$ is suitable for smoothing exponential and power layers and also the singularities of the type described by $\varphi_4(x,\epsilon)$.

The derivative $\mathrm{d}\varphi_i(x,\epsilon)/\mathrm{d}x$ of each local stretching mapping $\varphi_i(x,\epsilon)$ is large in the vicinity of the point $x = 0$ when the parameter ϵ is small, and decreases as x increases. The boundary of the layer for the function $\varphi_i(x,\epsilon)$ is defined to be at the point x_i where the modulus of the first derivative $(\mathrm{d}/\mathrm{d}x)\varphi_i(x,\epsilon)$ is limited by a constant $M > 0$ independent of the parameter ϵ, i.e.

$$\left|\frac{\mathrm{d}\varphi_i}{\mathrm{d}x}(x_i,\epsilon)\right| \le M \ .$$

The value of x_i coincides with the width of the layer, denoted by $\Delta_i(x,\epsilon)$, of the function $\varphi_i(x,\epsilon)$. So, from (4.40),

$$\begin{aligned}
x_1 &= \Delta_1(x,\epsilon) = \frac{\epsilon^\alpha}{b}\ln\epsilon^{-\alpha} \ , \qquad \frac{\mathrm{d}\varphi_1}{\mathrm{d}x}(x_1,\epsilon) = b/c \ , \\
x_2 &= \Delta_2(x,\epsilon) = \epsilon^{\alpha b/(b+1)} - \epsilon^\alpha \ , \qquad \frac{\mathrm{d}\varphi_2}{\mathrm{d}x}(x_2,\epsilon) = b/c \ , \\
x_3 &= \Delta_3(x,\epsilon) = d\epsilon^0 - \epsilon^\alpha \ , \qquad \frac{\mathrm{d}\varphi_3}{\mathrm{d}x}(x_3,\epsilon) = d^{(b-1)}b/c \ , \\
x_4 &= \Delta_4(x,\epsilon) = \frac{1}{\ln(1+\epsilon^{-\alpha})} - \epsilon^\alpha \ , \qquad \frac{\mathrm{d}\varphi_4}{\mathrm{d}x}(x_4,\epsilon) = 1/c \ .
\end{aligned} \tag{4.41}$$

These expressions evidently provide a rule for controlling, with the constant b, the width of the layers where the grid nodes are to be clustered. In order to make the layer wider, this constant needs to be reduced.

Also, from (4.41), one can obtain the maximum value $m_i > 0$ of the parameter ϵ for each stretching function $\varphi_i(x,\epsilon)$. The value of m is obtained from the obvious condition $x_i < 1, \ i = 1,2,3,4$. This value defines the range for ϵ for the application of the stretching, $0 < \epsilon \le m_i$, and consequently the contraction functions for the construction of the intermediate transformations. In the following discussion, we consider only those values of the parameter ϵ which are subject to the restriction $x_i < 1$.

The formulas for $\Delta_2(x,\epsilon), \ \Delta_3(x,\epsilon)$, and $\Delta_4(x,\epsilon)$ contain the quantity $-\epsilon^\alpha$, which, asymptotically, does not influence the width of the layers, but is included purely to simplify the expression for the first derivative of $\varphi_i(x,\epsilon)$ at the point x_i. Equation (4.41) clearly show that there exists a number $\epsilon_0 > 0$ such that

$$\Delta_3(x,\epsilon) > \Delta_4(x,\epsilon) > \Delta_2(x,\epsilon) > \Delta_1(x,\epsilon)$$

for all positive $\epsilon < \epsilon_0$.

The equations in (4.40) indicate that the length of the central part of the layer, where the first derivative reaches the maximum values $M\epsilon^{-\alpha}$ for $\varphi_i(x,\epsilon), \ i = 1,2,3$, and $M\epsilon^{-\alpha}/\ln\epsilon^{-\alpha}$ for $\varphi_4(x,\epsilon)$, is similar for all functions $\varphi_i(x,\epsilon)$ and equals $m\epsilon^\alpha$. However, the relations (4.41) state that the transitional part of the layer, between the center and the boundary, is considerably larger than $m\epsilon^\alpha$, especially for the functions $\varphi_2(x,\epsilon)$, $\varphi_3(x,\epsilon)$, and $\varphi_4(x,\epsilon)$. The first derivative in the transitional part of the layer is also large when the parameter ϵ is small, and therefore stretching of the central part is required as well, though to a lesser degree.

In contrast, the first derivative of each function $\varphi_i(x,\epsilon), \ \ i = 1,2,4$, is very small outside a layer when the parameter ϵ is small. Namely, from (4.41), for a point $x = \mathrm{const}$ lying outside a layer,

$$\begin{aligned}\frac{\mathrm{d}\varphi_1}{\mathrm{d}x}(x,\epsilon) &\sim \epsilon^{-\alpha}\exp(-b_1/\epsilon^{\alpha}) \ ,\\ \frac{\mathrm{d}\varphi_2}{\mathrm{d}x}(x,\epsilon) &\sim \epsilon^{\alpha m} \ ,\\ \frac{\mathrm{d}\varphi_4}{\mathrm{d}x}(x,\epsilon) &\sim \frac{1}{\ln \epsilon^{-\alpha}} \ .\end{aligned}$$

Of these three expressions for the first derivative, the last one has the least tendency to become zero outside the layer and, when the parameter ϵ is not too small, the function $\varphi_4(x,\epsilon)$ can be used over the whole interval [0, 1] to introduce a new coordinate variable to stretch the coordinate x in the layer.

The boundary point x_i of the layer for the transformation $\varphi_i(x,\epsilon)$ corresponds to the value $\varphi_i(x_i,\epsilon)=\varphi_i$ of the dependent variable $\varphi_i(x,\epsilon)$. So, φ_i defines the interval $[0,\varphi_i]$ which is transformed into the layer by the function inverse to $\varphi_i(x,\epsilon)$. The values of these points $\varphi_i(x_i,\epsilon)=\varphi_i$ corresponding to the values of x_i specified by (4.41) are given by

$$\begin{gathered}\varphi_1=\frac{1-\epsilon^{\alpha}}{c} \ , \quad \varphi_2=\frac{1-\epsilon^{\alpha b/(b+1)}}{c} \ ,\\ \varphi_3=\frac{d^b-\epsilon^{\alpha b}}{c} \ , \quad \varphi_4=\frac{\ln(\epsilon^{-\alpha})-\ln[\ln(1+\epsilon^{-\alpha})]}{c\ln(1+\epsilon^{-\alpha})} \ .\end{gathered} \tag{4.42}$$

These expressions imply that

$$\varphi_i \to 1/c \ , \quad i=1,2,4 \ , \quad \varphi_3 \to d^b/c \ ,$$

when ϵ tends to 0.

4.4.2 Basic Boundary Contraction Functions

The functions $\varphi_i(x,\epsilon)$ stretch the coordinate x within the narrow layers $[0,x_i]$, and therefore the mappings that are inverse to $\varphi_i(x,\epsilon)$ provide a contraction of the coordinate φ in the interval $[0,\varphi_i]$. Thus, these inverse functions can be used as the univariate local transformations $q^j(\xi^j)$, where $q^j=x$, $\xi^j=\varphi$, to build the intermediate n-dimensional transformations

$$\boldsymbol{q}(\boldsymbol{\xi}) : \varXi^n \to Q^n$$

which generate nodal clustering in the layers along the selected coordinates q^j.

Taking into account (4.36)–(4.39), the local inverse transformations $x_i(\xi,\epsilon)$ of the corresponding stretching functions $\varphi_i(x,\epsilon)$ have the following form:

$$\begin{aligned}
x_1(\xi,\epsilon) &= -\frac{\epsilon^\alpha}{b}\ln(1-c\xi)\ , \qquad \alpha>0\ , \quad b>0\ ,\\
x_2(\xi,\epsilon) &= \epsilon^\alpha\Big((1-c\xi)^{-1/b}-1\Big)\ , \qquad \alpha>0\ , \quad b>0\ ,\\
x_3(\xi,\epsilon) &= (\epsilon^{\alpha b}+c\xi)^{1/b}-\epsilon^\alpha\ , \qquad \alpha>0\ , \quad 1>b>0\ ,\\
x_4(\xi,\epsilon) &= \epsilon^\alpha\Big((1+\epsilon^{-\alpha})^{c\xi}-1\Big)\ , \qquad \alpha>0\ .
\end{aligned} \tag{4.43}$$

Differentiating these expressions p times gives

$$\begin{aligned}
\frac{\mathrm{d}^p x_1(\xi,\epsilon)}{\mathrm{d}\xi^p} &= d_1\epsilon^\alpha(1-c\xi)^{-p}\ ,\\
\frac{\mathrm{d}^p x_2(\xi,\epsilon)}{\mathrm{d}\xi^p} &= d_2\epsilon^\alpha(1-c\xi)^{-1/b-p}\ ,\\
\frac{\mathrm{d}^p x_3(\xi,\epsilon)}{\mathrm{d}\xi^p} &= d_3(\epsilon^{\alpha b}+c\xi)^{1/b-p}\ ,\\
\frac{\mathrm{d}^p x_4(\xi,\epsilon)}{\mathrm{d}\xi^p} &= d_4\epsilon^\alpha(1+\epsilon^{-\alpha})^{c\xi}\ln^p(1+\epsilon^{-\alpha})\ ,
\end{aligned}$$

where $|d_i| \le M,\ i=1,2,3,4$. Therefore, the points ξ_i^p and $x_i^p = x_i(\xi_i^p,\epsilon)$, $i=1,2,3,4$, such that on the interval $[0,\xi_i^p]$ the pth derivative of the mapping $x_i(\xi,\epsilon)$ is ϵ-uniformly bounded, are given by the equations

$$\begin{aligned}
\xi_1^p &= \frac{1-\epsilon^{\alpha p}}{c}\ , & x_1^p &= \frac{\epsilon^\alpha}{pb}\ln\epsilon^{-\alpha}\ ,\\
\xi_2^p &= \frac{1-\epsilon^\beta}{c}\ , & x_2^p &= \epsilon^{p\beta}-\epsilon^\alpha\ , \quad \beta=\frac{\alpha b}{1+pb}\ ,\\
\xi_3^p &= 1\ , & x_3^p &= 1 \text{ if } 1/b\ge p\ ,\\
\xi_4^p &= \frac{\ln\epsilon^{-\alpha}-p\ln[\ln(1+\epsilon^{-\alpha})]}{c\ln(1+\epsilon^{-\alpha})}\ , & x_4^p &= \frac{1}{\ln^p(1+\epsilon^{-\alpha})}-\epsilon^\alpha\ .
\end{aligned}$$

Since $\xi_i^{p_1} \le \xi_i^{p_2}$ and $x_i^{p_1} \le x_i^{p_2},\ i=1,2,3,4$, if $p_1 \ge p_2$, we conclude that

$$\left|\frac{\mathrm{d}^j x_i(\xi,\epsilon)}{\mathrm{d}\xi^j}\right| \le M\ , \quad 0\le\xi\le\xi_i^p\ , \quad j\le p\ , \quad i=1,2,3,4\ .$$

It is clear that the interval $[0,x_i^p],\ i=1,2,3,4$, is the layer of order $p-1$ of the corresponding singular function $\varphi_i(x,\epsilon),\ i=1,2,3,4$. Hence, the transformation $x_i(\xi,\epsilon),\ i=1,2,3,4$, maps the interval $[0,\varphi_i^p]$ onto the layer of order $p-1$ of the singular function $\varphi_i(x,\epsilon),\ i=1,2,3,4$.

Notice the functions $x_i(\xi,\epsilon),\ i=1,2,3,4$, can be used as local trial functions to treat singularly perturbed problems by Galerkin or Petrov–Galerkin finite-element methods.

The first derivative of any of the functions $x_i(\xi,\epsilon)$ is small at the points $\xi,\ 0\le\xi\le\varphi_i(x_i^1,\epsilon)=\varphi_i^1$, and therefore the magnitude of the grid spacing in the x

direction of the grid generated by the mapping $x_i(\xi,\epsilon)$ is also small in the layer; it is approximately of the order of $(\mathrm{d}/\mathrm{d}\xi)[x_i(\xi,\epsilon)]h$. The degree of grid clustering at the center of the layer reaches a value of ϵ^k and increases at the points near the boundary x_i^1 of the layer.

The stretching functions $\varphi_i(x,\epsilon)$ themselves describe the qualitative behavior of solutions within their zones of large gradients along the coordinate lines normal to the layers. Therefore, as any mapping $x_i(\xi,\epsilon)$, $i = 1,2,3,4$, is the inverse of the corresponding function $\varphi_i(x,\epsilon)$, the grids derived from the transformations $x_i(\xi,\epsilon)$ provide the optimum nonuniform resolution of the physical solution in the layers with an economy of nodal points. However, these functions, excluding the mapping $x_3(\xi,\epsilon)$, produce excessively sparse grids outside the layers, since their first derivative, satisfying the equation

$$\frac{\mathrm{d}x_i}{\mathrm{d}\xi}(\xi,\epsilon) = 1\Big/\frac{\mathrm{d}\varphi_i}{\mathrm{d}x}[x(\xi),\epsilon]$$

is very large when $\xi > \xi_i^1$ and tends to infinity as the parameter ϵ nears zero. Therefore, the contraction functions $x_i(\xi,\epsilon)$ can only be used to provide grid clustering in the layers; outside the layers, the grids must be generated through other mappings producing less coarse grids.

Formulas for Basic Layer-Damping Transformations

In order to define a layer-damping intermediate transformation $x(\xi,\epsilon)$ on the interval [0, 1] through the use of the local univariate mappings $x_i(\xi,\epsilon)$, $i = 1,2,3,4$, from (4.43) specified on the corresponding intervals $[0,\xi_i^p]$, to provide adequate clustering of grid points, these mappings need to be extended continuously or smoothly over the whole interval [0, 1] to map this interval monotonically onto the unit interval [0, 1]. This can be done by "gluing" these local nonuniform transformations $x_i(\xi,\epsilon)$ to other mappings in the interval $[\xi_i^p, 1]$ that are more uniform than the basic functions $x_i(\xi,\epsilon)$, for example, linear or power functions. The glued transformation extending $x_i(\xi,\epsilon)$ must be smooth, or at least continuous.

Continuous Mappings.

Nonsmooth continuous layer-damping mappings, denoted here as $x_{i,\mathrm{c}}^p(\xi,\epsilon)$, can be defined as

$$x_{i,\mathrm{c}}^p(\xi,\epsilon) = \begin{cases} x_i(\xi,\epsilon)\ , & 0 \le \xi \le \xi_i^p\ , \\ x_i^p + \dfrac{(1-x_i^p)(\xi-\xi_i^p)}{1-\xi_i^p}\ , & \xi_i^p \le \xi \le 1\ . \end{cases} \tag{4.44}$$

These functions are monotonically increasing, given a suitable choice of the interval $[0, m_i]$ for the parameter ϵ, and vary from 0 to 1. Therefore, they generate the individual univariate layer-damping transformations $q(\xi,\epsilon)$, assuming

$$q(\xi,\epsilon) = x_{j,\mathrm{c}}^p(\xi,\epsilon)\ , \qquad j = 1,2,3,4\ .$$

The pth derivative of the reference function $x_3(\xi, \epsilon)$ is limited uniformly, with respect to the parameter ϵ, for all $0 < \epsilon \le m_3$, if $1/b \ge p$, so the matching of this function to any other one to transform the interval [0, 1] onto the interval [0, 1] is, in general, not necessary. The proper function, monotonically increasing and varying from 0 to 1, denoted by $x^p_{3,\,\mathrm{S}}(\xi, \epsilon)$, is obtained by adjusting the constant c in (4.43):

$$x^p_{3,\mathrm{S}}(\xi, \epsilon) = (\epsilon^{\alpha b} + c\xi)^{1/b} - \epsilon^{\alpha} \ , \quad c = (\epsilon^{\alpha} + 1)^b - \epsilon^{\alpha b} \ , \quad 1/b \ge p \ .$$

The length of any interval $[0, \xi^p_i]$, $i = 1, 2, 4$, transformed onto the corresponding layer by the corresponding function $x_{i,\mathrm{C}}(\xi, \epsilon)$ defined by (4.44) approaches the constant $1/c$ as the parameter ϵ tends to zero. This quantity $1/c$ specifies that part of the uniform grid in the interval [0, 1] of the independent variable ξ which is transformed into the layer. Obviously, the value of the constant c must be more than 1.

Smooth Mappings.

Smooth basic univariate layer-damping transformations of the interval [0, 1] can be defined by matching the basic local transformations $x_i(\xi, \epsilon)$ at the corresponding points ξ^p_i to the function of the type $x_{3,\mathrm{S}}(\xi, \epsilon)$ since its derivatives up to order p are ϵ-uniformly bounded when $1/b \ge \alpha$. The matching is made at the cost of the constant c in the formulas for $x_i(\xi, \epsilon)$. These very smooth transformations, obtained by matching smoothly the local contraction functions $x_i(\xi, \epsilon)$, $i = 1, 2, 4$, with the function $a_1 + [a_2(\xi - \xi^p_i) + a_3]^p$ and denoted below by $x_{i,\mathrm{S}}(\xi, \epsilon)$, are given here.

The local basic contraction function $x_1(\xi, \epsilon)$ is extended by the procedure of smooth matching to

$$x^p_{1,\mathrm{S}}(\xi, \epsilon) = \begin{cases} -\dfrac{\epsilon^{\alpha}}{b} \ln(1 - c\xi) \ , & 0 \le \xi \le \xi^p_1 \ , \\ x^p_1 - \epsilon^{\alpha} + \left(\dfrac{c(\xi - \xi^p_1)}{bp} + \epsilon^{\alpha/p} \right)^p \ , & \xi^p_1 \le \xi \le 1 \ , \end{cases} \tag{4.45}$$

with

$$\begin{aligned} x^p_1 &= \frac{\epsilon^{\alpha}}{pb} \ln \epsilon^{-\alpha} \ , \quad c = bp(1 - x^p_1 + \epsilon^{\alpha})^{1/p} + 1 - (1 + bp)\epsilon^{\alpha/p} \ , \\ \xi^p_1 &= (1 - \epsilon^{\alpha/p})/c \ . \end{aligned}$$

The use of the local contraction function $x_2(\xi, \epsilon)$ yields

$$x^p_{2,\mathrm{S}}(\xi, \epsilon) = \begin{cases} \epsilon^{\alpha}[(1 - c\xi)^{-1/b} - 1] \ , & 0 \le \xi \le \xi^p_2 \ , \\ x^p_2 - \epsilon^{\beta p} + \left(\dfrac{c}{bp}(\xi - \xi^p_2) + \epsilon^{\beta} \right)^p \ , & \xi^p_2 \le \xi \le 1 \ , \end{cases} \tag{4.46}$$

where

$$\beta = \frac{b\alpha}{bp+1}, \quad x_2^p = \epsilon^{\beta p} - \epsilon^{\alpha}, \quad c = bp[(1 - x_2^p + \epsilon^{\beta p})^{1/p} - \epsilon^{\beta}] + 1 - \epsilon^{\beta},$$
$$\xi_2^p = \frac{1-\epsilon^{\beta}}{c}.$$

When $b = 1$, this transformation is as follows:

$$x_{2,\mathrm{s}}^p(\xi, \epsilon) = \begin{cases} \epsilon^{\alpha} \dfrac{c\xi}{1-c\xi} \ , & 0 \le \xi \le \xi_2^p \ , \\ x_2^p - \epsilon^{\beta p} + \left(\dfrac{c}{p}(\xi - \xi_2^p) + \epsilon^{\beta}\right)^p \ , & \xi_2^p \le \xi \le 1 \ , \end{cases} \tag{4.47}$$

where

$$\beta = \frac{\alpha}{p+1} \ , \quad x_2^p = \epsilon^{\beta p} - \epsilon^{\alpha} \ , \quad c = \frac{p}{1-\xi_2^p}[(1 - x_2^p + \epsilon^{\beta p})^{1/p} - \epsilon^{\beta}] \ ,$$
$$\xi_2^p = \frac{1-\epsilon^{\beta}}{c} \ .$$

The function $x_3(\xi, \epsilon)$ gives

$$x_{3,\mathrm{s}}^p(\xi, \epsilon) = (c\xi + \epsilon^{\alpha/p})^p - \epsilon^{\alpha} \ , \quad c = (1+\epsilon^{\alpha})^{1/p} - \epsilon^{\alpha/p} \ . \tag{4.48}$$

Finally, for the fourth basic mapping $x_4(\xi, \epsilon)$, we obtain, for $p > 1$,

$$x_{4,\mathrm{s}}^p(\xi, \epsilon) = \begin{cases} \epsilon^{\alpha}[(1+\epsilon^{-\alpha})^{c\xi} - 1] \ , & 0 \le \xi \le \xi_4^p \ , \\ x_4^p - y + \left[\dfrac{c}{p}(\xi - \xi_4^p) + y^{1/p}\right]^p \ , & \xi_4^p \le \xi \le 1 \ , \end{cases} \tag{4.49}$$

where

$$x_4^p = \ln^{-p}(1+\epsilon^{-\alpha}) - \epsilon^{\alpha} \ , \quad y = [(x_4^p + \epsilon^{\alpha})\ln(1+\epsilon^{-\alpha})]^{p/(p-1)} \ ,$$
$$c = p[(1 + y - x_4^p)^{1/p} - y^{1/p}] + \frac{\ln(1 + x_4^p/\epsilon^{\alpha})}{\ln(1+\epsilon^{-\alpha})} \ , \quad \xi_4^p = \frac{\ln(1 + x_4^p/\epsilon^{\alpha})}{c\ln(1+\epsilon^{-\alpha})} \ .$$

While, for $p = 1$, we find

$$x_{4,\mathrm{s}}^1(\xi, \epsilon) = \begin{cases} \epsilon^{\alpha}[(1+\epsilon^{-\alpha})^{c\xi} - 1] \ , & 0 \le \xi \le \xi_4^1 \ , \\ x_4^1 + c(\xi - \xi_4^1) \ , & \xi_4^1 \le \xi \le 1 \ , \end{cases} \tag{4.50}$$

where

$$x_4^1 = \ln^{-1}(1+\epsilon^{-\alpha}) - \epsilon^{\alpha} \ , \quad c = 1 - x_4^1 + \frac{\ln(\epsilon^{-\alpha}) - \ln[\ln(1+\epsilon^{-\alpha})]}{\ln(1+\epsilon^{-\alpha})} \ ,$$
$$\xi_4^1 = \frac{\ln(\epsilon^{-\alpha}) - \ln[\ln(1+\epsilon^{-\alpha})]}{c\ln(1+\epsilon^{-\alpha})} \ .$$

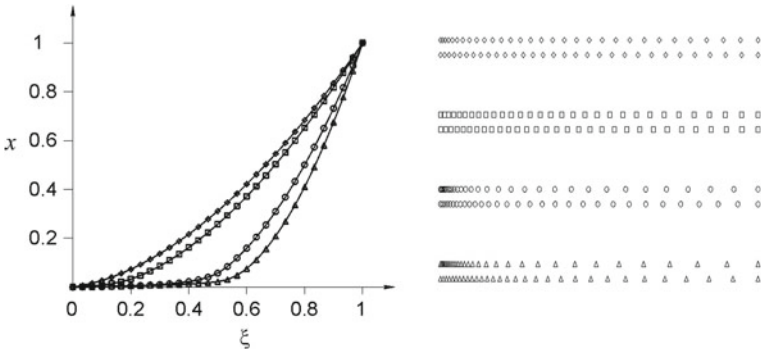


Fig. 4.5 Basic layer-damping functions (*left*) and corresponding layer-resolving meshes (*right*)

Figure 4.5 (left) illustrates the functions $x^p_{i,\,\mathrm{S}}(\xi, \epsilon)$ for $\epsilon = 2 \cdot 10^{-3}$ and $p = 2$. The right-hand side of Fig. 4.5 depicts the corresponding grid nodes derived by these functions for $\epsilon = 10^{-1}$ and $\epsilon = 1/30$. The symbols $\triangle$, O, $\square$, and $\Diamond$ correspond to the functions $x^2_{1,\,\mathrm{S}}(\xi, \epsilon)$, $x^2_{2,\,\mathrm{S}}(\xi, \epsilon)$, $x^2_{3,\,\mathrm{S}}(\xi, \epsilon)$, and $x^2_{4,\,\mathrm{S}}(\xi, \epsilon)$, respectively.

The first derivative of the local function

$$x_4(\xi, \epsilon) = \epsilon^k \Big((1 + \epsilon^{-\alpha})^{c\xi} - 1 \Big)$$

tends to $\ln \epsilon^{-\alpha}$ as $\xi > \xi^1_4$. This quantity is large when the parameter ϵ is very small; however, if ϵ is not too small, the magnitude of $\ln \epsilon^{-\alpha}$, which characterizes the grid spacing, is not very large and may be tolerable for grid generation. In this case, it is reasonable to use the local basic transformation $x_4(\xi, \epsilon)$ as a global one from [0, 1] to [0, 1] without matching it to any other one to generate a grid. By adjusting the constant c, we obtain the form

$$x_5(\xi, \epsilon) = \epsilon^\alpha \Big((1 + \epsilon^{-\alpha})^{\xi} - 1 \Big) \, .$$

The length of the interval $[0, \xi^p_1]$ transformed into the corresponding layer $[0, x^p_1]$ by the smooth function $x^p_{1,\,\mathrm{S}}(\xi, \epsilon)$ tends to $1/(1+pb)$ as the parameter ϵ tends to zero. For the function $x^p_{2,\,\mathrm{S}}(\xi, \epsilon)$, the length of the interval $[0, \xi^p_2]$ also tends to $1/(1 + pb)$ and, for the function $x^p_{4,\,\mathrm{S}}(\xi, \epsilon)$, the length of the interval $[0, \xi^p_4]$ tends to $1/(1 + p)$. Consequently, this part of the uniform grid on the interval [0, 1] is transformed into the corresponding layer. If there is a need for a larger proportion of the grid points to be distributed by smooth mappings into layers, the basic local contraction functions $x_i(\xi, \epsilon)$, $i = 1, 2, 4$, should be matched smoothly to polynomials. In this case, the point of matching can be chosen with less restriction and it will not be completely prescribed, unlike in the case of the functions $x^p_{i,\,\mathrm{S}}(\xi, \epsilon)$.

4.4.3 Other Univariate Transformations

Besides the univariate transformations described above, which depend on the parameter ϵ and are directly connected with the solutions to singularly perturbed equations, there are other monotonic functions, e.g. polynomials, hyperbolic functions, sines, and tangents, which are used as local contraction mappings to yield grid clustering in boundary and interior layers.

Eriksson Function

One such reference function was introduced by Eriksson (1982):

$$x_6(\xi) = \frac{\mathrm{e}^{d\xi} - 1}{\mathrm{e}^{d} - 1}\,, \qquad d > 0\,, \qquad 0 \le \xi \le 1\,. \tag{4.51}$$

This function provides a concentration of the grid towards the boundary $\xi = 0$.

There is a direct correspondence between $x_6(\xi)$ and the basic transformation $x_5(\xi, \epsilon)$. Namely, if d in (4.51) is equal to $\ln(1 + \epsilon^{-\alpha})$, then the Eriksson function (4.51) coincides with the contraction transformation $x_5(\xi, \epsilon)$, i.e.

$$x_6(\xi) = \epsilon^{\alpha}\Big((1 + \epsilon^{-\alpha})^{\xi} - 1\Big)\,.$$

This relation shows clearly how to adjust the grid spacing automatically to the physical small parameter ϵ by means of the Eriksson basic function.

Other functions, based on the inverse hyperbolic sines and tangents, were introduced by Vinokur (1983) to treat exponential singularities. Note that hyperbolic sines and tangents are defined through exponential functions and, therefore, in the case of narrow layers, are locally similar to the exponential function $\varphi_1(x, \epsilon)$.

Tangent Function

The basic function

$$y(\xi) = \tan \xi \tag{4.52}$$

is very popular for generating grid clustering. Using two parameters α and β, a monotonic function transforming the interval [0, 1] onto (0, 1] with an opportunity to control the contraction near the boundary $\xi = 0$ can be defined by

$$x(\xi) = d \tan(\beta\xi)\,. \tag{4.53}$$

The condition $x(1) = 1$ implies

$$d = 1/\tan \beta\,.$$

For the derivative of the function $x(\xi)$ with respect to ξ, we have

$$x'(\xi) = \frac{\beta}{\tan\beta\cos^2\beta\xi} \ .$$

In order to cope with any boundary layer quantity whose derivative with respect to x reaches values of $\epsilon^{-\alpha}$ at the point $x = 0$, the function $x'(\xi)$ must have a value of the order of ϵ^{α} at the point $\xi = 0$. This condition implies

$$\beta = \arctan\epsilon^{-\alpha}$$

and consequently

$$\begin{aligned} &d = \epsilon^{\alpha} \ , \\ &x'(0) = \epsilon^{\alpha}\arctan\epsilon^{-\alpha} \sim \frac{\pi}{2}\epsilon^{\alpha} \ , \qquad \text{for} \quad 0 < \epsilon \ll 1 \ . \end{aligned}$$

Thus, the required expression for the local contraction function (4.53) eliminating the boundary layer is

$$x(\xi) = \epsilon^{\alpha}\tan[\arctan(\epsilon^{-\alpha})\xi] \ . \tag{4.54}$$

The corresponding inverse local stretching function $\xi(x)$ has the form

$$\varphi(x) = \frac{\arctan(\epsilon^{-\alpha}x)}{\arctan(\epsilon^{-\alpha})} \ .$$

We have

$$\varphi'(x) = \frac{\epsilon^{\alpha}}{\arctan\epsilon^{-1}(\epsilon^{2\alpha} + x^2)}$$

and thus

$$M_1\frac{\epsilon^{\alpha}}{(\epsilon^{\alpha} + x)^2} \le |\xi'(x)| \le M_2\frac{\epsilon^{\alpha}}{(\epsilon^{\alpha} + x)^2} \ , \tag{4.55}$$

where $0 < \epsilon \le 1/2$, and the constants M_1 and M_2 are independent of the parameter ϵ. A comparison of the inequality (4.55) and the relations (4.40) shows that the local stretching function $\varphi(x)$ is qualitatively equivalent to the function $\varphi_2(x, \epsilon)$ described by (4.37) with $b = 1$. The function (4.54) is therefore suitable to cope with solutions that are close to step functions in layers of exponential and power types.

Procedure for the Construction of Local Contraction Functions

The features of the tangent function (4.52) and the procedure described give a clue as to how to build new functions which can generate local grid clustering near a boundary point. These functions are derived from some basic univariate mappings $y(\xi)$, satisfying, in analogy with $\tan\xi$, the following conditions:

(1) $y'(0) = 1$,
(2) $y(\xi)$ is a monotonical increasing function for $0 \le \xi < a$ for some $a > 0$,
(3) $y(\xi) \to \infty$ when $\xi \to a$.

If $y(\xi)$ is a function satisfying these properties, then, assuming in analogy with (4.53) and (4.54)

$$x(\xi) = \frac{y(\beta\xi)}{y(\beta)} = \epsilon^{\alpha} y[y^{-1}(\epsilon^{-\alpha})\xi] \ ,$$

where y^{-1} is the inverse of $y(\xi)$, we obtain a monotonic transformation of the interval [0, 1] onto [0, 1] with a contraction of the order of ϵ^{α} at the point $\xi = 0$.

Note that the transformations

$$\begin{aligned} x_1(\xi, \epsilon) &= -\frac{\epsilon^{\alpha}}{b} \ln(1 - c\xi) \\ x_2(\xi, \epsilon) &= \epsilon^{\alpha}[(1 - c\xi)^{(-1/b)} - 1] \end{aligned}$$

from (4.43) can be obtained in accordance with this scheme from the functions

$$y_1(\xi) = -\ln(1 - \xi)$$

and

$$y_2(\xi) = b[(1 - \xi)^{-1/b} - 1] \ ,$$

respectively.

The original function $y(\xi)$ can be formed as a ratio

$$y(\xi) = b_1(\xi)/b_2(\xi)$$

of two functions $b_1(\xi)$ and $b_2(\xi)$ which are strongly positive on the interval $[0, a]$. In addition, the function $b_1(\xi)$ must be monotonically increasing, while $b_2(\xi)$ is monotonically decreasing and satisfies the condition $b_2(a) = 0$. For example, the function

$$y(\xi) = \frac{\xi}{1 - \xi}$$

generates the local contraction transformation

$$x(\xi) = \frac{\epsilon^{\alpha}\xi}{1 - (1 - \epsilon^{\alpha})\xi} \ ,$$

which coincides with the transformation $x_2(\xi, \epsilon)$ from (4.43) for $b = 1, \ c = (1 - \epsilon^{\alpha})$.

4.4.4 Construction of Basic Intermediate Transformations

The basic functions $x_{i,\,\mathrm{c}}(\xi, \epsilon)$ and $x_{i,\,\mathrm{s}}(\xi, \epsilon)$ described above can be considered as construction elements for building intermediate transformations $\boldsymbol{q}(\boldsymbol{\xi}) : \Xi^n \rightarrow Q^n$

that serve to provide adequate grid clustering where necessary. Firstly, these basic functions can be used as separate transformations $q^i(\xi^i)$ of the coordinates ξ^i. The first derivatives of the basic mappings $x_{i,\,\mathrm{c}}(\xi,\epsilon)$ and $x_{i,\,\mathrm{s}}(\xi,\epsilon)$ are small near the point $\xi = 0$, and therefore the derived intermediate transformations produce grid clustering in the vicinity of the selected boundary surfaces $\xi^i = 0$.

Functions which provide grid clustering near arbitrary coordinate surfaces can be derived from these basic univariate mappings. For this purpose, it is sufficient to define monotonic scalar functions having a small first derivative near arbitrary boundary or interior points in the interval [0, 1]. Such mappings can be defined by simple procedures of scaling, shifting, and matching with the basic functions $x_{i,\,\mathrm{c}}(\xi,\epsilon)$ and $x_{i,\,\mathrm{s}}(\xi,\epsilon)$, as described below.

Functions with Boundary Contraction

For example, let $x(0,\xi)$ be one of these basic monotonically increasing functions varying from 0 to 1 and having a small value of the first derivative near the point $\xi = 0$, thus exhibiting a grid contraction near the point $x = 0$. Then, the mapping

$$x(1,\xi) = 1 - x(0, 1-\xi)$$

is also a monotonically increasing function transforming the interval [0, 1] onto itself and having the same small first derivative, but near the boundary point $x = 1$. Therefore, it performs a nodal concentration near the point $x = 1$.

Grid clustering near two boundary points 0 and 1 can be produced by the mapping $x(0,1,\xi)$ that is a composite of the two functions $x(0,\xi)$ and $x(1,\xi)$, say,

$$x(0,1,\xi) = x[0, x(1,\xi)]\ ,$$

or can be obtained by the following formula of scaling and matching of the functions $x(0,\xi)$ and $x(1,\xi)$:

$$x(0,1,\xi) = \begin{cases} x_0 x(0, \xi/x_0)\ , & 0 \le \xi \le x_0\ , \\ 1 - (1-x_0)\, x\left(0, \dfrac{1-\xi}{1-x_0}\right)\ , & x_0 \le \xi \le 1\ , \end{cases}$$

where x_0 is an interior matching point of the interval [0, 1].

Functions with Interior Contraction

Further, if x_0 is an inner point of the interval [0, 1], then the mapping

$$x(x_0,\xi) = \begin{cases} x_0[1 - x(0, 1-\xi/x_0)]\ , & 0 \le \xi \le x_0\ , \\ x_0 + (1-x_0)x\left(0, \dfrac{\xi - x_0}{1-x_0}\right)\ , & x_0 \le \xi \le 1\ , \end{cases}$$

is a monotonically increasing function from the interval [0, 1] onto the interval [0, 1] which provides grid clustering in the vicinity of the point $x = x_0$. The function $x(0, x_0, 1, \xi)$, defined as the composition of the two mappings $x(0, 1, \xi)$ and $x(x_0, \xi)$ introduced above, namely,

$$x(0, x_0, 1, \xi) = x[x_0, x(0, 1, \xi)],$$

provides a concentration of grid nodes in the vicinity of the boundary points $0,\ 1$ and of the interior point x_0.

A monotonically increasing function $x(x_0, x_1, \xi)$ performing grid clustering near two interior points x_0 and $x_1,\ x_0 < x_1$, can be defined as a composition of two functions of the type $x(x_0, \xi)$ or can be given by the formula

$$x(x_0, x_1, \xi) = \begin{cases} x_0[1 - x(0, 1 - \xi/x_0)]\ , & 0 \le \xi \le x_0\ , \\ x_0 + (d - x_0)\, x\left(0, \dfrac{\xi - x_0}{d - x_0}\right)\ , & x_0 \le \xi \le b\ , \\ x_1 + (d - x_1)\, x\left(0, \dfrac{x_1 - \xi}{x_1 - d}\right)\ , & b \le \xi \le x_1\ , \\ x_1 + (1 - x_1)\, x\left(0, \dfrac{\xi - x_1}{1 - x_1}\right)\ , & x_1 \le \xi \le 1\ , \end{cases}$$

where d is a specified number satisfying $x_0 < b < x_1$. The same procedures allow one to construct monotonically increasing functions providing grid clustering near an arbitrary number of points. If the original mapping $x(0, \xi)$ is smooth, then the functions derived by these procedures are smooth as well.

Analogously, there can be defined monotonically decreasing functions providing grid concentration near an arbitrary number of points using the basic decreasing transformations $1 - x(0, \xi)$. So, there is a broad range of possibilities in using the basic transformations to generate effective grid clustering.

Clustering Near Arbitrary Surfaces

The intermediate transformations $\boldsymbol{q}(\boldsymbol{\xi})$ constructed with the above approach through these modifications of the basic scalar functions provide grid clustering near the coordinate surfaces $\xi^i = \xi^i_l$. One drawback of such intermediate transformations is that they generate grid clustering only near these coordinate surfaces, with the same spacing in the vicinity of each of them. Therefore, some procedures are needed to construct intermediate functions with a broader range of possibilities.

In three-dimensional domains, for instance, there is often a need to define an intermediate transformation $\boldsymbol{q}(\boldsymbol{\xi})$ providing grid clustering near an arbitrary surface intersecting a coordinate direction, say ξ^3. Let the surface be prescribed by the function

$$\xi^3 = g(\xi^1, \xi^2)\ .$$

The required mapping $\boldsymbol{q}(\xi)$, providing grid concentration near this surface, is given by the formula

$$
\begin{aligned}
q^1(\boldsymbol{\xi}) &= \xi^1\ ,\\
q^2(\boldsymbol{\xi}) &= \xi^2\ ,\\
q^3(\boldsymbol{\xi}) &= \begin{cases} g(\xi^1,\xi^2)\{1-f[1-\xi^3/g(\xi^1,\xi^2)]\}, & 0\le \xi^3\le g(\xi^1,\xi^2),\\ g(\xi^1,\xi^2)+[1-g(\xi^1,\xi^2)]f\left(\dfrac{\xi^3-g(\xi^1,\xi^2)}{1-g(\xi^1,\xi^2)}\right), & g(\xi^1,\xi^2)\le\xi^3\le 1,\end{cases}
\end{aligned}
$$

where $f(\xi)$ is a basic monotonically increasing function with a small first derivative near the point $\xi = 0$. Compositions of such transformations produce maps that provide grid clustering near a number of surfaces intersecting different directions.

Nonuniform Clustering

The procedures described above provide adequate grid clustering in the vicinity of arbitrary surfaces, but with the same grid spacing around any one specified surface. However, in some cases, for example, when gridding a flow region around a body, there is a need for nonuniform grid clustering in the transverse direction with respect to different parts of the surface of the body. Such grid clustering can be realized by intermediate transformations constructed along the coordinate surface (line in two dimensions) through a combination of basic functions with different values of the parameter ϵ. For example, a two-dimensional intermediate transformation $\boldsymbol{q}(\boldsymbol{\xi})$ can be defined as

$$
\begin{aligned}
q^1(\xi^1,\xi^2) &= \xi^1\ ,\\
q^2(\xi^1,\xi^2) &= \xi^1\, x_1(\xi^2,\epsilon^k) + (1-\xi^1)\, x_2(\xi^2,\epsilon^d)\ ,
\end{aligned}
$$

where $x_i(\xi^2,\epsilon^m)$ is one of the basic functions. The mapping $\boldsymbol{q}(\boldsymbol{\xi})$ provides a nonuniform grid spacing along the coordinate $\xi^2 = \xi^2_0$ in the ξ^1 direction.

The procedures presented here can be applied to other mappings as well to construct intermediate transformations generating nonuniform grid clustering in the desired zones of the physical domain.

4.4.5 Multidirectional Equidistribution

The generation of grids in multidimensional domains commonly requires adaptation in several directions. One way to perform such adaptation is to extend the methods of univariate equidistribution. A generalization of the univariate equidistribution approach to generate grids in a multidimensional domain can be accomplished by using either a combination or a composition of univariate equidistributions along fixed families of coordinate lines which are specified or computed beforehand in the physical domain.

Combination of One-Dimensional Equidistributions

In the combination approach, the grid is derived by overlaying a series of grid lines obtained separately with univariate equidistributions in each direction (Fig. 4.6).

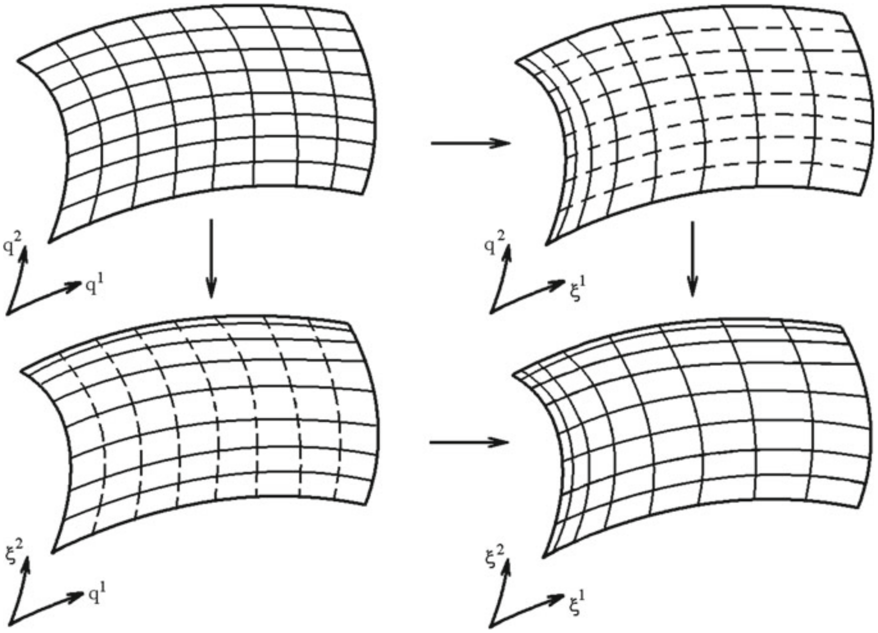


Fig. 4.6 Coordinate system derived by overlaying two independently deformed families (*solid curves*) of coordinate lines

Thus, let the physical region X^n have a coordinate system $q^1, \dots, q^n$ along which a univariate adaptation is supposed to be performed. Let the coordinate system q^i be determined by a coordinate transformation

$$\boldsymbol{x}(\boldsymbol{q}) : Q^n \to X^n \ .$$

Here, the parametric (intermediate) domain Q^n can be considered as the unit cube. The equidistribution of the grid points along any fixed family of coordinate lines is carried out by the formulas of univariate equidistribution. This process proceeds separately for each fixed coordinate family q^i by determining the respective weight function $w_i(\boldsymbol{q})$ and finding the coordinate transformation $q^i(\xi^i)$ for each fixed value $(q^1, \dots, q^{i-1}, q^{i+1}, q^n)$ through the following equation, of the form (4.23):

$$\begin{aligned}&\frac{\partial}{\partial \xi^i}\left(\frac{\partial q^i}{\partial \xi^i} w_i(\boldsymbol{q})\right) = 0 \ , \qquad 0 < \xi^i < 1 \ , \qquad i \quad \text{fixed} \ , \\ &q^i(0) = 0 \ , \qquad\qquad q^i(1) = 1 \ ,\end{aligned} \tag{4.56}$$

and then forming the final grid by overlaying the coordinate curves obtained. In fact, the grid is built by solving the system (4.56) for $i = 1, \dots, n$. Of course,

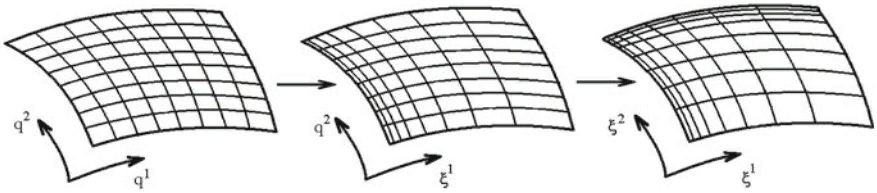


Fig. 4.7 Coordinate system derived by the composition of two successive deformations

unlike the one-dimensional case, the solution to this system may not be a one-to-one transformation.

Composition of Univariate Equidistributions

In the composition approach, the process is split into a sequence of one-directional adaptations, in which the grid is obtained through successive application of one-dimensional equidistribution techniques (Fig. 4.7). In two dimensions, this succession can be represented as

$$\boldsymbol{x}(q^1, q^2) \to \boldsymbol{x}[q^1(\xi^1, q^2), q^2] \to \boldsymbol{x}\{q^1[\xi^1, q^2(\xi^1, \xi^2)], q^2(\xi^1, \xi^2)\} .$$

This approach produces one-to-one coordinate transformations.

4.5 Comments

Reviews of adaptive methods for the generation of structured grids have been published by Anderson (1983), Thompson (1985), Eiseman (1987), Hawken et al. (1991), and Liseikin (1996).

Equidistribution approaches of various kinds have been reported by a number of researchers. The original one-dimensional integral formulation of the equidistribution principle was proposed by Boor (1974), while the differential versions were presented by Danaev (1979), Yanenko et al. (1977), Tolstykh (1978), and Dwyer et al. (1980).

The most popular and general forms of the weight functions were reviewed by Thompson (1985). These functions were proposed by Russell and Christiansen (1978), Ablow and Schechter (1978), White (1979), Dwyer et al. (1980), Nakamura (1983), and Anderson and Steinbrenner (1986). Acharya and Moukalled (1990) used a normalized second derivative of the solution as the weighting function. A linear combination of the first and second derivatives of the solution was used as a measure of the weighting function by Dwyer et al. (1980), who successfully applied equidistribution along one family of grid lines within two-dimensional problems. A combination of first and second derivatives and the curvature of the solution variables to specify weight functions was applied by Ablow and Schechter (1978) and Noack and Anderson (1990). A selection of monitor functions in the form of weighted Boolean sums of various solution characteristics was defined by Weatherill and Soni (1991) and Soni and Yang (1992).

In a one-dimensional application, Gnoffo (1983) used a tension spring analogy in which the adapted grid point spacing along a family of coordinate lines resulted from the minimization of the spring system's potential energy. The idea of the spring analogy, represented in the equation for the weighting function, was extended by Nakahashi and Deiwert (1985) to demonstrate the feasibility and versatility of the equidistribution method. They also utilized in their considerations the notion of a torsion spring attached to each node in order to control the inclination of the grid lines.

Rai and Anderson (1981, 1982) and Eiseman (1985) have each used the idea of moving grid points under the influence of forcing or weighting functions that either attract or repel grid points relative to each other. Thus, points with forcing (or weighting) functions greater than a specified average value attract each other, and those with values less than the average value repel each other.

Examples of the numerical solution of singularly perturbed problems on an equidistant mesh were studied by Andrew and Whrite (1979). The equidistant method was also advocated by Ablow (1982) and Catheral (1991), who applied it to some gas-dynamic calculations.

Dorfi and Drury (1987) used a very effective technique for incorporating smoothness into the univariate equidistribution principle. Their one-dimensional technique ensures that the ratio of adjacent grid intervals is restricted, thus controlling clustering and grid expansion. The power of this smoothing capability was clearly demonstrated in the valuable comparative studies by Furzland et al. (1990) and Zegeling (1993). A multidimensional generalization of the approach of Dorfi and Drury (1987) was presented by Huang and Sloan (1994), who introduced control of concentration, scaling, and smoothness.

Morrison (1962) was apparently the first who managed to show analytically the efficiency of the error equidistribution principle for the generation of grids for the numerical solution of ordinary differential equations. Babuŝka and Rheinboldt (1978) proposed an error estimator based on the solution of a local variational problem. A truncation error measure for generating optimal grids was applied by Denny and Landis (1972), Liseikin and Yanenko (1977), White (1979, 1982), Ablow and Schechter (1978), Miller (1981), Miller and Miller (1981), Davis and Flaherty (1982), Adjerid and Flaherty (1986), and Petzold (1987). The approaches based on the equidistribution of the truncation error were developed by Pereyra and Sewell (1975) and Davis and Flaherty (1982), while the equidistribution of the residual was developed by Carey (1979), Pierson and Kutler (1980), and Rheinboldt (1981). An analysis of the strategies based on a uniform error distribution was also undertaken by Chen (1994). However, the numerical experiments by Blom and Verwer (1989) show that the mesh generated from the error measurement monitor may be of poor quality.

Some methods which control the movement of the grid nodes in accordance with the equidistribution of the residuals of equations were developed by Miller (1981). Several versions of the moving-mesh method were also studied by

Huang et al. (1994) to demonstrate their ability to accurately tracking rapid spatial and temporal transitions.

A curve-by-curve grid line equidistribution approach in the computational space was described by Eiseman (1987). The combination and composition versions of the multidimensional equidistribution principle were proposed by Darmaev and Liseikin (1987).

An orthogonalization technique of Potter and Tuttle (1973) for two-di-mensional grid control, whereby the grids are adapted in one direction and orthogonality is imposed on the second, was proposed by Anderson and Rajendran (1984) and Dwyer and Onyejekwe (1985).

Certain anomalies which may arise in the process of the numerical solution of the nonlinear equations modeling the equidistribution principle and ways for surmounting them were discussed by Steinberg and Roache (1990) and Knupp (1991, 1992). Also, some adverse effects of dynamic grid adaptation on the numerical solution of physical problems were noted by Sweby and Yee (1990).

The method of equidistribution and minimization of the heuristically determined error at each time step was used for calculations of nonstationary problems by Dorfi and Drury (1987), Dwyer et al. (1980), Klopfer and McRae (1981), Miller (1983), Wathen (1990), and White (1982).

There are three basic approaches to treating problems with boundary and interior layers. The classical approach relies on expansion of the solution in a series of singular and slowly changing functions. The second technique applies special approximations of equations. The third one is based on the implementation of local stretching functions to stretch the coordinates and, correspondingly, provide clustered grids.

The approach using stretching functions appears to be more effective in comparison with the other techniques because it requires only rough information about the qualitative properties of the solution and enables one to interpolate the solution uniformly over the entire physical region. The application of interactive procedures using the basic intermediate transformations allows one to generate efficient grids in arbitrary zones even without preliminary information about the qualitative features of the solution.

Estimates of the derivatives of the solution to the problem of the type (4.10) with exponential layers were obtained by Brish (1954). Investigation of the qualitative properties of the solution to the linear problem (4.1) in interior layers was carried out by Berger et al. (1984).

The asymptotic location of the interior layers of the solution to the problem (4.6) was found by Lorenz (1982, 1984). The asymptotic expansion of the linear version of the problem (4.7) for $a(0) > 1$ was considered by Lomov (1964). A qualitative investigation of the solutions to the problems (4.1), (4.6), (4.7) in arbitrary boundary and interior layers was carried out by Liseikin (1984, 1986, 1993). In these papers, estimates of the derivatives of the solution were obtained. A detailed description of the estimates of derivatives of the solutions of singularly perturbed equations is presented in the monograph by Liseikin and Petrenko (1989) and Liseikin (2001).

The logarithmic transformation $x_1(\xi, \epsilon)$ in (4.43) with $\alpha = 1/2$ was introduced by Bahvalov (1969) for the generation of clustered grids in the vicinity of exponential boundary layers of singularly perturbed equations. The mappings $x_i(\xi, \epsilon)$, $i = 2, 3, 4$, were proposed by Liseikin (1984, 1986) for the construction of nonuniform clustered grids within power and combined boundary and interior layers. A particular shape of the basic contraction mapping $x_2(\xi, \epsilon)$ for $b = 1$, having the form

$$x_2(\xi, \epsilon) = \epsilon^{\alpha} \frac{c\xi}{1 - c\xi} ,$$

was also proposed independently by Vulanovic (1984) to generate grids within some exponential layers for boundary layer functions of the type described by (4.36).

Stretching functions based on inverse hyperbolic sines were employed by Thomas et al. (1972) in a numerical solution of inviscid supersonic flow.

A two-sided stretching function of the logarithmic type (4.39) was introduced by Roberts (1971) to study boundary layer flows.

A family of tangent mappings of the form

$$x = x_0 + \alpha \tan[(s - s_0)\beta\xi] ,$$

suitable primarily for internal layers, was introduced by Vinokur (1983) to generate grids. These mappings were also employed by Bayliss and Garbey (1995) as part of the adaptive pseudospectral method.

Physical quantities were used as new coordinates for stretching boundary and interior layers by Tolstykh (1973).

References

Ablow, C. M. (1982). Equidistant mesh for gas dynamics calculations. *Applied Mathematics and Computation*, *10*(11), 859–863.

Ablow, C. M., & Schechter, S. (1978). Campylotropic coordinates. *Journal of Computational Physics*, *27*, 351–363.

Acharya, S., & Moukalled, F. H. (1990). An adaptive grid solution procedure for convection-diffusion problems. *Journal of computational physics*, *91*, 32–54.

Adjerid, S., & Flaherty, J. F. (1986). A moving finite element method with error estimation and refinement for one-dimensional time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, *23*, 778–796.

Anderson, D. A. (1983). Adaptive grid methods for partial differential equations. In K. N. Ghia & U. Ghia (Eds.), *Advances in grid generation* (pp. 1–15). Houston: ASME.

Anderson, D. A., & Rajendran, N. (1984). Two approaches toward generating orthogonal adaptive grids. AIAA Paper 84-1610.

Anderson, D. A., & Steinbrenner, J. (1986). Generating adaptive grids with a conventional grid scheme. AIAA Paper 86-0427.

Andrew, B., & Whrite, J. R. (1979). On selection of equidistributing meshes for two-point boundary-value problems. *SIAM Journal on Numerical Analysis*, *16*(3), 472–502.

Babuska, I., & Rheinboldt, W. C. (1978). A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, *12*, 1597–1615.

Bahvalov, N. S. (1969). On optimization of the methods of the numerical solution of boundary-value problems with boundary layers. *Journal of Computational Mathematics and Mathematical Physics*, *9*(4), 842–859 (Russian) [English transl.: *USSR Computational Mathematics and Mathematical Physics*, *9* (1969)].

Bayliss, A., & Garbey, M. (1995). Adaptive pseudo-spectral domain decomposition and the approximation of multiple layers. *Journal of Computational Physics*, *119*(1), 132–141.

Berger, A. E., Han, H., & Kellog, R. B. (1984). A priori estimates of a numerical method for a turning point problem. *Mathematics of Computation*, *42*(166), 465–492.

Blom, J. G., & Verwer, J. G. (1989). On the use of arclength and curvature in a moving-grid method which is based on the methods of lines. Report NM-N8402, CWI, Amsterdam.

Boor, C. (1974). Good approximation by splines with variable knots. *Lecture Notes in Mathematics*, *363*, 12–20.

Brish, N. J. (1954). On boundary-value problems for the equation $\epsilon y'' = f(x, y, y')$ with small ϵ. *Doklady Akademii Nauk USSR*. *XCV*(3), 429–432 (Russian).

Carey, G. F. (1979). Adaptive refinement and nonlinear fluid problems. *Computer Methods in Applied Mechanics and Engineering*, *17/18*, 541–560.

Catheral, D. (1991). The adaption of structured grids to numerical solutions for transonic flow. *International Journal for Numerical Methods in Engineering*, *32*, 921–937.

Chen, K. (1994). Error equidistribution and mesh adaptation. *SIAM Journal on Scientific Computing*, *15*(4), 798–818.

Danaev, N. T. (1979). One approach for generating grids with node clustering in the zones of large gradients. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *10*(4), 60–74 (Russian).

Darmaev, T. G., & Liseikin, V. D. (1987). A method for generating multidimensional adaptive grids. *Modelirovanie v mekhan. Novosibirsk.*, *1*(1), 49–57 (Russian).

Davis, S. F., & Flaherty, J. E. (1982). An adaptive finite element method for initial boundary-value problems for partial differential equations. *SIAM Journal on Scientific and Statistical Computing*, *3*, 6–27.

Denny, V. E., & Landis, R. B. (1972). A new method for solving two-point boundary-value problems using optimal node distribution. *Journal of Computational Physics*, *9*(1), 120–137.

Dorfi, E. A., & Drury, L. O'C. (1987). Simple adaptive grids for 1-D initial value problems. *Journal of Computational Physics*, *69*, 175–195.

Dwyer, H. A., & Onyejekwe, O. O. (1985). Generation of fully adaptive and/or orthogonal grids. In *Proceedings of the 9th International Conference on Numerical Methods in Fluid Dynamics* (pp. 422–426). Saclay.

Dwyer, H. A., Kee, R. J., & Sanders, B. R. (1980). Adaptive grid method for problems in fluid mechanics and heat transfer. *AIAA Journal*, *18*(10), 1205–1212.

Eiseman, P. R. (1985). Grid generation for fluid mechanics computations. *Annual Review of Fluid Mechanics*, *17*, 487–522.

Eiseman, P. R. (1987). Adaptive grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 321–376.

Eriksson, L. E. (1982). Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation. *AIAA Journal*, *20*, 1313–1320.

Furzland, R. M., Verwer, J. G., & Zegeling, P. A. (1990). A numerical study of three moving grid methods for 1-D PDEs which are based on the method of lines. *Journal of Computational Physics*, *89*, 349–388.

Gnoffo, P. A. (1983). A finite-volume, adaptive grid algorithm applied to planetary entry flowfields. *AIAA Journal*, *21*(9), 1249–1254.

Hawken, D. F., Gottlieb, J. J., & Hansen, J. S. (1991). Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. *Journal of Computational Physics*, *95*, 254–302.

Huang, W., & Sloan, D. M. (1994). A simple adaptive grid method in two dimensions. *SIAM Journal on Scientific Computing*, *15*(4), 776–797.

Huang, W., Ren, Y., & Russel, R. D. (1994). Moving mesh PDEs based on the equidistribution principle. *SIAM Journal on Numerical Analysis*, *31*, 709–730.

Klopfer, G. H., & McRae, D. D. (1981). The nonlinear modified equation approach to analysing finite difference schemes. AIAA Paper 81-1029.

Knupp, P. M. (1991). The direct variational grid generation method extended to curves. *Applied Mathematics and Computation*, *43*, 65–78.

Knupp, P. M. (1992). A robust elliptic grid generator. *Journal of Computational Physics*, *100*, 409–418.

Liseikin, V. D. (1984). On the numerical solution of singularly perturbed equations with a turning point. *Journal of Computational Mathematics and Mathematical Physics*, *24*(12), 1812–1818 (Russian) [English transl.: *USSR Computational Mathematics and Mathematical Physics*, *24* (1984)]

Liseikin, V. D. (1986). On the numerical solution of equations with a power boundary layer. *Journal of Computational Mathematics and Mathematical Physics*, *26*(12), 1813–1820 (Russian).

Liseikin, V. D. (1993). Estimates for derivatives of solutions to differential equations with boundary and interior layers. *Siberian Mathematical Journal*, *July*, 1039–1051.

Liseikin, V. D. (1996). Construction of structured adaptive grids - a review. *Computational Mathematics and Mathematical Physics*, *36*(1), 1–32.

Liseikin, V. D. (2001). *Layer resolving grids and transformations for singular perturbation problems*. Utrecht: VSP.

Liseikin, V. D., & Petrenko, V. E. (1989). Adaptive invariant method for the numerical solution of problems with boundary and interior layers. Computer Center SD AS USSR, Novosibirsk (Russian).

Liseikin, V. D., & Yanenko, N. N. (1977). Selection of optimal numerical grids. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *8*(7), 100–104 (Russian).

Lomov, S. A. (1964). Power boundary layer in problems with a small parameter. *Doklady Akademii Nauk USSR*, *184*(3), 516–519 (Russian).

Lorenz, J. (1982). Nonlinear boundary-value problems with turning points and properties of difference schemes. *Lecture Notes in Mathematics*, *942*, 150–169.

Lorenz, J. (1984). Analysis of difference schemes for a stationary shock problems. *SIAM Journal on Numerical Analysis*, *21*(6), 1038–1053.

Miller, K. (1981). Moving finite elements II. *SIAM Journal on Numerical Analysis*, *18*, 1033–1057.

Miller, K. (1983). Alternate codes to control the nodes in the moving finite element method. *Adaptive Computing Methods for Partial Differential Equations* (pp. 165–182). Philadelphia: SIAM.

Miller, K., & Miller, R. N. (1981). Moving finite elements I. *SIAM Journal on Numerical Analysis*, *18*, 1019–1032.

Morrison, D. (1962). Optimal mesh size in the numerical integration of an ordinary differential equation. *Journal of the Association for Computing Machinery*, *9*, 98–103.

Nagumo, M. (1937). Uber die Differentialgleichung $y'' = f(x, y, y')$. *Proceedings of the Physico-Mathematical Society of Japan*, *19*, 861–866.

Nakahashi, K., & Deiwert, G. S. (1985). A three-dimensional adaptive grid method. AIAA Paper 85-0486.

Nakamura, S. (1983). Adaptive grid relocation algorithm for transonic full potential calculations using one-dimensional or two-dimensional diffusion equations. In K. N. Ghia & U. Ghia (Eds.), *Advances in grid generation* (pp. 49–58). Houston: ASME.

Noack, R. W., & Anderson, D. A. (1990). Solution adaptive grid generation using parabolic partial differential equations. *AIAA Journal*, *28*(6), 1016–1023.

Pereyra, V., & Sewell, E. G. (1975). Mesh selection for discrete solution of boundary-value problems in ordinary differential equations. *Numerische Mathematik*, *23*, 261–268.

Petzold, L. R. (1987). Observations on an adaptive moving grid method for one-dimensional systems of partial differential equations. *Applied Numerical Mathematics*, *3*, 347–360.

Pierson, B. L., & Kutler, P. (1980). Optimal nodal point distribution for improved accuracy in computational fluid dynamics. *AIAA Journal*, *18*, 49–53.

Polubarinova-Kochina, P. Ya. (1977). *Theory of motion of phreatic water*. Moscow (Russian): Nauka.

Potter, D. E., & Tuttle, G. H. (1973). The construction of discrete orthogonal coordinates. *Journal of Computational Physics*, *13*, 483–501.

Rai, M. M., & Anderson, D. A. (1981). Grid evolution in time asymptotic problems. *Journal of Computational Physics*, *43*, 327–344.

Rai, M. M., & Anderson, D. A. (1982). Application of adaptive grids to fluid flow problems with asymptotic solution. *AIAA Journal*, *20*, 496–502.

Rheinboldt, W. C. (1981). Adaptive mesh refinement process for finite element solutions. *International Journal for Numerical Methods in Engineering*, *17*, 649–662.

Roberts, G. O. (1971). In *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics*. Berlin, Heidelberg: Springer.

Russell, R. D., & Christiansen, J. (1978). Adaptive mesh selection strategies for solving boundary-value problems. *SIAM Journal on Numerical Analysis*, *15*, 59–80.

Soni, B. K., & Yang, J.-C. (1992). General purpose adaptive grid generation system. AIAA Paper 92-0664.

Steinberg, S., & Roache, P. J. (1990). Anomalies in grid generation on curves. *Journal of Computational Physics*, *91*, 255–277.

Sweby, P. K., & Yee, H. C. (1990). On the dynamics of some grid adaptation schemes. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulation and Related Fields* (p. 467). Swansea: Pineridge.

Thomas, P. D., Vinokur, M., Bastianon, R. A., & Conti, R. J. (1972). Numerical solution for three-dimensional inviscid supersonic flow. *AIAA Journal*, *10*(7), 887–894.

Thompson, J. F. (1985). A survey of dynamically-adaptive grids in the numerical solution of partial differential equations. *Applied Numerical Mathematics*, *1*, 3–27.

Tolstykh, A. I. (1973). A method for the numerical solution of Navier-Stokes equations of a compressible gas flow for a wide range of Reynolds number. *Doklady Akademii Nauk SSSR*, *210*, 48–51 (Russian).

Tolstykh, A. I. (1978). Concentration of grid points in the process of solving and using high-accuracy schemes for the numerical investigation of viscous gas flows. *Computational Mathematics and Mathematical Physics*, *18*, 139–153 (Russian) [English transl.: *USSR Computational Mathematics and Mathematical Physics*, *18*, 134–138 (1979)].

Vinokur, M. (1983). On one-dimensional stretching functions for finite-difference calculations. *Journal of Computational Physics*, *50*, 215–234.

Vulanovic, R. (1984). Mesh construction for numerical solution of a type of singular perturbation problems. *Numerical Methods and Approximation Theory*, 137–142.

Wathen, A. J. (1990). Optimal moving grids for time-dependent partial differential equations. *Journal of Computational Physics*, *101*, 51–54.

Weatherill, N. P., & Soni, B. K. (1991). Grid adaptation and refinement in structured and unstructured algorithms. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical grid generation in computational fluid dynamics and related fields* (pp. 143–158). New York: North-Holland.

White, A. B. (1979). On selection of equidistributing meshes for two-point boundary-value problems. *SIAM Journal on Numerical Analysis*, *16*(3), 472–502.

White, A. B. (1982). On the numerical solution of initial boundary-value problems in one space dimension. *SIAM Journal on Numerical Analysis*, *19*, 683–697.

Yanenko, N. N., Danaev, N. T., & Liseikin, V. D. (1977). A variational method for grid generation. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *8*(4), 157–163 (Russian).

Zamaraev, K. K., Khairutdinov, P. F., & Zhdanov, V. P. (1985). *Electron tunneling in chemistry. Chemical reactions at large distances*. Novosibirsk (Russian): Nauka.

Zegeling, P. A. (1993). *Moving-grid methods for time-dependent partial differential equations*. CWI Tract 94, Amsterdam: Centrum voor Wiskund en Informatica.

Chapter 5
Algebraic Grid Generation

5.1 Introduction

The algebraic grid generation approach relies chiefly on an explicit construction of coordinate transformations through the formulas of transfinite interpolation. Of central importance in the method are blending functions (univariate quantities, each depending on one chosen coordinate only). These provide matching of the grid distribution on, and grid directions from, the boundaries and specified interior surfaces of an arbitrary domain. Direct control of the essential properties of the coordinate transformations in the vicinity of the boundaries and interior surfaces is carried out by the specification of the out-of-surface-direction derivatives and blending functions.

The purpose of this chapter is to describe common techniques of algebraic grid generation.

Nearly all of the formulas of transfinite interpolation include both repeated indices over which a summation is carried out and one repeated index, usually i, that is fixed. Therefore, in this chapter, we do not use the convention of summation of repeated indices, but instead use the common notation $\sum$ to indicate summation.

5.2 Transfinite Interpolation

This section describes some general three-dimensional formulas of transfinite interpolation which are used to define algebraic coordinate transformations from a standard three-dimensional cube Ξ^3 with Cartesian coordinates ξ^i, $i = 1, 2, 3$, onto a physical domain X^3 with Cartesian coordinates x^i, $i = 1, 2, 3$. The formulation of the three–dimensional interpolation is based on a particular operation of Boolean summation over unidirectional interpolations. So, first, the general formulas of unidirectional interpolation are reviewed.

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_5

5.2.1 *Unidirectional Interpolation*

General Formulas

For the unit cube Ξ^3, let there be chosen one coordinate direction ξ^i and some sections of the cube orthogonal to this direction, defined by the planes $\xi^i = \xi^i_l$, $l = 1, \dots, L_i$. Furthermore, on each section $\xi^i = \xi^i_l$, let there be given the values of some vector-valued function $\boldsymbol{r}(\boldsymbol{\xi})$, $\boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)$, and of its derivatives with respect to ξ^i up to order P^i_l. Then, the unidirectional interpolation of the function $\boldsymbol{r}(\boldsymbol{\xi})$ is a vector-valued function $\boldsymbol{P}^i[\boldsymbol{r}](\boldsymbol{\xi})$ from Ξ^3 into R^3 defined by the formula

$$\boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi}) = \sum_{l=1}^{L^i} \sum_{n=0}^{P^i_l} \alpha^i_{l,n}(\xi^i) \frac{\partial^n}{(\partial \xi^i)^n} \boldsymbol{r}(\boldsymbol{\xi}|_{\xi^i = \xi^i_l}) \ . \tag{5.1}$$

Here, the smooth scalar functions $\alpha^i_{l,n}(\xi^i)$, depending on one independent variable ξ^i, are subject to the following restrictions:

$$\frac{\mathrm{d}^m}{(\mathrm{d}\xi^i)^m} \alpha^i_{l,n}(\xi^i_k) = \delta^l_k \delta^n_m \ , \qquad l, k = 1, \dots, L^i \ , \qquad m, n = 0, 1, \dots P^i_l \ , \tag{5.2}$$

where δ^j_i is the Kronecker delta function, i.e. $\delta^j_i = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$

The expression $(\boldsymbol{\xi}|_{\xi^i = \xi^i_l})$ in (5.1) designates a point that is a projection of $\boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)$ on the section $\xi^i = \xi^i_l$, i.e. the ith coordinate ξ^i of $\boldsymbol{\xi}$ is fixed and equal to ξ^i_l; for example,

$$(\boldsymbol{\xi}|_{\xi^1 = \xi^1_l}) = (\xi^1_l, \xi^2, \xi^3) \ .$$

It is also assumed in (5.1) and below that the operator for the zero-order derivative is the identity operator, i.e.

$$\frac{\partial^0}{(\partial \xi^i)^0} f(\boldsymbol{\xi}) = f(\boldsymbol{\xi}) \ , \qquad \frac{\mathrm{d}^0}{(\mathrm{d}\xi^i)^0} g(\xi^i) = g(\xi^i) \ .$$

The coefficients $\alpha^i_{l,n}(\xi^i)$ in (5.1) are referred to as the blending functions. They serve to propagate the values of the vector-valued function $\boldsymbol{r}(\boldsymbol{\xi})$ from the specified sections of the cube $\Xi^{\mathbf{3}}$ into its interior. It is easily shown that the conditions (5.2) imposed on the blending functions $\alpha^i_{l,n}(\xi^i)$ provide matching at the sections $\xi^i = \xi^i_l$ of the values of the function $\boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi})$ and $\boldsymbol{r}(\boldsymbol{\xi})$, as well as the values of their derivatives with respect to ξ^i, namely,

$$\frac{\partial^n \boldsymbol{P}_i[\boldsymbol{r}]}{(\partial \boldsymbol{\xi}^i)^n}(\boldsymbol{\xi}|_{\xi^i = \xi^i_l}) = \frac{\partial^n \boldsymbol{r}}{(\partial \xi^i)^n}(\boldsymbol{\xi}|_{\xi^i = \xi^i_l}) \ , \qquad n = 0, \dots, P^i_l \ .$$

Two-Boundary Interpolation

A very important interpolation for grid generation applications is the one which matches the values of the vector-valued function $\boldsymbol{r}(\boldsymbol{\xi})$ and of its derivatives exclusively at the boundary planes of the cube Ξ^3. In this case, $L^i = 2$, $\xi^i_1 = 0$, and $\xi^i_2 = 1$, and the relations (5.1) and (5.2) have the form

$$\begin{aligned}\boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi}) = &\sum_{n=0}^{P^i_1} \alpha^1_{1,n}(\xi^i)\frac{\partial^n}{(\partial\xi^i)^n}\boldsymbol{r}(\boldsymbol{\xi}|_{\xi^i=0}) \\ &+\sum_{n=0}^{P^i_2} \alpha^i_{2,n}(\xi^i)\frac{\partial^n}{(\partial\xi^i)^n}\boldsymbol{r}(\boldsymbol{\xi}|_{\xi^i=1})\ ,\end{aligned} \tag{5.3}$$

$$\frac{\mathrm{d}^m}{(\mathrm{d}\xi^i)^m}\alpha^i_{l,n}(\xi^i_k) = \delta^l_k\delta^n_m\ , \qquad l,k = 1,2\ , \qquad m,n = 0,1,\dots P^i_l\ . \tag{5.4}$$

The interpolation described by (5.3) is referred to as the two-boundary interpolation.

5.2.2 *Tensor Product*

The composition of two unidirectional interpolations $\boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi})$ and $\boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi})$ of $\boldsymbol{r}(\boldsymbol{\xi})$ in the directions ξ^i and ξ^j, respectively, is called their tensor product. This operation is denoted by $\boldsymbol{P}_i[r] \otimes \boldsymbol{P}_j[r](\xi)$ and, in accordance with (5.1), we obtain

$$\begin{aligned}\boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) &= \boldsymbol{P}_i\big[\boldsymbol{P}_j[\boldsymbol{r}]\big](\boldsymbol{\xi}) \\ &= \sum_{l=1}^{L^i}\sum_{n=0}^{P^i_l}\alpha^i_{l,n}(\xi^i)\frac{\partial^n \boldsymbol{P}_j[\boldsymbol{r}]}{(\partial\xi^i)^n}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) \\ &= \sum_{k=1}^{L^j}\sum_{m=0}^{P^j_k}\sum_{l=1}^{L^i}\sum_{n=0}^{P^i_l}\alpha^i_{l,n}(\xi^i)\alpha^j_{k,m}(\xi^j)\frac{\partial^{n+m}\boldsymbol{r}}{(\partial\xi^i)^n(\partial\xi^j)^m}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k})\ .\end{aligned} \tag{5.5}$$

Here, by the notation $(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k})$, we mean the point which is the projection of $\boldsymbol{\xi}$ on the intersection of the planes $\xi^i = \xi^i_l$ and $\xi^j = \xi^j_k$, e.g.

$$(\boldsymbol{\xi}|_{\xi^1=\xi^1_l,\xi^3=\xi^3_k}) = (\xi^1_l, \xi^2, \xi^3_k)\ .$$

Equation (5.5) shows clearly that the tensor product is a commutative operation, i.e.

$$\boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}] = \boldsymbol{P}_j[\boldsymbol{r}] \otimes \boldsymbol{P}_i[\boldsymbol{r}]\ .$$

Using the relations (5.1), (5.2), and (5.5), we obtain

$$\begin{aligned}
&\frac{\partial}{\partial \xi^i} \boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}|_{\xi^i=\xi^i_s, \xi^j=\xi^j_t}) \\
&= \sum_{m=1}^{L^j} \sum_{k=0}^{P^j_k} \sum_{l=1}^{L^i} \sum_{p=0}^{P^i_l} \frac{\mathrm{d}}{\mathrm{d}_{\xi^i}} \alpha^i_{m,k}(\xi^i_s) \alpha^j_{l,p}(\xi^j_t) \frac{\partial^{k+p} \boldsymbol{r}}{(\partial \xi^i)^k (\partial \xi^j)^p} (\boldsymbol{\xi}|_{\xi^i=\xi^i_m, \xi^j=\xi^j_l}) \\
&= \frac{\partial \boldsymbol{r}}{\partial \xi^i} (\boldsymbol{\xi}|_{\xi^i=\xi^i_s, \xi^j=\xi^j_t}) \ .
\end{aligned}$$

Analogously,

$$\frac{\partial^{k+p}}{(\partial \xi^i)^k (\partial \xi^j)^p} (\boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}])(\boldsymbol{\xi}|_{\xi^i=\xi^i_s, \xi^j=\xi^j_t}) = \frac{\partial^{k+p}}{(\partial \xi^i)^k (\partial \xi^j)^p} \boldsymbol{r}(\boldsymbol{\xi}|_{\xi^i=\xi^i_s, \xi^j=\xi^j_t}) \ .$$

Thus, the derivatives of the tensor product $\boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}]$ with respect to ξ^i and ξ^j match the derivatives of the function $\boldsymbol{r}(\boldsymbol{\xi})$ at the intersections of the planes $\xi^i = \xi^i_s$ and $\xi^j = \xi^j_t$.

5.2.3 Boolean Summation

Bidirectional Interpolation

The bidirectional interpolation matching the values of the function $\boldsymbol{r}(\boldsymbol{\xi})$ and of its derivatives at the sections in the directions ξ^i and ξ^j is defined through the Boolean summation $\oplus$:

$$\boldsymbol{P}_i[\boldsymbol{r}] \oplus \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) = \boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi}) + \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) - \boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) \ . \tag{5.6}$$

Using (5.1) and (5.5), we obtain

$$\begin{aligned}
\boldsymbol{P}_i[\boldsymbol{r}] \oplus \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) &= \sum_{l=1}^{L^i} \sum_{n=0}^{P^i_l} \alpha^i_{l,n}(\xi^i) \frac{\partial^n \boldsymbol{r}}{(\partial \xi^i)^n} (\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) \\
&\quad + \sum_{k=1}^{L^j} \sum_{m=0}^{P^j_k} \alpha^j_{k,m}(\xi^i) \frac{\partial^m \boldsymbol{r}}{(\partial \xi^j)^m} (\boldsymbol{\xi}|_{\xi^j=\xi^j_k})
\end{aligned}$$

$$- \sum_{k=1}^{L^j} \sum_{m=0}^{P^j_k} \sum_{l=1}^{L^i} \sum_{n=0}^{P^i_l} \alpha^i_{l,n}(\xi^i) \alpha^j_{k,m}(\xi^j) \frac{\partial^{n+m} \boldsymbol{r}}{(\partial \xi^i)^n (\partial \xi^j)^m} (\boldsymbol{\xi}|_{\xi^i=\xi^i_l, \xi^j=\xi^j_k}) \ . \tag{5.7}$$

Taking into account the relation

$$\boldsymbol{P}_j[\boldsymbol{r}] - \boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}] = \boldsymbol{P}_j\big[\boldsymbol{r} - \boldsymbol{P}_i[\boldsymbol{r}]\big] \tag{5.8}$$

we obtain the result that the formulas (5.6) and (5.7) for the Boolean summation can be written as the ordinary sum of two unidirectional interpolants $\boldsymbol{P}_i[\boldsymbol{r}]$ and $\boldsymbol{P}_j\big[\boldsymbol{r} - \boldsymbol{P}_i[\boldsymbol{r}]\big]$. Thus, using (5.1), we obtain

$$\boldsymbol{P}_i[\boldsymbol{r}] \oplus \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) = \boldsymbol{P}_j[\boldsymbol{r}](\boldsymbol{\xi}) + \sum_{l=1}^{L^i}\sum_{n=0}^{P_l^i} \alpha_{l,n}^i(\xi^i)\frac{\partial^n(\boldsymbol{r} - \boldsymbol{P}_j[\boldsymbol{r}])}{(\partial\xi^i)^n}(\boldsymbol{\xi}|_{\xi^i=\xi_l^i})\ . \tag{5.9}$$

From (5.7), it is evident that

$$\boldsymbol{P}_i[\boldsymbol{r}] \oplus \boldsymbol{P}_j[\boldsymbol{r}] = \boldsymbol{P}_j[\boldsymbol{r}] \oplus \boldsymbol{P}_i[\boldsymbol{r}]\ ,$$

so the indices i and j in (5.7), (5.9) can be interchanged.

The Boolean summation (5.6) matches $\boldsymbol{r}(\boldsymbol{\xi})$ and its derivatives at all sections $\xi^i = \xi_k^i$ and $\xi^j = \xi_l^j$, i.e.

$$\frac{\partial^{k+p}}{(\partial\xi^i)^k(\partial\xi^j)^p}(\boldsymbol{P}_i[\boldsymbol{r}] \otimes \boldsymbol{P}_j[\boldsymbol{r}])(\boldsymbol{\xi}|_{\xi^t=\xi_l^t}) = \frac{\partial^{k+p}}{(\partial\xi^i)^k(\partial\xi^j)^p}\boldsymbol{r}(\boldsymbol{\xi}|_{\xi^t=\xi_l^t})\ ,$$

where either $t = i$ or $t = j$.

Three-Dimensional Interpolation

A multidirectional interpolation $\boldsymbol{P}[\boldsymbol{r}](\boldsymbol{\xi})$ of $\boldsymbol{r}(\boldsymbol{\xi})$, which matches the values of the function $\boldsymbol{r}(\boldsymbol{\xi})$ and of its derivatives at the sections $\xi^i = \xi_l^i, l = 1, \dots, L_i$, in all directions $\xi^i, i = 1, 2, 3$, is defined through the Boolean summation of all unidirectional interpolations $\boldsymbol{P}_i[\boldsymbol{r}], i = 1.2, 3$:

$$\boldsymbol{P}[\boldsymbol{r}] = \boldsymbol{P}_1[\boldsymbol{r}] \oplus \boldsymbol{P}_2[\boldsymbol{r}] \oplus \boldsymbol{P}_3[\boldsymbol{r}]\ . \tag{5.10}$$

Taking into account (5.6), we obtain

$$\begin{aligned}\boldsymbol{P}[\boldsymbol{r}] = \boldsymbol{P}_1[\boldsymbol{r}] + \boldsymbol{P}_2[\boldsymbol{r}] + \boldsymbol{P}_3[\boldsymbol{r}]&\\ -\boldsymbol{P}_1[\boldsymbol{r}] \otimes \boldsymbol{P}_2[\boldsymbol{r}] - \boldsymbol{P}_1[\boldsymbol{r}] \otimes \boldsymbol{P}_3[\boldsymbol{r}] - \boldsymbol{P}_2[\boldsymbol{r}] \otimes \boldsymbol{P}_3[\boldsymbol{r}]&\\ +\boldsymbol{P}_1[\boldsymbol{r}] \otimes \boldsymbol{P}_2[\boldsymbol{r}] \otimes \boldsymbol{P}_3[\boldsymbol{r}]\ .&\end{aligned} \tag{5.11}$$

Recursive Form of Transfinite Interpolation

Using the relation (5.8), we can easily show that (5.11) is also equal to the following equation:

$$\boldsymbol{P}[\boldsymbol{r}] = \boldsymbol{P}_1[\boldsymbol{r}] + \boldsymbol{P}_2\big[\boldsymbol{r} - \boldsymbol{P}_1[\boldsymbol{r}]\big] + \boldsymbol{P}_3\Big[\boldsymbol{r} - \boldsymbol{P}_1[\boldsymbol{r}] - \boldsymbol{P}_2\big[\boldsymbol{r} - \boldsymbol{P}_1[\boldsymbol{r}]\big]\Big]\ . \tag{5.12}$$

This represents the formula (5.10) for multidirectional interpolation as the sum of the three unidirectional interpolations $\boldsymbol{P}_1[\boldsymbol{r}]$, $\boldsymbol{P}_2\Big[\boldsymbol{r} - \boldsymbol{P}_1[\boldsymbol{r}]\Big]$, and $\boldsymbol{P}_3\Big[\boldsymbol{r} - \boldsymbol{P}_1[\boldsymbol{r}] - \boldsymbol{P}_2\Big[\boldsymbol{r} - \boldsymbol{P}_1[\boldsymbol{r}]\Big]\Big]$. Therefore, the expression (5.12) for $\boldsymbol{P}[\boldsymbol{r}]$ gives a recursive form of the interpolation (5.10) through a sequence of the unidirectional interpolations (5.1):

$$\begin{aligned}
\boldsymbol{F}_1[\boldsymbol{r}] &= \boldsymbol{P}_1[\boldsymbol{r}]\ ,\\
\boldsymbol{F}_2[\boldsymbol{r}] &= \boldsymbol{F}_1[\boldsymbol{r}] + \boldsymbol{P}_2\Big[\boldsymbol{r} - \boldsymbol{F}_1[\boldsymbol{r}]\Big]\ ,\\
\boldsymbol{P}[\boldsymbol{r}]\ &= \boldsymbol{F}_2[\boldsymbol{r}] + \boldsymbol{P}_3\Big[\boldsymbol{r} - \boldsymbol{F}_2[\boldsymbol{r}]\Big]
\end{aligned}$$

which is usually applied in constructing algebraic coordinate transformations. Using (5.1), we obtain

$$\begin{aligned}
\boldsymbol{F}_1[\boldsymbol{r}](\boldsymbol{\xi}) &= \sum_{l=1}^{L^1}\sum_{n=0}^{P_l^1} \alpha_{l,n}^1(\xi^1)\frac{\partial^n \boldsymbol{r}}{(\partial\xi^1)^n}(\xi_l^1, \xi^2, \xi^3)\ ,\\
\boldsymbol{F}_2[\boldsymbol{r}](\boldsymbol{\xi}) &= \boldsymbol{F}_1[\boldsymbol{r}](\boldsymbol{\xi}) + \sum_{l=1}^{L^2}\sum_{n=0}^{P_l^2} \alpha_{l,n}^2(\xi^2)\frac{\partial^n (\boldsymbol{r} - \boldsymbol{F}_1[\boldsymbol{r}])}{(\partial\xi^2)^n}(\xi^1, \xi_l^2, \xi^3)\ ,\\
\boldsymbol{P}[\boldsymbol{r}](\boldsymbol{\xi}) &= \boldsymbol{F}_2[\boldsymbol{r}](\boldsymbol{\xi}) + \sum_{l=1}^{L^3}\sum_{n=0}^{P_l^3} \alpha_{l,n}^3(\xi^3)\frac{\partial^n (\boldsymbol{r} - \boldsymbol{F}_2[\boldsymbol{r}])}{(\partial\xi^3)^n}(\xi^1, \xi^2, \xi_l^3)\ .
\end{aligned} \qquad (5.13)$$

It is easy to see, taking advantage of (5.2), that the multiple summation matches the function $\boldsymbol{r}(\boldsymbol{\xi})$ and its derivatives with respect to ξ^1, ξ^2, and ξ^3 on all sections $\xi^i = \xi_l^i$, $i = 1, 2, 3$, of the cube Ξ^3.

Outer Boundary Interpolation

Equation (5.13) shows that the outer boundary interpolation based on the two-boundary unidirectional interpolations described by (5.4) has the following form:

$$\begin{aligned}
\boldsymbol{F}_1[\boldsymbol{r}](\boldsymbol{\xi}) &= \sum_{n=0}^{P_1^1} \alpha_{1,n}^1(\xi^1)\frac{\partial^n \boldsymbol{r}}{(\partial\xi^1)^n}(0, \xi^2, \xi^3)\\
&\quad + \sum_{n=0}^{P_2^1} \alpha_{2,n}^1(\xi^1)\frac{\partial^n \boldsymbol{r}}{(\partial\xi^1)^n}(1, \xi^2, \xi^3)\ ,\\
\boldsymbol{F}_2[\boldsymbol{r}](\boldsymbol{\xi}) &= \boldsymbol{F}_1[\boldsymbol{r}](\boldsymbol{\xi}) + \sum_{n=0}^{P_1^2} \alpha_{1,n}^2(\xi^2)\frac{\partial^n (\boldsymbol{r} - \boldsymbol{F}_1[\boldsymbol{r}])}{(\partial\xi^2)^n}(\xi^1, 0, \xi^3)\\
&\quad + \sum_{n=0}^{P_2^2} \alpha_{2,n}^2(\xi^2)\frac{\partial^n (\boldsymbol{r} - \boldsymbol{F}_1[\boldsymbol{r}])}{(\partial\xi^2)^n}(\xi^1, 1, \xi^3)\ ,
\end{aligned}$$

$$\begin{aligned}\boldsymbol{P}[\boldsymbol{r}](\boldsymbol{\xi}) = \boldsymbol{F}_2[\boldsymbol{r}](\boldsymbol{\xi}) &+ \sum_{n=0}^{P_1^3} \alpha_{1,n}^3(\xi^3)\frac{\partial^n(\boldsymbol{r}-\boldsymbol{F}_2[\boldsymbol{r}])}{(\partial\xi^3)^n}(\xi^1,\xi^2,0)\\ &+ \sum_{n=0}^{P_2^3} \alpha_{2,n}^3(\xi^3)\frac{\partial^n(\boldsymbol{r}-\boldsymbol{F}_2[\boldsymbol{r}])}{(\partial\xi^3)^n}(\xi^1,\xi^2,1)\ . \end{aligned} \tag{5.14}$$

Two-Dimensional Interpolation

The formulas for two-dimensional transfinite interpolation of a two-dimensional vector-valued function $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^2 \to X^2$ are obtained from (5.13) and (5.14) by assuming $\boldsymbol{F}_2(\boldsymbol{r}) = \boldsymbol{P}(\boldsymbol{r})$, $\alpha_{l,k}^3 = 0$, and omitting ξ^3. For example, we obtain, from (5.13), the following formula for two-dimensional transfinite interpolation:

$$\begin{aligned}\boldsymbol{F}_1[\boldsymbol{r}](\xi^1,\xi^2) &= \sum_{l=1}^{L^1}\sum_{k=0}^{P_l^1} \alpha_k^1(\xi^1)\frac{\partial^k \boldsymbol{r}}{(\partial\xi^1)^k}(\xi_l^1,\xi^2)\ ,\\ \boldsymbol{P}[\boldsymbol{r}](\xi^1,\xi^2) &= \boldsymbol{F}_1[\boldsymbol{r}](\xi^1,\xi^2)\\ &\quad+\sum_{l=1}^{L^2}\sum_{m=0}^{P_l^2} \alpha_{l,m}^2(\xi^2)\frac{\partial^m(\boldsymbol{r}-\boldsymbol{F}_1[\boldsymbol{r}])}{(\partial\xi^2)^m}(\xi^1,\xi_l^2)\ .\end{aligned} \tag{5.15}$$

5.3 Algebraic Coordinate Transformations

This section sets out the definitions of the algebraic coordinate transformations appropriate for the generation of coordinate grids through the formulas of transfinite interpolation.

5.3.1 Formulation of Algebraic Coordinate Transformation

The formulas of transfinite interpolation described above give clear guidance on how to define an algebraic coordinate transformation

$$\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^3 \to X^3\ , \qquad \boldsymbol{x}(\boldsymbol{\xi}) = (x^1(\boldsymbol{\xi}), x^2(\boldsymbol{\xi})\ , x^3(\boldsymbol{\xi}))\ , \qquad \boldsymbol{\xi} = (\xi^1,\xi^2,\xi^3)$$

from the cube Ξ^3 onto a domain $X^3 \subset R^3$ which matches, at the boundary and some chosen intermediate coordinate planes of the cube, the prescribed values and the specified derivatives of $\boldsymbol{x}(\boldsymbol{\xi})$ along the coordinate directions emerging from the coordinate surfaces (Fig. 5.1).

Let there be chosen, in each direction ξ^i, some coordinate planes $\xi^i = \xi_l^i$, $l = 1,\dots,L^i$, of the cube Ξ^3, including two opposite boundary planes $\xi^i = \xi_1^i = 0$,

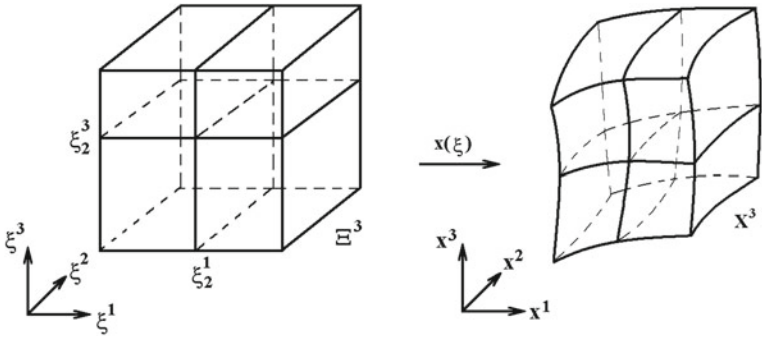


Fig. 5.1 Coordinate transformation

$\xi^i = \xi^i_{L^i} = 1$. Furthermore, let there be specified, at each section $\xi^i = \xi^i_l$, a smooth three-dimensional vector-valued function denoted by $\boldsymbol{A}^i_{l,0}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})$, which is assumed to represent the values of the function $\boldsymbol{x}(\boldsymbol{\xi})$ being constructed at the points of this section. Also, let there be specified, at this section, three-dimensional vector-valued functions denoted by $\boldsymbol{A}^i_{l,n}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})$ which represent derivatives with respect to ξ^i of the function $\boldsymbol{x}(\boldsymbol{\xi})$ on the respective sections $\xi^i = \xi^i_l$. Thus, it is assumed that

$$\begin{aligned}
\boldsymbol{A}^i_{l,0}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) &= \frac{\partial^0}{(\partial\xi^i)^0}\boldsymbol{x}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) = \boldsymbol{x}(\boldsymbol{\xi}|_{\xi^l=\xi^i_l})\ , \qquad l = 1, \dots, L_i\ ,\\
\boldsymbol{A}^i_{l,n}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) &= \frac{\partial^n}{(\partial\xi^i)^n}\boldsymbol{x}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})\ , \qquad n = 1, \dots, P^i_l\ .
\end{aligned}$$

Since

$$\frac{\partial^m}{(\partial\xi^j)^m}\Big(\frac{\partial^n\boldsymbol{x}}{(\partial\xi^i)^n}\Big) = \frac{\partial^n}{(\partial\xi^j)^n}\Big(\frac{\partial^m\boldsymbol{x}}{(\partial\xi^j)^m}\Big),$$

we find that the vector functions $\boldsymbol{A}^i_{l,n}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})$ and $\boldsymbol{A}^j_{k,m}(\boldsymbol{\xi}|_{\xi^j=\xi^j_k})$ specifying the corresponding derivatives on the planes $\xi^i = \xi^i_l$ and $\xi^j = \xi^j_k$, respectively, must be compatible at the intersection of these planes, i.e.

$$\begin{aligned}
&\frac{\partial^m}{(\partial\xi^j)^m}\boldsymbol{A}^i_{l,n}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) = \frac{\partial^n}{(\partial\xi^j)^n}\boldsymbol{A}^j_{k,m}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k})\ ,\\
&n = 0, \dots, P^i_l\ , \qquad m = 0, \dots, P^j_k\ . \qquad (5.16)
\end{aligned}$$

When the vector-valued functions $\boldsymbol{A}^i_{l,k}$ satisfying (5.16) are specified, the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is obtained by substituting the functions $\boldsymbol{A}^i_{l,0}$ and $\boldsymbol{A}^i_{l,n}$ for the values of $\boldsymbol{r}(\boldsymbol{\xi})$ and of its derivatives $\partial^n\boldsymbol{r}/\partial(\xi^i)^n(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})$, respectively, in the above formulas for transfinite interpolation. Hence, the transformation based on the unidirectional interpolation given by (5.1) has the form

$$\boldsymbol{P}_i(\boldsymbol{\xi}) = \sum_{l=1}^{L^i} \sum_{n=0}^{P_l^i} \alpha_{l,n}^i(\xi^i) \boldsymbol{A}_{l,n}^i(\boldsymbol{\xi}|_{\xi^i=\xi_l^i}) \ . \tag{5.17}$$

This mapping matches the values of $\boldsymbol{A}_{l,n}^i$ only at the coordinate planes $\xi^i = \xi_l^i$ crossing the chosen coordinate ξ^i.

The formula (5.5) for the tensor product $\otimes$ of the two mappings $\boldsymbol{P}_i(\boldsymbol{\xi})$ and $\boldsymbol{P}_j(\boldsymbol{\xi})$ obtained from (5.17) then gives the transformation

$$\begin{aligned} &\boldsymbol{P}_i \otimes \boldsymbol{P}_j(\boldsymbol{\xi}) \\ &= \sum_{k=1}^{L^j} \sum_{m=0}^{P_k^j} \sum_{l=1}^{L^i} \sum_{n=0}^{P_l^i} \alpha_{l,n}^i(\xi^i) \alpha_{k,m}^j(\xi^j) \frac{\partial^n}{(\partial \xi^i)^n} \boldsymbol{A}_{k,m}^j(\boldsymbol{\xi}|_{\xi^i=\xi_l^i, \xi^j=\xi_k^j}) \ , \end{aligned} \tag{5.18}$$

which matches the values of $\boldsymbol{A}_{l,n}^i$ and $\boldsymbol{A}_{k,m}^j$ at the intersection of the planes $\xi^i = \xi_l^i$ and $\xi^j = \xi_k^j$. According to the consistency conditions (5.16), the operation of the tensor product is commutative, i.e.

$$\boldsymbol{P}_i \otimes \boldsymbol{P}_j(\boldsymbol{\xi}) = \boldsymbol{P}_j \otimes \boldsymbol{P}_i(\boldsymbol{\xi}) \ ,$$

which is indispensable for an appropriate definition of the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$.

5.3.2 General Algebraic Transformations

The general formula for the three-dimensional coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ that provides a matching with $\boldsymbol{A}_{l,n}^i$ in all directions and at all chosen coordinate planes $\xi^i = \xi_l^i$ is given by the replacement of the values of the function $\boldsymbol{r}(\boldsymbol{\xi})$ and of its derivatives in the recursive formula (5.13) by the functions $\boldsymbol{A}_{l,n}^i$. Thus, we obtain

$$\begin{aligned} \boldsymbol{F}_1(\boldsymbol{\xi}) &= \sum_{l=1}^{L^1} \sum_{n=0}^{P_l^1} \alpha_{l,n}^1(\xi^1) \boldsymbol{A}_{l,n}^1(\xi_l^1, \xi^2, \xi^3) \ , \\ \boldsymbol{F}_2(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\xi) + \sum_{l=1}^{L^2} \sum_{n=0}^{P_l^2} \alpha_{l,n}^2(\xi^2) \Big(\boldsymbol{A}_{l,n}^2 - \frac{\partial^n \boldsymbol{F}_1}{(\partial \xi^2)^n} \Big)(\xi^1, \xi_l^2, \xi^3) \ , \\ \boldsymbol{x}(\boldsymbol{\xi}) &= \boldsymbol{F}_2(\boldsymbol{\xi}) + \sum_{l=1}^{L^3} \sum_{n=0}^{P_l^3} \alpha_{l,n}^3(\xi^3) \Big(\boldsymbol{A}_{l,n}^3 - \frac{\partial^n \boldsymbol{F}_2}{(\partial \xi^3)^n} \Big)(\xi^1, \xi^2, \xi_l^3) \ . \end{aligned} \tag{5.19}$$

As the specified functions $\boldsymbol{A}^i_{l,n}$ are consistent on the intersections of the planes $\xi^i = \xi^i_l$ and $\xi^j = \xi^j_k$ and, therefore, the tensor product of the transformations $\boldsymbol{P}_i(\boldsymbol{\xi})$ and $\boldsymbol{P}_j(\boldsymbol{\xi})$ is commutative, the result (5.19) is independent of the specific ordering of the successive interpolation directions ξ^i.

The formula for the two-dimensional algebraic coordinate transformation is obtained in a corresponding way from (5.15):

$$\begin{aligned}
\boldsymbol{F}_1(\boldsymbol{\xi}) &= \sum_{l=1}^{L^1}\sum_{n=0}^{P^1_l} \alpha^1_{l,n}(\xi^1)\boldsymbol{A}^1_{l,n}(\xi^1_l, \xi^2) \ , \\
\boldsymbol{x}(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\boldsymbol{\xi}) + \sum_{l=1}^{L^2}\sum_{n=0}^{P^2_l} \alpha^2_{l,n}\Big(\boldsymbol{A}^2_{l,n} - \frac{\partial^n \boldsymbol{F}_1}{(\partial \xi^2)^n}\Big)(\xi^1, \xi^2_l) \ ,
\end{aligned} \tag{5.20}$$

where $\boldsymbol{A}^i_{l,n}$ are two-dimensional vector-valued functions representing $\boldsymbol{x}(\boldsymbol{\xi})$ for $n = 0$ and its derivatives for $P^i_l \geq n > 0$ at the sections

$$\xi^i = \xi^i_l \ , \qquad i = 1, 2 \ , \qquad l = 1, \ldots, L^i \ .$$

These functions must satisfy the relations (5.16) at the points $(\xi^1_l,\ \xi^2_m)$, $l = 1, \ldots, L^1,\ m = 1, \ldots, L^2$.

The vector-valued function $\boldsymbol{x}(\boldsymbol{\xi})$ defined by (5.19) maps the unit cube Ξ^3 onto the physical region X^3 bounded by the six coordinate surfaces specified by the parametrizations $\boldsymbol{A}^i_{1,0}(\boldsymbol{\xi}|_{\xi^i=0})$ and $\boldsymbol{A}^i_{L^i,0}(\boldsymbol{\xi}|_{\xi^i=1})$, $i = 1, 2, 3$, from the respective boundary intervals of Ξ^3. The introduction of the intermediate planes $\xi^i = \xi^i_l$, $0 < \xi^i_l < 1$, into the formulas of transfinite interpolation allows one to control the grid distribution and grid spacing in the vicinity of some selected interior surfaces of the domain X^3. A similar result is achieved by joining, at the selected boundary surfaces, a series of transformations $\boldsymbol{x}(\boldsymbol{\xi})$ constructed using the above described outer boundary interpolation equation (5.14):

$$\begin{aligned}
\boldsymbol{F}_1(\boldsymbol{\xi}) &= \sum_{n=0}^{P^1_1} \alpha^1_{1,n}(\xi^1)\boldsymbol{A}^1_{1,n}(0, \xi^2, \xi^3) \\
&\quad + \sum_{n=0}^{P^1_2} \alpha^1_{2,n}(\xi^1)\boldsymbol{A}^1_{2,n}(1, \xi^2, \xi^3) \ , \\
\boldsymbol{F}_2(\xi) &= \boldsymbol{F}_1(\xi) + \sum_{n=0}^{P^2_1} \alpha^2_{1,n}(\xi^2)\Big(\boldsymbol{A}^2_{1,n} - \frac{\partial^n \boldsymbol{F}_1}{(\partial \xi^2)^n}(\xi^1, 0, \xi^3)\Big) \\
&\quad + \sum_{n=0}^{P^2_2} \alpha^2_{2,n}(\xi^2)\Big(\boldsymbol{A}^2_{2,n} - \frac{\partial^n \boldsymbol{F}_1}{(\partial \xi^2)^n}\Big)(\xi^1, 1, \xi^3) \ ,
\end{aligned}$$

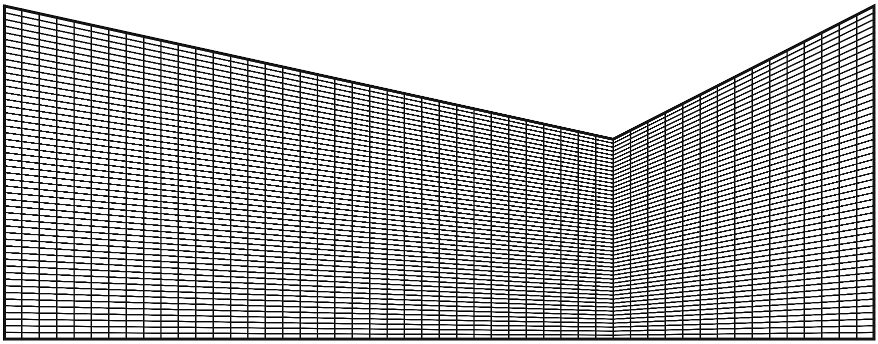


Fig. 5.2 Two-dimensional nonsmooth grid generated by means of transfinite interpolation

$$\boldsymbol{x}(\boldsymbol{\xi}) = \boldsymbol{F}_2(\xi) + \sum_{n=0}^{P_1^3} \alpha_{1,n}^3(\xi^3)\Big(\boldsymbol{A}_{1,n}^3 - \frac{\partial^n \boldsymbol{F}_2}{(\partial \xi^3)^n}\Big)(\xi^1, \xi^2, 0)$$
$$+ \sum_{n=0}^{P_2^3} \alpha_{2,n}^3(\xi^3)\Big(\boldsymbol{A}_{2,n}^3 - \frac{\partial^n \boldsymbol{F}_2}{(\partial \xi^3)^n}\Big)(\xi^1, \xi^2, 1) \, . \qquad (5.21)$$

This boundary interpolation transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is widely applied to generate grids in regions around bodies. These domains cannot be successfully gridded by one global mapping $\boldsymbol{x}(\boldsymbol{\xi})$ from the unit cube Ξ^3 because of the inevitable singularities pertinent to such global maps. An approach based on the matching of a series of boundary-interpolated transformations is thus preferable. It only requires the consistent specification of the parametrizations and coordinate directions at the corresponding boundary surfaces.

Equations (5.18)–(5.21) use the same set of blending functions $\alpha_{l,n}^i(\xi^i)$ to define each component $x^i(\xi)$ of the transformation $\boldsymbol{x}(\boldsymbol{\xi})$. These formulas can be generalized by introducing an individual set of blending functions $\alpha_{l,n}^i(\xi^i)$ for the definition of each component $x^i(\boldsymbol{\xi})$ of the map $\boldsymbol{x}(\boldsymbol{\xi})$ being built. Such a generalization gives broader opportunities to define appropriate algebraic coordinate transformations $\boldsymbol{x}(\boldsymbol{\xi})$ and, therefore, to generate grids more successfully.

One of the drawbacks of the method of transfinite interpolation for generating structured grids is that it carries boundary-sharp bends inside a domain (Fig. 5.2).

5.4 Lagrange and Hermite Interpolations

The recursive formula (5.19) represents a general form of transfinite interpolation which includes the prescribed values of the constructed coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ and of its derivatives up to order P_l^i at the sections $\xi^i = \xi_l^i$ of the cube Ξ^3.

However, most grid generation codes require, as a rule, only specification of the values of the function $\boldsymbol{x}(\boldsymbol{\xi})$ being sought and sometimes, in addition, the values of its first derivatives at the selected sections. Such sorts of algebraic coordinate transformation are described in this section.

5.4.1 Coordinate Transformations Based on Lagrange Interpolation

A Lagrange interpolation matches only the values of the function $\boldsymbol{r}(\boldsymbol{\xi})$ at some prescribed sections $\xi^i = \xi^i_l$, $l = 1, \dots, L^i$, of the cube Ξ^3. So, in accordance with (5.1), the unidirectional Lagrange interpolation has the following form:

$$\boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi}) = \sum_{l=1}^{L^i} \alpha^i_l(\xi^i)\boldsymbol{r}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) \ .$$

The blending function $\alpha^i_l(\xi^i)$ in this equation corresponds to $\alpha^i_{l,0}(\xi^i)$ in the formula (5.1). Taking into account (5.2), the blending functions $\alpha^i_l(\xi^i)$, $\quad l = 1, \dots, L^i$, depending on one independent variable ξ^i, must be subject to the following restrictions:

$$\alpha^i_l(\xi^i_k) = \delta^l_k \ , \qquad l, k = 1, \dots, L^i \ . \tag{5.22}$$

These restrictions imply that the blending function α^i_l for a fixed l equals 1 at the point $\xi^i = \xi^i_l$ and equals zero at all other points ξ^i_m, $m \neq l$. The formula for the construction of a three-dimensional coordinate mapping $\boldsymbol{x}(\boldsymbol{\xi})$ based on the Lagrangian interpolation is obtained from (5.19) as

$$\begin{aligned}
\boldsymbol{F}_1(\boldsymbol{\xi}) &= \sum_{l=1}^{L^1} \alpha^1_l(\xi^1)\boldsymbol{A}^1_l(\boldsymbol{\xi}|_{\xi^1=\xi^1_l}) \ , \\
\boldsymbol{F}_2(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\boldsymbol{\xi}) + \sum_{l=1}^{L^2} \alpha^2_l(\xi^2)\Big(\boldsymbol{A}^2_l - \boldsymbol{F}_1\Big)(\boldsymbol{\xi}|_{\xi^2=\xi^2_l}) \ , \\
\boldsymbol{x}(\boldsymbol{\xi}) &= \boldsymbol{F}_2(\boldsymbol{\xi}) + \sum_{l=1}^{L^3} \alpha^3_l(\xi^3)\Big(\boldsymbol{A}^3_l - \boldsymbol{F}_2\Big)(\boldsymbol{\xi}|_{\xi^3=\xi^3_l}) \ ,
\end{aligned} \tag{5.23}$$

where the blending functions $\alpha^i_l(\xi^i)$ satisfy (5.22), and the functions $\boldsymbol{A}^i_l(\xi|_{\xi^i=\xi^i_l})$ corresponding to $\boldsymbol{A}^i_{l,0}$ in (5.22) specify the values of the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ being sought. In accordance with (5.16), the specified functions $\boldsymbol{A}^i_l$ must coincide at the intersection of their respective coordinate planes $\xi^i = \xi^i_l$, i.e.

$$\boldsymbol{A}^i_l(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) = \boldsymbol{A}^j_k(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) \ .$$

When $L^1 = L^2 = L^3 = 2$, i.e. the prescribed interior sections are absent, then the ith component of the transformation (5.23) has the following form:

$$\begin{aligned} F^i_1(\xi^1,\xi^2,\xi^3) &= \alpha^1_1(\xi^1)\psi^i(0,\xi^2,\xi^3) + \alpha^1_2(\xi^1)\psi^i(1,\xi^2,\xi^3), \\ F^i_2(\xi^1,\xi^2,\xi^3) &= F^i_1(\xi^1,\xi^2,\xi^3) + \alpha^2_1(\xi^2)[\psi^i(\xi^1,0,\xi^3) - F^i_1(\xi^1,0,\xi^3)] + \\ &+ \alpha^2_2(\xi^2)[\psi^i(\xi^1,1,\xi^3) - F^i_1(\xi^1,1,\xi^3)], \\ x^i(\xi^1,\xi^2,\xi^3) &= F^i_2(\xi^1,\xi^2,\xi^3) + \alpha^3_1(\xi^3)[\psi^i(\xi^1,\xi^2,0) - F^i_2(\xi^1,\xi^2,0)] + \\ &+ \alpha^3_2(\xi^3)[\psi^i(\xi^1,\xi^2,1) - F^i_2(\xi^1,\xi^2,1)], \quad i=1,2,3, \end{aligned} \tag{5.24}$$

where the function $\psi^i(\xi^1,\xi^2,\xi^3)$, $i = 1,2,3$ is the ith component of a specified boundary transformation $\boldsymbol{\psi}(\boldsymbol{\xi}) : \partial\Xi^3 \to \partial X^3$; $\alpha^i_l(t) : [0,1] \to \mathbb{R}$, $i = 1,2,3$, $l = 1,2$ are scalar blending functions subject to the following restrictions:

$$\alpha^i_1(0) = 1, \quad \alpha^i_1(1) = 0, \quad \alpha^i_2(0) = 0, \quad \alpha^i_2(1) = 1, \quad i = 1,2,3. \tag{5.25}$$

In particular, when $\alpha^i_1(t) = 1 - t$, $\alpha^i_2(t) = t$, $i = 1,2,3$, we obtain the simplest formula of transfinite interpolation in a vector form

$$\begin{aligned} \boldsymbol{F}_1(\boldsymbol{\xi}) &= (1-\xi^1)\boldsymbol{\psi}(0,\xi^2,\xi^3) + \xi^1\boldsymbol{\psi}(1,\xi^2,\xi^3), \\ \boldsymbol{F}_2(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\boldsymbol{\xi}) + (1-\xi^2)[\boldsymbol{\psi}(\xi^1,0,\xi^3) - \boldsymbol{F}_1(\xi^1,0,\xi^3)] \\ &+ \xi^2[\boldsymbol{\psi}(\xi^1,1,\xi^3) - \boldsymbol{F}_1(\xi^1,1,\xi^3)], \\ \boldsymbol{x}(\boldsymbol{\xi}) &= \boldsymbol{F}_2(\boldsymbol{\xi}) + (1-\xi^3)[\boldsymbol{\psi}(\xi^1,\xi^2,0) - \boldsymbol{F}_2(\xi^1,\xi^2,0)] \\ &+ \xi^3[\boldsymbol{\psi}(\xi^1,\xi^2,1) - \boldsymbol{F}_2(\xi^1,\xi^2,1)]. \end{aligned} \tag{5.26}$$

The conditions (5.25) provide the identity

$$\boldsymbol{x}(\boldsymbol{\xi})|_{\partial\Xi^3} = \boldsymbol{\psi}(\boldsymbol{\xi})|_{\partial\Xi^3} \tag{5.27}$$

for the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ obtained by (5.24).

Now we consider some examples of the blending functions used in Lagrange interpolations.

Lagrange Polynomials

The best-known blending functions $\alpha^i_l(\xi^i)$ satisfying (5.22) are defined as Lagrange polynomials applied to the points $\xi^i_1, \ldots, \xi^i_{L^i}$:

$$\alpha^i_l(\xi^i) = \prod_{j=1}^{L^i} \frac{\xi^i - \xi^i_j}{\xi^i_l - \xi^i_j} \ , \qquad j \neq l \ . \tag{5.28}$$

For example, when $L^i = 2$, then, from (5.28),

$$\alpha_1^i(\xi^i) = \frac{\xi^i - \xi_2^i}{\xi_1^i - \xi_2^i}\ , \qquad \alpha_2^i(\xi^i) = \frac{\xi^i - \xi_1^i}{\xi_2^i - \xi_1^i} = 1 - \alpha_1^i(\xi^i)\ . \tag{5.29}$$

Therefore, for the boundary interpolation, i.e. when $\xi_1^i = 0$, $\xi_2^i = 1$, we obtain

$$\alpha_1^i(\xi^i) = 1 - \xi^i\ , \qquad \alpha_2^i(\xi^i) = \xi^i\ . \tag{5.30}$$

Spline Functions

The Lagrange polynomials become polynomials of a high-order when a large number of intermediate sections $\xi^i = \xi_l^i$ is applied to control the grid distribution in the interior of the domain X^3. These polynomials of high order may cause oscillations. One way to overcome this drawback is to use splines as blending functions $\alpha_l^i(\xi^i)$. The splines are defined as polynomials of low-order between each of the specified points $\xi^i = \xi_{L^i}^i$, with continuity of some derivatives at the interior points.

Piecewise-continuous splines satisfying (5.22) can be derived by means of linear polynomials. The simplest pattern of such blending functions in the form of splines consists of piecewise linear functions:

$$\alpha_l(\xi^i) = \begin{cases} 0\ , & \xi^i \le \xi_{l-1}^i, \\ \dfrac{\xi^i - \xi_{l-1}^i}{\xi_l^i - \xi_{l-1}^i}\ , & \xi_{l-1}^i \le \xi^i \le \xi_l^i, \\ \dfrac{\xi_{l+1}^i - \xi^i}{\xi_{l+1}^i - \xi_l^i}\ , & \xi_l^i \le \xi^i \le \xi_{l+1}^i, \\ 0\ , & \xi^i \ge \xi_{l+1}^i \end{cases}$$

However, the use of these blending functions results in a nonsmooth point distribution, since they themselves are not smooth.

Continuity of the first derivative of a spline blending function can be achieved with polynomials of the third-order, regardless of the number of interior sections.

Construction Based on General Functions

The application of polynomials in the Lagrange interpolation gives only a poor opportunity to control the grid spacing near the selected boundary and interior surfaces. In this subsection, we describe a general approach, originally, proposed by Liseikin (1999), to constructing the blending functions $\alpha_l^i(\xi^i)$ through the use of a wide range of basic functions, which provides a real opportunity to control the grid point distribution.

The formulation of the blending functions on the interval $0 \le \xi^i \le 1$, with L^i specified points,

$$0 = \xi_1^i < \cdots < \xi_{L^i}^i = 1\ ,$$

requires only the specification of some univariate smooth positive function

$$\phi(x) : [0, \infty) \to [0, \infty)\ ,$$

satisfying the restrictions $\phi(0) = 0$, $\phi(1) = 1$. This function can be used as a basic element to derive the blending functions satisfying (5.22) through the following standard procedure.

First, we define two series of functions

$$\phi_l^f(\xi^i) \quad \text{and} \quad \phi_l^b(\xi^i) \,, \qquad l = 1, \ldots, L^i \;.$$

The functions $\phi_l^f(\xi^i)$ are defined for $l = 1$ by

$$\phi_1^f(\xi^i) = \phi(1 - \xi^i) \,, \qquad 0 \le \xi \le 1 \,,$$

and for $1 < l \le L^i$ by

$$\phi_l^f(\xi^i) = \begin{cases} 0 \,, & 0 \le \xi^i \le \xi^i_{l-1} \,, \\ \phi\Big(\dfrac{\xi^i - \xi^i_{l-1}}{\xi^i_l - \xi^i_{l-1}}\Big) \,, & \xi^i_{l-1} \le \xi^i \le 1 \,. \end{cases}$$

The functions $\phi_l^b(\xi^i)$ are determined similarly:

$$\phi^b_{L^i}(\xi^i) = \phi(\xi^i)$$

and for $1 \le l < L^i$,

$$\phi_l^b(\xi^i) = \begin{cases} 0 \,, & 1 \ge \xi^i \ge \xi^i_{l+1} \,, \\ \phi\Big(\dfrac{\xi^i_{l+1} - \xi^i}{\xi^i_{l+1} - \xi^i_l}\Big) \,, & 0 \le \xi^i \le \xi^i_{l+1} \,. \end{cases}$$

Using the functions $\phi_l^f(\xi^i)$ and $\phi_l^b(\xi^i)$, the blending coefficients $\alpha_l^i(\xi^i)$ satisfying (5.22) are defined by

$$\alpha_l^i(\xi^i) = \phi_l^f(\xi^i)\phi_l^b(\xi^i) \,, \qquad l = 1, \ldots, L^i \;. \tag{5.31}$$

Each of these blending functions vanishes outside some interval, and thus it affects the interpolation function only locally (Fig. 5.3).

Note that this procedure for constructing blending functions for the Lagrange interpolations will yield splines if the original function ϕ is a polynomial. This construction may be extended by using various original functions for the terms ϕ_l^f and ϕ_l^b in (5.31).

The simplest example of the basic function is $\phi(x) = x$. However, this function generates nonsmooth blending coefficients $\alpha_l^i(\xi^i)$ at the points ξ^i_{l-1} and ξ^i_{l+1}, since $\alpha_l^i(\xi^i) \equiv 0$ outside the interval $(\xi^i_{l-1},\ \xi^i_{l+1})$. If the derivative of $\phi(x)$ at the point $x = 0$ is zero, then the blending functions derived by the procedure described are smooth. One example of such a function is $\phi(x) = x^2$. It can readily be shown that in this case, the blending functions α_l^i are of the class $C^1[0, 1]$.

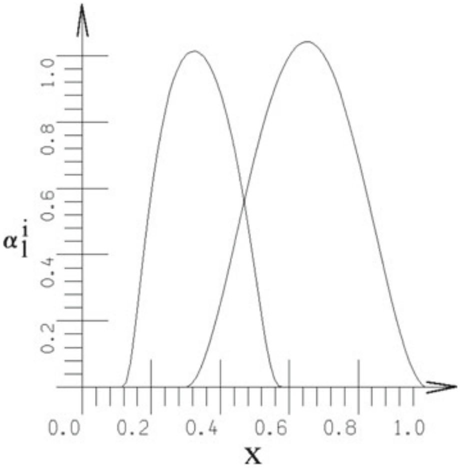


Fig. 5.3 Smooth blending functions

Continuity of the higher-order derivatives of the blending functions (5.31) is obtained when the basic function $\phi(x)$ satisfies the condition $\phi^{(k)}(0) = 0,\ k > 1$, in particular, if $\phi(x) = x^{k+1}$. The function $\phi(x) = \varphi(x)$, where

$$\varphi(x) = \begin{cases} 0\,, & x = 0\,, \\ a^{1-1/x}\,, & x > 0\,, \end{cases}$$

with $a > 1$, generates an infinitely differentiable blending function $\alpha_l^i(\xi^i)$ on the interval $[0, 1]$. Figure 5.3 demonstrates the blending functions constructed for $\phi(x) = \varphi(x)$ (left) and $\phi(x) = x^2$ (right).

Relations Between Blending Functions

Now we point out some relations between blending functions which can be useful for their construction. If the functions $\alpha_l^i(\xi^i)$ are blending functions for Lagrangian interpolation, namely, they are subject to the restrictions (5.22), then the functions $\beta_l^i(\xi^i)$ defined below satisfy the condition (5.22) as well:

$$\begin{aligned}
&(1)\quad \beta_l^i(\xi^i) = \alpha_l^i(\xi^i) f(\xi) \quad \text{if} \quad f(\xi_l^i) = 1\,, \\
&(2)\quad \beta_l^i(\xi^i) = \alpha_l^i[f(\xi^i)] \quad \text{if} \quad f(\xi_l^i) = \xi_l^i\,, \\
&(3)\quad \beta_l^i(\xi^i) = f[\alpha_l^i(\xi^i)] \quad \text{if} \quad f(0) = 0,\ f(1) = 1\,, \\
&(4)\quad \beta_l^i(\xi^i) = \alpha_l^i(\xi^i) + f(\xi^i) \quad \text{if} \quad f(\xi_l^i) = 0\,, \\
&(5)\quad \beta_l^i(\xi^i) = 0.5[\alpha_l^i(\xi^i) + \gamma_l^i(\xi^i)] \quad \text{if} \quad \gamma_l^i(\xi) \text{ satisfies (5.22)}\,.
\end{aligned} \tag{5.32}$$

5.4.2 Transformations Based on Hermite Interpolation

Hermite interpolation matches the values of both the function $\boldsymbol{r}(\boldsymbol{\xi})$, and its first derivatives $\partial \boldsymbol{r}/\partial \xi^i(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})$ at each section $\xi^i = \xi^i_l, l = 1, \dots, L^i$, and therefore, the unidirectional interpolation (5.1) takes the following form:

$$\boldsymbol{P}_i[\boldsymbol{r}](\boldsymbol{\xi}) = \sum_{l=1}^{L^i} \Big(\alpha^i_{l,0}(\xi^i)\boldsymbol{r}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) + \alpha^i_{l,1}(\xi^i)\frac{\partial \boldsymbol{r}}{\partial \xi^i}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l}) \Big) \,. \tag{5.33}$$

The formula (5.19) in the case of a Hermite coordinate mapping $\boldsymbol{x}(\boldsymbol{\xi})$ which matches the specified values of $\boldsymbol{x}(\boldsymbol{\xi})$, denoted by $\boldsymbol{A}^i_{l,0}$, and of its first derivatives, denoted by $\boldsymbol{A}^i_{l,1}$, at all sections $\xi^i = \xi^i_l$, $l = 1, \dots, L^i$, and in all directions $\xi^i, i = 1, 2, 3$, is thus reduced to

$$\begin{aligned}
\boldsymbol{F}_1(\boldsymbol{\xi}) &= \sum_{l=1}^{L^1} \Big(\alpha^1_{l,0}(\xi^1)\boldsymbol{A}^1_{l,0}(\xi^1_l, \xi^2, \xi^3) + \alpha^1_{l,1}(\xi^1)\boldsymbol{A}^1_{l,1}(\xi^1_l, \xi^2, \xi^3) \Big) \,, \\
\boldsymbol{F}_2(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\boldsymbol{\xi}) + \sum_{l=1}^{L^2} \Big(\alpha^2_{l,0}(\xi^2)(\boldsymbol{A}^2_{l,0} - \boldsymbol{F}_1)(\xi^1, \xi^2_l, \xi^3) \\
&\quad + \alpha^2_{l,1}(\xi^2)(\boldsymbol{A}^2_{l,1} - \frac{\partial \boldsymbol{F}_1}{\partial \xi^2})(\xi^1, \xi^2_l, \xi^3) \Big) \,, \\
\boldsymbol{x}(\boldsymbol{\xi}) &= \boldsymbol{F}_2(\boldsymbol{\xi}) + \sum_{l=1}^{L^3} \Big(\alpha^3_{l,0}(\xi^3)\big(\boldsymbol{A}^3_{l,0} - \boldsymbol{F}_2\big)(\xi^1, \xi^2, \xi^3_l) \\
&\quad + \alpha^3_{l,1}(\xi^3)\big(\boldsymbol{A}^3_{l,1} - \frac{\partial \boldsymbol{F}_2}{\partial \xi^3})(\xi^1, \xi^2, \xi^3_l) \Big) \,,
\end{aligned} \tag{5.34}$$

where, in accordance with (5.2), the blending functions $\alpha^i_{l,0}$, $\alpha^i_{l,1}$ satisfy the conditions

$$\begin{aligned}
\alpha^i_{l,0}(\xi^i_k) &= \delta^l_k \,, & \alpha^i_{l,1}(\xi^i_k) &= 0 \,, \\
\frac{\mathrm{d}}{\mathrm{d}\xi^i}\alpha^i_{l,1}(\xi^i_k) &= \delta^l_k \,, & \frac{\mathrm{d}}{\mathrm{d}\xi^i}\alpha^i_{l,0}(\xi^i_k) &= 0 \,, \\
l, k &= 1, \dots, L^i \,, & i &= 1, 2, 3 \,,
\end{aligned} \tag{5.35}$$

and the vector-valued functions $\boldsymbol{A}^i_{l,n}(\xi|_{\xi^i=\xi^i_l})$ satisfy the consistency conditions (5.16):

$$\begin{aligned}
\boldsymbol{A}^i_{l,0}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) &= \boldsymbol{A}^j_{k,0}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) \,, \\
\frac{\partial}{\partial \xi^j}\boldsymbol{A}^i_{l,0}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) &= \boldsymbol{A}^j_{k,1}(\boldsymbol{\xi}|_{\xi^i=\xi^i_l,\xi^j=\xi^j_k}) \,.
\end{aligned} \tag{5.36}$$

Construction of Blending Functions

The blending functions $\alpha^i_{l,m}(\xi^i), m = 0, 1$, for Hermite interpolations can be obtained from the smooth blending functions defined for Lagrange interpolations. Namely, let $\alpha^i_l(\xi^i)$, $l = 1, \dots, L^i$, be some smooth scalar functions meeting the conditions (5.22). The functions $\alpha^i_{l,m}$, $m = 0, 1$, determined by the relations

$$\begin{aligned}
\alpha^i_{l,0} &= \Big(1 - 2(\xi^i - \xi^i_l)\frac{\mathrm{d}\alpha^i_l}{\mathrm{d}\xi^i}(\xi_i)\Big)[\alpha^i_l(\xi^i)]^2 \; , \\
\alpha^i_{l,1} &= (\xi^i - \xi^i_l)[\alpha^i_l(\xi^i)]^2 \; ,
\end{aligned} \tag{5.37}$$

then satisfy (5.35) and, therefore, are the blending functions for the Hermite interpolations. For example, if $L^i = 2$ and the Lagrangian blending functions are defined through (5.29), then, from (5.37),

$$\begin{aligned}
\alpha^i_{1,0}(\xi^i) &= \Big(1 - 2\frac{\xi^i - \xi^i_1}{\xi^i_1 - \xi^i_2}\Big)\Big(\frac{\xi^i - \xi^i_2}{\xi^i_1 - \xi^i_2}\Big)^2 \; , \\
\alpha^i_{2,0}(\xi^i) &= \Big(1 - 2\frac{\xi^i - \xi^i_2}{\xi^i_2 - \xi^i_1}\Big)\Big(\frac{\xi^i - \xi^i_1}{\xi^i_2 - \xi^i_1}\Big)^2 \; , \\
\alpha^i_{1,1}(\xi^i) &= (\xi^i - \xi^i_1)\Big(\frac{\xi^i - \xi^i_2}{\xi^i_1 - \xi^i_2}\Big)^2 \; , \\
\alpha^i_{2,1}(\xi^i) &= (\xi^i - \xi^i_2)\Big(\frac{\xi^i - \xi^i_1}{\xi^i_2 - \xi^i_1}\Big)^2 \; .
\end{aligned} \tag{5.38}$$

So, if $\xi^i_1 = 0$, $\xi^i_2 = 1$, then, from these relations,

$$\begin{aligned}
\alpha^i_{1,0}(\xi^i) &= (1 + 2\xi^i)(\xi^i - 1)^2 \; , \\
\alpha^i_{2,0}(\xi^i) &= (3 - 2\xi^i)(\xi^i)^2 = 1 - \alpha^i_{1,0}(\xi^i) \; , \\
\alpha^i_{1,1}(\xi^i) &= \xi^i(1 - \xi^i)^2 \; , \\
\alpha^i_{2,1}(\xi^i) &= (\xi^i - 1)(\xi^i)^2 \; .
\end{aligned} \tag{5.39}$$

If the blending functions for Lagrange interpolation satisfy the condition

$$\frac{\mathrm{d}\alpha^i_l}{\mathrm{d}\xi^i}(\xi^i) \equiv 0 \; , \quad \text{if} \quad \xi^i \geq \xi^i_{l+1} \quad \text{and} \quad \xi^i \leq \xi^i_{l-1} \; , \tag{5.40}$$

then the blending functions $\alpha^i_{l,n}(\xi^i)$ for the Hermite interpolation can be derived from $\alpha^i_l(\xi^i)$ by the relations

$$\begin{aligned}
\alpha^i_{l,0}(\xi^i) &= \Big(1 + (\xi^i - \xi^i_l)\frac{\mathrm{d}\alpha^i_l}{\mathrm{d}\xi^i}(\xi^i_l)\Big)\alpha^i_l(\xi^i) \; , \\
\alpha^i_{l,1}(\xi^i) &= (\xi^i - \xi^i_l)\alpha^i_l(\xi^i) \; .
\end{aligned} \tag{5.41}$$

It is readily shown that the blending functions $\alpha^i_{l,n}(\xi^i),\ n = 0, 1$, satisfy the restriction (5.35). Note that the approach described above for the general construction of the blending functions for Lagrange interpolation yields the smooth blending functions $\alpha^i_l(\xi^i),\ l = 1, \ldots, L^i$, in the form (5.31), which, in addition to (5.22), are also subject to (5.40).

Deficient Form of Hermite Interpolation

Often, it is not reasonable to specify the values of the first derivative with respect to ξ^i of the sought coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ at all sections $\xi^i = \xi^i_l, l = 1, \ldots, L^i$, but only at some selected ones. By omitting the corresponding terms

$$\alpha^1_{l,1}(\xi^1)\boldsymbol{A}^1_{l,1}(\boldsymbol{\xi}|_{\xi^1=\xi^1_l})$$

and/or

$$\alpha^i_{l,1}(\xi^i)(\boldsymbol{A}^i_{l,1} - \frac{\partial \boldsymbol{F}_{i-1}}{\partial \xi^i})(\boldsymbol{\xi}|_{\xi^i=\xi^i_l})\ , \quad i = 2, 3\ ,$$

in (5.34), a deficient form of Hermite interpolation is obtained which matches the values of the first derivatives at the selected sections only. For example, the outer boundary interpolation which contains the outer boundary specifications on all boundaries but the outward derivative with respect to ξ^1 on the boundary $\xi^1 = 0$ only has, in accordance with (5.34), the form

$$\begin{aligned} \boldsymbol{F}_1(\boldsymbol{\xi}) &= \alpha^1_{1,0}(\xi^1)\boldsymbol{A}^1_{1,0}(0, \xi^2, \xi^3) + \alpha^1_{2,0}(\xi^1)\boldsymbol{A}^1_{2,0}(1, \xi^2, \xi^3) \\ &\quad + \alpha^1_{1,1}(\xi^1)\boldsymbol{A}^1_{1,1}(0, \xi^2, \xi^3)\ , \\ \boldsymbol{F}_2(\boldsymbol{\xi}) &= \boldsymbol{F}_1(\boldsymbol{\xi}) + \alpha^2_{1,0}(\xi^2)(\boldsymbol{A}^2_{1,0} - \boldsymbol{F}_1)(\xi^1, 0, \xi^3) \\ &\quad + \alpha^2_{2,0}(\xi^2)(\boldsymbol{A}^2_{2,0} - \boldsymbol{F}_1)(\xi^1, 1, \xi^3)\ , \\ \boldsymbol{x}(\boldsymbol{\xi}) &= \boldsymbol{F}_2(\boldsymbol{\xi}) + \alpha^3_{1,0}(\xi^3)(\boldsymbol{A}^3_{1,0} - \boldsymbol{F}_2)(\xi^1, \xi^2, 0) \\ &\quad + \alpha^3_{2,0}(\xi^3)(\boldsymbol{A}^3_{2,0} - \boldsymbol{F}_2)(\xi^1, \xi^2, 1)\ . \end{aligned} \tag{5.42}$$

Specification of Normal Directions

In the outer boundary interpolation technique, the outward derivatives $\boldsymbol{A}^i_{1,1}(\xi|_{\xi^i=0}),\ \boldsymbol{A}^i_{2,1}(\xi|_{\xi^i=1})$ along the lines emerging from the boundary surfaces are usually required to be performed as normals to the corresponding boundary surfaces in order to generate orthogonal grids near the boundaries. The boundary surfaces are parametrized by the specified boundary transformations $\boldsymbol{A}^i_{1,0}(\xi|_{\xi^i=0})$ and $\boldsymbol{A}^i_{2,0}(\xi|_{\xi^i=1})$, respectively. Therefore, these normals can be computed from the cross product of the vectors tangential to the boundary surfaces. For example, the ξ^1 coordinate direction $\boldsymbol{A}^1_{l,1}(\xi^1_l, \xi^2, \xi^3)$ can be specified as

$$\boldsymbol{A}^1_{1,1}(0, \xi^2, \xi^3) = g(\xi^2, \xi^3)\Big(\frac{\partial}{\partial \xi^2}\boldsymbol{A}^1_{1,0}(0, \xi^2, \xi^3) \times \frac{\partial}{\partial \xi^3}\boldsymbol{A}^1_{1,0}(0, \xi^2, \xi^3)\Big)\ ,$$

where $g(\xi^2, \xi^3)$ is a scalar function that can be used to control the spacing of the grid lines emerging from the boundary surface represented by the parametrization $\boldsymbol{A}^1_{1,0}(0, \xi^2, \xi^3)$. Such a specification of the first derivatives can be chosen on the interior sections as well.

Parametrization of Cells

Formulas of transfinite interpolation can also be used for definition of the shape of a grid cell and the parametrization of this cell by values of the coordinates of its vertices. For this purpose, a transformation of the cube Ξ^3 into X^3 is written out in such a way that the vertices of the cube are transformed on the vertices of the cell, while the edges of the cube are transformed by the formula of the one–dimensional transformation. After this, the faces of the cube are transformed on the faces of the cell with the formula of the two–dimensional transfinite interpolation, and finally the interior of the cube Ξ^3 is mapped on the interior of the cell through the formula of the three–dimensional transfinite interpolation. The formula of such a transformation has the following form:

$$\begin{aligned}
&\boldsymbol{x}(\boldsymbol{\xi}) = (1-\xi^1)(1-\xi^2)(1-\xi^3)\boldsymbol{x}_{000} + (1-\xi^1)(1-\xi^2)\xi^3\boldsymbol{x}_{001} + \\
&+ (1-\xi^1)\xi^2(1-\xi^3)\boldsymbol{x}_{010} + (1-\xi^1)\xi^2\xi^3\boldsymbol{x}_{011} + \xi^1(1-\xi^2)(1-\xi^3)\boldsymbol{x}_{100} + \\
&+ \xi^1(1-\xi^2)\xi^3\boldsymbol{x}_{101} + \xi^2\xi^2(1-\xi^3)\boldsymbol{x}_{110} + \xi^1\xi^2\xi^3\boldsymbol{x}_{111},
\end{aligned}$$

where $\boldsymbol{x}_{i_1 i_2 i_3} = (x^1(i_1, i_2, i_3), x^2(i_1, i_2, i_3), x^3(i_1, i_2, i_3)), i_1, i_2, i_3 = 0, 1,$ are the vertices of the cell. The edges of this cell are the straight lines connecting its vertices, while its faces are surfaces of second order.

5.5 Control Techniques

Commonly, all algebraic schemes are computationally efficient but require a significant amount of user interaction and control techniques to define workable meshes. This section delineates some control approaches applied to algebraic grid generation.

The spacing between the grid points and the skewness of the grid cells in the physical domain is controlled in the algebraic method, primarily by the blending functions $\alpha^i_{l,n}(\xi^i)$, by the representations of the boundary and intermediate surfaces $\boldsymbol{A}^i_{l,0}(\xi|_{\xi^i=\xi^i_l})$, and by the values of the first derivatives $\boldsymbol{A}^i_{l,1}(\xi|_{\xi^i=\xi^i_l})$ in the interpolation equations.

As was stated in Chap. 4, an effective approach which significantly simplifies the control of grid generation relies on the introduction of an intermediate control domain between the computational and the physical regions. The control domain is a unit cube Q^3 with the Cartesian coordinates $q^i, \ i = 1, 2, 3$. In this approach, the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ from the unit cube Ξ^3 onto the physical region X^3 is defined as a composition of two transformations: $\boldsymbol{q}(\boldsymbol{\xi})$ from Ξ^3 onto Q^3 and $g(\boldsymbol{q}), \ \boldsymbol{q} = (q^1, q^2, q^3)$, from Q^3 onto X^3, that is,

$$\boldsymbol{x}(\boldsymbol{\xi}) = g[\boldsymbol{q}](\boldsymbol{\xi}) : \varXi^3 \to X^3 \ .$$

The functions $g(\boldsymbol{q})$ and $\boldsymbol{q}(\boldsymbol{\xi})$ can be constructed through the formulas of transfinite interpolation or by other techniques. As both the computational domain $\varXi^3$ and the intermediate domain Q^3 are the standard unit cubes, the formulas of transfinite interpolation for the generation of the intermediate transformations $\boldsymbol{q}(\boldsymbol{\xi})$ are somewhat simpler than the original expressions. In these formulas, it can be assumed, without any loss of generality, that the boundary planes $\xi^i = 0$ and $\xi^i = 1$ for each $i = 1, 2, 3$ are transformed by the function $\boldsymbol{q}(\boldsymbol{\xi})$ onto the boundary planes $q^i = 0$ and $q^i = 1$, respectively, so that

$$\begin{aligned}
\boldsymbol{q}(\boldsymbol{\xi}|_{\xi^1=0}) &= [0, q^2(0, \xi^2, \xi^3), q^3(0, \xi^2, \xi^3)] \ , \\
\boldsymbol{q}(\boldsymbol{\xi}|_{\xi^1=1}) &= [1, q^2(1, \xi^2, \xi^3), q^3(1, \xi^2, \xi^3)] \ .
\end{aligned}$$

Therefore, the first component $q^1(\xi)$ of the Lagrangian boundary interpolation for the intermediate mapping $\boldsymbol{q}(\boldsymbol{\xi})$ has the form

$$\begin{aligned}
F_1(\boldsymbol{\xi}) &= \alpha_2^1(\xi^1) \ , \\
F_2(\boldsymbol{\xi}) &= F_1(\boldsymbol{\xi}) + \alpha_1^2(\xi^2)\Big(u^1(\xi^1, 0, \xi^3) - \alpha_2^1(\xi^1)\Big) \\
&\quad + \alpha_2^2(\xi^2)\ \Big(u^1(\xi^1, 1, \xi^3 - \alpha_2^1(\xi^1)\Big) \ , \\
q^1(\boldsymbol{\xi}) &= F_2(\boldsymbol{\xi}) + \alpha_1^3(\xi^3)\Big(u^1(\xi^1, \xi^2, 0) - F_2(\xi^1, \xi^2, 0)\Big) \\
&\quad + \alpha_2^3(\xi^3)\Big(u^1(\xi^1, \xi^2, 1) - F_2(\xi^1, \xi^2, 1)\Big) \ .
\end{aligned} \tag{5.43}$$

Analogous equations can be defined for the other components of the intermediate transformation $\boldsymbol{q}(\boldsymbol{\xi})$.

The functions based on the reference univariate transformations $x_{i,\mathrm{c}}(\varphi, \epsilon)$ and $x_{i,\mathrm{s}}(\varphi, \epsilon)$ introduced in Chap. 4 can be used very successfully as blending functions to construct intermediate transformations by Lagrange and Hermite interpolations in the two-boundary technique. In the case of Lagrange interpolation, the blending function $\alpha_{1,0}^i(\xi^i)$ satisfies the conditions $\alpha_{1,0}^i(0) = 1, \ \alpha_{1,0}^i(1) = 0$. Therefore, any monotonically decreasing function derived by applying the procedures described in Sect. 4.4 to the reference univariate functions can be used as the blending function $\alpha_{1,0}^i(\xi^i)$. Analogously, the blending function $\alpha_{2,0}^i(\xi^i)$ can be represented by any monotonically increasing mapping based on one of the standard local contraction functions $x_i(\varphi, \epsilon)$. The blending functions $\alpha_{i,1}^i(\xi^i)$ for Hermite interpolations can also use these standard transformations through applying the operation described by (5.37) to the blending functions $\alpha_{1,0}^i(\xi^i)$. By choosing the proper functions, one has an opportunity to construct intermediate transformations that provide adequate grid clustering in the zones where it is necessary.

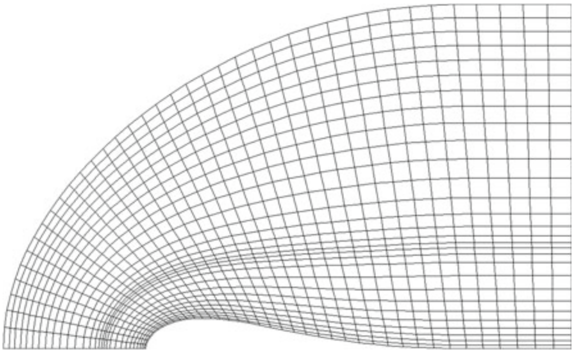


Fig. 5.4 Quadrangular adaptive grid

One example of a two-dimensional adaptive quadrangular grid, with the intermediate grid generated in such a manner through the basic stretching functions, is presented in Fig. 5.4.

5.6 Transfinite Interpolation from Triangles and Tetrahedrons

The formulas of transfinite interpolation define a coordinate transformation from the unit cube Ξ^3 (the square Ξ^2 in two dimensions and line Ξ^1 in one dimension) onto a physical domain X^3 (or X^2 or X^1). The application of this interpolation may lead to singularities of the type pertaining to polar transformations when any boundary segment of the physical domain, corresponding to a boundary segment of the computational domain, is contracted into a point. An example is when the boundary of a physical two-dimensional domain X^2 is composed of three smooth segments, as shown in Fig. 5.5. One way to treat such regions is to use coordinate transformations from triangular computational domains in two dimensions and tetrahedral domains in three dimensions. It can be seen that the transfinite interpolation approach can be modified to generate triangular or tetrahedral grids by mapping a standard triangular or tetrahedral domain, respectively. The formulation of a transfinite interpolation to obtain these transformations from the standard unit tetrahedron (triangle in two dimensions) is based on the composition of an operation of scaling (stretching) the coordinates to deform the tetrahedron into the unit cube Ξ^3 and an algebraic transformation constructed by the equations given above.

This procedure is readily clarified in two dimensions by the scheme depicted in Fig. 5.5. Suppose that the boundary segments AB, BC, and CD of the unit triangle T^2 are mapped onto the corresponding boundary segments AB, BC, and CD of the domain X^2. Then, in this procedure, the standard triangle T^2 with a uniform triangular grid is expanded to a square by a deformation $\boldsymbol{\xi}(\boldsymbol{t})$ uniformly stretching each horizontal line of the triangle to make it a rectangle, and afterwards, the rectangle

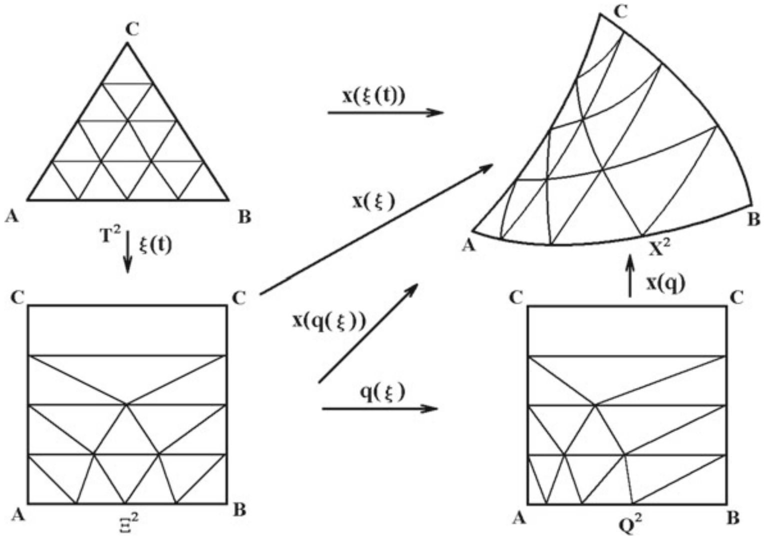


Fig. 5.5 Scheme for gridding triangular curvilinear domains with triangles

is uniformly stretched in the vertical direction to make it the unit square Ξ^2, as shown in Fig. 5.5. This operation is the inverse of the contraction $\boldsymbol{t}(\boldsymbol{\xi})$ of the square along the horizontal and vertical lines to transform it into the triangle. As a result, we obtain a square Ξ^2 with triangular cells on all horizontal levels except the top one. The number of these cells in each horizontal band reduces from the lower levels to the upper ones. The top level consists of one rectangular cell. With this deformation of T^2, the transformation between the boundaries of T^2 and X^2 generates the transformation

$$\boldsymbol{x}(\boldsymbol{\xi}) : \partial\Xi^2 \to \partial X^2 \ ,$$

which is the composition of $\boldsymbol{t}(\boldsymbol{\xi})$ and the assumed mapping of the boundary of T^2 onto the boundary of X^2. This boundary transformation maps the top segment of Ξ^2 onto the point C in X^2. Now, applying the formulas of transfinite interpolation to a square Ξ^2 with such grid cells, and the specified boundary transformation, one generates the algebraic transformation

$$\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^2 \to X^2$$

and consequently

$$\boldsymbol{x}[\boldsymbol{\xi}(\boldsymbol{t})] : T^2 \to X^2$$

from the triangle to the physical region X^2 with the prescribed values of the transformation at the boundary segments of the triangle. Note that the composition $\boldsymbol{x}[\boldsymbol{\xi}(\boldsymbol{t})]$

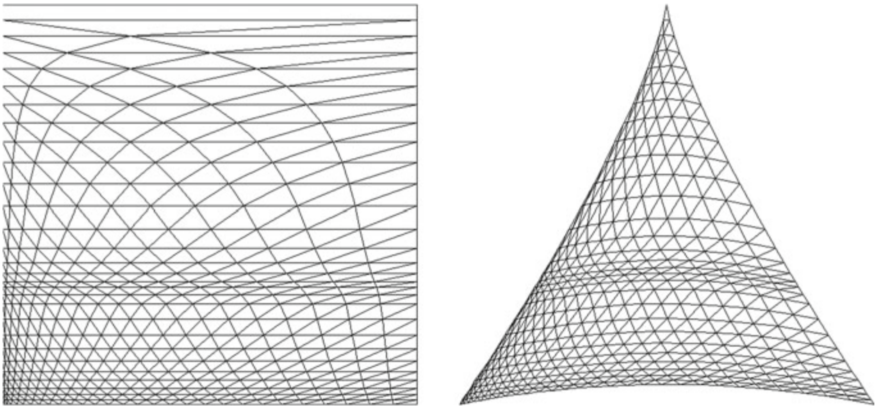


Fig. 5.6 Example of an adaptive algebraic triangular grid (*right*) and the corresponding grid on the intermediate domain (*left*) generated by the algebraic method

is continuous as the upper segment of Ξ^2 is transformed by $\boldsymbol{x}(\boldsymbol{\xi})$ onto one point C in X^2.

In fact, such a triangular grid in the physical domain can be generated directly by mapping the nonuniform grid constructed in the unit square Ξ^2 as described above onto X^2 with a standard algebraic coordinate transformation defined by transfinite interpolation.

The generation of grids by this approach is very well justified for regions shaped like curvilinear triangles, i.e. their boundaries are composed of three smooth curves intersecting at angles θ less than π. By dividing an arbitrary domain into triangular curvilinear domains, one can generate a composite triangular grid in the entire domain through the procedure described above.

An analogous procedure using transfinite interpolation is readily formulated for generating tetrahedral grids in regions with shapes similar to that of a tetrahedron.

The approach for generating triangular or tetrahedral meshes described above can be extended to include grid adaptation by adding to the scheme an intermediate domain and intermediate transformation $\boldsymbol{q}(\boldsymbol{\xi})$, as illustrated in Fig. 5.6, and special blending functions, as in the case of generating hexahedral (or quadrilateral) grids. Here, an adaptive triangular grid is generated through the composition of the transformations $\boldsymbol{q}(\boldsymbol{\xi})$ and $\boldsymbol{x}(\boldsymbol{q})$, where $\boldsymbol{q}(\boldsymbol{\xi})$ is an intermediate mapping providing grid adaptation and $\boldsymbol{x}(\boldsymbol{q})$ is an algebraic transformation.

Note that the procedure described above for generating triangular grids (tetrahedral and prismatic ones in three dimensions) can be realized analogously in other techniques based on coordinate transformations from the unit cube.

5.7 Drag and Sweeping Methods

Through the methods described above, the interior grid nodes are obtained by interpolating the grid points from all boundary segments. For domains of relatively simple geometries, the grid nodes may be obtained by dragging or sweeping the grid nodes of one boundary section. Suppose these nodes are $\boldsymbol{x}_i$, $i = 1, \dots, N$; then, both the interior nodes and another boundary grid nodes $\boldsymbol{x}_i^j$, $i = 1, \dots, N,\ j = 1, \dots, M$, may be defined by the formula

$$\boldsymbol{x}_i^j = \boldsymbol{x}_i^{j-1} + \boldsymbol{v}_i^j, \quad \boldsymbol{x}_i^0 = \boldsymbol{x}_i,$$

with specified incremental vectors $\boldsymbol{v}_i^j$. If the vectors $\boldsymbol{v}_i^j$ are constant, then this algorithm is referred to as a drag method. However, the vectors $\boldsymbol{v}_i^j$ may be different, in which case the approach is referred to as a sweeping method. These methods were developed at the early stage of mesh generation by Park and Washam (1979).

5.8 Comments

The standard formulas of multivariate transfinite interpolation using Boolean operations were described by Gordon (1969, 1971), although a two-dimensional interpolation formula with the simplest blending functions for the construction of the boundaries of hexahedral patches from CAD data was proposed by Coons (1967) and Ahuja and Coons (1968). The construction of coordinate transformations through the formulas of transfinite interpolation was formulated by Gordon and Hall (1973) and Gordon and Thiel (1982). The Hermite interpolation was presented by Smith (1982).

The multisurface method was described by Eiseman (1980) and was, in its original form, a univariate formula for grid generation based on the specification of two boundary surfaces and an arbitrary number of interior control surfaces. The blending functions were implicitly derived from global and/or local interpolants which result from an expression for the tangential derivative spanning between the exterior boundary surfaces. The multisurface transformation can be described in the context of transfinite interpolation.

A two-boundary technique was introduced by Smith (1981). It is based on the description of two opposite boundary surfaces, tangential derivatives on the boundary surfaces which are used to compute normal derivatives, and Hermite cubic blending functions.

The construction of some special blending functions aimed at grid clustering at boundaries was performed by Eriksson (1982) and Smith and Eriksson (1987). A detailed description of various forms of blending functions with the help of splines was presented in a monograph by Thompson et al. (1985).

The procedures described above for generating smooth blending functions and algebraic triangulations were developed by Liseikin (1999).

References

Ahuja, D. V., & Coons, S. A. (1968). Geometry for construction and display. *IBM Systems Journal*, *7*, 188–205.

Coons, S. A. (1967). Surfaces for computer aided design of space forms. Project MAC, Design Division Department of Mechanical Engineering, MII, 1964; Revised to MAC-TR-41, June, 1967.

Eiseman, P. R. (1980). Geometric methods in computational fluid dynamics. ICASE Report 80-11 and Von Karman Institute for Fluid Dynamics Lecture Series Notes.

Eriksson, L. E. (1982). Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation. *AIAA Journal*, *20*, 1313–1320.

Gordon, W. J. (1969). Distributive lattices and the approximation of multivatiate functions. In I. J. Shwenberg (Ed.), *Symposium on approximation with special emphasis on spline functions* (pp. 223–277). Madison: Academic Press.

Gordon, W. J. (1971). Blending-function methods of bivariate and multivariate interpolation and approximation. *SIAM Journal on Numerical Analysis*, *8*, 158–177.

Gordon, W. J., & Hall, C. A. (1973). Construction of curvilinear coordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, *7*, 461–477.

Gordon, W. J., & Thiel, L. C. (1982). Transfinite mappings and their application to grid generation. In J. F. Thompson (Ed.), *Numerical grid generation* (pp. 171–192). New York: North-Holland.

Liseikin, V. D. (1999). *Grid generation methods*. Berlin: Springer.

Park, S., & Washam, C. J. (1979). Drag method as a finite element mesh generation sheme. *Composite Structures*, *10*(1–2), 343–346.

Smith, R. E. (1981). Two-boundary grid generation for the solution of the three-dimensional Navier–Stokes equations. NASA TM-83123.

Smith, R. E. (1982). Algebraic grid generation. In J. F. Thompson (Ed.), *Numerical grid generation* (pp. 137–170). New York: North-Holland.

Smith, R. E., & Eriksson, L. E. (1987). Algebraic grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 285–300.

Thompson, J. F., Warsi, Z. U. A., & Mastin, C. W. (1985). *Numerical grid generation foundations and applications*. New York: North-Holland.

Chapter 6
Grid Generation Through Differential Systems

6.1 Introduction

Grid techniques based on the use of systems of partial differential equations to derive coordinate transformations are very popular in mapping approaches for generating grids. The choice of the systems of equations relies on numerical principles and careful analysis of the required properties of the equations. They must have intrinsic abilities to cope with complex geometries and to produce grids which are locally compressed by large factors compared with uniform grids. The equations should be computationally efficient, i.e. easy to model numerically and solve. Therefore, the task of formulating satisfactory grid equations is not simple.

The present chapter describes the most typical systems of equations for grid generation in a physical domain X^n: elliptic, hyperbolic, and parabolic.

6.2 Elliptic Equations

At present, elliptic methods of grid generation have widespread applications. The formulation of an elliptic method for grid generation relies on the utilization of an elliptic system whose solution defines a coordinate transformation

$$\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n, \quad \boldsymbol{x}(\boldsymbol{\xi}) = [x^1(\boldsymbol{\xi}), \dots, x^n(\boldsymbol{\xi})], \quad \boldsymbol{\xi} = (\xi^1, \dots, \xi^n), \quad \boldsymbol{\xi} \in \Xi^n \tag{6.1}$$

from the computational domain $\Xi^n \subset R^n$ specified by the user onto the physical one $X^n \subset R^n$. The values of the vector-valued function $\boldsymbol{x}(\boldsymbol{\xi})$ at the points of a reference grid in Ξ^n define the nodes of the elliptic grid in X^n (Fig. 13.2 for $n = 3$, $X^3 \equiv S^3$). However, in practice, the grid nodes are obtained through the numerical solution of a boundary value problem for the elliptic system on the reference grid in the domain Ξ^n (Fig. 6.1).

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_6

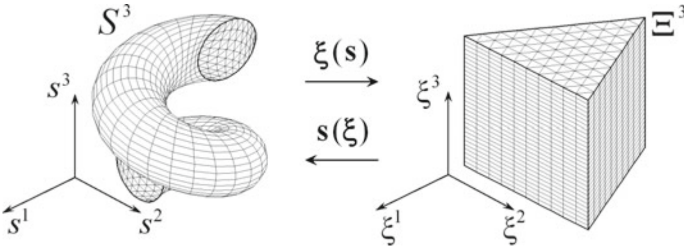


Fig. 6.1 Prismatic grid generated by a mapping approach

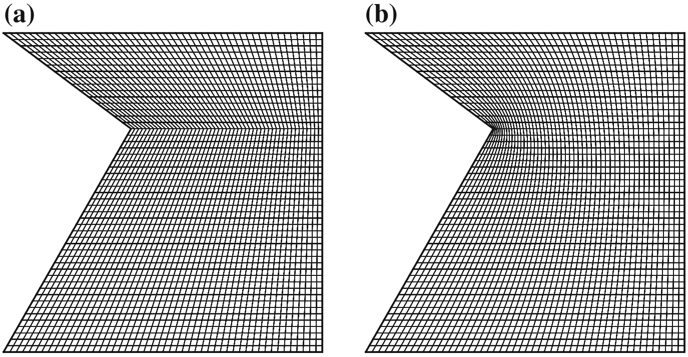


Fig. 6.2 Two-dimensional grids: **a** – grid obtained by means of transfinite interpolation; **b** – grid obtained by means of elliptic equations (inverted Laplace equations)

Elliptic equations are attractive for generating curvilinear coordinates because of some of their properties. First, elliptic equations which obey the extremum principle, i.e. the extrema of solutions cannot be within the domain, are readily formulated and numerically implemented. With this property, there is less tendency for folding of the resulting grid cells. Another important property of any elliptic system is the inherent smoothness of its solution, and consequently of the resulting coordinate curves in the interior of the domain and even on smooth segments of the boundary. Moreover, the smoothness can be propagated over the whole boundary with slope discontinuities if the boundary conditions are consistent with the equations of the elliptic system (Fig. 6.2 for $n = 2$). The third advantageous feature of elliptic systems is that they allow one to specify the coordinate points (and/or coordinate-line slopes) on the whole boundary of the domain. Finally, well-established methods are available to solve elliptic equations.

A disadvantage of elliptic systems is the cost of the numerical solution, especially because the commonly applied equations, considered in the transformed space, are nonlinear and require iteration.

Commonly, the elliptic equations for generating curvilinear coordinates are formulated in two ways:

(1) in the computational domain Ξ^n with the physical Cartesian coordinates x^i as the dependent variables;
(2) in the physical domain X^n with the curvilinear coordinates ξ^i as the dependent variables.

In the second way, the coordinate mapping $\boldsymbol{x}(\boldsymbol{\xi})$ is constructed by solving the elliptic system obtained by a transformation of the original system so as to interchange the dependent and independent variables.

The initial elliptic systems for the generation of grids are generally chosen in the form

$$L_1(x^i) \equiv a_1^{kj} \frac{\partial^2 x^i}{\partial \xi^k \partial \xi^j} + b_1^j \frac{\partial x^i}{\partial \xi^j} + c_1 x^i = f_1^i \ , \quad i, j, k = 1, \dots, n \ , \tag{6.2}$$

and

$$L_2(\xi^i) \equiv a_2^{kj} \frac{\partial^2 \xi^i}{\partial x^k \partial x^j} + b_2^j \frac{\partial \xi^i}{\partial x^j} + c_2 \xi^i = f_2^i \ , \quad i, j, k = 1, \dots, n \ . \tag{6.3}$$

Recall that repeated indices in formulas mean a summation over them unless otherwise noted. The condition of ellipticity puts a restriction on the coefficients a_l^{ij} in (6.2) and (6.3):

$$a_l^{ij} b^i b^j \geq c_l b^k b^k \ , \qquad c_l > 0 \ , \qquad i, j, k = 1, \dots, n \ , \qquad l = 1, 2 \ ,$$

for an arbitrary vector $\boldsymbol{b} = (b^1, \dots, b^n)$.

6.2.1 Laplace Systems

The most simple elliptic systems for generating grids are represented by the uncoupled Laplace equations, either in the computational domain,

$$\nabla^2 x^i \equiv \frac{\partial}{\partial \xi^j} \frac{\partial x^i}{\partial \xi^j} = 0 \ , \qquad i, j = 1, \dots, n \ , \tag{6.4}$$

with the dependent variables x^i, or in the physical domain,

$$\nabla^2 \xi^i \equiv \frac{\partial}{\partial x^j} \frac{\partial \xi^i}{\partial x^j} = 0 \ , \qquad i, j = 1, \dots, n \ , \tag{6.5}$$

with the dependent variables ξ^i.

Multiplying the system (6.5) by $\partial x^p / \partial \xi^i$ and summing over i, we readily obtain the inverse Laplace system with the dependent and independent quantities interchanged, in the form

$$g^{ij}\frac{\partial^2 x^p}{\partial \xi^i \partial \xi^j} = 0\ , \qquad i, j, p = 1, \ldots, n\ , \tag{6.6}$$

where, in accordance with (2.18),

$$g^{ij} = \nabla \xi^i \cdot \nabla \xi^j = \frac{\partial \xi^i}{\partial x^k}\frac{\partial \xi^j}{\partial x^k}\ , \qquad i, j, k = 1, \ldots, n\ , \tag{6.7}$$

is the (ij)th contravariant metric element of X^n in the coordinates $\xi_1, \ldots, \xi_n$. These equations shape the coupled quasilinear elliptic system in the computational domain Ξ^n. A grid in the domain X^n is generated by solving (6.4) or (6.6), with the Cartesian values of $\boldsymbol{x}$ on the physical boundaries used as the boundary conditions along the corresponding boundary segments of Ξ^n.

The maximum principle is valid for the Laplace equations (6.4) and (6.5). In the case of the system (6.4), it guarantees that the image of Ξ^n produced by the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ will be contained in X^n if the domain X^n is convex. Analogously, the image of X^n produced by the transformation $\boldsymbol{\xi}(\boldsymbol{x})$ satisfying the Laplace system (6.5) will be contained in Ξ^n if Ξ^n is a convex domain. In the latter case, the restriction of convexity is not imposed on the physical domain X^n. As the convex computational domain Ξ^n can be specified by the user, the system (6.5), and correspondingly (6.6), has been more consistently favored in applications than the system (6.4) for generating grids in general regions.

The Eq. (6.4) seem to be more natural than the equations in (6.6) for the implementation into numerical codes, as they are linear and of divergent form with respect to the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$. However, such a divergent model, owing to the maximum principle, does not guarantee that all grid points will be inside of the physical domain X^n when it is not convex (Fig. 6.3, left-hand), let alone that the grid cells may be folded.

The considerations mentioned are the major reasons why the formulation of grid systems is reasonable to make with respect to the function $\boldsymbol{\xi}(\mathbf{x}) : X^n \to \Xi^n$ that is the inverse of the intermediate transformation $\mathbf{x}(\boldsymbol{\xi}) : \Xi^n \to X^n$.

The Eq. (6.5) are also preferred because the physical-space formulation provides direct control of grid spacing and orthogonality. For these reasons, the formulation of many other elliptic grid generators is also commonly performed in terms of the inverse of the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$.

The main problem in grid generation is to make the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^n \to X^n$ a diffeomorphism, i.e. one-to-one mapping with the Jacobian J not vanishing. In the case $n = 2$, the mathematical foundation of the technique, based on the Laplace system (6.5) with a convex computational domain Ξ^2, is solid. It is founded on the following result, derived from a theorem by Rado.

Let X^2 be a simply connected bounded domain in R^2. In this case, the Jacobian of the transformation $\boldsymbol{\xi}(\boldsymbol{x})$ generated by the system (6.5) does not vanish in the interior of X^2, if Ξ^2 is a rectangle and $\boldsymbol{\xi}(\boldsymbol{x}) : \partial X^2 \to \partial \Xi^2$ is a homeomorphism. Note, in the case $n > 2$, this property may be breached (see Farrel and Jones 1996).

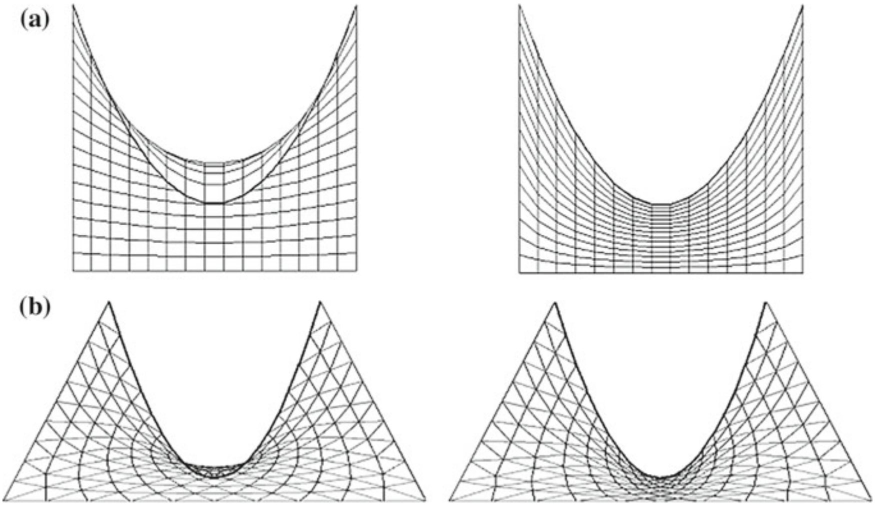


Fig. 6.3 Quadrilateral (**a**) and triangular (**b**) grids in concave domains generated by the solution of equations (6.4) (*left*) and by the solution of equations (6.6) (*right*)

The system (6.5) for $n = 2$ was introduced by Crowley (1962) and Winslow (1967), and owing to the properties noted above, it has been the most widely used system for generating fixed coordinate grids in general regions.

Some features of the coordinate transformations and corresponding grids derived from the system (6.5) and, correspondingly, (6.6) are considered in the next two subsections.

Two-Dimensional Equations

In this subsection, we discuss the qualitative behavior of the coordinate lines generated by the two-dimensional Laplace system (6.5) near the boundary curves. We assume that Ξ^2 is a unit square, X^2 is a simply connected bounded domain, and the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is defined as a solution to the Dirichlet problem for the system (6.6) with a specified one-to-one boundary transformation

$$\boldsymbol{x}(\boldsymbol{\xi}) : \partial\Xi^2 \to \partial X^2 .$$

It is obvious from the theorem above that the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ is the inverse of the transformation $\boldsymbol{\xi}(\boldsymbol{x})$, that is, a solution to the Laplace system (6.5) with the Dirichlet boundary conditions

$$\boldsymbol{\xi}(\boldsymbol{x}) : \partial X^2 \to \partial\Xi^2 .$$

From (2.21), the two-dimensional contravariant metric elements g^{ij} in (6.6) are connected with the covariant elements $g_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j}$ by the relation

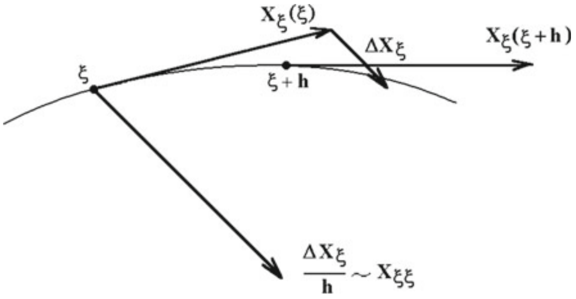


Fig. 6.4 Direction of the derivative of the tangential vector

$$g^{ij} = (-1)^{i+j} \frac{g_{3-i\ 3-j}}{g} , \qquad i = 1, 2 ,$$

with fixed i and j. Therefore, the system (6.6) for $n = 2$ is equivalent to

$$g_{22} \frac{\partial^2 x^i}{\partial \xi^1 \partial \xi^1} - 2g_{12} \frac{\partial^2 x^i}{\partial \xi^1 \partial \xi^2} + g_{11} \frac{\partial^2 x^i}{\partial \xi^2 \partial \xi^2} = 0 , \qquad i = 1, 2 . \tag{6.8}$$

We now demonstrate that the spacing between coordinate lines, say $\xi^2 =$ const., in the vicinity of the respective boundary curve $\xi^2 = \xi^2_0$, increases toward it if the boundary line is convex and, conversely, the spacing decreases when the boundary line is concave.

Let us consider, for clarity, a family of the coordinate curves $\xi^2 =$ const. Then, the boundary curve of this family is defined by the relation $\xi^2 = \xi^2_0$ with $\xi^2_0 = 0$ or $\xi^2_0 = 1$.

First, we note that the vector $\boldsymbol{x}_{\xi^1\xi^1}$, which is the derivative with respect to ξ^1 of the tangential vector $\boldsymbol{x}_{\xi^1}$, $\boldsymbol{x} = (x^1, x^2)$, is directed, as shown in Fig. 6.4, toward the concavity of the coordinate line $\xi^2 = \xi^2_0$. Another important gradient vector of $\xi^2(\boldsymbol{x})$,

$$\nabla \xi^2 = \Bigl(\frac{\partial \xi^2}{\partial x^1}, \frac{\partial \xi^2}{\partial x^2} \Bigr) ,$$

is orthogonal to the tangent vector $\boldsymbol{x}_{\xi^1}$. The dot product of the vector $\nabla \xi^2$ and the tangential vector $\boldsymbol{x}_{\xi^2}$ equals 1. Therefore, these vectors are always directed to one side of the line $\xi^2 = \xi^2_0$ (see Figs. 2.2 and 6.4); in particular, they are directed into the domain X^2 if this coordinate line is the boundary curve $\xi^2 = 0$. Thus, the sign of the quantity

$$Q = \nabla \xi^2 \cdot \boldsymbol{x}_{\xi^1\xi^1} = \Gamma^2_{11}$$

can serve as a criterion of the local shape of the boundary $\xi^2 = 0$. Namely, if $Q < 0$, which means the vectors $\boldsymbol{x}_{\xi^1\xi^1}$ and $\nabla \xi^2$ are directed toward different sides of the coordinate curve $\xi^2 = 0$, then the domain X^2 is concave (if $Q > 0$ the domain is convex) near that part of the boundary $\xi^2 = 0$ where this inequality is satisfied (see Fig. 6.5).

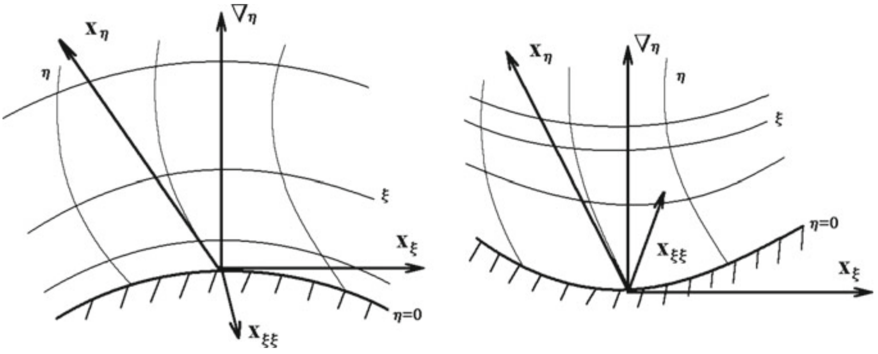


Fig. 6.5 Grid concentration near a concave boundary curve (*left*) and grid rarefaction near a convex part of the boundary (*right*)

We have

$$\begin{aligned} Q &= \boldsymbol{x}_{\xi^1\xi^1} \cdot \nabla\xi^2 = -\boldsymbol{x}_{\xi^1} \cdot \frac{\partial}{\partial\xi^1}(\nabla\xi^2) \\ &= -\Big(\frac{\partial x^1}{\partial\xi^1}\Big)^2 \frac{\partial^2\xi^2}{\partial x^1\partial x^1} - 2\frac{\partial x^1}{\partial\xi^1}\frac{\partial x^2}{\partial\xi^1}\frac{\partial^2\xi^2}{\partial x^1\partial x^2} - \Big(\frac{\partial x^2}{\partial\xi^1}\Big)^2 \frac{\partial^2\xi^2}{\partial x^2\partial x^2} \ . \end{aligned} \tag{6.9}$$

The vector $\boldsymbol{n} = \nabla\xi^2/|\nabla\xi^2|$ is the unit normal to the tangential vector $\boldsymbol{x}_{\xi^1}$. It is valid that

$$\boldsymbol{n} \cdot \boldsymbol{x}_{\xi^2} = 1/|\nabla\xi^2| = 1/\sqrt{g^{22}} \ ,$$

where

$$g^{22} = \nabla\xi^2 \cdot \nabla\xi^2 = \frac{\partial\xi^2}{\partial x^i}\frac{\partial\xi^2}{\partial x^i} \ , \qquad i = 1, 2 \ .$$

Let us denote by l_h the distance between the two coordinate lines $\xi^2 = 0$ and $\xi^2 = h$. Using the above equation for $\boldsymbol{n} \cdot \boldsymbol{x}_{\xi^2}$, we have

$$l_h = (\boldsymbol{n} \cdot \boldsymbol{x}_{\xi^2})h + O(h)^2 = h/\sqrt{g^{22}} + O(h)^2 \ .$$

So, the quantity

$$s_2 = 1/\sqrt{g^{22}}$$

reflects the relative spacing between the coordinate grid lines $\xi^2 = \text{const}$.

The vector $\boldsymbol{n}$ is orthogonal to the boundary coordinate line $\xi^2 = 0$, and therefore the rate of change of the relative spacing s_2 of the coordinate curves $\xi^2 = \text{const}$ near this boundary line is computed in the $\boldsymbol{n}$ direction. Since

$$\boldsymbol{n} = \frac{1}{\sqrt{g^{22}}}\Big(\frac{\partial \xi^2}{\partial x^1}, \ \frac{\partial \xi^2}{\partial x^2}\Big) \ ,$$

we obtain

$$\begin{aligned}\frac{\partial s_2}{\partial \boldsymbol{n}} &= \frac{1}{\sqrt{g^{22}}}\Big(\frac{\partial s_2}{\partial x^1}\frac{\partial \xi^2}{\partial x^1} + \frac{\partial s_2}{\partial x^2}\frac{\partial \xi^2}{\partial x^2}\Big)\\ &= -\frac{1}{2(g^{22})^2}\Big(\Big(\frac{\partial \xi^2}{\partial x^1}\Big)^2\frac{\partial^2 \xi^2}{\partial x^1 \partial x^1} + 2\frac{\partial \xi^2}{\partial x^1}\frac{\partial \xi^2}{\partial x^2}\frac{\partial^2 \xi^2}{\partial x^1 \partial x^2}\\ &\quad + \Big(\frac{\partial \xi^2}{\partial x^2}\Big)^2\frac{\partial^2 \xi^2}{\partial x^2 \partial x^2}\Big) \ .\end{aligned}$$

Using in this equation the relation

$$\frac{\partial \xi^i}{\partial x^j} = (-1)^{i+j}\frac{\partial x^{3-j}}{\partial \xi^{3-i}}\Big/ J \ , \qquad i, j, = 1, 2 \ ,$$

with fixed i and j, we obtain for the rate of change of the relative spacing s_2

$$\begin{aligned}\frac{\partial s_2}{\partial \boldsymbol{n}} = -\frac{1}{2J^2(g^{22})^2}\Big[\Big(\frac{\partial x^2}{\partial \xi^1}\Big)^2\frac{\partial^2 \xi^2}{\partial x^1 \partial x^1} - 2\frac{\partial x^2}{\partial \xi^1}\frac{\partial x^1}{\partial \xi^1}\frac{\partial^2 \xi^2}{\partial x^1 \partial x^2}\\ + \Big(\frac{\partial x^1}{\partial \xi^1}\Big)^2\frac{\partial^2 \xi^2}{\partial x^2 \partial x^2}\Big] \ . \qquad (6.10)\end{aligned}$$

Equation (6.5) for $n = 2$ implies

$$\frac{\partial^2 \xi^2}{\partial x^2 \partial x^2} = -\frac{\partial^2 \xi^2}{\partial x^1 \partial x^1} \ ,$$

and therefore we see, from (6.9) and (6.10), that

$$\frac{\partial s_2}{\partial \boldsymbol{n}} = -\frac{1}{2J^2(g^{22})^2} Q \ .$$

Thus, we find that the quantities $\partial s_2/\partial \boldsymbol{n}$ and Q have different signs. So, if the boundary line $\xi^2 = 0$ is convex or concave at some point $\boldsymbol{\xi}_0$, i.e. $Q > 0$ or $Q < 0$, respectively, then

$$\frac{\partial s_2}{\partial \boldsymbol{n}} < 0 \quad \text{or} \quad \frac{\partial s_2}{\partial \boldsymbol{n}} > 0 \ ,$$

at this point. These inequalities mean that the spacing of the grid lines $\xi^2 = \text{const}$ decreases or increases, respectively, from the boundary curve $\xi^2 = 0$ in the vicinity of the boundary point $\boldsymbol{\xi}_0$.

Analogous computations are readily carried out for the coordinates near the boundary curves $\xi^2 = 1$ and $\xi^1 = 0$ or $\xi^1 = 1$, which result in the conclusion that the grid lines obtained from the Laplace system (6.5) for $n = 2$ are attracted to the concave part of the boundary and repelled near the convex part (Fig. 6.5).

Three-Dimensional Equations

In contrast to two-dimensional domains, the problem of generating one-to-one three-dimensional transformations through the Laplace system (6.5) has not yet been solved theoretically. One of the reasons is the fact that the technique used for the two-dimensional case cannot extended to higher dimensions. This observation was made by Liao (1991). However, we assume that the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ obtained as a solution to the Dirichlet problem for the system (6.6) with $n = 3$ on the unit cube Ξ^3 is a diffeomorphism, and hence the inverse transformation $\boldsymbol{\xi}(\boldsymbol{x})$ is a solution to the Laplace system (6.5). In this case, the analogous property of the concentration of the coordinate surfaces toward the concave part of the boundary and their rarefaction toward the convex part is valid. This subsection gives a detailed proof of this fact.

First, we note that in the three-dimensional case, the gradient vector

$$\nabla\xi^3 = \Big(\frac{\partial\xi^3}{\partial x^1}, \frac{\partial\xi^3}{\partial x^2}, \frac{\partial\xi^3}{\partial x^3}\Big)$$

is orthogonal to the tangent vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$, $\boldsymbol{x} = (x^1, x^2, x^3)$. The vectors $\nabla\xi^3$ and $\boldsymbol{x}_{\xi^3}$ are directed toward one side of the coordinate surface $\xi^3 = \xi^3_0$. And the quantity

$$s_3 = 1/\sqrt{g^{33}}\ ,$$

where, in accordance with (6.7) for $n = 3$,

$$g^{33} = \nabla\xi^3 \cdot \nabla\xi^3 = \frac{\partial\xi^3}{\partial x^i}\frac{\partial\xi^3}{\partial x^i}\ , \qquad i = 1, 2, 3\ ,$$

means, as in the two-dimensional case, the relative grid spacing between the coordinate surfaces $\xi^3 = \text{const}$ in the normal direction $\boldsymbol{n}$, where

$$\boldsymbol{n} = \nabla\xi^3/|\nabla\xi^3| = \nabla\xi^3/\sqrt{g^{33}}\ .$$

The rate of change of the relative spacing in this direction $\boldsymbol{n}$ equals

$$\begin{aligned}\frac{\partial s_3}{\partial \boldsymbol{n}} &= \frac{1}{\sqrt{g^{33}}}\Big(\frac{\partial s_3}{\partial x^1}\frac{\partial\xi^3}{\partial x^1} + \frac{\partial s_3}{\partial x^2}\frac{\partial\xi^3}{\partial x^2} + \frac{\partial s_3}{\partial x^3}\frac{\partial\xi^3}{\partial x^3}\Big)\\ &= -\frac{1}{2(g^{33})^2}\Big(\frac{\partial\xi^3}{\partial x^i}\frac{\partial\xi^3}{\partial x^j}\frac{\partial^2\xi^3}{\partial x^i\partial x^j}\Big)\ , \qquad i, j = 1, 2, 3\ . \qquad (6.11)\end{aligned}$$

Using the general identity (2.49),

$$\frac{\partial^2 \xi^i}{\partial x^k \partial x^m} = -\frac{\partial^2 x^p}{\partial \xi^l \partial \xi^j} \frac{\partial \xi^j}{\partial x^k} \frac{\partial \xi^l}{\partial x^m} \frac{\partial \xi^i}{\partial x^p} \ ,$$

$$i, j, k, l, m, p = 1, 2, 3 \ ,$$

for $i = 3$, between the second derivatives of the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^3 \to X^3$ and $\boldsymbol{\xi}(\boldsymbol{x}) : X^3 \to \Xi^3$ in (6.11), we obtain

$$\frac{\partial s_3}{\partial \boldsymbol{n}} = \frac{1}{2(g^{33})^2} g^{3j} g^{3l} \frac{\partial^2 x^p}{\partial \xi^l \partial \xi^j} \frac{\partial \xi^3}{\partial x^p} \ , \qquad j, l, p = 1, 2, 3 \ .$$

Now we write out the right-hand side of this equation as the sum of two parts, one of which contains all terms of the kind

$$\frac{\partial^2 x^p}{\partial \xi^l \partial \xi^j} \ , \qquad i, j = 1, 2 \ , \qquad p = 1, 2, 3 \ ,$$

namely,

$$\begin{aligned}
\frac{\partial s_3}{\partial \boldsymbol{n}} &= \frac{1}{2(g^{33})^2} (Q_1 + Q_2) \ , \\
Q_1 &= \Big((g^{31})^2 \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^1} + 2g^{31} g^{32} \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^2} + (g^{32})^2 \frac{\partial^2 x^p}{\partial \xi^2 \partial \xi^2}\Big) \frac{\partial \xi^3}{\partial x^p} \ , \\
Q_2 &= g^{33}\Big(2g^{31} \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^3} + 2g^{32} \frac{\partial^2 x^p}{\partial \xi^2 \partial \xi^3} + g^{33} \frac{\partial^2 x^p}{\partial \xi^3 \partial \xi^3}\Big) \frac{\partial \xi^3}{\partial x^p} \ ,
\end{aligned} \tag{6.12}$$

where $p = 1, 2, 3$. Multiplying the elliptic system (6.6) for $n = 3$ by $\partial \xi^m / \partial x^p$ and summing the result over p, we obtain, for $m = 3$,

$$g^{ij} \frac{\partial^2 x^p}{\partial \xi^i \partial \xi^j} \frac{\partial \xi^3}{\partial x^p} = 0 \ , \qquad i, j, p = 1, 2, 3 \ .$$

Using this equation in the expression (6.12) for the quantity Q_2, we readily obtain

$$Q_2 = -g^{33}\Big(g^{11} \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^1} + 2g^{12} \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^2} + g^{22} \frac{\partial^2 x^p}{\partial \xi^2 \partial \xi^2}\Big) \frac{\partial \xi^3}{\partial x^p} \ , \ p = 1, 2, 3 \ .$$

Therefore,

$$\begin{aligned}
Q_1 + Q_2 = \Big(& [(g^{31})^2 - g^{33} g^{11}] \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^1} \\
& + [(g^{32})^2 - g^{33} g^{22}] \frac{\partial^2 x^p}{\partial \xi^2 \partial \xi^2} \\
& + 2[g^{12} g^{33} - g^{13} g^{23}] \frac{\partial^2 x^p}{\partial \xi^1 \partial \xi^2}\Big) \frac{\partial \xi^3}{\partial x^p} \ .
\end{aligned}$$

And, in accordance with the relation

$$g_{ij} = J^2(g^{i+1\ j+1}g^{i+2\ j+2} - g^{i+1\ j+2}g^{i+2\ j+1})\ , \qquad i, j = 1, 2, 3\ ,$$

from (2.22), where any superscript index k can be identified with $k \pm 3$, we have

$$Q_1 + Q_2 = -\frac{g_{3-i\ 3-j}}{J^2}\frac{\partial^2 x^p}{\partial\xi^i\partial\xi^j}\frac{\partial\xi^3}{\partial x^p}\ , \quad i, j = 1, 2\ , \quad p = 1, 2, 3\ . \tag{6.13}$$

Now we consider the value of $Q_1 + Q_2$ at the boundary surface $\xi^3 = 0$. Let $\boldsymbol{\xi}_0$ be a point at this surface. The derivative of the vector

$$\boldsymbol{b} = a_1\boldsymbol{x}_{\xi^1} + a_2\boldsymbol{x}_{\xi^2}\ , \qquad a_i = \text{const}\ , \qquad i = 1, 2\ ,$$

along the direction $\boldsymbol{t} = \boldsymbol{b}(\boldsymbol{\xi}_0)$ is the vector

$$\frac{\partial\boldsymbol{b}}{\partial\boldsymbol{t}} = (a_1)^2\boldsymbol{x}_{\xi^1\xi^1} + 2a_1a_2\boldsymbol{x}_{\xi^1\xi^2} + (a_2)^2\boldsymbol{x}_{\xi^2\xi^2}\ .$$

If $\boldsymbol{\xi}_0$ is a point of local convexity of the boundary surface $\xi^3 = 0$, then in analogy with the vector $\boldsymbol{x}_{\xi^1\xi^1}$ considered previously in the two-dimensional case, the vector $\partial\boldsymbol{b}/\partial\boldsymbol{t}(\boldsymbol{\xi}_0)$ is directed into the domain X^3. The vector $\nabla\xi^3$ at the point $\boldsymbol{\xi}_0$ and the vector $\boldsymbol{x}_{\xi^3}$ are also directed into the domain X^3. Therefore, the dot product of the vectors $\nabla\xi^3$ and $\partial\boldsymbol{b}/\partial\boldsymbol{t}$ is positive at the point under consideration, i.e.

$$\nabla\xi^3 \cdot \frac{\partial\boldsymbol{b}}{\partial\boldsymbol{t}} = \Big((a_1)^2\frac{\partial^2 x^p}{\partial\xi^1\partial\xi^1} + 2a_1a_2\frac{\partial^2 x^p}{\partial\xi^1\partial\xi^2} + (a_2)^2\frac{\partial^2 x^p}{\partial\xi^2\partial\xi^2}\Big)\frac{\partial\xi^3}{\partial x^p} > 0\ , \tag{6.14}$$

where $p = 1, 2, 3$. Considering the three cases

$$(a_1, a_2) = (1, 0), \qquad (a_1, a_2) = (0, 1), \qquad (a_1, a_2) = (1, 1)$$

in (6.14), we find that at the point $\boldsymbol{\xi}_0$,

$$\frac{\partial^2 x^p}{\partial\xi^1\partial\xi^1}\frac{\partial\xi^3}{\partial x^p} > 0\ , \qquad p = 1, 2, 3\ ,$$

$$\frac{\partial^2 x^p}{\partial\xi^2\partial\xi^2}\frac{\partial\xi^3}{\partial x^p} > 0\ , \qquad p = 1, 2, 3\ ,$$

and

$$\Big|\frac{\partial^2 x^p}{\partial\xi^1\partial\xi^2}\frac{\partial\xi^3}{\partial x^p}\Big| < \Big(\frac{\partial^2 x^p}{\partial\xi^1\partial\xi^1}\frac{\partial\xi^3}{\partial x^p}\ \frac{\partial^2 x^p}{\partial\xi^2\partial\xi^2}\frac{\partial\xi^3}{\partial x^p}\Big)^{1/2}\ , \qquad p = 1, 2, 3\ .$$

As the quadratic form $\{g_{ij}\}$, $i, j = 1, 2$, is positive, we obtain, from (6.13),

$$Q_1 + Q_2 < 0 ,$$

and, correspondingly, $\partial s_3/\partial \boldsymbol{n} < 0$, i.e. the spacing between the coordinate surfaces $\xi^3 = \text{const}$ decreases from the convex part of the boundary $\xi^3 = 0$.

Analogously, we have $Q_1 + Q_2 > 0$ at a point on a concave part of the boundary surface $\xi^3 = 0$, which implies the observation that the grid surface spacing increases locally from a concave part of the boundary surface.

The same facts are obviously true for the corresponding grid spacings near the boundary surfaces $\xi^3 = 1$, $\xi^i = 0$, and $\xi^i = 1$, $i = 1, 2$.

Thus, we find that the coordinate surfaces of the coordinate system derived from the Laplace equations (6.5) are clustered near the concave parts of the boundary and coarser near its convex parts (Figs. 6.6 and 6.7).

6.2.2 Poisson Systems

The Laplace system provides little opportunity to control the properties of the grid, in particular, to adapt the mesh to the geometry of the boundary or to the features of the solution of the physical equations in regions of the domain where this is necessary. Only one opportunity is given, by the specification of the boundary conditions. However, the grid point distribution on the boundaries noticeably affects only the disposition of the nearby interior grid nodes. The distribution of the nodes over most of the interior is influenced more by the form of the elliptic equations than by the boundary values.

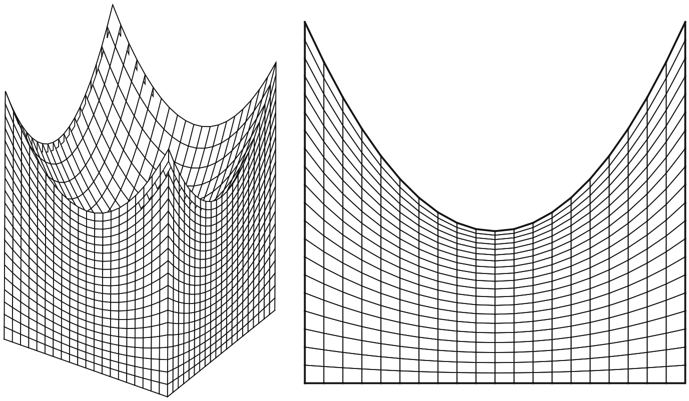


Fig. 6.6 Three-dimensional numerical grid with node clustering near a concave segment of the boundary (*left*), its section (*right*)

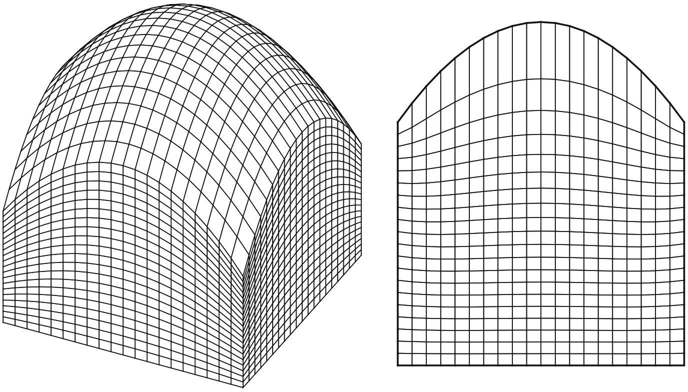


Fig. 6.7 Three-dimensional numerical grid with node rarefaction near a convex segment of the boundary (*left*), its section (*right*)

Therefore, in order to provide global control of the grid node distribution, the Laplace system is replaced by a more general elliptic system with variable coefficients. The simplest way to obtain such a generalization, suggested by Godunov and Prokopov (1972) for the generation of two-dimensional grids, consists of adding right-hand terms to the Laplace system (6.5), thus making it a Poisson system.

The actual generation of the grid is done through the numerical solution of the inverted Poisson system in the computational domain Ξ^n, where the curvilinear coordinates ξ^i are the independent variables and the Cartesian coordinates x^i are the dependent variables.

An elliptic method of grid generation based on the numerical solution of a system of inverted Poisson equations is being used for a broad range of practical applications. The method allows the users to generate numerical grids in fairly complicated domains and on surfaces that arise while analyzing multidimensional fluid-flow problems. Practically all big grid generation codes incorporate it as a basic tool for generating structured grids. Other techniques (algebraic, hyperbolic, etc.) play an auxiliary role in the codes, serving as an initial guess for the elliptic solver, or as a technique for generating grids in regions with simple geometry.

Formulation of the System

The system of Poisson equations for generating grids has the form

$$\nabla^2 \xi^i \equiv \frac{\partial}{\partial x^j}\frac{\partial \xi^i}{\partial x^j} = P^i \,, \qquad i, j = 1, \ldots, n \,. \tag{6.15}$$

The quantities P^i are called either control functions or source terms. The source terms are essential for providing an effective control of the grid point distribution, although the choice of the proper control functions P^i is difficult, especially for multicomponent geometries.

Since

$$\frac{\partial}{\partial x^j}\left(\frac{\partial \xi^i}{\partial x^j}\right)\frac{\partial x^k}{\partial \xi^i} = -\frac{\partial^2 x^k}{\partial \xi^i \partial \xi^m}\frac{\partial \xi^i}{\partial x^j}\frac{\partial \xi^m}{\partial x^j} = -g^{im}\frac{\partial^2 x^k}{\partial \xi^i \partial \xi^m} ,$$

by multiplying the Poisson system (6.15) by $\partial x^k/\partial \xi^i$ and summing over i, an inverted system of the equations is obtained:

$$g^{ij}\frac{\partial^2 x^k}{\partial \xi^i \partial \xi^j} = -P_i \frac{\partial x^k}{\partial \xi^i} , \qquad k, i, j = 1, \dots, n . \tag{6.16}$$

Note that the left-hand part of these equations comprises the left-hand part of the system of inverted Laplace equations (6.6). The system (6.16) can also be represented in the following vector notation:

$$g^{ij}\boldsymbol{x}_{\xi^i \xi^j} = -P^i \boldsymbol{x}_{\xi^i} , \qquad i, j = 1, \dots, n . \tag{6.17}$$

For one-dimensional space, we obtain, from (6.17),

$$\frac{\mathrm{d}^2 x}{\mathrm{d}\xi^2} = -P\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^3 .$$

Assuming

$$P = \left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2 \frac{\partial w}{\partial \xi}\Big/ w ,$$

where w is some positive function, playing the role of a weight in applications, we have

$$w\frac{\mathrm{d}}{\mathrm{d}\xi}\left(w\frac{\mathrm{d}x}{\mathrm{d}\xi}\right) = 0 .$$

This equation is related to the equation in (4.23), thus giving a clue as to how to generate univariate grid clustering with the control function P.

Using the relations (2.21) and (2.22), we obtain, from (6.17), two- and three-dimensional systems of inverted Poisson grid equations:

$$B_n[\boldsymbol{x}](\boldsymbol{\xi}) = -gP^i \boldsymbol{x}_{\xi^i} , \quad i = 1, \dots, n , \tag{6.18}$$

where

$$g = \det\{g_{ij}\} , \quad B_n[y] = gg^{ij}\frac{\partial^2 y}{\partial \xi^i \partial \xi^j} , \quad i, j = 1, \dots, n ,$$

i.e. for $n = 2$ and $n = 3$

$$\begin{aligned}
B_2[y] &= g_{22}\frac{\partial^2 y}{\partial\xi^1\partial\xi^1} - 2g_{12}\frac{\partial^2 y}{\partial\xi^1\partial\xi^2} + g_{11}\frac{\partial^2 y}{\partial\xi^2\partial\xi^2}\ ,\\
B_3[y] &= [g_{22}g_{33} - (g_{23})^2]\frac{\partial^2 y}{\partial\xi^1\partial\xi^1} + [g_{11}g_{33} - (g_{13})^2]\frac{\partial^2 y}{\partial\xi^2\partial\xi^2}\\
&\quad + [g_{11}g_{22} - (g_{12})^2]\frac{\partial^2 y}{\partial\xi^3\partial\xi^3} + 2[g_{23}g_{13} - g_{12}g_{33}]\frac{\partial^2 y}{\partial\xi^1\partial\xi^2}\\
&\quad + 2[g_{12}g_{23} - g_{22}g_{13}]\frac{\partial^2 y}{\partial\xi^1\partial\xi^3} + 2[g_{13}g_{12} - g_{23}g_{11}]\frac{\partial^2 y}{\partial\xi^2\partial\xi^3}\ .
\end{aligned} \tag{6.19}$$

Justification for the Poisson System

The idea of using the Poisson system to provide efficient control of grid generation was justified by the fact that the system of the Poisson type is obtained from the Laplace system, for intermediate coordinates which are transformed to other coordinates. Let every component $q^i,\ i = 1,\dots,n$, of the coordinate transformation $\boldsymbol{q}(\boldsymbol{x})$ satisfy the Laplace equation

$$\nabla^2 q^i = \frac{\partial}{\partial x^j}\Big(\frac{\partial q^i}{\partial x^j}\Big) = 0\ , \qquad i, j = 1,\dots,n\ .$$

Furthermore, let $\boldsymbol{\xi}(\boldsymbol{q})$ be a new intermediate one-to-one smooth coordinate transformation. Then, every new coordinate ξ^i will satisfy the inhomogeneous elliptic system

$$\begin{aligned}
\nabla^2\xi^i &= \frac{\partial}{\partial x^j}\Big(\frac{\partial\xi^i}{\partial q^k}\frac{\partial q^k}{\partial x^j}\Big) = \frac{\partial^2\xi^i}{\partial q^k\partial q^m}\overline{g}^{km} + \frac{\partial\xi^i}{\partial q^k}\frac{\partial^2 q^k}{\partial x^j\partial x^j}\\
&= \overline{g}^{km}\frac{\partial^2\xi^i}{\partial q^k\partial q^m}\ , \qquad i, j, k, m = 1,\dots,n\ ,
\end{aligned} \tag{6.20}$$

where $\overline{g}^{km}$ is the (k, m) element of the contravariant metric tensor of the domain X^n in the coordinates $q^1,\dots,q^n$, i.e.

$$\overline{g}^{km} = \frac{\partial q^k}{\partial x^j}\frac{\partial q^m}{\partial x^j}\ , \qquad j, k, m = 1,\dots,n\ .$$

The elements $g^{ij},\ i, j = 1,\dots,n$, of the contravariant metric tensor of the domain X^n in the coordinates $\xi^1,\dots,\xi^n$ are connected with $\overline{g}^{ij},\ i, j = 1,\dots,n$, by

$$\overline{g}^{km} = g^{lj}\frac{\partial q^k}{\partial\xi^l}\frac{\partial q^m}{\partial\xi^j}\ , \qquad k, l, m, j = 1,\dots,n\ .$$

Thus, taking into account this relation, the system (6.20) has the form (6.15), i.e.

$$\nabla^2\,\xi^i = P^i\ , \qquad i = 1,\dots,n\ ,$$

where

$$P^i = g^{lj} \frac{\partial q^k}{\partial \xi^l} \frac{\partial q^m}{\partial \xi^j} \frac{\partial^2 \xi^i}{\partial q^k \partial q^m} , \qquad i, j, k, l, m = 1, \dots, n . \tag{6.21}$$

From the identity

$$\frac{\partial q^k}{\partial \xi^l} \frac{\partial q^m}{\partial \xi^j} \frac{\partial^2 \xi^i}{\partial q^k \partial qm} \equiv -\frac{\partial \xi^i}{\partial q^m} \frac{\partial^2 q^m}{\partial \xi^l \partial \xi^j},$$

we also have

$$P^i = -g^{lj} \frac{\partial \xi^i}{\partial q^m} \frac{\partial^2 q^m}{\partial \xi^l \partial \xi^j} . \tag{6.22}$$

Thus, by applying the intermediate coordinate transformation $\boldsymbol{q}(\boldsymbol{\xi})$ to a grid generated as a solution of the Laplace system, we obtain a grid which could have been generated directly as the solution of the Poisson system (6.15) with the appropriate control functions defined by (6.21) and (6.22).

The general Poisson system (6.15) does not obey the maximum principle. And, in contrast to the Laplace system (6.5) for $n = 2$, there is no guarantee that the generated grid is not folded. In fact, any smooth but folded coordinate transformation $\boldsymbol{\xi}(\boldsymbol{x})$ can be obtained from the system (6.15) by computing P^i directly from the Laplacian of $\xi^i(\boldsymbol{x})$. If these P^i are used in the Poisson system (6.15), then the folded transformation $\boldsymbol{\xi}(\boldsymbol{x})$ will be reproduced.

One way to make the Poisson system satisfy the maximum principle is to replace the control functions P^i with other functions which guarantee the maximum principle. One appropriate approach is to define the control functions P^i in the form

$$P^i = g^{jk} P^i_{jk} , \qquad i, j, k = 1, \dots, n .$$

Such an expression for P^i is prompted by (6.21) and (6.22) with

$$P^i_{jk} = \frac{\partial \xi^i}{\partial q^m} \frac{\partial^2 q^m}{\partial \xi^j \partial \xi^k} , \qquad i, j, k, m = 1, \dots, n , \tag{6.23}$$

defined by the transformation from the intermediate coordinates q^i to the final computational coordinates ξ^i. According to the theory of elliptic equations, the factors g^{ij} in the expressions for P^i guarantee the maximum principle for the Poisson system (6.15).

Thus, an appropriate Poisson system can be defined by the equations

$$\nabla^2 \xi^i = g^{jk} P^i_{jk} , \qquad i, j, k = 1, \dots, n , \tag{6.24}$$

where the control functions P^i_{jk} are considered to be specified. The inverse of (6.24) then has the form

$$g^{ij}(\boldsymbol{x}_{\xi^{ij}} + P^k_{ij} \boldsymbol{x}_{\xi^k}) = 0 . \tag{6.25}$$

When the intermediate transformation $\boldsymbol{q}(\boldsymbol{\xi})$ is composed of separate one-dimensional mappings $q^i(\xi^i)$ for each coordinate direction ξ^i, then, from (6.23),

$$P^i_{jk} = \delta^i_j \delta^i_k P^i$$

so that the generation system (6.24) becomes

$$\nabla^2 \xi^i = g^{ii} P^i \tag{6.26}$$

for each fixed $i = 1, \ldots, n$. The inverted system then has the form

$$g^{ij} \boldsymbol{x}_{\xi^i \xi^j} + g^{ii} P^i \boldsymbol{x}_{\xi^i} = 0 \ . \tag{6.27}$$

The selection of the control functions P^i_{jk} is a difficult task. Equations (6.23) show that these functions are not independent if the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is defined as the composition of an intermediate mapping $\boldsymbol{q}(\boldsymbol{\xi})$ and an exterior mapping $\boldsymbol{x}(\boldsymbol{q})$ which satisfies the inverted Laplace equation (6.6).

Equivalent Forms of the Poisson System

Taking into account the general identity (2.57) for arbitrary smooth functions A^i, $i = 1, \ldots, n$,

$$\frac{\partial}{\partial x^j}\left(A^j\right) \equiv \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(J A^m \frac{\partial \xi^j}{\partial x^m}\right) , \qquad j, m = 1, \ldots, n \ ,$$

we obtain, assuming $A^j = \partial \xi^i / \partial x^j$,

$$\nabla^2 \xi^i = \frac{\partial}{\partial x^j}\left(\frac{\partial \xi^i}{\partial x^j}\right) \equiv \frac{1}{J}\frac{\partial}{\partial \xi^j}\left(J \frac{\partial \xi^i}{\partial x^m}\frac{\partial \xi^j}{\partial x^m}\right) \equiv \frac{1}{J}\frac{\partial}{\partial \xi^j}(J g^{ij}) \ , \tag{6.28}$$

where $i, j, m = 1, \ldots, n$. Therefore, the Poisson system (6.15) is equivalent to the following system of equations:

$$\frac{1}{J}\frac{\partial}{\partial \xi^j}(J g^{ij}) = P^i \ , \qquad i, j = 1, \ldots, n \ , \tag{6.29}$$

which is derived from the elements of the metric tensors only. The left-hand part of (6.29) can be expressed through the Christoffel symbols. For this purpose, we consider the identity

$$\frac{1}{J}\frac{\partial}{\partial \xi^j}\left(J g^{ij} \boldsymbol{x}_{\xi^i}\right) \equiv 0 \ , \tag{6.30}$$

which is a result of (2.48), since $g^{ij}\boldsymbol{x}_{\xi^i} = \boldsymbol{\nabla}\xi^j$. Performing the differentiation on the left-hand part of (6.30), we obtain

$$g^{ij}\boldsymbol{x}_{\xi^i\xi^j} + \frac{1}{J}\frac{\partial}{\partial\xi^j}(Jg^{ij})\boldsymbol{x}_{\xi^i} \equiv 0 \,, \qquad i, j = 1, \dots, n \,. \tag{6.31}$$

The dot product of (6.31) and $\nabla\xi^k$ results in

$$\nabla^2\xi^k \equiv -g^{ij}\Gamma^k_{ij} \,, \qquad i, j, k = 1, \dots, n \,, \tag{6.32}$$

using (6.28) and (2.39). The identity (6.32) demonstrates that the value of $\nabla^2\xi^k$ is expressed through the metric elements and the space Christoffel symbols of the second kind.

The utilization of (6.32) generates the following equivalent form of the Poisson system (6.15):

$$-g^{ij}\Gamma^k_{ij} = P^k \,, \qquad i, j, k = 1, \dots, n \,. \tag{6.33}$$

In order to define the value of the forcing terms on the boundaries, we use an alternative, equivalent system of equations

$$P^k = -g^{ij}g^{lk}[ij, l] \,, \qquad i, j, k, l = 1, \dots, n \,, \tag{6.34}$$

which is obtained from (6.33) and (2.43), with

$$[ij, l] = \boldsymbol{x}_{\xi^i\xi^j} \cdot \boldsymbol{x}_{\xi^l} = \frac{1}{2}\Big(\frac{\partial g_{il}}{\partial\xi^j} + \frac{\partial g_{jl}}{\partial\xi^i} - \frac{\partial g_{ij}}{\partial\xi^l}\Big) \,,$$

$$i, j, l = 1, \dots, n \,.$$

In particular, when the coordinate system ξ^i is orthogonal, then (6.34) results in

$$P^k = -g^{ii}g^{kk}[ii, k] = g^{kk}\Big(\frac{1}{2}g^{ii}\frac{\partial g_{ii}}{\partial\xi^k} - g^{kk}\frac{\partial g_{kk}}{\partial\xi^k}\Big) \,,$$

$$i = 1, \dots, n \,, \qquad k \text{ fixed.}$$

Orthogonality at Boundaries

The grid point distribution in the immediate neighborhood of the boundaries of two-dimensional and three-dimensional regions has a strong influence on the accuracy of the algorithms developed for the numerical solution of partial differential equations. In particular, it is often desirable to have orthogonal or nearly orthogonal grid lines emanating from some boundary segments.

Consider, for example, the evaluation of the outward normal derivative of an arbitrary function f at the boundary of a two-dimensional region X^2:

$$\frac{\partial f}{\partial \boldsymbol{n}} = \boldsymbol{n} \cdot \nabla f.$$

If the boundary is a line of constant ξ^1, then

$$\boldsymbol{n} = \frac{1}{\sqrt{g^{22}}} \nabla \xi^2 ,$$

and so

$$\frac{\partial f}{\partial \boldsymbol{n}} = \frac{1}{\sqrt{g^{22}}} \frac{\partial f}{\partial x^i} \frac{\partial \xi^2}{\partial x^i} = \frac{g^{2k}}{\sqrt{g^{22}}} \frac{\partial f}{\partial \xi^k}, \qquad k = 1, 2 . \tag{6.35}$$

If the coordinates $\xi^1,\ \xi^2$ are orthogonal, (6.35) reduces to just

$$\frac{\partial f}{\partial \boldsymbol{n}} = \frac{1}{\sqrt{g_{22}}} \frac{\partial f}{\partial \xi^2}.$$

Obviously, this equation is much simpler than (6.35) and is to be preferred for most analytical purposes. Less obviously, but of importance to numerical schemes, (6.35) couples the ξ^1 and ξ^2 variations of the function f, and thus the application of a Neumann boundary condition to f may involve the difference of two large numbers, with a possible loss of numerical accuracy.

The Poisson system provides two opportunities to satisfy the requirement of orthogonality or near orthogonality of the coordinate lines emanating from the boundary segments, either by imposing Neumann boundary conditions or by specifying the source terms P^i through the boundary values of the coordinate transformation.

The commonly used approach to the specification of the source terms P^i to provide boundary orthogonality relies on the computation of the values of $\nabla^2 \xi^i$ on boundary segments, provided the coordinate lines ξ^i are orthogonal to these segments. These computed data generate the boundary conditions for P^i. Expansion of the boundary values of P^i over the whole region through algebraic or differential approaches produces the specification of the control functions. The coincidence of P^i and the computed values of $\nabla^2 \xi^i$ on the boundary provides some grounds for the expectation that the solution of the Poisson system with the specified P^i will yield a coordinate system which is nearly orthogonal in the vicinity of the boundary segments.

In this subsection, we find some necessary conditions for the boundary values of the control functions P^i to generate coordinates which emanate orthogonally or nearly orthogonally from the respective boundary segments.

Two-Dimensional Equations

Now we consider a two-dimensional case. Let a coordinate curve $\xi^2 = \xi_0^2$ be orthogonal to the opposite family of coordinate lines $\xi^1 = \text{const}$. In this case,

$$\begin{aligned} &g_{12} = 0 , \qquad &&J = \sqrt{g_{11}\, g_{22}} , \\ &g^{12} = 0 , \qquad &&g^{11} = 1/g_{11} , \qquad g^{22} = 1/g_{22} , \end{aligned}$$

along this coordinate line $\xi^2 = \xi_0^2$. With these equations, the relations (6.34) for the definition of $P^i,\ i = 1, 2$, on the coordinate line $\xi^2 = \xi_0^2$ have the form

$$\begin{aligned} P^1 &= -(g^{11})^2[11,1] - g^{22}g^{11}[22,1] \\ &= -\frac{1}{g_{11}}\Big(\frac{1}{2g_{11}}\frac{\partial g_{11}}{\partial \xi^1} + \frac{1}{g_{22}}(\boldsymbol{x}_{\xi^2\xi^2}\cdot \boldsymbol{x}_{\xi^1})\Big)\ . \end{aligned} \tag{6.36}$$

Analogously, if the coordinate curve $\xi^1 = \xi^1_0$ is orthogonal to the opposite family of coordinate curves, then for the source term P^2 on the curve $\xi^1 = \xi^1_0$, we obtain

$$\begin{aligned} P^2 &= -\frac{1}{2(g_{22})^2}\frac{\partial g_{22}}{\partial \xi^2} - \frac{1}{J^2}\frac{\partial^2 x^i}{\partial \xi^1\,\partial \xi^1}\frac{\partial x^i}{\partial \xi^2} \\ &= -\frac{1}{g_{22}}\Big(\frac{1}{2g_{22}}\frac{\partial g_{22}}{\partial \xi^2} + \frac{1}{g_{11}}[11,2]\Big)\ . \end{aligned} \tag{6.37}$$

If the curve $\xi^2 = \xi^2_0$ is the boundary segment, then all of the quantities in (6.36) are known except g_{22} and $\boldsymbol{x}_{\xi^2\xi^2}$. The metric term g_{22} is connected with the relative grid spacing $|\boldsymbol{x}_{\xi^2}|$ of the coordinate lines $\xi^2 =$ const by the relation $g_{22} = |\boldsymbol{x}_{\xi^2}|^2$. If the spacing $|\boldsymbol{x}_{\xi^2}|$ is specified on the boundary curve $\xi^2 = \xi^2_0$, then only $\boldsymbol{x}_{\xi^2\xi^2}$ is an unknown quantity in the specification of the control function P^1 on this boundary. In the same way, on the boundary segment $\xi^1 = \xi^1_0$, only $\boldsymbol{x}_{\xi^1\xi^1}$ is an unknown quantity in (6.37) for P^2. One way to define $\boldsymbol{x}_{\xi^1\xi^1}$ and $\boldsymbol{x}_{\xi^2\xi^2}$, and consequently P^1 and P^2, on the respective boundary segments is to apply an iterative procedure which utilizes the Eq. (6.18) with the term $-2g_{12}\boldsymbol{x}_{\xi^1\xi^2}$ omitted because of the orthogonality condition:

$$g_{22}\boldsymbol{x}_{\xi^1\xi^1} + g_{11}\boldsymbol{x}_{\xi^2\xi^2} = -g_{11}g_{22}P^i\boldsymbol{x}_{\xi^i}\ . \tag{6.38}$$

Every step allows one to evaluate the control function P^1 on the boundary curves $\xi^2 = \xi^2_0$ and the control function P^2 on the boundary lines $\xi^1 = \xi^1_0$. By expansion from the boundary values, the control functions P^1 and P^2 are evaluated in the domain X^2. By solving the system (6.38) with the obtained control functions P^i, the grid corresponding to the next step is generated in the domain X^2. If convergence is achieved, the final grid is generated satisfying the condition of orthogonality and the specified spacing at the boundary.

Local Straightness at the Boundary

The Eqs. (6.36) and (6.37), which serve to define the control functions P^i on the boundary, are simplified if an additional requirement of local straightness of coordinate lines is imposed. To demonstrate this, we note that the vector

$$\boldsymbol{b} = \Big(\frac{\partial x^2}{\partial \xi^1}, -\frac{\partial x^1}{\partial \xi^1}\Big)$$

is orthogonal to the tangential vector $\boldsymbol{x}_{\xi_1}$. From the assumed condition of orthogonality of the coordinate system along the curve $\xi^2 = \xi^2_0$, we find that the vector $\boldsymbol{x}_{\xi_2}$ is parallel to the vector $\boldsymbol{b}$:

$$\boldsymbol{x}_{\xi^2} = F\boldsymbol{b}\ ,$$

i.e.

$$\begin{aligned}\frac{\partial x^1}{\partial \xi^2} &= -F\frac{\partial x^2}{\partial \xi^1}\ ,\\ \frac{\partial x^2}{\partial \xi^2} &= F\frac{\partial x^1}{\partial \xi^1}\ .\end{aligned} \tag{6.39}$$

Let $\partial x^2/\partial \xi^2 \neq 0$ at the boundary point $\boldsymbol{\xi}_0$. After squaring every equation of the system (6.39) and summing them, we find that $F = \sqrt{g_{22}/g_{11}}$. Therefore,

$$\begin{aligned}[22, 1] &= \frac{\partial x^i}{\partial \xi^1}\frac{\partial^2 x^i}{\partial \xi^2\,\partial \xi^2}\\ &= \frac{1}{F}\Big(\frac{\partial x^2}{\partial \xi^2}\frac{\partial^2 x^1}{\partial \xi^2\,\partial \xi^2} - \frac{\partial x^1}{\partial \xi^2}\frac{\partial^2 x^2}{\partial \xi^2\,\partial \xi^2}\Big)\\ &= \frac{1}{F}(\frac{\partial x^2}{\partial \xi^2})^2\frac{\partial}{\partial \xi^2}\Big(\frac{\partial x^1}{\partial \xi^2}\Big/\frac{\partial x^2}{\partial \xi^2}\Big)\end{aligned} \tag{6.40}$$

at the point $\boldsymbol{\xi}_0$. The substitution of this relation in (6.36) yields

$$P^1 = -\frac{1}{2(g_{11})^2}\frac{\partial g_{11}}{\partial \xi^1} - \frac{1}{J^2 F}\Big(\frac{\partial x^2}{\partial \xi^2}\Big)^2\frac{\partial}{\partial \xi^2}\Big(\frac{\partial x^1}{\partial \xi^2}\Big/\frac{\partial x^2}{\partial \xi^2}\Big)\ . \tag{6.41}$$

The ratio $(\partial x^1/\partial \xi^2)/(\partial x^2/\partial \xi^2)$ is merely the slope $\mathrm{d}x^1/\mathrm{d}x^2$ of the family of the coordinate curves $\xi^1 = \text{const}$, which are transverse to the coordinate $\xi^2 = \xi^2_0$. The imposition of the condition that these transverse coordinate lines $\xi^1 = \text{const}$ are locally straight (i.e. have zero curvature) in the neighborhood of the coordinate $\xi^2 = \xi^2_0$ leads to the equation

$$\frac{\partial}{\partial \xi^2}\Big(\frac{\partial x^1}{\partial \xi^2}\Big/\frac{\partial x^2}{\partial \xi^2}\Big) = 0 \tag{6.42}$$

on the coordinate line $\xi^2 = \xi^2_0$. So, in this case, we obtain, from (6.36), the following expression for P^1,

$$P^1 = -\frac{1}{2(g_{11})^2}\frac{\partial}{\partial \xi^1}g_{11} \tag{6.43}$$

which the source term P^1 must satisfy along the coordinate curve $\xi^2 = \xi^2_0$ if it is orthogonal to the family of locally straight coordinate lines $\xi^1 = \text{const}$. This equation can be used to compute the numerical value of P^1 at each grid point on the horizontal boundaries where the transformation $\boldsymbol{x}(\boldsymbol{\xi})$ and consequently the metric element g_{11}, is specified.

Analogously, if the coordinate line $\xi^1 = \xi^1_0$ is orthogonal to the family of locally straight coordinate curves $\xi^2 = \text{const}$, we have

$$P^2 = -\frac{1}{2(g_{22})^2}\frac{\partial}{\partial\xi^2}g_{22} \qquad (6.44)$$

along this coordinate.

Once the control function P^1 is defined at each mesh point of the horizontal boundaries $\xi^2 = 0$ and $\xi^2 = 1$, its value at the interior mesh points can be computed by unidirectional interpolation along the vertical mesh lines $\xi^1 = \text{const}$ between the horizontal boundaries. Similarly, the control function P^2 can be computed by unidirectional interpolation along the horizontal mesh lines $\xi^2 = \text{const}$.

Three-Dimensional Equations

Now we find the values of the system (6.28) on a coordinate surface, say $\xi^3 = \xi^3_0$, when it is orthogonal to the family of coordinates ξ^3. These values define the specification of the control functions $P^i,\ i = 1, 2$, on the coordinate surface to obtain three-dimensional grids nearly orthogonal about this surface through the system (6.16).

From the condition of orthogonality, we have the following relations on the surface $\xi^3 = \xi^3_0$:

$$\begin{aligned} &g_{13} = g_{23} = 0\ , \qquad g^{33} = 1/g_{33}\ , \\ &J = \sqrt{g_{33}\overline{g}}\ , \qquad \overline{g} = \det\{g_{ij}\}, \qquad i, j = 1, 2\ . \end{aligned} \qquad (6.45)$$

It is also clear that the orthogonality condition on the coordinate surface $\xi^3 = \xi^3_0$ implies that the matrix $\{g^{ij}\},\ i, j = 1, 2$, is inverse to the tensor $\{g_{ij}\},\ i, j = 1, 2$. In fact, the matrix $\{g_{ij}\},\ i, j = 1, 2$, is the covariant metric tensor of the surface $\xi^3 = \xi^3_0$ in the coordinates $\xi^1,\ \xi^2$ represented by the parametrization

$$\boldsymbol{r}(\boldsymbol{\xi}) : \varXi^2 \to R^3\ , \qquad \boldsymbol{\xi} = (\xi^1, \xi^2)\ , \qquad \boldsymbol{r} = (x^1, x^2, x^3)\ ,$$

where

$$\boldsymbol{r}(\xi^1, \xi^2) = \boldsymbol{x}(\xi^1, \xi^2, \xi^3_0)\ .$$

Correspondingly, the matrix $\{g^{ij}\},\ i, j = 1, 2$, is the contravariant metric tensor of the surface $\xi^3 = \xi^3_0$ in the coordinates $\xi^1,\ \xi^2$.

The forcing terms $P^i,\ i = 1, 2$, are expressed by the system of equations (6.29). We will write out the equations for $i = 1, 2$ as a sum of two parts. The first part contains only the terms with the superscripts 1 and 2, which thus are related to the coordinate surface $\xi^3 = \xi^3_0$. The second part includes the terms with the superscript 3. Thus, we assume

$$\begin{aligned} P^i &= P^i_1 + P^i_2\ , \\ P^i_1 &= \frac{1}{J}\frac{\partial}{\partial\xi^j}(Jg^{ij})\ , \qquad i, j = 1, 2\ , \\ P^i_2 &= \frac{1}{J}\frac{\partial}{\partial\xi^3}(Jg^{i3})\ , \qquad i = 1, 2\ . \end{aligned} \qquad (6.46)$$

Let us consider the case $i = 1$. In accordance with the formula (2.22), we obtain for the element g^{13} in (6.46)

$$g^{13} = \frac{g_{21}g_{32} - g_{31}g_{22}}{g} \ .$$

So, taking into account the relations (6.45) valid on the surface $\xi^3 = \xi^3_0$, we obtain

$$g^{13} = -\frac{1}{g_{33}}(g^{21}g_{32} + g^{11}g_{13}) \ .$$

Also, as a result of the condition of orthogonality, we have, on the surface,

$$g^{ij} = \frac{(-1)^{i+j}g_{3-i\ 3-j}}{\overline{g}} \ , \qquad i, j = 1, 2 \ ,$$

with $i,\ j$ fixed, and thus

$$P^1_2 = \frac{1}{J}\frac{\partial}{\partial \xi^3}(Jg^{13}) = -\frac{1}{g_{33}}\Big(g^{21}\frac{\partial}{\partial \xi^3}g_{23} + g^{11}\frac{\partial}{\partial \xi^3}g_{13}\Big) \tag{6.47}$$

on $\xi^3 = \xi^3_0$. For the term P^i_1 on the surface $\xi^3 = \xi^3_0$, we find, using (6.45) and (6.46),

$$\begin{aligned} P^i_1 &= \frac{1}{\sqrt{g_{33}\overline{g}}}\frac{\partial}{\partial \xi^j}(\sqrt{g_{33}\overline{g}}g^{ij}) \\ &= \frac{1}{\sqrt{\overline{g}}}\frac{\partial}{\partial \xi^j}(\sqrt{\overline{g}}g^{ij}) + \frac{g^{ij}}{2g_{33}}\frac{\partial}{\partial \xi^j}g_{33}, \qquad i, j = 1, 2 \ . \end{aligned} \tag{6.48}$$

The relations (6.47) and (6.48) yield

$$\begin{aligned} P^1 = &\frac{1}{\sqrt{\overline{g}}}\frac{\partial}{\partial \xi^j}(\sqrt{\overline{g}}g^{1j}) \\ &+\frac{1}{g_{33}}\Big[g^{11}\Big(\frac{1}{2}\frac{\partial}{\partial \xi^1}g_{33} - \frac{\partial}{\partial \xi^3}g^{13}\Big) + g^{12}\Big(\frac{1}{2}\frac{\partial}{\partial \xi^2}g_{33} - \frac{\partial}{\partial \xi^3}g^{23}\Big)\Big] \\ = &\frac{1}{\sqrt{\overline{g}}}\frac{\partial}{\partial \xi^j}(\sqrt{\overline{g}}g^{1j}) + \frac{1}{g_{33}}\Big(g^{11}\frac{\partial x^k}{\partial \xi^1}\frac{\partial^2 x^k}{\partial \xi^3 \partial \xi^3} + g^{12}\frac{\partial x^k}{\partial \xi^2}\frac{\partial^2 x^k}{\partial \xi^3 \partial \xi^3}\Big) \ . \end{aligned}$$

Analogously,

$$P^2 = \frac{1}{\sqrt{\overline{g}}}\frac{\partial}{\partial \xi^j}(\sqrt{\overline{g}}g^{2j}) + \frac{1}{g_{33}}\Big(g^{21}\frac{\partial x^k}{\partial \xi^1}\frac{\partial^2 x^k}{\partial \xi^3 \partial \xi^3} + g^{22}\frac{\partial x^k}{\partial \xi^2}\frac{\partial^2 x^k}{\partial \xi^3 \partial \xi^3}\Big).$$

So, a general formula for $P^i,\ i = 1, 2$, on the coordinate surface $\xi^3 = \xi^3_0$ is

$$P^i = \frac{1}{\sqrt{\overline{g}}}\frac{\partial}{\partial \xi^j}(\sqrt{\overline{g}}g^{ij}) + \frac{1}{g_{33}}\Big(g^{ij}\frac{\partial x^k}{\partial \xi^j}\frac{\partial^2 x^k}{\partial \xi^3 \partial \xi^3}\Big) \ , \tag{6.49}$$

where $i, j = 1, 2, \ k = 1, 2, 3$. Equation (6.49) can be used in the same manner as (6.36) and (6.37) to define the values of the control functions at the boundary segments for the purpose of providing orthogonality at the boundary with a specified normal spacing, through an iterative procedure.

If the vector $\boldsymbol{r}_{\xi^3\xi^3}, \ \boldsymbol{r} = (x^1, x^2, x^3)$, is parallel to the vector $\boldsymbol{r}_{\xi^3}$, for example, when the curvature of the coordinate lines ξ^3 vanishes on the surface $\xi^3 = \xi^3_0$, then, from the condition of orthogonality, the second sum of (6.49) vanishes, which implies

$$P^i = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} (\sqrt{g} g^{ij}), \qquad i = 1, 2 , \tag{6.50}$$

on the surface $\xi^3 = \xi^3_0$.

As was mentioned, the covariant and contravariant elements of the surface $\xi^3 = \xi^3_0$ in the coordinates $\xi^1, \ \xi^2$ coincide with the elements g_{ij} and g^{ij}, respectively, for $i, \ j = 1, 2$. So, the expression (6.50) for P^i is the value obtained by applying the Beltrami operator Δ_{B},

$$\Delta_{\mathrm{B}} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left(\sqrt{g} g^{kj} \frac{\partial}{\partial \xi^k} \right), \qquad j, k = 1, 2 , \tag{6.51}$$

to the function $\xi^i(\boldsymbol{x})$, i.e.

$$\Delta_{\mathrm{B}} \xi^i = P^i \ , \qquad i = 1, 2 \ .$$

Analogous equations for P^i are valid for the coordinate surfaces $\xi^i = \xi^i_0$, $i = 1$ or $i = 2$.

Control of the Angle of Intersection

Now, for two dimensions, we find how to use the source terms P^i to control the angle at which each grid line transverse to the boundary intersects it.

First, we note that the maximum principle for the Laplace operator guarantees that if

$$\nabla^2 f \geq \nabla^2 g$$

and

$$f|_{\partial X^n} = g|_{\partial X^n} \ ,$$

then

$$f(x) \leq g(x) \ , \qquad x \in X^n \ .$$

Therefore, in the two-dimensional case, a decrease in the values of P^i causes an increase in the values of ξ^i, and, correspondingly, a reduction of the intersection angle of the boundary line with the opposite family of coordinate curves. For example, if P^i is negative at a grid point obtained through the inverted Laplace system, the point is moved towards the side where ξ^i is less. The opposite effect is produced when the values of P^i are increased. This observation allows one to influence the angle of intersection by choosing larger or smaller values of P^i.

A more sophisticated procedure for controlling the angle of intersection relies on a study of the dependence of the source terms P^i on the boundary distribution and the angle of intersection.

In place of the orthogonality condition $g_{12} = 0$, we use, therefore, the condition

$$\boldsymbol{x}_{\xi^1} \cdot \boldsymbol{x}_{\xi^2} = g_{12} = |\boldsymbol{x}_{\xi^1}|\,|\boldsymbol{x}_{\xi^2}| \cos\theta = \sqrt{g_{11}g_{22}} \cos\theta \ , \tag{6.52}$$

where θ denotes the angle of intersection between a coordinate line, say $\xi^2 = \xi^2_0$, and the corresponding transverse coordinate curves $\xi^1 = \text{const}$. So, θ is a function depending on ξ^1. A more convenient representation of this condition is

$$g_{12} = J \cot\theta \ , \tag{6.53}$$

which follows from (6.52) and the equation

$$J = |\boldsymbol{x}_{\xi^1}|\,|\boldsymbol{x}_{\xi^2}| \sin\theta = \sqrt{g_{11}g_{22}} \sin\theta \ .$$

Since

$$J = \sqrt{g_{11}g_{22} - (g_{12})^2} \ ,$$

we obtain, from (6.52) and (6.53),

$$\begin{aligned} g^{11} &= \frac{1}{g_{11}\sin^2\theta} \ , \\ g^{22} &= \frac{1}{g_{22}\sin^2\theta} \ , \\ g^{12} &= -\frac{\cos\theta}{\sqrt{g_{11}g_{22}}\sin^2\theta} \ . \end{aligned} \tag{6.54}$$

The angle between the tangent vectors $\boldsymbol{x}_{\xi^2}$ and $\boldsymbol{x}_{\xi^1}$ is θ, and therefore the vector $\boldsymbol{x}_{\xi^2}$ intersects the vector

$$\boldsymbol{b} = (-\frac{\partial x^2}{\partial \xi^1}, \frac{\partial x^1}{\partial \xi^1}) \ ,$$

which is orthogonal to the vector $\boldsymbol{x}_{\xi^1}$, at an angle of $\pi/2 - \theta$ (Fig. 6.8). The vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{b}$ are orthogonal and have the same length; therefore,

$$\boldsymbol{x}_{\xi^2} = F(\cos\theta\, \boldsymbol{x}_{\xi^1} + \sin\theta\, \boldsymbol{b}) \ ,$$

where $F = \sqrt{g_{22}/g_{11}}$. Hence,

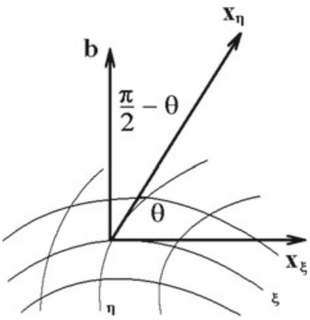


Fig. 6.8 Angle between the normal and tangential vectors

$$\begin{aligned}\frac{\partial x^1}{\partial \xi^2} &= F\Big(\cos\theta \frac{\partial x^1}{\partial \xi^1} - \sin\theta \frac{\partial x^2}{\partial \xi^1}\Big) ,\\ \frac{\partial x^2}{\partial \xi^2} &= F\Big(\cos\theta \frac{\partial x^2}{\partial \xi^1} + \sin\theta \frac{\partial x^1}{\partial \xi^1}\Big)\end{aligned} \tag{6.55}$$

on the coordinate curve $\xi^2 = \xi^2_0$. It is clear that (6.55) is a generalization of (6.39).

Now we compute the forcing term P^1 on the boundary curves $\xi^2 = 0$ and $\xi^2 = 1$ required to provide control of the angle θ at these segments. For this purpose, we consider the representation of the Poisson system (6.15) in the form (6.32).

For $i = 1,\ n = 2$, the system (6.32) implies

$$\begin{aligned}P^1 &= -g^{11}\Gamma^1_{11} - 2g^{12}\Gamma^1_{12} - g^{22}\Gamma^1_{22}\\ &= -g^{11}\Big(\frac{\partial^2 x^1}{\partial \xi^1 \partial \xi^1}\frac{\partial \xi^1}{\partial x^1} + \frac{\partial^2 x^2}{\partial \xi^1 \partial \xi^1}\frac{\partial \xi^1}{\partial x^2}\Big)\\ &\quad -2g^{12}\Big(\frac{\partial^2 x^1}{\partial \xi^1 \partial \xi^2}\frac{\partial \xi^1}{\partial x^1} + \frac{\partial^2 x^2}{\partial \xi^1 \partial \xi^2}\frac{\partial \xi^1}{\partial x^2}\Big)\\ &\quad -g^{22}\Big(\frac{\partial^2 x^1}{\partial \xi^2 \partial \xi^2}\frac{\partial \xi^1}{\partial x^1} + \frac{\partial^2 x^2}{\partial \xi^2 \partial \xi^1}\frac{\partial \xi^1}{\partial x^2}\Big) .\end{aligned} \tag{6.56}$$

In the two-dimensional case, we have

$$\begin{aligned}\frac{\partial \xi^1}{\partial x^1} &= \frac{\partial x^2}{\partial \xi^2}\Big/ J,\\ \frac{\partial \xi^1}{\partial x^2} &= -\frac{\partial x^2}{\partial \xi^1}\Big/ J,\end{aligned} \tag{6.57}$$

so, using the relation (6.55) valid along the coordinate $\xi^2 = \xi^2_0$, we obtain

$$\begin{aligned}\frac{\partial\xi^1}{\partial x^1} &= \frac{1}{g_{11}}\Big(\cot\theta\frac{\partial x^2}{\partial\xi^1}+\frac{\partial x^1}{\partial\xi^1}\Big)\ ,\\ \frac{\partial\xi^1}{\partial x^2} &= \frac{1}{g_{11}}\Big(\frac{\partial x^2}{\partial\xi^1}-\cot\theta\frac{\partial x^1}{\partial\xi^1}\Big)\ .\end{aligned}\tag{6.58}$$

Let the quantities $\partial x^1/\partial\xi^1$ and $\partial x^2/\partial\xi^2$ not vanish at the point under consideration. Then, the (6.56) is given by

$$\begin{aligned}g^{11}\Gamma^1_{11} &= g^{11}\Big(\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}\frac{\partial\xi^1}{\partial x^1}+\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1}\frac{\partial\xi^1}{\partial x^2}\Big)\\ &= \frac{1}{(g_{11})^2\sin^2\theta}\Big[\Big(\cot\theta\frac{\partial x^2}{\partial\xi^1}+\frac{\partial x^1}{\partial\xi^1}\Big)\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}\\ &\qquad+\Big(-\cot\theta\frac{\partial x^1}{\partial\xi^1}+\frac{\partial x^2}{\partial\xi^1}\Big)\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1}\Big]\\ &= \frac{1}{(g_{11})^2\sin^2\theta}\Big[\frac{1}{2}\frac{\partial}{\partial\xi^1}g_{11}-\cot\theta\Big(\frac{\partial x^1}{\partial\xi^1}\Big)^2\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^2}{\partial\xi^1}\Big/\frac{\partial x^1}{\partial\xi^1}\Big)\Big]\ .\end{aligned}\tag{6.59}$$

In order to compute the second term of (6.56), we note first that, from (6.55),

$$\frac{\partial x^1}{\partial\xi^2}\Big/\frac{\partial x^2}{\partial\xi^2} = \frac{\cos\theta\dfrac{\partial x^1}{\partial\xi^1}-\sin\theta\dfrac{\partial x^2}{\partial\xi^1}}{\sin\theta\dfrac{\partial x^1}{\partial\xi^1}+\cos\theta\dfrac{\partial x^2}{\partial\xi^1}}\ .$$

Therefore,

$$\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^1}{\partial\xi^2}\Big/\frac{\partial x^2}{\partial\xi^2}\Big) = F^2\Big(\frac{\partial x^2}{\partial\xi^2}\Big)^2[f_1(\xi^1)+f_2(\xi^1)]\ ,\tag{6.60}$$

where

$$\begin{aligned}f_1(\xi_1) &= -\Big[\Big(\sin\theta\frac{\partial x^1}{\partial\xi^1}+\cos\theta\frac{\partial x^2}{\partial\xi^1}\Big)^2+\Big(\cos\theta\frac{\partial x^1}{\partial\xi^1}-\sin\theta\frac{\partial x^2}{\partial\xi^1}\Big)^2\Big]\theta_{\xi^1}\\ &= -g_{11}\theta_{\xi^1}\ ,\end{aligned}$$

$$\begin{aligned}f_2(\xi_1) &= \Big(\cos\theta\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}-\sin\theta\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1}\Big)\Big(\sin\theta\frac{\partial x^1}{\partial\xi^1}+\cos\theta\frac{\partial x^2}{\partial\xi^1}\Big)\\ &\quad-\Big(\sin\theta\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}+\cos\theta\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1}\Big)\Big(\cos\theta\frac{\partial x^1}{\partial\xi^1}-\sin\theta\frac{\partial x^2}{\partial\xi^1}\Big)\\ &= \frac{\partial^2 x^1}{\partial\xi^1\partial\xi^1}\frac{\partial x^2}{\partial\xi^1}-\frac{\partial^2 x^2}{\partial\xi^1\partial\xi^1}\frac{\partial x^1}{\partial\xi^1}\\ &= -\Big(\frac{\partial x^1}{\partial\xi^1}\Big)^2\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^2}{\partial\xi^1}\Big/\frac{\partial x^1}{\partial\xi^1}\Big).\end{aligned}$$

So, using the relations (6.56) and (6.60), we obtain

$$\begin{aligned} g^{12}\Gamma^1_{12} &= g^{12}\Big(\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^2}\frac{\partial\xi^1}{\partial x^1} + \frac{\partial^2 x^2}{\partial\xi^1\partial\xi^2}\frac{\partial\xi^1}{\partial x^2}\Big) \\ &= \frac{g^{12}}{J}\Big(\frac{\partial^2 x^1}{\partial\xi^1\partial\xi^2}\frac{\partial x^2}{\partial\xi^2} - \frac{\partial^2 x^2}{\partial\xi^1\partial\xi^2}\frac{\partial x^1}{\partial\xi^2}\Big) \\ &= \frac{1}{J}g^{12}\Big(\frac{\partial x^2}{\partial\xi^2}\Big)^2\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^1}{\partial\xi^2}\Big/\frac{\partial x^2}{\partial\xi^2}\Big) \\ &= \frac{\cot\theta}{(g_{11})^2\sin^2\theta}\Big[g_{11}\theta_{\xi^1} + \Big(\frac{\partial x^1}{\partial\xi^1}\Big)\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^2}{\partial\xi^1}\Big/\frac{\partial x^1}{\partial\xi^1}\Big)\Big] . \end{aligned} \tag{6.61}$$

Analogously, the third term of (6.56) is given by

$$\begin{aligned} g^{22}\Gamma^1_{22} &= \frac{\partial^2 x^1}{\partial\xi^2\partial\xi^2}\frac{\partial\xi^1}{\partial x^1} + \frac{\partial^2 x^2}{\partial\xi^2\partial\xi^1}\frac{\partial\xi^1}{\partial x^2} \\ &= \frac{1}{J}\Big(\frac{\partial^2 x^1}{\partial\xi^2\partial\xi^2}\frac{\partial x^2}{\partial\xi^2} - \frac{\partial^2 x^2}{\partial\xi^2\partial\xi^2}\frac{\partial x^1}{\partial\xi^2}\Big) \\ &= \frac{1}{J}\Big(\frac{\partial x^2}{\partial\xi^2}\Big)^2\frac{\partial}{\partial\xi^2}\Big(\frac{\partial x^1}{\partial\xi^2}\Big/\frac{\partial x^2}{\partial\xi^2}\Big) . \end{aligned} \tag{6.62}$$

Now, using the relations (6.59), (6.61), and (6.62) in (6.56), we obtain

$$\begin{aligned} P^1 = &-\frac{1}{(g_{11})^2\sin^2\theta} \\ &\times\Big[\frac{1}{2}\frac{\partial}{\partial\xi^1}g_{11} + \cot\theta\Big(\frac{\partial x^1}{\partial\xi^1}\Big)^2\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^2}{\partial\xi^1}\Big/\frac{\partial x^1}{\partial\xi^1}\Big) + 2g_{11}\cot\theta\ \theta_{\xi^1}\Big] \\ &-\frac{1}{Jg_{22}\sin^2\theta}\Big(\frac{\partial x^2}{\partial\xi^2}\Big)^2\frac{\partial}{\partial\xi^2}\Big(\frac{\partial x^1}{\partial\xi^2}\Big/\frac{\partial x^2}{\partial\xi^2}\Big) \end{aligned} \tag{6.63}$$

along the coordinate curve $\xi^2 = \xi^2_0$.

Analogously, for the second source term P^2 along the coordinate line $\xi^1 = \xi^1_0$,

$$\begin{aligned} P^2 = &-\frac{1}{(g_{22})^2\sin^2\theta} \\ &\times\Big[\frac{1}{2}\frac{\partial}{\partial\xi^2}g_{22} + \cot\theta\Big(\frac{\partial x^2}{\partial\xi^2}\Big)^2\frac{\partial}{\partial\xi^2}\Big(\frac{\partial x^1}{\partial\xi^2}\Big/\frac{\partial x^2}{\partial\xi^2}\Big) + 2g_{11}\cot\theta\ \theta_{\xi^2}\Big] \\ &-\frac{1}{Jg_{11}\sin^2\theta}\Big(\frac{\partial x^1}{\partial\xi^1}\Big)^2\frac{\partial}{\partial\xi^1}\Big(\frac{\partial x^2}{\partial\xi^1}\Big/\frac{\partial x^1}{\partial\xi^1}\Big) . \end{aligned} \tag{6.64}$$

If the coordinate curves $\xi^i = \mathrm{const}$ are locally straight at the points of their intersection with the opposite coordinate lines, then the last lines in (6.63) and (6.64) will vanish. Using the conventional notation x, y and ξ, η instead of x^1, x^2 and ξ^1, ξ^2, we have for the source terms P^1 and P^2 in this case

$$P^1 = -\frac{1}{(g_{11})^2 \sin^2\theta}\Big[\frac{1}{2}\frac{\partial}{\partial\xi}g_{22} + \cot\theta\Big(\frac{\partial x}{\partial\xi}\Big)^2 \frac{\partial}{\partial\xi}\Big(\frac{\partial y}{\partial\xi}\Big/\frac{\partial x}{\partial\xi}\Big) \\ +2g_{11}\cot\theta\,\theta_\xi\Big], \qquad (6.65)$$

$$P^2 = -\frac{1}{(g_{22})^2 \sin^2\theta}\Big[\frac{1}{2}\frac{\partial}{\partial\eta}g_{22} + \cot\theta\Big(\frac{\partial y}{\partial\eta}\Big)^2 \frac{\partial}{\partial\eta}\Big(\frac{\partial x}{\partial\eta}\Big/\frac{\partial y}{\partial\eta}\Big) \\ +2g_{11}\cot\theta\,\theta_\eta\Big]. \qquad (6.66)$$

Adaptation Through Control Functions

In the elliptic method based on Poisson equations, the grid adaptation may be realized through the choice of the control functions P^i in the inverted system of equations

$$g^{ij}\frac{\partial^2 x^k}{\partial\xi^i\partial\xi^j} = -P^i\frac{\partial x^k}{\partial\xi^i}, \qquad i, j, k = 1, \dots, n. \qquad (6.67)$$

In one dimension, this system is reduced to

$$\Big(\frac{\mathrm{d}\xi}{\mathrm{d}x}\Big)^2\frac{\mathrm{d}^2x}{\mathrm{d}\xi^2} = -P\frac{\mathrm{d}x}{\mathrm{d}\xi}. \qquad (6.68)$$

One way of determining the source function P in (6.68) so as to provide an opportunity to control the attraction or repulsion of grid points is to use an analogy between the one-dimensional equation in (4.23), i.e.

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\Big(\frac{\mathrm{d}x}{\mathrm{d}\xi}\psi(x,\epsilon)\Big) = 0, \qquad 0 < \xi < 1, \qquad (6.69)$$

implementing the one-dimensional equidistribution principle, and (6.68). In accordance with the univariate equidistribution approach, the control function P in (6.68) realizing the equidistribution of the monitor function $w(x)$ should have the form

$$P = \Big(\frac{\mathrm{d}\xi}{\mathrm{d}x}\Big)^2\frac{\mathrm{d}w}{\mathrm{d}\xi}\Big/ w, \qquad (6.70)$$

since, with this expression for P, the one-dimensional inverted Poisson equation (6.68) is transformed into the equation

$$w\frac{\mathrm{d}}{\mathrm{d}\xi}\left(w\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)=0\ ,$$

which is equivalent to Eq. (6.69).

The simplest approach to the specification of the control functions P^i in (6.67) lies in the generalization of (6.70). Commonly, this is realized by substituting g^{ii} for $(\mathrm{d}\xi/\mathrm{d}x)^2$ and $\partial w/\partial\xi^i$ for $\mathrm{d}w/\mathrm{d}\xi$, by the following representation of the control functions P^i, for example:

$$P^i=g^{ii}\frac{\partial w}{\partial\xi_i}\Big/w\ ,\qquad \text{for } i \text{ fixed}\ . \tag{6.71}$$

A more general form also suggests

$$P^i=\sum_j g^{ij}\frac{\partial w^i}{\partial\xi^j}\Big/w^i\ ,\qquad \text{for } i \text{ fixed}\ . \tag{6.72}$$

The functions $w,\ w^i$ in (6.71) and (6.72) are defined in terms of the components of the solution of the physical problem and their derivatives. Thus, in problems of the motion of a liquid in a reservoir, w is defined in terms of the depth function H:

$$w=1+H\ .$$

In gas-dynamics investigations, P^i and w are defined through a number of salient physical quantities. For flows with shock waves, use is made of the gas density, the Mach number, and the internal energy and pressure, while in boundary-layer calculations, the velocity or vorticity is used to formulate the control functions P^i. For instance, in supersonic compressible flows, the pressure p is commonly identified as such a quantity, since shock waves are detected by its rapid variation. To study these flows on a grid controlled by source terms P^i determined through (6.71), the weight function w is usually specified by the equation

$$w=1+\parallel\nabla p\parallel\ .$$

Another example of the specification of the control functions P^i applied to the study of two-dimensional flows can be presented through the density ρ:

$$\begin{aligned}
P^1&=g^{11}\left(\frac{\partial w}{\partial\xi^1}-c_1\frac{\partial w}{\partial\xi^2}\right)w^{-1}\ ,\\
P^2&=g^{22}\left(-c_2\frac{\partial w}{\partial\xi^1}+\frac{\partial w}{\partial\xi^2}\right)w^{-1}\ ,\\
w&=1+f(\rho)\ .
\end{aligned}$$

6.2.3 Other Elliptic Equations

Elliptic equations of the form

$$\sum_{i,j=1}^{n} g^{ij} \frac{\partial}{\partial \xi^i} \left(b_{ij} \frac{\partial x^\alpha}{\partial \xi^j} \right) = 0 \, , \qquad \alpha = 1, 2, \ldots, n \, , \tag{6.73}$$

in which adaptation is carried out by choosing the coefficients b_{ij}, defined as a rule in terms of the gradients of the solution, are also used. Thus, in calculations of two-dimensional gas flows, adaptive grids are constructed using (6.73) with

$$\begin{aligned} &b_{11} = f_1 \, , \qquad b_{22} = f_2 \, , \qquad b_{12} = b_{21} = \sqrt{f_1 f_2} \, , \\ &f_i = f_i(p, g_{11}, g_{22}) \, , \qquad i = 1, 2 \, . \end{aligned}$$

The simplest form of (6.73) is obtained for

$$g^{ij} = \triangle^i_j \, , \qquad b_{ij} = f(\rho) \triangle^j_i \, ,$$

resulting in

$$\frac{\partial}{\partial \xi^i} \left(f(\rho) \frac{\partial \mathbf{x}}{\partial \xi^i} \right) = 0 \, , \qquad i = 1, \ldots, n \, .$$

6.3 Biharmonic Equations

The main drawback of a grid generation method based on a second-order elliptic differential equation is the limitation in controlling the boundary grid distribution and the direction of the coordinate lines emanating from the boundary. This results in considerable numerical difficulties in the solution of problems involving boundary conditions in the normal direction, for example, problems of heat transfer and inviscid aerodynamics. Thus, the technique described above, which utilizes the forcing terms of the Poisson system to control the directions of the grid lines, is not always acceptable.

A more reliable approach to this problem is the use of differential equations of increased order, in particular, biharmonic equations. A system of biharmonic equations provides an efficient opportunity to simultaneously satisfy both Dirichlet and Neumann conditions on the boundaries. This provides the flexibility necessary to smoothly patch together the subgrids and control the locations of grid points.

6.3.1 Formulation of the Approach

As for the elliptic system of second order, the most acceptable system of biharmonic equations for producing suitable grids is formulated in terms of the coordinates x^i of the physical domain, used as independent variables, through a composition of Laplace operators:

$$\nabla^2(\nabla^2\xi^i) = 0 \, , \qquad i = 1, \ldots, n \, . \tag{6.74}$$

This system is extended to the mixed-boundary-value problem by imposing the boundary conditions

$$\xi^i(\boldsymbol{x}) = f^i(\boldsymbol{x}) \, , \qquad \frac{\partial \xi^i}{\partial \boldsymbol{n}}(\boldsymbol{x}) = 0 \, , \qquad \boldsymbol{x} \in \partial X^n \, . \tag{6.75}$$

The derivative $\partial/\partial \boldsymbol{n}$ is taken in the outward normal direction on the boundary of X^n. Applying the coupled approach, the problem of (6.74) and (6.75) yields the system

$$\begin{aligned} &\nabla^2\xi^i = p^i \, , \\ &\nabla^2 p^i = 0 \, , \qquad i = 1, \ldots, n, \end{aligned} \tag{6.76}$$

and the boundary conditions for ξ^i and p^i

$$\begin{aligned} &\boldsymbol{\xi}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) \, , \qquad \boldsymbol{x} \in \partial X^n \, , \\ &\boldsymbol{p}(\boldsymbol{x}) = \nabla^2\boldsymbol{\xi}(\boldsymbol{x}) - c\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{n}}(\boldsymbol{x}) \, , \qquad \boldsymbol{x} \in \partial X^n \, , \end{aligned} \tag{6.77}$$

where c is an arbitrary nonzero constant, and

$$\boldsymbol{f} = (f^1, \ldots, f^n) \, , \qquad \boldsymbol{p} = (p^1, \ldots, p^n) \, .$$

6.3.2 Transformed Equations

In the computational domain Ξ^n with the dependent and independent variables interchanged, the original equations (6.76) become

$$\begin{aligned} &g^{ij}\boldsymbol{x}_{\xi^i\xi^j} + p^i\boldsymbol{x}_{\xi^i} = 0 \, , \\ &g^{ij}\boldsymbol{p}_{\xi^i\xi^j} = 0 \, , \qquad i, j = 1, \ldots, n \, . \end{aligned} \tag{6.78}$$

The boundary conditions for (6.78) are, in accordance with (6.77) and (6.32),

$$\boldsymbol{x}\Big|_{\partial \Xi^n} = \boldsymbol{f}^{-1}(\boldsymbol{\xi}) \ ,$$
$$p^k\Big|_{\partial \Xi^n} = g^{ij}\Gamma^k_{ij} - c\frac{\partial \xi^k}{\partial \boldsymbol{n}} \ , \qquad i, j, k = 1, \ldots, n \ . \tag{6.79}$$

6.4 Orthogonal Systems

A system of equations suitable for generating orthogonal grids is commonly obtained in two ways. In the first approach, the system is derived from the following equations representing the condition of orthogonality:

$$g_{ij} \equiv \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j} = 0 \ , \qquad i \neq j \ .$$

The second approach is based on any differential identity which can be derived for a suitable system by eliminating the terms $g_{ij}, \ i \neq j$.

6.4.1 *Derivation from the Condition of Orthogonality*

One example of a differential system, considered by Haussling and Coleman (1981) to generate two-dimensional orthogonal and nearly orthogonal grids, is

$$\frac{\partial}{\partial \xi^1} g_{12} = 0 \ ,$$
$$\frac{\partial}{\partial \xi^2} g_{12} = 0 \ . \tag{6.80}$$

The constant solution $g_{12} = \text{const}$ to (6.80) exists only if it is consistent with the boundary data. Only at the corners of the computational region Ξ^2 can g_{12} be specified in advance. Thus, the system (6.80) is suitable for obtaining an orthogonal grid when the region X^2 has right angles at the corners. Now we change to the customary notations x, y for x^1, x^2 and ξ, η for ξ^1, ξ^2.

Expanding (6.80) yields

$$x_\xi x_{\xi\eta} + x_{\xi\xi}x_\eta + y_\xi y_{\xi\eta} + y_{\xi\xi}y_\eta = 0 \ , \tag{6.81}$$

$$x_\xi x_{\eta\eta} + x_{\xi\eta}x_\eta + y_\xi y_{\eta\eta} + y_{\xi\eta}y_\eta = 0 \ . \tag{6.82}$$

To compute the transformation $\boldsymbol{r}(\xi, \eta) : \Xi^2 \to X^2, \ \boldsymbol{r} = (x, y)$, these equations are combined as follows. The product of (6.81) and x_η is added to the product of (6.82) and x_ξ, giving

$$(x_\eta)^2 x_{\xi\xi} + (x_\xi)^2 x_{\eta\eta} + 2x_\xi x_\eta x_{\xi\eta} + x_\eta y_\eta y_{\xi\xi} + x_\xi y_\xi y_{\eta\eta} \\ +(x_\eta y_\xi + x_\xi y_\eta) y_{\xi\eta} = 0\ . \qquad (6.83)$$

The product of (6.81) and y_η is added to the product of (6.82) and y_ξ, yielding

$$(y_\eta)^2 y_{\xi\xi} + (y_\xi)^2 y_{\eta\eta} + 2y_\xi y_\eta y_{\xi\eta} + x_\eta y_\eta x_{\xi\xi} + x_\xi y_\xi x_{\eta\eta} \\ +(x_\eta y_\xi + x_\xi y_\eta) x_{\xi\eta} = 0\ . \qquad (6.84)$$

The systems (6.83) and (6.84) are approximated by central differences and the resulting algebraic systems are solved iteratively using successive overrelaxation. The reason for replacing (6.81) and (6.82) with (6.83) and (6.84) is to obtain a nonzero coefficient for x_{ij} and y_{ij} in the finite-difference forms of (6.84) and (6.84). This eliminates the possibility of dividing by zero in the iteration process.

6.4.2 Multidimensional Equations

A multidimensional differential system for generating orthogonal and nearly orthogonal grids is usually obtained by the second approach, using some differential identities and then eliminating the terms $g_{ij},\ i \neq j$. One example gives the identity (2.48),

$$\frac{\partial}{\partial \xi^j}\left(J \frac{\partial \xi^j}{\partial x^k}\right) \equiv 0\ , \qquad j, k = 1, \dots, n\ .$$

In accordance with (2.24),

$$\frac{\partial \xi^j}{\partial x^k} = g^{ij} \frac{\partial x^k}{\partial \xi^i}\ , \qquad i, j, k = 1, \dots, n\ .$$

Using this relation in the above equation, another form of the identity is obtained:

$$\frac{\partial}{\partial \xi^j}\left(J g^{ij} \frac{\partial x^k}{\partial \xi^i}\right) \equiv 0\ , \qquad i, j, k = 1, \dots, n\ , \qquad (6.85)$$

which also follows from the Beltrami equations

$$\frac{\partial^2 x^k}{\partial x^j \partial x^j} \equiv \frac{1}{J} \frac{\partial}{\partial \xi^j}\left(J g^{ij} \frac{\partial x^k}{\partial \xi^i}\right) \equiv 0\ , \qquad i, j, k = 1, \dots, n\ . \qquad (6.86)$$

Substituting the condition of orthogonality

$$g^{ij} = 0\ , \qquad i \neq j\ , \qquad i, j = 1, \dots, n,$$

in the Eq. (6.85) for g^{ij}, $i \neq j$, we obtain the system of elliptic equations required to generate orthogonal coordinates:

$$\frac{\partial}{\partial \xi^j}\left(J g^{jj} \frac{\partial x^k}{\partial \xi^j}\right) = 0 , \qquad j, k = 1, \ldots, n , \tag{6.87}$$

where

$$\begin{aligned} J &= \sqrt{\prod_{i=1}^{n} g_{ii}} , \\ g^{jj} &= 1/g_{jj} , \qquad \text{for each fixed} \qquad j = 1, \ldots, n . \end{aligned}$$

In two dimensions, using the common notations x, y for the dependent variables and ξ, η for the independent variables, the system (6.87) is expressed as

$$\begin{aligned} \frac{\partial}{\partial \xi}\left(F \frac{\partial x}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{F} \frac{\partial x}{\partial \eta}\right) &= 0 , \\ \frac{\partial}{\partial \xi}\left(F \frac{\partial y}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{F} \frac{\partial y}{\partial \eta}\right) &= 0 , \end{aligned} \tag{6.88}$$

with $F = \sqrt{g_{22}/g_{11}}$.

Analogously, for the three-dimensional system, we obtain, from (6.87),

$$\frac{\partial}{\partial \xi^i}\left(F_i \frac{\partial \mathbf{x}}{\partial \xi^i}\right) = 0 , \qquad i = 1, 2, 3 , \tag{6.89}$$

where

$$F_i = g_{kk} g_{ll}/g_{ii} , \qquad (i, k, l) \text{ cyclic and fixed} ,$$

i.e.

$$\begin{aligned} F_1 &= \sqrt{g_{22} g_{33}/g_{11}} , \\ F_2 &= \sqrt{g_{33} g_{11}/g_{22}} , \\ F_3 &= \sqrt{g_{11} g_{22}/g_{33}} . \end{aligned}$$

6.5 Hyperbolic and Parabolic Systems

Hyperbolic and parabolic methods of grid generation imply the numerical solution of hyperbolic and parabolic differential equations, respectively. Both types of system of equations are solved by marching in the direction of one selected curvilinear coordinate. These procedures are much faster than an elliptic scheme, producing a grid in an order of magnitude less computational time.

Hyperbolic grid generation relies on the numerical solution of hyperbolic systems of equations. The hyperbolic equations allow one to use a marching numerical solution without any iteration or initial guess, which makes their use very simple and inexpensive.

Hyperbolic methods are efficient for generating grids in domains around bodies. The solution marches from the inner boundary toward the outer field, generating loops of grids one by one, so the computational time is almost equal to that of one iteration of solving elliptic grid generation equations by an iterative scheme. So, the computational time required to generate the grid by the marching algorithm is only a very small fraction of that for the elliptic grid generation equations, and the fast-memory space required during grid generation can be substantially reduced from that required by the elliptic grid generation method. Furthermore, hyperbolic equations are very suitable for providing grid orthogonality and grid node clustering.

However, hyperbolic grid systems also have their inherent undesirable properties:

(1) since the hyperbolic methods are essentially a marching procedure, the specification of the entire boundary is not allowed, and therefore the methods are not appropriate for the computation of internal and closed systems;
(2) the techniques propagate singularities of the boundary into the interior of the domain;
(3) grid oscillation or even overlapping of grid lines is often encountered in hyperbolic grid generation unless artificial damping terms for stability are appropriately added to the equations.

There are two major approaches in formulating hyperbolic systems. In the first approach, the Jacobian of the transformation is specified. The second imposes a specification of the cell aspect ratio.

Parabolic methods possess some of the advantages of both elliptic and hyperbolic techniques. The advantages of using parabolic partial differential equations to generate structured grids are as follows:

(1) parabolic equations allow for formulating initial-value problems, so grids are generated through a marching algorithm, as in the hyperbolic grid generation method;
(2) the parabolic equations have most of the properties of the elliptic equations, in particular, the diffusion effect which smooths out any singularity of the inner boundary condition, and prescribed outer boundary conditions may be satisfied.

6.5.1 Specification of Aspect Ratio

The condition of orthogonality $g^{ij} = \boldsymbol{x}_{\xi^1} \cdot \boldsymbol{x}_{\xi^2} = 0$ alone is not sufficient for obtaining the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^2 \to X^2$. Two equations are needed, since both the x^1 and x^2 coordinates of the transformed grid points are to be found.

Initial-Value Problems

Here, a method presented by Starius (1977) for determining orthogonal grids, based on nonlinear hyperbolic initial-value problems which are formally related to the

Cauchy–Riemann equations, is considered. For convenience, the ordinary notations x, y for x^1, x^2 and ξ, η for ξ^1, ξ^2 are utilized in this subsubsection.

The orthogonality requirement $g_{12} = 0$ yields the initial-value problem

$$\begin{aligned} x_\eta &= -y_\xi F \ , \qquad x(\xi, 0) = x(\xi) \ , \\ y_\eta &= x_\xi F \ , \qquad y(\xi, 0) = y(\xi) \ , \end{aligned} \tag{6.90}$$

where F is a positive function which is selected by meeting the following set of requirements:

(1) $F = F(\xi, \eta, x, y, x_\xi, y_\xi, x_\eta, y_\eta)$;
(2) a condition of invariance;
(3) a condition on the hyperbolic type of the system (6.90);
(4) geometrical conditions depending on the region X^2;
(5) sufficient conditions for well-posedness of the nonlinear hyperbolic initial-value problems.

The invariance conditions are simply invariance under transitions and rotations. Let (x, y) be a solution to the equations (6.90); then, $(\overline{x}, \overline{y})$, defined by either

$$\begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}$$

or

$$\begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = Q \begin{pmatrix} x \\ y \end{pmatrix}$$

where

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

is the matrix of rotation, and a, b, θ are arbitrary constants, is also a solution. The first equation implies that F does not depend on (x, y), and the second implies the following partial differential equation for F:

$$-y_\xi F_{x_\xi} + x_\xi F_{y_\xi} - y_\eta F_{x_\eta} + x_\eta F_{y_\eta} = 0.$$

The solution to this equation is given by

$$F = F(\xi, \eta, g_{11}, g_{22}, g_{12}).$$

This is, in fact, the general solution. The last argument of this function F is zero because of the orthogonality requirement. Furthermore, by squaring (6.90) and adding, we find that g_{22} is connected with g_{11} by the relation

$$g_{22} = F^2 g_{11},$$

i.e. F is the aspect ratio. Therefore, only positive functions F depending on ξ, η and g_{11} have to be considered, i.e.

$$F = F(\xi, \eta, z)\ , \qquad z = \sqrt{g_{11}}.$$

The quantity z has an evident geometrical interpretation, namely, it is the length of the tangential vector $\boldsymbol{x}_\xi$.

In order for the initial-value problem (6.90) to be well-posed , it is necessary that the system be hyperbolic. The type of the system (6.90) is defined by the eigenvalues of the matrix

$$M = \begin{pmatrix} -\frac{x_\xi y_\xi}{z} F_z & -F - \frac{y_\xi^2}{z} F_z \\ F + \frac{x_\xi^2}{z} F_z & \frac{x_\xi y_\xi}{z} F_z \end{pmatrix}$$

obtained through the linearization of the system (6.90). The system is hyperbolic if the matrix M has real eigenvalues. The characteristic equation for M is given by

$$\lambda^2 = -F(F + zF_z) = -\frac{f f_z}{z},$$

where

$$f = Fz.$$

Thus, $f_z < 0$ if $F > 0$, and so the function $f = zF$ is strictly decreasing in z. The inequality $f_z < 0$ implies that M has distinct real eigenvalues. For $f_z = 0$, we have a multiple real eigenvalue but only one eigenvector.

Before a specific f can be chosen, the quantity z must be normalized in some sense. Let ξ be the arc length of ∂X^2; then, $z = 1$ there. This means an equidistant grid spacing on the boundary $\eta = \eta_0$. A graded grid in the tangential direction is obtained with a suitable choice of the function $\xi(t)$. Grading in the other direction can be achieved in the same way, which implies that f contains a factor depending on the direction of η. When different gradings in η are required for different values of ξ, the factor must depend on ξ as well.

In order to specify the function f, two cases are considered:

(1) the spacing of the mesh in the η direction is about the same along the whole boundary;
(2) the spacing is variable.

The spacing of the mesh along the η curve is expressed by

$$\int_0^{\eta_0} \sqrt{g_{22}}\, \mathrm{d}\eta = \int_0^{\eta_0} f\, \mathrm{d}\eta$$

for all ξ. From this equation, it is seen that in the first case, f does not need to depend explicitly on ξ, i.e. it will depend only on z.

When solving nonlinear hyperbolic problems, discontinuities and shocks generally appear in the solution or in its derivatives; f must be chosen so that this cannot happen as long as $z > 0$. It was proved by Starius that a suitable f is

$$f(z) = \frac{A + Bz^2}{C + z^2}, \tag{6.91}$$

where A, B, and C are constants such that $A > BC$, and $A + B = C + 1$. Since $f(0) = A/C$, the divergence factor in the η direction from the outer to the inner boundary of the curvilinear mesh can never exceed this quantity.

For the second case, it was assumed by Starius that

$$f(\xi, z) = a(\xi) f_0(\xi, z),$$

where $a(\xi)$ is a periodic function in ξ such that $0 < a \leq 1$. By using this f in (6.90), we can see that $a(\xi)$ is a relative measure of the grid spacing in the η direction.

6.5.2 Specification of Jacobian

Orthogonal Grids in Two Dimensions

In the two-dimensional case, the hyperbolic coordinate equations obtained by specifying the Jacobian and a measure of orthogonality have the form

$$\begin{aligned} &\boldsymbol{x}_{\xi^1} \cdot \boldsymbol{x}_{\xi^2} = 0\ , \\ &|\boldsymbol{x}_{\xi^1} \times \boldsymbol{x}_{\xi^2}| = J\ , \end{aligned} \tag{6.92}$$

that is,

$$\begin{aligned} &\frac{\partial x^1}{\partial \xi^1}\frac{\partial x^1}{\partial \xi^2} + \frac{\partial x^2}{\partial \xi^1}\frac{\partial x^2}{\partial \xi^2} = 0\ , \\ &\frac{\partial x^1}{\partial \xi^1}\frac{\partial x^2}{\partial \xi^2} - \frac{\partial x^1}{\partial \xi^2}\frac{\partial x^2}{\partial \xi^1} = J\ , \end{aligned} \tag{6.93}$$

where J is a specified area source term. These equations form a system of nonlinear partial differential equations whose solution is based upon solving the system

$$A\boldsymbol{x}_{\xi^1} + B\boldsymbol{x}_{\xi^2} = \boldsymbol{f}, \tag{6.94}$$

where

$$A = \begin{pmatrix} \frac{1}{\sqrt{g^0_{11}g^0_{22}}}\frac{\partial x^1_0}{\partial \xi^2} - \frac{1}{g^0_{11}}\frac{\partial x^1_0}{\partial \xi^1} & \frac{1}{\sqrt{g^0_{11}g^0_{22}}}\frac{\partial x^2_0}{\partial \xi^2} - \frac{1}{g^0_{11}}\frac{\partial x^2_0}{\partial \xi^1} \\ \frac{\partial x^2_0}{\partial \xi^2} & -\frac{\partial x^1_0}{\partial \xi^2} \end{pmatrix},$$

$$B = \begin{pmatrix} \frac{1}{\sqrt{g^0_{11}g^0_{22}}}\frac{\partial x^1_0}{\partial \xi^1} - \frac{1}{g^0_{22}}\frac{\partial x^1_0}{\partial \xi^2} & \frac{1}{\sqrt{g^0_{11}g^0_{22}}}\frac{\partial x^2_0}{\partial \xi^1} - \frac{1}{g^0_{22}}\frac{\partial x^2_0}{\partial \xi^2} \\ -\frac{\partial x^2_0}{\partial \xi^1} & \frac{\partial x^1_0}{\partial \xi^1} \end{pmatrix},$$

$$\boldsymbol{x} = (x_1, x_2)^T \ , \qquad \boldsymbol{f} = (0,\ J + J_0)^T \ .$$

Equation (6.94) represents the linearization of (6.93) about the state $\boldsymbol{x}_0$. Taking ξ^2 as a marching direction, we obtain, from (6.94),

$$\boldsymbol{x}_{\xi^2} + B^{-1}A\boldsymbol{x}_{\xi^1} = B^{-1}\boldsymbol{f} \ . \tag{6.95}$$

For the eigenvalues $\lambda_1,\ \lambda_2$ of the two-dimensional metric $B^{-1}A$, we have

$$\begin{aligned} &\lambda_1\lambda_2 = \det(B^{-1}A), \\ &\lambda_1 + \lambda_2 = \operatorname{tr}(B^{-1}A). \end{aligned}$$

As

$$\begin{aligned} &\det(B^{-1}A) = F = |\boldsymbol{x}_{\xi^2}|/|\boldsymbol{x}_{\xi^1}| = \sqrt{g_{22}/g_{11}}, \\ &\operatorname{tr}(B^{-1}A) = 0, \end{aligned}$$

we obtain $\lambda_1 = F,\ \lambda_2 = -F$. Hence, the system (6.95) is hyperbolic and the local solution consists of one left- and one right-running wave. Equation (6.95) is typically modified by adding an artificial term $\epsilon\boldsymbol{x}_{\xi^1\xi^1}$ to stabilize the numerical scheme:

$$\boldsymbol{x}_{\xi^2} + B^{-1}A\boldsymbol{x}_{\xi^1} + \epsilon\boldsymbol{x}_{\xi^1\xi^1} = B^{-1}\boldsymbol{f}. \tag{6.96}$$

Two-Dimensional Nonorthogonal Grids

A more general hyperbolic system which does not include any constraints on the angle θ between the tangential vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$ is obtained from the identities

$$\begin{aligned} &g_{12} = \sqrt{g_{11}g_{22}}\cos\theta \ , \\ &\sqrt{g_{11}g_{22}}\sin\theta = J \ , \end{aligned} \tag{6.97}$$

where θ and J can be user-specified. Choosing the ξ^2 direction to be the marching direction and solving the system (6.96) for $\partial x^1/\partial\xi^2,\ \partial x^2/\partial\xi^2$, we obtain

$$\begin{aligned}\frac{\partial x^1}{\partial \xi^2} &= \sqrt{\frac{g_{22}}{g_{11}}}\Big(\frac{\partial x^1}{\partial \xi^1}\cos\theta - \frac{\partial x^2}{\partial \xi^1}\sin\theta\Big)\,,\\ \frac{\partial x^2}{\partial \xi^2} &= \sqrt{\frac{g_{22}}{g_{11}}}\Big(\frac{\partial x^2}{\partial \xi^1}\cos\theta + \frac{\partial x^1}{\partial \xi^1}\sin\theta\Big).\end{aligned} \tag{6.98}$$

The linearization of these equations produces the system (6.94) with

$$A = \begin{pmatrix} \dfrac{1}{\sqrt{g^0_{11}g^0_{22}}}\dfrac{\partial x^1_0}{\partial \xi^2} - \dfrac{1}{g^0_{11}}\cos\theta\dfrac{\partial x^1_0}{\partial \xi^1} & \dfrac{1}{\sqrt{g^0_{11}g^0_{22}}}\dfrac{\partial x^2_0}{\partial \xi^2} - \dfrac{1}{g^0_{11}}\cos\theta_0\dfrac{\partial x^2_0}{\partial \xi^1} \\ \dfrac{\partial x^2_0}{\partial \xi^2} & -\dfrac{\partial x^1_0}{\partial \xi^2} \end{pmatrix},$$

$$B = \begin{pmatrix} \dfrac{1}{\sqrt{g^0_{11}g^0_{22}}}\dfrac{\partial x^1_0}{\partial \xi^1} - \dfrac{1}{g^0_{22}}\cos\theta\dfrac{\partial x^1_0}{\partial \xi^2} & \dfrac{1}{\sqrt{g^0_{11}g^0_{22}}}\dfrac{\partial x^2_0}{\partial \xi^1} - \dfrac{1}{g^0_{22}}\cos\theta_0\dfrac{\partial x^2_0}{\partial \xi^2} \\ -\dfrac{\partial x^2_0}{\partial \xi^1} & \dfrac{\partial x^1_0}{\partial \xi^1} \end{pmatrix},$$

$$\boldsymbol{f} = (\cos\theta + \cos\theta_0,\ J + J_0)^T.$$

The matrix B^{-1} exists when $\sin\theta \neq 0$, and

$$\lambda = \pm\sqrt{\frac{g_{22}}{g_{11}}}.$$

Hence, the system (6.98) in this case is also hyperbolic.

The introduction of the angle θ into the system (6.98) allows one to solve the initial-value problem, i.e. to specify grid data on the initial boundary $\xi^2 = 0$ and the side boundaries $\xi^1 = 0,\ \xi^1 = 1$. For (6.98), the boundary curves $\xi^2 \rightarrow \boldsymbol{x}(\xi^1_0, \xi^2),\ \xi^1_0 = 0$ or $\xi^1_0 = 1$, need not intersect the initial curve $\xi^2 = 0$ orthogonally, and so the initial-value problem is typically ill-posed. Equation (6.98), however, give an opportunity to choose the angle terms near the boundary so that a consistent problem results.

Three-Dimensional Version

The three-dimensional hyperbolic grid generation approach, where the marching direction is, say, ξ^3, is based on two orthogonality relations and an additional equation to control the Jacobian as follows:

$$\begin{aligned}
&\boldsymbol{x}_{\xi^1}\cdot\boldsymbol{x}_{\xi^3}=0\ ,\\
&\boldsymbol{x}_{\xi^2}\cdot\boldsymbol{x}_{\xi^3}=0\ ,\\
&\det\left\{\frac{\partial x^i}{\partial\xi^j}\right\}=J\ ,\qquad i,j=1,2,3\ .
\end{aligned}\tag{6.99}$$

The local linearization of the system (6.99) with respect to $(\boldsymbol{x}-\boldsymbol{x}_0)$, where $\boldsymbol{x}_0(\xi)$ is a known state, neglecting products of small quantities that are second order in $(\boldsymbol{x}-\boldsymbol{x}_0)$, yields

$$A_0(\boldsymbol{x}-\boldsymbol{x}_0)_{\xi^1}+B_0(\boldsymbol{x}-\boldsymbol{x}_0)_{\xi^2}+C_0(\boldsymbol{x}-\boldsymbol{x}_0)_{\xi^3}=\boldsymbol{f},\tag{6.100}$$

where $\boldsymbol{x}=(x^1,x^2,x^3)^T,\ \boldsymbol{x}_0=(x_0^1,x_0^2,x_0^3)^T,\ A_0,\ B_0$, and C_0 are coefficient matrices that are evaluated from $\boldsymbol{x}_0(\boldsymbol{\xi})$, and the subscripts $s^i,\ i=1,2,3$, denote partial derivatives.

6.5.3 Parabolic Equations

The parabolic grid approach lies between the elliptic and hyperbolic ones.

The two-dimensional parabolic grid generation equation where the marching direction is ξ^2 may be written in the following form:

$$\boldsymbol{x}_{\xi^2}=A_1\boldsymbol{x}_{\xi^1\xi^1}-B_1\boldsymbol{x}+\boldsymbol{S},\tag{6.101}$$

where $A_1,\ B_1$, and B are matrix coefficients, and $\boldsymbol{S}$ is a source vector that contains the information about the outer boundary configuration. Analogously, the three-dimensional parabolic equations may be written as follows:

$$\boldsymbol{x}_{\xi^3}=A_i\boldsymbol{x}_{\xi^i\xi^i}-B_1\boldsymbol{x}+\boldsymbol{S}\ ,\qquad i=1,2\ .\tag{6.102}$$

6.5.4 Hybrid Grid Generation Scheme

The combination of the hyperbolic and parabolic schemes into a single scheme is attractive because it can use the advantages of both schemes. These advantages are: first, it is a noniterative scheme; second, the orthogonality of the grid near the initial boundary is well controlled; and third, the outer boundary can be prescribed.

A hybrid grid generation scheme in two dimensions for the particular marching direction ξ^2 can be derived by combining (6.95) and (6.101), in particular, as the sum of Eqs. (6.95) and (6.100) multiplied by weights α and $1-\alpha$, respectively:

$$\begin{aligned}\alpha(B^{-1}A\boldsymbol{x}_{\xi^1}+\boldsymbol{x}_{\xi^2})+(1-\alpha)(\boldsymbol{x}_{\xi^2}-A_1\boldsymbol{x}_{\xi^1\xi^1}+B_1\boldsymbol{x})\\ =\alpha B^{-1}\boldsymbol{f}+(1-\alpha)\boldsymbol{S}\ .\end{aligned}\qquad(6.103)$$

The parameter α can be changed as desired to control the proportions of the two methods. If α approaches 1, (6.103) becomes the hyperbolic grid generation equation, while if α approaches zero, it becomes the parabolic grid generation equation. In practical applications, α is set to 1, when the grid generation starts from the initial boundary curve $\xi^2=0$, but it gradually decreases and approaches zero when the grid reaches the outer boundary.

An analogous combination of (6.100) and (6.102) can be used to generate three-dimensional grids through a hybrid of parabolic and hyperbolic equations.

6.6 Grid Equations for Nonstationary Problems

In the finite-difference techniques developed for the numerical solution of nonstationary problems, the time variable t must be discretized in order to provide computation of all physical variables at each time slice t^n. As the physical solution is dependent on the variable t, it is reasonable to adjust the grid to the solution to follow the trajectories of severe variations in the physical quantities. The goal is to compute as accurately as desired all physical quantities of interest at each grid location in space and time. As a result, the placement of the grid points obtained with such an adjustment depends on the time t. The motion of the grid points is demonstrated in the transformed equations by the grid velocity $\boldsymbol{x}_t$ appearing in the transformed time derivative. In the process of numerical solution, the grid speed is either found using the differences of the values of the function $\boldsymbol{x}(\boldsymbol{\xi},\ t)$ on the $(n+1)$th and nth layers or specified in advance.

This section reviews some techniques aimed at the generation of nonstationary grids for time-dependent partial differential equations which, in addition to spatial derivatives, also contain derivatives with respect to the time variable t. An example is the following form:

$$\frac{\partial \boldsymbol{u}}{\partial t}=L(\boldsymbol{u},\boldsymbol{x},t)\ ,\qquad \boldsymbol{x}\in X^n\ ,\qquad t>0\ ,\qquad(6.104)$$

where L is a differential operator involving spatial derivatives.

The accuracy of the numerical solution of time-dependent partial differential equations is significantly dependent on the time-step size as well as the spatial mesh size. However, high resolution in local regions of large solution variations allows larger time steps to be taken.

There are two basic approaches to solving nonstationary problems: the method of lines and the methods of moving grids.

6.6.1 Method of Lines

The idea of the method of lines consists in converting nonstationary equations into a system of ordinary differential equations with time as the independent variable. This is carried out by first approximating only the space derivatives on a specified space grid chosen a priori. This operation yields a system of ordinary differential equations with respect to the independent variable t. The system obtained is then approximated on a one-dimensional grid, generally nonuniform, discretizing the variable t. In this approach, the spatial grid is fixed, and therefore it is applicable only to the numerical solution of problems with stationary solution singularities. However, many solutions to time-dependent problems have large variations which change with time. Moreover, the solutions may have narrow layers of rapid change with respect to both space and time, for example, a solution of the simple running-wave form

$$u(x, t) = f(x - ct) \ ,$$

where $f(y)$ is a scalar function, of boundary or interior-layer type.

6.6.2 Moving-Grid Techniques

There are two major approaches to controlling the movement of the points of the numerical grid with time. With the first, stationary grids are constructed through the same method on each time layer but the grid points move over time because of changes either of the control functions in the equation or of the boundary conditions for the coordinate transformations applying on the boundary of the computation region Ξ^n. With the second, an equation for the velocity $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$ of grid point movement is determined and the boundary value problem is then solved for these equations.

Specification of Spatial Grid Distribution

The simplest way to generate grids for nonstationary problems is to use the equidistribution approach with a time-dependent weight function $w(x, t)$, for instance, in the form

$$\int_{x_i}^{x_{i+1}} w(x, t)\mathrm{d}x = c(t) = \frac{1}{N}\int_a^b w(x, t)\mathrm{d}x \ , \qquad i = 0, \ldots, N - 1 \ .$$

This form can be readily reformulated as a boundary value problem of the type (6.69). In particular, in the case of one-dimensional nonstationary problems, it may be realized by the replacement of the variables $x,\ t$ by new variables $\xi,\ t$, where ξ is an arc-length-like coordinate, i.e. the result is the equidistant mesh. The advantage of this procedure is that in the variables $\xi,\ t$, the solution $u(\xi, t)$ cannot develop large spatial gradients $\partial u/\partial \xi$.

Grid Movement Induced by Boundary Movement

Equations of the type (6.15) in which the control functions P^i depend on t are often used at each time step of the solution of gas-dynamics problems, the movement of grid nodes on the boundary of the region being determined by the motion of the medium; in particular, the velocity of the grid and the velocity of the medium are the same on the boundary. The movement of interior points in this approach depends solely upon the boundary motion. The boundaries of the regions may be shock waves, interior boundaries which separate the different regions of flow, or free surfaces.

Algebraic methods of constructing moving grids by interpolation from a moving boundary can also be considered in the same fashion.

Specification of Grid Speed

In multidimensional nonstationary problems, it is often necessary to generate adaptive grids which are adjusted to a moving solution of a problem. As a result, the grid distribution is also nonstationary, that is, the derivative $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$, which represents the speed of the grid movement , is nonzero in the general case. The expression $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$ appears in the equations for nonstationary physical problems rewritten in the independent variables $\boldsymbol{\xi}, \ t$.

This subsubsection delineates some typical approaches for formulating grid equations by specification of the velocity of grid point movement to treat nonstationary problems. In our considerations, we discuss a simplified form of the transformation $\boldsymbol{x}(\boldsymbol{\xi}, \tau)$, assuming $t = \tau$. In this case, the first temporal derivatives $\partial/\partial t$ and $\partial/\partial \tau$ are subject to the relation

$$\frac{\partial}{\partial \tau} = \frac{\partial}{\partial t} + \frac{\partial}{\partial x^i}\frac{\partial x^i}{\partial \tau} \ .$$

Thus, assuming the identification $t = \tau$, (7.44) is transformed into

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\frac{\partial \boldsymbol{u}}{\partial x^i}\frac{\partial x^i}{\partial t} + L[\boldsymbol{u}, \boldsymbol{x}(\boldsymbol{\xi}, t), t] \ , \qquad \boldsymbol{\xi} \in \Xi^n \ , \qquad t > 0 \ , \tag{6.105}$$

with $\boldsymbol{u} = \boldsymbol{u}[\boldsymbol{x}(\boldsymbol{\xi}, t), t]$.

The simplest way to obtain grid equations which include the speed of grid movement is to add the term $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$ to the equations developed for the generation of fixed grids, in particular, to the inverted Poisson equations (6.16).

The equations for the grid node velocities $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$ can be determined, for example, from the conditions for minimizing functionals of measure of the deviation from Lagrangian properties. These functionals will be discussed in Chap. 7.

It is often proposed to solve nonstationary problems by determining equations for $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$ from the condition that the solution to the physical problem in the new variables $(\boldsymbol{\xi}, t)$ is stationary. Also, the grid velocity can be formulated on the basis of providing conditions of stability in the difference scheme in nonstationary problems.

In one more approach, for the equations for multidimensional problems of the mechanics of a continuous medium, the equations for the grid velocities are determined from the condition that the convective terms are either a minimum or zero. For example, if the operator $L(\boldsymbol{u}, \boldsymbol{x}, t)$ in (6.104) has the form

$$L(\boldsymbol{u}, \boldsymbol{x}, t) = a^i \frac{\partial \boldsymbol{u}}{\partial x^i} + L_1(\boldsymbol{u}, \boldsymbol{x}, t) \ , \qquad i = 1, \dots, n \ ,$$

where L_1 is an operator without first derivatives, then (6.105) in the new coordinates $t, \xi^1, \dots, \xi^n$ determined by the transformation $\boldsymbol{x}(\boldsymbol{\xi}, t)$ is expressed as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} = \left(a^i - \frac{\partial x^i}{\partial t}\right) + L_1[\boldsymbol{u}, \boldsymbol{x}(\boldsymbol{\xi}, t), t] \ , \qquad i = 1, \dots, n \ . \tag{6.106}$$

Thus, the condition for zero convective terms in (6.106) gives the following equations for the components of the grid velocity $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$:

$$\frac{\partial x^i}{\partial t} = a^i \ , \qquad i = 1, \dots, n \ .$$

In such a way, for example, the Lagrangian coordinates are generated, so the approach aimed at the elimination of convective terms in the transformed equations by nonstationary coordinate mappings is referred to as the Lagrangian method.

The inverted Poisson equations (6.16), formally differentiated with respect to t, are also used to obtain equations for $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$. Also, differentation with respect to time of the equations modeling the equidistribution principle gives differential equations for the grid node velocity. This operation provides the necessary grid velocity equations, which are then integrated to obtain the grid motion as a function of time.

6.6.3 Time-Dependent Deformation Method

The deformation method for generating multidimensional structured grids in an n-dimensional domain X^n coinciding with the computational domain Ξ^n can be extended to produce time-dependent grids with nonstationary weight functions. This subsection describes such an extension, using for this purpose weight functions $w(\boldsymbol{x}, t) > 0$ that satisfy the following normalization properties:

$$\int_{\Xi^n} \left(\frac{1}{w} - 1\right) \mathrm{d}\boldsymbol{\xi} = 0 \ , \tag{6.107}$$

and $w(\boldsymbol{x}, t_0) = 1$ for all $\boldsymbol{x} \in \Xi^n$. The nonstationary transformation $\boldsymbol{x}(\boldsymbol{\xi}, t)$ satisfying the equidistribution condition is found from the following initial-value problem for ordinary differential equations, formulated for each fixed point $\boldsymbol{\xi}$ in Ξ^n :

$$\begin{aligned}\frac{\partial}{\partial t}\boldsymbol{x}(\boldsymbol{\xi},t) &= \boldsymbol{v}[\boldsymbol{x}(\boldsymbol{\xi},t),t]w[\boldsymbol{x}(\boldsymbol{\xi},t),t]\ , \qquad t > t_0\ ,\\ \boldsymbol{x}(\boldsymbol{\xi},t_0) &= \xi\ , \qquad t = t_0\ ,\end{aligned} \tag{6.108}$$

where the vector field $\boldsymbol{v}(\boldsymbol{x},t)$ is computed from the following system:

$$\begin{aligned}\mathrm{div}_x\boldsymbol{v}(\boldsymbol{x},t) &= -\frac{\partial}{\partial t}\Big(\frac{1}{w(\boldsymbol{x},t)}\Big) \quad \text{in} \quad \Xi^n\ ,\\ \mathrm{curl}\ \boldsymbol{v}(\boldsymbol{x},t) &= 0 \quad \text{in} \quad \Xi^n\ ,\\ \boldsymbol{v}\cdot\boldsymbol{n} &= 0 \quad \text{on} \quad \partial\Xi^n\ .\end{aligned} \tag{6.109}$$

We consider the derivative with respect to t of the function

$$H(\boldsymbol{\xi},t) = Jw[\boldsymbol{x}(\boldsymbol{\xi},t),t]\ ,$$

where $J = \det(\partial x^i/\partial\xi^j)$. Since the relation (2.88) is valid for the transformation $\boldsymbol{x}(\boldsymbol{\xi},t)$, we obtain

$$\frac{\partial H}{\partial t} = J\Big(w\ \mathrm{div}_x\frac{\partial\boldsymbol{x}}{\partial t} + \frac{\partial w}{\partial x^i}\frac{\partial x^i}{\partial t} + \frac{\partial w}{\partial t}\Big)\ .$$

Application of (6.107) and (6.108) to this equation yields

$$\begin{aligned}\frac{\partial H}{\partial t} &= J\Big[w\ \mathrm{div}_x\Big(\frac{\boldsymbol{v}}{w}\Big) + \frac{\partial w}{\partial x^i}\frac{v^i}{w} + \frac{\partial w}{\partial t}\Big]\\ &= J\Big(-\frac{\partial w}{\partial t} - \frac{v^i}{w}\frac{\partial w}{\partial x^i} + \frac{\partial w}{\partial x^i}\frac{v^i}{w} + \frac{\partial w}{\partial t}\Big) = 0\ .\end{aligned}$$

Since $H(\boldsymbol{\xi},t_0) = 1$, we obtain

$$H(\boldsymbol{\xi},t) = Jw[\boldsymbol{x}(\boldsymbol{\xi},t),t] = 1\ ,$$

i.e. the multidimensional equidistribution principle is obeyed by the transformation $\boldsymbol{x}(\boldsymbol{\xi},t)$.

6.7 Comments

A two-dimensional Laplace system (6.4) which implied the physical coordinates to be solutions in the logical domain Ξ^2 was introduced by Godunov and Prokopov (1967), Barfield (1970), and Amsden and Hirt (1973). A general two-dimensional elliptic system of the type (6.2) for generating structured grids was considered by Chu (1971).

A two-dimensional Laplace system (6.5) using the logical coordinates ξ^i as dependent variables was proposed by Crowley (1962) and Winslow (1967). The technique presented in this chapter to analyzing the qualitative behavior near boundary segments of the coordinate lines obtained through the inverted Laplace equations was introduced by the author of this book. A rather geometric approach was described for this purpose in the monograph by Thompson et al. (1985). A more detailed analysis of grid behavior near the boundary of both surfaces and two- and three-dimensional domains was performed by Liseikin (2004, 2007).

Godunov and Prokopov (1972) obtained a system of the Poisson type (6.15) assuming that its solution is a composition of conformal and stretching transformations. The general Poisson system presented in the current book was justified by Thompson et al. (1974) and Thompson et al. (1985) in their monograph.

The algorithm aimed at grid clustering at a boundary and forcing grid lines to intersect the boundary in a nearly normal fashion through the source terms of the Poisson system was developed by Steger and Sorenson (1979), Visbal and Knight (1982), and White (1990). Thomas and Middlecoff (1980) described a procedure for controlling the local angle of intersection between transverse grid lines and the boundary through the specification of the control functions. Control of grid spacing and orthogonality was performed by Tamamidis and Assanis (1991) by introducing a distortion function (the ratio of the diagonal metric elements) into the system of Poisson equations. Warsi (1982) replaced the source terms P^i in (6.15) by $g^{ii}P^i$ (i fixed) to improve the numerical behavior of the generator. As a result, the modified system acquired the property of satisfying the maximum principle.

The technique based on setting to zero the off-diagonal elements of the elliptic system was proposed by Lin and Shaw (1991) to generate nearly orthogonal grids, while Soni et al. (1993) used a specification of the control functions for this purpose.

A composition of Poisson's and Laplace's equations in the computational domain to derive biharmonic equations of fourth order was used to generate smooth block-structured grids via the specification of grid line slopes and boundary point distributions by Bell et al. (1982). Schwarz (1986) used for this purpose equations of sixth order, which were composed of Poisson and Laplace systems with respect to the dependent physical coordinates x^i. An alternative method, based on the solution of biharmonic equations in the physical domain, was introduced by Sparis (1985). A recent implementation of the biharmonic equations to provide boundary orthogonality and off-boundary spacing as boundary conditions was presented by Sparis and Karkanis (1992).

A combination of elliptic and algebraic techniques was applied by Spekreijse (1995) to generate two- and three-dimensional grids. An approach to formulating an orthogonal system by differentiating nondiagonal metric elements was developed by Haussling and Coleman (1981). Ryskin and Leal (1983), in two dimensions, and Theodoropoulos and Bergeles (1989) in three dimensions, have developed elliptic methods for nearly orthogonal grid generation.

The first systematic analysis of the use of two-dimensional hyperbolic equations to generate orthogonal grids was made by Starius (1977) and Steger and Chaussee (1980), although hyperbolic grid generation can be traced back to McNally (1972). This system was generalized by Cordova and Barth (1988). They developed a two-dimensional hyperbolic system with an angle-control source term which allows one to constrain a grid with more than one boundary. A combination of grids using the hyperbolic technique of Steger and Chaussee (1980), which starts from each boundary segment, was generated by Jeng and Shu (1995). The extension to three dimensions was performed by Steger and Rizk (1985), Chan and Steger (1992), and Tai et al. (1996), who introduced grid smoothing as well.

The generation of grids based on a parabolic scheme approximating the inverted Poisson equations was first proposed for two-dimensional grids by Nakamura (1982). A variation of Nakamura's method was developed by Noack (1985) to use in space-marching solutions to the Euler equations. Extensions of this parabolic technique for generating solution adaptive grids were performed by Edwards (1985) and Noack and Anderson (1990).

A combination of hyperbolic and parabolic schemes that uses the advantages of the two but eliminates the drawbacks of each was proposed by Nakamura and Suzuki (1987).

The idea of defining the control functions of the Poisson system through weight functions was formulated by Anderson (1983, 1987) and extended by Eiseman (1987). Some versions of the specification of the control functions through sums of derivatives of physical quantities and quality measures of the domain geometry were presented by Dannenhoffer (1990), Kim and Thompson (1990), Tu and Thompson (1991), Soni (1991), and Hall and Zingg (1995), while Hodge et al. (1987) applied analytical expressions for this purpose.

A deformation method for generating multidimensional unfolded grids has been developed by Liao and Anderson (1992) and Semper and Liao (1995) on the basis of a deformation scheme originally introduced by Moser (1965). This deformation method was improved and applied to practical problems by Wan and Turek (2006, 2007), Chu et al. (2008), and Liao et al. (2008).

A large number of important moving-grid methods for the numerical solution of unsteady equations can be found in the survey by Hawken et al. (1991) and in the monograph by Zegeling (1993).

The movement of the nonstationary grid considered by Godunov and Prokopov (1972) was caused by the boundary point speeds, while in the interior of the domain, the grid nodes were found by solving the inverted Poisson system. Hindman et al. (1981) formulated the grid motion in time by taking the time derivative of the inverted Poisson equations.

One more approach involving grid speed equations, but based on time differentiation of a more complicated set of Euler–Lagrange equations derived from the minimization of a Brackbill–Saltzman-type functional, was presented by Slater et al. (1995).

A strategy for automatic time step selection based on equidistributing the local truncation error in both the time and the space discretization was proposed by Chen et al. (1993).

White (1979) suggested a technique for numerically integrating systems of time-dependent first-order partial differential equations in one space variable x. His technique replaces the variables $x,\ t$ with the new variables $s,\ t$, where s is an arc-length-like coordinate.

Some questions arising from coupling upwinding schemes with moving equidistributed meshes were discussed by Li and Petzold (1997). The stability problems related to an equidistributed mesh of the systems of differential equations for the grid velocities were studied by Coyle et al. (1986).

One approach to generating one-dimensional time-dependent grids was proposed by Dar'in et al. (1988) with the help of the system of evolutionary equations

$$\begin{aligned}\frac{\partial x(\xi,t)}{\partial \xi} &= \Psi\ ,\\ \frac{\partial \Psi}{\partial t} &= -\frac{\partial P}{\partial \xi}\ .\end{aligned}$$

In order to concentrate the grid nodes in the high-gradient zones, various expressions for P containing the derivatives with respect to ξ of the physical quantities were considered. This approach was extended to the construction of two-dimensional adaptive grids by Dar'in and Mazhukin (1989).

The method of equidistribution and minimization of the heuristically determined error at each time step was used for calculations of nonstationary problems by Dorfi and Drury (1987), Dwyer et al. (1980), Klopfer and McRae (1981), Miller (1983), Wathen (1990), and White (1982).

Formal addition of the velocity function $\boldsymbol{x}_t(\boldsymbol{\xi}, t)$ to (6.14), to the equations obtained from variational methods based on the minimization of grid quality functionals, or to expressions for the errors determined heuristically in terms of the spatial derivatives was analyzed by Rai and Anderson (1981), Bell et al. (1982), Harten and Hyman (1983), and Greenberg (1985).

Various physical analogies, such as those of springs (Bell and Shubin (1983), Rai and Anderson (1982)), chemical reactions (Greenberg (1985)) and concepts from continuum mechanics (Jacquotte (1987), Knupp (1995)) have also been used to construct moving adaptive grids.

References

Amsden, A. A., & Hirt, C. W. (1973). A simple scheme for generating general curvilinear grids. *Journal of Computational Physics*, *11*, 348–359.

Anderson, D. A. (1983). Adaptive grid methods for partial differential equations. In K. N. Ghia & U. Ghia (Eds.), *Advances in Grid Generation* (pp. 1–15). Houston: ASME.

Anderson, D. A. (1987). Equidistribution schemes, Poisson generators, and adaptive grids. *Applied Mathematics and Computation*, *24*, 211–227.

Barfield, W. D. (1970). An optimal mesh generator for Lagrangian hydrodynamic calculations in two space dimensions. *Journal of Computational Physics*, *6*, 417–429.

Bell, J. B., Shubin, G. R., & Stephens, A. B. (1982). A segmentation approach to grid generation using biharmonics. *Journal of Computational Physics*, *47*(3), 463–472.

Bell, J. B., & Shubin, G. R. (1983). An adaptive grid finite-difference method for conservation laws. *Journal of Computational Physics*, *52*, 569–591.

Chan, W. M., & Steger, L. G. (1992). Enhancement of a three-dimensional hyperbolic grid generation scheme. *Applied Mathematics and Computation*, *51*(1), 181–205.

Chen, K., Baines, M. J., & Sweby, P. K. (1993). On an adaptative time stepping strategy for solving nonlinear diffusion equations. *Journal Computational of Physics*, *105*, 324–332.

Chu, W. H. (1971). Development of a general finite difference approximation for a general domain. *Journal of Computational Physics*, *8*, 392–408.

Cordova, J. Q., & Barth, T. J. (1988). Grid generation for general 2-D regions using hyperbolic equations, AIAA Paper 88-0520.

Coyle, J. M., Flaherty, J. E., & Ludwig, R. (1986). On the stability of mesh equidistribution strategies for time-dependent partial differential equations. *Journal of Computational Physics*, *62*, 26–39.

Crowley, W. P. (1962): An equipotential zoner on a quadrilateral mesh. Memo, Lawrence Livermore National Lab., 5 July 1962.

Dannenhoffer, J. F. (1990). A comparison of adaptive-grid redistribution and embedding for steady transonic flows. AIAA Paper 90-1965.

Dar'in, N., & Mazhukin, V. I. (1989). Mathematical modelling of non-stationary two-dimensional boundary-value problems on dynamic adaptive grids. *Matematicheskoe Modelirovanie*, *1*(3), 29–30. (Russian).

Dar'in, N., Mazhukin, V. I., & Samarskii, A. A. (1988). Finite-difference solution of gas dynamics equations using adaptive grids which are dynamically associated with the solution. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, *28*, 1210–1225. (Russian).

Dorfi, E. A., & Drury, L. O'C. (1987). Simple adaptive grids for 1-D initial value problems. *Journal of Computational Physics*, *69*, 175–195.

Dwyer, H. A., Kee, R. J., & Sanders, B. R. (1980). Adaptive grid method for problems in fluid mechanics and heat transfer. *AIAA Journal*, *18*(10), 1205–1212.

Edwards, T. A. (1985). Noniterative three-dimensional grid generation using parabolic partial differential equations. AIAA Paper 85-0485.

Eiseman, P. R. (1987). Adaptive grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 321–376.

Farrel, F. T., & Jones, L. E. (1996). Some non-homeomorphic harmonic homotopy equivalences. *Bulletin of the London Mathematical Society*, *28*, 177–182.

Godunov, S. K., & Prokopov, G. P. (1967). Calculation of conformal mappings in the construction of numerical grids. *Journal of Computational Mathematics and Mathematical Physics*, *7*, 1031–1059. (Russian).

Godunov, S. K., & Prokopov, G. P. (1972). On utilization of moving grids in gasdynamics computations. *J. Vychisl. Matem. Matem. Phys. 12*, 429–440 (Russian) [English transl.: USSR Computational Mathematics and Mathematical Physics *12* (1972), 182–195].

Greenberg, J. B. (1985). A new self-adaptive grid method. *AIAA Journal*, *23*, 317–320.

Hall, D. J., & Zingg, D. W. (1995). Viscous airfoil computations adaptive grid redistribution. *AIAA Journal*, *33*(7), 1205–1210.

Harten, A., & Hyman, J. M. (1983). A self-adjusting grid for the computation of weak solutions of hyperbolic conservation laws. *Journal of Computational Physics*, *50*, 235–269.

Haussling, H. J., & Coleman, R. M. (1981). A method for generation of orthogonal and nearly orthogonal boundary-fitted coordinate systems. *Journal of Computational Physics*, *43*, 373–381.

Hawken, D. F., Gottlieb, J. J., & Hansen, J. S. (1991). Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations. *Journal of Computational Physics*, *95*, 254–302.

Hindman, R. G., Kutler, P., & Anderson, D. (1981). Two-dimensional unsteady Euler equation solver for arbitrary shaped flow regions. *AIAA Journal*, *19*(4), 424–431.

Hodge, J. K., Leone, S. A., & McCarry, R. L. (1987). Non-iterative parabolic grid generation for parabolized equations. *AIAA Journal*, *25*(4), 542–549.

Jacquotte, O.-P. (1987). A mechanical model for a new grid generation method in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, *66*, 323–338.

Jeng, Y. N., & Shu, Y.-L. (1995). Grid combination method for hyperbolic grid solver in regions with enclosed boundaries. *AIAA Journal*, *33*(6), 1152–1154.

Kim, H. J., & Thompson, J. F. (1990). Three-dimensional adaptive grid generation on a composite block grid. *AIAA Journal*, *28*, 470–477.

Klopfer, G. H., & McRae, D. D. (1981). The nonlinear modified equation approach to analysing finite difference schemes. AIAA Paper 81-1029.

Knupp, P. M. (1995). Mesh generation using vector-fields. *Journal of Computational Physics*, *119*, 142–148.

Liao, G. (1991). On harmonic maps. In Castilio, J.E. (ed.),*Mathematical Aspects of Numerical Grid Generation.* Frontiers in Applied Mathematics, vol. 8 (pp. 123–130). SIAM, Philadelphia.

Liao, G., & Anderson, D. (1992). A new approach to grid generation. *Applicable Analysis*, *44*, 285–298.

Liao, G. G., Cai, X., Fleitas, D., Luo, X., Wang, J., & Xue, J. (2008). Optimal control approach to data aligment. *Applied Mathematics Letters*, *21*, 898–905.

Lin, K. L., & Shaw, H. J. (1991). Two-dimensional orthogoal grid generation techniques. *Computers & Structures*, *41*(4), 569–585.

Li, S., & Petzold, L. (1997). Moving mesh methods with upwinding schemes for time-dependent PDEs. *Journal of Computational Physics*, *131*(2), 368–377.

Liseikin, V. D. (2004). *A Computational Differential Geometry Approach to Grid Generation*. Berlin: Springer.

Liseikin, V. D. (2007). *A Computational Differential Geometry Approach to Grid Generation* (2nd ed.). Berlin: Springer.

McNally, D. (1972). FORTRAN program for generating a two-dimensional orthogonal mesh between two arbitrary boundaries. NASA, TN D-6766.

Miller, K. (1983). Alternate codes to control the nodes in the moving finite element method. In *Adaptive Computing Methods for Partial Differential Equations.* (pp. 165–182). Philadelphia: SIAM.

Moser, J. (1965). The volume elements on a manifold. *Transactions of American Mathematical Society*, *120*, 286.

Nakamura, S. (1982). Marching grid generation using parabolic partial differential equations. *Applied Mathematics and Computation*, *10*(11), 775–786.

Nakamura, S., & Suzuki, M. (1987). Noniterative three-dimensional grid generation using a parabolic-hyperbolic hybrid scheme. AIAA Paper 87-0277.

Noack, R. W. (1985). Inviscid flow field analysis of maneuvering hypersonic vehicles using the SCM formulation and parabolic grid generation. AIAA Paper 85-1682.

Noack, R. W., & Anderson, D. A. (1990). Solution adaptive grid generation using parabolic partial differential equations. *AIAA Journal*, *28*(6), 1016–1023.

Rai, M. M., & Anderson, D. A. (1981). Grid evolution in time asymptotic problems. *Journal of Computational Physics*, *43*, 327–344.

Rai, M. M., & Anderson, D. A. (1982). Application of adaptive grids to fluid flow problems with asymptotic solution. *AIAA Journal*, *20*, 496–502.

Ryskin, G., & Leal, L. G. (1983). Orthogonal mapping. *Journal of Computational Physics*, *50*, 71–100.

Schwarz, W. (1986). Elliptic grid generation system for three-dimensional configurations using Poisson's equation. In J. Hauser & C. Taylor (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics* (pp. 341–351). Swansea: Pineridge.

Semper, B., & Liao, G. (1995). A moving grid finite-element method using grid deformation. *Numerical Methods for Partial Differential Equations*, *11*, 603–615.

Slater, J. W., Liou, M. S., & Hindman, R. G. (1995). Approach for dynamic grids. *AIAA Journal*, *33*(1), 63–68.

Soni, B. K. (1991). Grid optimization: a mixed approach. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (p. 617). New York: North-Holland.

Soni, B. K., Huddleston, D. K., Arabshahi, A., & Yu, B. (1993). A study of CFD algorithms applied to complete aircraft configurations. AIAA Paper 93-0784.

Sparis, P. D. (1985). A method for generating boundary-orthogonal curvilinear coordinate systems using the biharmonic equation. *Journal of Computational Physics*, *61*(3), 445–462.

Sparis, P. N., & Karkanis, A. (1992). Boundary-orthogonal biharmonic grids via preconditioned gradient methods. *AIAA Journal*, *30*(3), 671–678.

Spekreijse, S. P. (1995). Elliptic grid generation based on Laplace equations and algebraic transformations. *Journal of Computational Physics*, *118*, 38–61.

Starius, G. (1977). Constructing orthogonal curvilinear meshes by solving initial value problems. *Numerische Mathematlk*, *28*, 25–48.

Steger, J. L., & Chaussee, D. S. (1980). Generation of body fitted coordinates using hyperbolic differential equations. *SIAM Journal of Scientific Statistical Computing*, *1*(4), 431–437.

Steger, J. L., & Rizk, Y. M. (1985). Generation of three-dimensional body-fitted coordinates using hyperbolic partial differential equations. NASA, TM 86753, June.

Steger, J. L., & Sorenson, R. L. (1979). Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *Journal of Computational Physics*, *33*, 405–410.

Tai, C. H., Chiang, D. C., & Su, Y. P. (1996). Three-dimensional hyperbolic grid generation with inherent dissipation and Laplacian smoothing. *AIAA Journal*, *34*(9), 1801–1806.

Tamamidis, P., & Assanis, D. N. (1991). Generation of orthogonal grids with control of spacing. *Journal of Computational Physics*, *94*, 437–453.

Theodoropoulos, T., & Bergeles, G. C. (1989). A Laplacian equation method for numerical generation of boundary-fitted 3d orthogonal grids. *Journal of Computational Physics*, *82*, 269–288.

Thomas, P. D., & Middlecoff, J. F. (1980). Direct control of the grid point distribution in meshes generated by elliptic equations. *AIAA Journal*, *18*(6), 652–656.

Thompson, J. F., Thames, F. C., & Mastin, C. W. (1974). Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, *15*, 299–319.

Thompson, J. F., Warsi, Z. U. A., & Mastin, C. W. (1985). *Numerical Grid Generation: Foundations and Applications*. New York: North-Holland.

Tu, Y., & Thompson, J. F. (1991). Three-dimensional solution-adaptive grid generation on composite configurations. *AIAA Journal*, *29*, 2025–2026.

Visbal, M., & Knight, D. (1982). Generation of orthogonal and nearly orthogonal coordinates with grid control near boundaries. *AIAA Journal*, *20*(3), 305–306.

Wan, D., & Turek, S. (2006). Fictitious boundary and moving mesh methods for the numerical simulation of rigid particulate flows. *Journal of Computational Physics*, *22*, 28–56.

Wan, D., & Turek, S. (2007). An efficient multigrid-FEM method for the simulation of solidliquid two phase flows. *Journal of Computational Applied Mathematics*, *203*(2), 561–580.

Warsi, Z. U. A. (1982). Basic differential models for coordinate generation. In J. F. Thompson (Ed.), *Numerical Grid Generation* (pp. 41–78). New York: North-Holland.

Wathen, A. J. (1990). Optimal moving grids for time-dependent partial differential equations. *Journal of Computational Physics*, *101*, 51–54.

White, A. B. (1979). On selection of equidistributing meshes for two-point boundary-value problems. *SIAM Journal on Numerical Analysis*, *16*(3), 472–502.

White, A. B. (1982). On the numerical solution of initial boundary-value problems in one space dimension. *SIAM Journal on Numerical Analysis*, *19*, 683–697.

White, A. B. (1990). Elliptic grid generation with orthogonality and spacing control on an arbitrary number of boundaries. AIAA Paper 90-1568

Winslow, A. M. (1967). Equipotential zoning of two-dimensional meshes. *Journal of Computational Physics*, *1*, 149–172.

Zegeling, P. A. (1993). *Moving-grid methods for time-dependent partial differential equations*. CWI Tract 94, Amsterdam: Centrum voor Wiskund en Informatica.

Chapter 7
Variational Methods

7.1 Introduction

The calculus of variations provides an excellent opportunity to create new techniques for the generation of grids with mapping approaches by utilizing the idea of optimization of grid characteristics modeled through appropriate functionals. The grid characteristics include grid smoothness, departure from orthogonality or conformality, cell skewness, and cell volume. The minimization of a combination of the functionals representing the desired grid features generates the equations for those coordinate transformations which yield a grid with optimally balanced grid quality measures. The relative contributions of the functionals are determined by the user-prescribed weights.

The major task of the variational approach to grid generation is to describe all basic measures of the desired grid features in an appropriate functional form and to formulate a combined functional that provides a well-posed minimization problem. This chapter describes some basic functionals representing the grid quality properties and measures of grid features. These functionals can provide mathematical feedback in an automatic grid procedure.

7.2 Calculus of Variations

The goal of the calculus of variations is to find the functions which are optimal in terms of specified functionals. The optimal functions also are referred to as critical or stationary points of the respective functionals. The theory of the calculus of variations has been developed to formulate and describe the laws and relations concerned with the critical points of functionals. One of the most important achievements of this theory is the discovery that the optimal functions satisfy some easily formulated equations called the Euler–Lagrange equations. Thus, the problem of computing the optimal functions is related to the problem of the solution of these equations. This

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_7

section presents the Euler–Lagrange equations derived through the minimization of functionals suitable for the purpose of grid generation.

The condition of the convexity of the functionals is of paramount importance to the well-posedness of both the minimization problem for the functionals and the boundary value problems for the resulting Euler–Lagrange equations. Therefore, this section also discusses questions concerned with the convexity of functionals.

7.2.1 General Formulation

Commonly, in the calculus of variations, any functional over some admissible set of functions $\boldsymbol{f} : D^n \to R^m$ is defined by the integral

$$I(\boldsymbol{f}) = \int_{D^n} G(\boldsymbol{f})\mathrm{d}V \ , \tag{7.1}$$

where D^n is a bounded n-dimensional domain, and $G(\boldsymbol{f})$ is some operator specifying, for each vector-valued function $\boldsymbol{f} : D^n \to R^m$, a scalar function $G(\boldsymbol{f}) : D^n \to R$. The admissible set is composed of those functions $\boldsymbol{f}$ which satisfy a prescribed boundary condition

$$\boldsymbol{f} \mid_{\partial D^n} = \boldsymbol{\phi}$$

and for which the integral (7.1) is limited.

In the application of the calculus of variations to grid generation, this set of admissible functions is a set of sufficiently smooth invertible coordinate transformations

$$\boldsymbol{\xi}(\boldsymbol{x}) : X^n \to \Xi^n$$

between the physical domain X^n and the computational domain Ξ^n or, vice versa, a set of sufficiently smooth invertible coordinate transformations from the computational domain Ξ^n onto the physical region X^n :

$$\boldsymbol{x}(\boldsymbol{\xi}) : Q^n \to X^n \ .$$

The integral (7.1) is defined over the domain X^n or Ξ^n, respectively.

In grid generation applications, the operator G is commonly chosen as a combination of weighted local grid characteristics which are to be optimized. The choice depends, of course, on what is expected from the grid. Some forms of local grid characteristics were formulated in Chap. 3 through the coordinate transformations and their first and second derivatives. Therefore, for the purpose of grid generation, it can be supposed that the most widely acceptable formula for the operator G in (7.1) is one which is derived from some expressions containing the first and second derivatives of the coordinate transformations. Thus, we can assume that the functional (7.1), depending on the coordinate transformation $\boldsymbol{\xi}(\boldsymbol{x})$, is of the form

$$I(\boldsymbol{\xi}) = \int_{X^n} G(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\xi}_{x^i}, \boldsymbol{\xi}_{x^i x^j}) \mathrm{d}\boldsymbol{x} \ , \tag{7.2}$$

where G is a smooth function of its variables

$$\boldsymbol{x} \ , \quad \boldsymbol{\xi} \ , \quad \boldsymbol{\xi}_{x^i} = \frac{\partial \boldsymbol{\xi}(\boldsymbol{x})}{\partial x^i} \ , \quad \text{and} \quad \boldsymbol{\xi}_{x^i x^j} = \frac{\partial^2 \boldsymbol{\xi}(\boldsymbol{x})}{\partial x^i \partial x^j} \ .$$

The admissible set for this functional is a set of the invertible vector-valued functions $\boldsymbol{\xi}(\boldsymbol{x}) : X^n \to \Xi^n$ satisfying the condition of smoothness up to the fourth order, i.e.

$$\xi^i(\boldsymbol{x}) \in C^4(X^n) \ , \qquad i = 1, \ldots, n \ .$$

Analogously, the functional (7.1) formulated over a set of invertible coordinate transformations $\boldsymbol{x}(\boldsymbol{\xi})$ from $C^4(\Xi^n)$ has the form

$$I(\boldsymbol{x}) = \int_{\Xi^n} G(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{x}_{\xi^i}, \boldsymbol{x}_{\xi^i \xi^j}) \mathrm{d}\boldsymbol{\xi} \ . \tag{7.3}$$

In accordance with the assumption that the admissible set of functions for the functional (7.2) or (7.3) is composed of the corresponding invertible coordinate transformations, we can reformulate either of these two functionals in terms of the other through the following transition formulas:

$$\begin{aligned} \int_{X^n} f \mathrm{d}\boldsymbol{x} &= \int_{\Xi^n} (Jf) \mathrm{d}\boldsymbol{\xi} \ , \\ \int_{\Xi^n} f \mathrm{d}\boldsymbol{\xi} &= \int_{X^n} (f/J) \mathrm{d}\boldsymbol{x} \ , \end{aligned} \tag{7.4}$$

where $J = \det\{\partial x^i / \partial \xi^j\}$. Thus, for the functional (7.3), we obtain

$$\begin{aligned} I(\boldsymbol{x}) &= \int_{X^n} \Big(\frac{1}{J} G(\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{x}_{\xi^i}, \boldsymbol{x}_{\xi^i \xi^j}) \Big) \mathrm{d}\boldsymbol{x} \\ &= \int_{X^n} G_1(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\xi}_{x^i}, \boldsymbol{\xi}_{x^i x^j}) \mathrm{d}\boldsymbol{x} = I_1(\boldsymbol{\xi}) \end{aligned}$$

with an implied transition from $\boldsymbol{x}_{\xi^i}$ and $\boldsymbol{x}_{\xi^i \xi^j}$ to $\boldsymbol{\xi}_{x^l}$ and $\boldsymbol{\xi}_{x^l x^k}$.

7.2.2 Euler–Lagrange Equations

To be definite, we consider here the variational principle for grid generation in the form of the functional (7.2) over the set of invertible smooth coordinate transformations $\boldsymbol{\xi}(\boldsymbol{x})$ from the physical domain X^n onto the computational domain Ξ^n. In general, the functionals are formulated in the physical space X^n rather than in the parametric space Ξ^n. This is preferred because the physical-space formulation can be used more simply to obtain grid generation techniques that provide the necessary grid properties.

If the transformation $\boldsymbol{\xi}(\boldsymbol{x})$ is optimal for the functional (7.2), then it satisfies a system of Euler–Lagrange equations in the interior points of the domain X^n :

$$G_{\xi^i} - \frac{\partial}{\partial x^j} G_{\frac{\partial \xi^i}{\partial x^j}} + \frac{\partial^2}{\partial x^j \partial x^k} G_{\frac{\partial^2 \xi^i}{\partial x^j \partial x^k}} = 0 \ , \qquad i, j, k, = 1, \dots, n \ , \tag{7.5}$$

where the subscripts ξ^i, $\partial \xi^i / \partial x^j$, and $\partial^2 \xi^i / \partial x^j \partial x^k$ mean the corresponding partial derivatives of G. We remind the reader that the repeated indices here and below imply summation over them unless otherwise noted.

In many applications, the integrand G is dependent only on $\boldsymbol{x}$, the function $\boldsymbol{\xi}(\boldsymbol{x})$, and its first derivatives, i.e. $G = G(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\xi}_{x^i})$. In this case, the admissible set of functions $\boldsymbol{\xi}^i(\boldsymbol{x})$ can be from the class $C^2(X^n)$, and the system of Euler–Lagrange equations (7.5) is reduced to

$$G_{\xi^i} - \frac{\partial}{\partial x^j} G_{\frac{\partial \xi^i}{\partial x^j}} = 0 \ , \qquad i, j = 1, \dots, n \ . \tag{7.6}$$

We give a schematic deduction of (7.6). Equation (7.5) are obtained in a similar manner.

Let the transformation $\boldsymbol{\xi}(\boldsymbol{x})$ be a critical point of the functional (7.2) with $G = G(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\xi}_{x^j})$. In order to prove that $\boldsymbol{\xi}(\boldsymbol{x})$ satisfies (7.6), we first choose a scalar smooth function from $C^2(X^n)$ which equals zero on the boundary of the domain X^n. Let this function be denoted by $\phi(\boldsymbol{x})$. Now, using $\phi(\boldsymbol{x})$, we define for a fixed index i a vector-valued function $\boldsymbol{\psi}(\boldsymbol{x}) = [\psi^1(\boldsymbol{x}), \dots, \psi^n(\boldsymbol{x})]$ dependent on $\boldsymbol{x}$ as follows:

$$\begin{aligned} \psi^j(\boldsymbol{x}) &= 0 \ , \qquad j \neq i \ , \\ \psi^i(\boldsymbol{x}) &= \phi(\boldsymbol{x}) \ , \ j = i \ , \end{aligned}$$

i.e.

$$\boldsymbol{\psi}(\boldsymbol{x}) = \phi(\boldsymbol{x}) \boldsymbol{e}_i \ ,$$

where $\boldsymbol{e}_i$ is the ith basic Cartesian vector. As was assumed, the transformation $\boldsymbol{\xi}(\boldsymbol{x})$ is critical for the functional (7.2), and therefore the following scalar smooth function

$$y(\epsilon) = I(\boldsymbol{\xi} + \epsilon \boldsymbol{\psi}) = \int_{X^n} G(\boldsymbol{x} \ , \ \boldsymbol{\xi} + \epsilon \boldsymbol{\psi} \ , \ \boldsymbol{\xi}_{x^i} + \epsilon \boldsymbol{\psi}_{x^i}) \mathrm{d}\boldsymbol{x} \ , \tag{7.7}$$

where ϵ is a real variable, has an extremum at the point $\epsilon = 0$. This results in the relation $y'(0) = 0$. In accordance with the rule of differentiation of integrals, we obtain

$$y'(0) = \int_{X^n} \Big(\phi(\boldsymbol{x}) G_{\xi^i} + \frac{\partial \phi}{\partial x^j} G_{\frac{\partial \xi^i}{\partial x^j}} \Big) \mathrm{d}\boldsymbol{x} \ , \qquad j = 1, \dots, n \ .$$

Taking into account that

$$\frac{\partial\phi}{\partial x^j}G_{\frac{\partial\xi^i}{\partial x^j}} = \frac{\partial}{\partial x^j}\Big(\phi G_{\frac{\partial\xi^i}{\partial x^j}}\Big) - \phi\frac{\partial}{\partial x^j}G_{\frac{\partial\xi^i}{\partial x^j}} \ , \qquad j = 1, \dots, n \ ,$$

we have

$$y'(0) = \int_{X^n}\Big[\phi\Big(G_{\xi^i} - \frac{\partial}{\partial x^j}G_{\frac{\partial\xi^i}{\partial x^j}}\Big) + \frac{\partial}{\partial x^j}\Big(\phi G_{\frac{\partial\xi^i}{\partial x^j}}\Big)\Big]\mathrm{d}\boldsymbol{x} = 0 \ . \tag{7.8}$$

Using the divergence theorem

$$\int_{X^n}\Big(\sum_{i=1}^{n}\frac{\partial A^i}{\partial x^i}\Big)\mathrm{d}\boldsymbol{x} = \int_{\partial X^n}(\boldsymbol{A}\cdot\boldsymbol{n})\mathrm{d}S \ , \tag{7.9}$$

valid for a smooth arbitrary vector function $\boldsymbol{A}(\boldsymbol{x}) = [A^1(\boldsymbol{x}), \dots, A^n(\boldsymbol{x})]$, we conclude that

$$\int_{X^n}\frac{\partial}{\partial x^j}\Big(\phi G_{\frac{\partial\xi^i}{\partial x^j}}\Big)\mathrm{d}\boldsymbol{x} = \int_{\partial X^n}\phi(\boldsymbol{G}^i\cdot\boldsymbol{n})\mathrm{d}S = 0 \ ,$$

where

$$\boldsymbol{G}^i = \Big(G_{\frac{\partial\xi^i}{\partial x^1}}, \dots, G_{\frac{\partial\xi^i}{\partial x^n}}\Big) \ ,$$

since the selected function ϕ equals zero at all points of ∂X^n. Thus, we find that the second summation term in the integral (7.8) can be omitted, and consequently we find that

$$y'(0) = \int_{X^n}\phi\Big(G_{\xi^i} - \frac{\partial}{\partial x^j}G_{\frac{\partial\xi^i}{\partial x^j}}\Big)\mathrm{d}\boldsymbol{x} = 0 \tag{7.10}$$

for every smooth function $\phi(\boldsymbol{x})$ satisfying the condition proposed above, that $\phi(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in \partial X^n$. From this relation, we readily find that the optimal coordinate transformation $\boldsymbol{\xi}(\boldsymbol{x})$ obeys (7.6) at every interior point of the domain X^n. If this were not so, then there would exist an interior point $\boldsymbol{x}_0$ such that the function

$$f(\boldsymbol{x}) = \Big(G_{\xi^i} - \frac{\partial}{\partial x^j}G_{\frac{\partial\xi^i}{\partial x^j}}\Big)(\boldsymbol{x})$$

does not vanish at this point, i.e. $f(\boldsymbol{x}_0) \neq 0$, say, $f(\boldsymbol{x}_0) > 0$. The function $f(\boldsymbol{x})$ is continuous, so there exists a positive number $r > 0$ such that $f(\boldsymbol{x})$ does not change its sign for all $\boldsymbol{x}$ satisfying $|\boldsymbol{x} - \boldsymbol{x}_0| \le r$, i.e. $f(\boldsymbol{x}) > 0$ at these points. Now we use for the function $\phi(\boldsymbol{x})$ in (7.10) a nonnegative mapping which equals zero at all points $\boldsymbol{x}$ that are outside the subdomain $|\boldsymbol{x} - \boldsymbol{x}_0| \le r$, and $\phi(\boldsymbol{x}_0) > 0$. For example, one such function $\phi(\boldsymbol{x})$ is expressed as follows:

$$\phi(\boldsymbol{x}) = \begin{cases} \exp\left(\dfrac{1}{(\boldsymbol{x}-\boldsymbol{x}_0)^2 - r^2}\right), & |\boldsymbol{x}-\boldsymbol{x}_0| < r \\ 0\,, & |\boldsymbol{x}-\boldsymbol{x}_0| \geq r \end{cases} .$$

This function is from the class $C^\infty(X^n)$. In accordance with the assumed property of $f(\boldsymbol{x})$, we find that (7.10) is not satisfied for the function $\phi(\boldsymbol{x})$ specified above, namely $y'(0) > 0$, i.e. the function $\boldsymbol{\xi}(\boldsymbol{x})$ is not critical for the functional (7.2), which is contrary to the initial assumption about the extremum of the functional at the point $\boldsymbol{\xi}(\boldsymbol{x})$. From this contradiction, we conclude that the optimal transformation $\boldsymbol{\xi}(\boldsymbol{x})$ is a solution to (7.6) at every interior point of the domain X^n.

Analogously, there can be obtained a system of Euler–Lagrange equations for the functional (7.3),

$$G_{x^i} - \frac{\partial}{\partial \xi^j} G_{\frac{\partial x^i}{\partial \xi^j}} + \frac{\partial^2}{\partial \xi^j \partial \xi^k} G_{\frac{\partial^2 x^i}{\partial \xi^j \partial \xi^k}} = 0\,, \qquad i, j = 1, \dots, n\,, \tag{7.11}$$

which is satisfied by the optimal coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$.

7.2.3 Convexity Condition

Convexity is a very important property imposed on functionals in the calculus of variations. In the case of the functional (7.2), it is formulated by the condition of positiveness of the tensors $G^i,\ i = 1, \dots, n$:

$$G^i = \left\{ G_{\frac{\partial \xi^i}{\partial x^j} \frac{\partial \xi^i}{\partial x^k}} \right\}, \qquad \text{with} \quad i \quad \text{fixed}\,.$$

Namely, every tensor $G^i,\ i = 1, \dots, n$, must be strongly positive. Recall that a matrix is strongly positive if every principal minor is larger then zero. In this case, there exists a constant $c_i > 0$ for every fixed index $i = 1, \dots, n$, such that

$$G_{\frac{\partial \xi^i}{\partial x^j} \frac{\partial \xi^i}{\partial x^k}} b^j b^k \geq c_i b^l b^l\,, \qquad j, k, l = 1, \dots, n\,, \tag{7.12}$$

for an arbitrary vector $\boldsymbol{b} = (b^1, \dots, b^n)$. The inequality (7.12) means that the system of Euler–Lagrange equations (7.6) is elliptic.

Convex functionals generate well-posed problems for their minimization and for the solution of the Dirichlet boundary value problem for the corresponding Euler–Lagrange equations (7.5) or (7.6). In particular, the relation (7.12) guarantees that there exists a unique, isolated optimal transformation which satisfies the system of Euler–Lagrange equations with Dirichlet boundary conditions.

7.2.4 Functionals Dependent on Metric Elements

The most common interior characteristics of grid cells were defined in Chap. 3 by the elements of the metric tensors. As the covariant elements can be derived from the contravariant ones, we can assume that the functional (7.2) representing an integral measure of some grid feature is defined by the integrand G, depending on $\boldsymbol{x}$ and the contravariant elements

$$g^{ij} = \nabla\xi^i \cdot \nabla\xi^j = \frac{\partial\xi^i}{\partial x^k}\frac{\partial\xi^j}{\partial x^k} , \qquad i, j, k = 1, \ldots, n ,$$

only, i.e.

$$I(\boldsymbol{\xi}) = \int_{X^n} G(\boldsymbol{x}, g^{ij})\mathrm{d}\boldsymbol{x} .$$

In this case, the corresponding system of the Euler–Lagrange equations (7.6) has the following divergent form:

$$\frac{\partial}{\partial x^j}\Big(\frac{\partial G}{\partial g^{lk}}\frac{\partial g^{lk}}{\partial(\partial\xi^i/\partial x^j)}\Big) = 0 , \qquad i, j, k, l = 1, \ldots, n . \tag{7.13}$$

As

$$\frac{\partial g^{lk}}{\partial(\partial\xi^i/\partial x^j)} = \delta^i_l\frac{\partial\xi^k}{\partial x^j} + \delta^i_k\frac{\partial\xi^l}{\partial x^j} , \qquad i, j, k, l = 1, \ldots, n , \tag{7.14}$$

we have the result that

$$\frac{\partial G}{\partial g^{lk}}\frac{\partial g^{lk}}{\partial(\partial\xi^i/\partial x^j)} = \Big(\frac{\partial G}{\partial g^{ik}} + \frac{\partial G}{\partial g^{ki}}\Big)\frac{\partial\xi^k}{\partial x^j} , \qquad i, j, k, l = 1, \ldots, n .$$

Therefore, the system of the Euler–Lagrange equations (7.13) is equivalent to

$$\frac{\partial}{\partial x^j}\Big[\Big(\frac{\partial G}{\partial g^{ik}} + \frac{\partial G}{\partial g^{ki}}\Big)\frac{\partial\xi^k}{\partial x^j}\Big] = 0 , \qquad i, j, k = 1, \ldots, n , \tag{7.15}$$

with summation over the repeated indices j and k. Taking advantage of the identity (2.57), the system (7.15) can be converted to

$$\frac{\partial}{\partial\xi^j}\Big[Jg^{kj}\Big(\frac{\partial G}{\partial g^{ik}} + \frac{\partial G}{\partial g^{kj}}\Big)\Big] = 0 , \qquad i, j, k = 1, \ldots, n , \tag{7.16}$$

written with respect to the independent variables ξ^i. In particular, when the integrand G is defined by the diagonal elements g^{ii} of the contravariant metric tensor $\{g^{ij}\}$, then (7.15) and (7.16) are as follows:

$$\frac{\partial}{\partial x^j}\Big(\frac{\partial G}{\partial g^{ii}}\frac{\partial \xi^i}{\partial x^j}\Big) = 0\ ,$$
$$\frac{\partial}{\partial \xi^j}\Big(J g^{ij}\frac{\partial G}{\partial g^{ii}}\Big) = 0\ , \qquad i, j = 1, \dots, n$$

with fixed index i.

7.2.5 Functionals Dependent on Tensor Invariants

Chapter 3 presents a description of some local grid quality properties which are defined by the invariants $I_1, \dots, I_n$ of the metric tensor $g_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j}$ in the coordinates $\xi^1, \dots, \xi^n$. The integration of these properties over the physical or computational domain represents the global grid properties in the form of functionals depending on the invariants. Taking into account the general identity

$$\int_{\Xi^n} G \mathrm{d}\boldsymbol{\xi} = \int_{X^n} [G/(I_n^{1/2})]\mathrm{d}\boldsymbol{x}\ ,$$

where

$$I_n = g = \det\{g_{ij}\} = \overset{2}{\det}\Big\{\frac{\partial x^i}{\partial \xi^j}\Big\}\ ,$$

we can consider all these functionals as integrals over the domain X^n in the form

$$I(\boldsymbol{\xi}) = \int_{X^n} G(\boldsymbol{x}, I_1, \dots, I_n)\mathrm{d}\boldsymbol{x}\ . \tag{7.17}$$

The Euler–Lagrange equations (7.6) in this case are represented as follows:

$$\frac{\partial}{\partial x^j}\Big(G_{I_k}\frac{\partial I_k}{\partial(\partial \xi^i/\partial x^j)}\Big) = 0\ , \qquad i, j, k = 1, \dots, n\ . \tag{7.18}$$

Two-Dimensional Tensor

For two-dimensional coordinate transformations $\boldsymbol{x}(\boldsymbol{\xi}) : \Xi^2 \to X^2$, the invariants are defined by (3.32). Since we consider the functionals depending on the invariants as integrals over the domain X^n, we need to rewrite the invariants through the terms of the contravariant metric tensor $\{g^{ij}\}$. This is readily accomplished by applying (2.21). Thus, we have in two dimensions

$$I_1 = g(g^{11} + g^{22})\ ,$$
$$I_2 = g = 1/\det\{g^{ij}\} = 1\Big/\Big[\det{}^2\Big\{\frac{\partial \xi^i}{\partial x^j}\Big\}\Big]\ .$$

From these relations, we obtain

$$\begin{aligned}\frac{\partial I_2}{\partial(\partial\xi^i/\partial x^j)} &= -2J^3(-1)^{i+j}\frac{\partial\xi^{3-i}}{\partial x^{3-j}} = -2I_2\frac{\partial x^j}{\partial\xi^i}\ ,\\ \frac{\partial I_1}{\partial(\partial\xi^i/\partial x^j)} &= 2g\frac{\partial\xi^i}{\partial x^j} - 2J^3(g^{11}+g^{22})(-1)^{i+j}\frac{\partial\xi^{3-i}}{\partial x^{3-j}}\\ &= 2\Big(I_2\frac{\partial\xi^i}{\partial x^j} - I_1\frac{\partial x^j}{\partial\xi^i}\Big)\ , \qquad i,j=1,2\ ,\end{aligned} \tag{7.19}$$

without summation over the repeated indices i and j. Thus, we find that in two dimensions, the Euler–Lagrange system (7.18) can be expressed as follows:

$$\frac{\partial}{\partial x^j}\Big[G_{I_1}\Big(I_2\frac{\partial\xi^i}{\partial x^j} - I_1\frac{\partial x^j}{\partial\xi^i}\Big) - G_{I_2}I_2\frac{\partial x^j}{\partial\xi^i}\Big] = 0\ , \qquad i,j=1,2\ . \tag{7.20}$$

The application of the identity (2.57) to each equation of (7.20) leads to the following system:

$$\frac{\partial}{\partial\xi^j}\{J[g^{ij}G_{I_1}I_2 - \delta^i_j(G_{I_1}I_1 + G_{I_2}I_2)]\} = 0\ , \qquad i,j=1,2\ . \tag{7.21}$$

In particular, for the integral measure of the two-dimensional grid density expressed locally by (3.75) with $n=2$, we assume

$$I_{\text{cn}} = \int_{X^2}(I_1/I_2)\mathrm{d}\boldsymbol{x} = \int_{X^2}(g^{11}+g^{22})\mathrm{d}\boldsymbol{x}\ ,$$

and therefore the system of Euler–Lagrange equations (7.20) for I_{cn} is the system of Laplace equations

$$\frac{\partial}{\partial x^j}\frac{\partial\xi^i}{\partial x^j} = 0\ , \qquad i,j=1,2\ .$$

Three-Dimensional Tensor

As in two dimensions, the invariants (3.33) of the three-dimensional tensor $\{g_{ij}\}$ can be expressed through the elements of the contravariant tensor $\{g^{ij}\}$. Using for this purpose (2.22), we obtain

$$\begin{aligned}&I_1 = g(g^{11}g^{22} + g^{11}g^{33} + g^{22}g^{33} - g^{12}g^{21} - g^{13}g^{31} - g^{23}g^{32})\ ,\\ &I_2 = (g^{11}+g^{22}+g^{33})/\det\{g^{ij}\}\ ,\\ &I_3 = g = 1/\det\{g^{ij}\}\ .\end{aligned}$$

Therefore, we have for $i, j = 1, 2, 3$

$$\begin{aligned}
\frac{\partial I_3}{\partial g^{ij}} &= -g^2 \text{ cofactor of } g^{ij} = -g g_{ij} = -g_{ij} I_3 \ , \\
\frac{\partial I_2}{\partial g^{ij}} &= \delta^i_j g - g g_{ij}(g^{11} + g^{22} + g^{33}) = \delta^i_j I_3 - g_{ij} I_2 \ , \\
\frac{\partial I_1}{\partial g^{ij}} &= \delta^i_j g(g^{11} + g^{22} + +g^{33}) - g^{ij} - g^{ij} I_3 = \delta^i_j I_2 - g_{ij} I_3 - g^{ij} \ .
\end{aligned} \tag{7.22}$$

Taking into account (7.14), we obtain

$$\frac{\partial I_k}{\partial(\partial \xi^i / \partial x^j)} = \Big(\frac{\partial I_k}{\partial g^{im}} + \frac{\partial I_k}{\partial g^{mi}}\Big)\frac{\partial \xi^m}{\partial x^j} \ , \qquad i, j, k, m = 1, 2, 3 \ .$$

Therefore, the use of (7.22) leads to

$$\begin{aligned}
\frac{\partial I_3}{\partial(\partial \xi^i / \partial x^j)} &= -2 I_3 \frac{\partial x^j}{\partial \xi^i} \ , \qquad i, j = 1, 2, 3 \ , \\
\frac{\partial I_2}{\partial(\partial \xi^i / \partial x^j)} &= -2 I_3 \frac{\partial \xi^i}{\partial x^j} - 2 I_2 \frac{\partial x^j}{\partial \xi^i} \ , \qquad i, j = 1, 2, 3 \ , \\
\frac{\partial I_1}{\partial(\partial \xi^i / \partial x^j)} &= 2\Big(I_2 \frac{\partial \xi^i}{\partial x^j} - I_3 \frac{\partial x^j}{\partial \xi^i} - g^{im}\frac{\partial \xi^m}{\partial x^j}\Big) \ , \qquad i, j, m = 1, 2, 3 \ .
\end{aligned}$$

Using these relations in (7.18), we find that the three-dimensional Euler–Lagrange equations for the functional (7.17) can be written as

$$\begin{aligned}
&\frac{\partial}{\partial x^j}\Big[G_{I_1}\Big(I_2 \frac{\partial \xi^i}{\partial x^j} - I_3 \frac{\partial x^j}{\partial \xi^i} - g^{im}\frac{\partial \xi^m}{\partial x^j}\Big) \\
&-G_{I_2}\Big(I_3 \frac{\partial \xi^i}{\partial x^j} + I_2 \frac{\partial x^j}{\partial \xi^i}\Big) - G_{I_3} I_3 \frac{\partial x^j}{\partial \xi^i}\Big] = 0 \ , \qquad i, j, m = 1, 2, 3 \ .
\end{aligned} \tag{7.23}$$

This system, after application of (2.57) to every equation, is transformed into

$$\begin{aligned}
&\frac{\partial}{\partial \xi^j}\{J[g^{ij}(G_{I_1} I_2 - G_{I_2} I_3) - g^{im} g^{mj} - \delta^i_j (G_{I_1} I_3 + G_{I_2} I_2 + G_{I_3} I_3)]\} = 0 \ , \\
&i, j, m = 1, 2, 3 \ ,
\end{aligned} \tag{7.24}$$

written with respect to the independent variables ξ^i.

For generating quasi-isometric grids in a three-dimensional domain X^3, there was applied in Garanzha (2000) and Garanzha and Kudryavtseva (2012) the following functional dependent on the invariants I_1 and I_3:

$$I_{\text{q-is}}[\boldsymbol{x}] = \int_{\Xi^n} \Big\{(1-\alpha)\frac{1}{3}\frac{(I_1)^{3/2}}{\sqrt{I_3}} + \alpha \frac{1}{2}\Big(\frac{1}{\sqrt{I_3}} + \sqrt{I_3}\Big)\Big\} \mathrm{d}\boldsymbol{\xi}, \quad \alpha > 0. \tag{7.25}$$

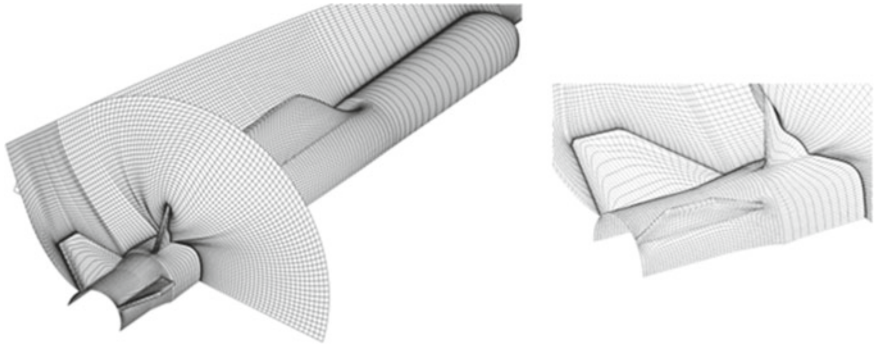


Fig. 7.1 Section of a quasi-isometric grid (*left*) and fragment of the grid (*right*)

Figure 7.1 exhibits a grid generated through the minimization of this functional.

7.3 Integral Grid Characteristics

Grid properties can play an extremely important role in influencing the accuracy and efficiency of the numerical solutions of partial differential equations. In particular, the truncation error is affected by the grid skewness, grid size, grid size ratio, angle between the grid lines, grid nonuniformity, and consistency of the grid with the features of the physical solution. Thus, by controlling these grid quantities, one can control the efficiency of the numerical solution of boundary value problems. The calculus of variations allows one to formulate, through appropriate functionals, natural techniques which can serve as tools for controlling various grid properties.

A description of some local grid characteristics was given in Chap. 3 through the elements of the metric tensors. The procedure of integration of these characteristics defines functionals which reflect global properties of the grid. In this section, some basic functionals modeling global grid characteristics are formulated. It needs to be emphasized that some local characteristics are dimensionally homogeneous. Therefore, in order to preserve this quality globally, the integration of the corresponding quantities should be carried out over a scaled region. If we assume that the logical domain Ξ^n is the unit cube, we can utilize it as such a normalized domain.

7.3.1 Dimensionless Functionals

Dimensionless functionals are formed by integrating the dimensionless grid characteristics reviewed in Chap. 3 over the computational domain Ξ^n.

Grid Skewness

The integrated measures of three-dimensional grid skewness are obtained from the formulas (3.67–3.69), derived by means of the cosines and cotangents of the angles between the tangential and normal vectors. These characteristics are dimensionally homogeneous and, in accordance with the argument about the integration of dimensionless quantities over the domain Ξ^n, we formulate the global grid skewness measures as

$$\begin{aligned}
I_{\mathrm{sk},1} &= \int_{\Xi^3}\Big(\frac{(g_{12})^2}{g_{11}g_{22}}+\frac{(g_{13})^2}{g_{11}g_{33}}+\frac{(g_{23})^2}{g_{22}g_{33}}\Big)\mathrm{d}\boldsymbol{\xi}\ ,\\
I_{\mathrm{sk},2} &= \int_{\Xi^3}\frac{1}{g}\Big(\frac{(g_{12})^2}{g^{33}}+\frac{(g_{13})^2}{g^{22}}+\frac{(g_{23})^2}{g^{11}}\Big)\mathrm{d}\boldsymbol{\xi}\\
&= \int_{\Xi^3}\Big(\frac{(g_{12})^2}{g_{11}g_{22}-(g_{12})^2}+\frac{(g_{13})^2}{g_{11}g_{33}-(g_{13})^2}\\
&\quad+\frac{(g_{23})^2}{g_{22}g_{33}-(g_{23})^2}\Big)\mathrm{d}\boldsymbol{\xi}\ ,\\
I_{\mathrm{sk},3} &= \int_{\Xi^3}\Big(\frac{(g^{12})^2}{g^{11}g^{22}}+\frac{(g^{13})^2}{g^{11}g^{33}}+\frac{(g^{23})^2}{g^{22}g^{33}}\Big)\mathrm{d}\boldsymbol{\xi}\ ,\\
I_{\mathrm{sk},4} &= \int_{\Xi^3}g\Big(\frac{(g^{12})^2}{g_{33}}+\frac{(g^{13})^2}{g_{22}}+\frac{(g^{23})^2}{g_{11}}\Big)\mathrm{d}\boldsymbol{\xi}\\
&= \int_{\Xi^3}\Big(\frac{(g^{12})^2}{g^{11}g^{22}-(g^{12})^2}+\frac{(g^{13})^2}{g^{11}g^{33}-(g^{13})^2}\\
&\quad+\frac{(g^{23})^2}{g^{22}g^{33}-(g^{23})^2}\Big)\mathrm{d}\boldsymbol{\xi}\ .
\end{aligned}\tag{7.26}$$

Since the elements g^{ij} of the contravariant metric tensor are expressed directly through the derivatives of the functions $\partial\xi^i/\partial x^j$, we see that the Euler–Lagrange equations for the functionals $I_{\mathrm{sk},3}$ and $I_{\mathrm{sk},4}$ can be obtained more easily if these functionals are reformulated over the domain X^3. This can be accomplished by using the relation (7.4). For example, we have, for the functional $I_{\mathrm{sk},3}$,

$$I_{\mathrm{sk},3} = \int_{X^3}\sqrt{\det(g^{ij})}\Big(\frac{(g^{12})^2}{g^{11}g^{22}}+\frac{(g^{13})^2}{g^{11}g^{33}}+\frac{(g^{23})^2}{g^{22}g^{33}}\Big)\mathrm{d}\boldsymbol{x}\ .$$

The functionals $I_{\mathrm{sk},1}$ and $I_{\mathrm{sk},2}$ can be transformed into functionals dependent on $\boldsymbol{\xi}(\boldsymbol{x})$ by the rule of transition (2.22) from the elements of $\{g_{ij}\}$ in the integrand to the elements of $\{g^{ij}\}$.

In two-dimensions, we obtain, from (7.26), only two functionals of dimensionally homogeneous skewness:

$$I_{\mathrm{sk},1}=\int_{\Xi^2}\frac{(g_{12})^2}{g_{11}g_{22}}\mathrm{d}\boldsymbol{\xi}=\int_{X^2}\sqrt{\det\{g^{ij}\}}\frac{(g^{12})^2}{g^{11}g^{22}}\mathrm{d}\boldsymbol{x}\ ,$$
$$I_{\mathrm{sk},2}=\int_{\Xi^2}g(g^{12})^2\mathrm{d}\boldsymbol{\xi}=\int_{X^2}\frac{1}{\sqrt{\det\{g^{ij}\}}}(g^{12})^2\mathrm{d}\boldsymbol{x}\ . \tag{7.27}$$

The first functional is defined through the cosines of the angles while the second is determined by the cotangents of the angles.

Deviation from Orthogonality

Dimensionally homogeneous functionals indicating the global grid nonorthogonality in three dimensions can be derived from the local nonorthogonality measures (3.70). As a result, we have in three dimensions

$$\begin{aligned}I_{\mathrm{o},1}&=\int_{\Xi^3}\frac{g_{11}g_{22}g_{33}}{g}\mathrm{d}\boldsymbol{\xi}\\&=\int_{X^3}\frac{[g^{22}g^{33}-(g^{23})^2][g^{11}g^{33}-(g^{13})^2][g^{11}g^{22}-(g^{12})^2]}{\sqrt{\det^5\{g^{ij}\}}}\mathrm{d}\boldsymbol{x}\ ,\\I_{\mathrm{o},2}&=\int_{\Xi^3}g(g^{11}g^{22}g^{33})\mathrm{d}\boldsymbol{\xi}=\int_{X^3}\frac{1}{\sqrt{\det\{g^{ij}\}}}(g^{11}g^{22}g^{33})\mathrm{d}\boldsymbol{x}\ .\end{aligned} \tag{7.28}$$

In two dimensions, (7.28) yields only one functional of departure from orthogonality:

$$I_{\mathrm{o},1}=\int_{\Xi^2}\frac{g_{11}g_{22}}{g}\mathrm{d}\boldsymbol{\xi}=\int_{X^2}\frac{g^{11}g^{22}}{\sqrt{\det\{g^{ij}\}}}\mathrm{d}\boldsymbol{x}\ . \tag{7.29}$$

Deviation from Conformality

Integration of the dimensionless characteristics (3.76), (3.85), and (3.88) over Ξ^2 or Ξ^3 generates a quantity which reflects an integral departure of the grid from a conformal grid. Thus, we obtain, in two dimensions,

$$I_{\mathrm{cf},1}=\int_{\Xi^2}(I_1/\sqrt{I_2})\mathrm{d}\boldsymbol{\xi}=\int_{X^2}(I_1/I_2)\mathrm{d}\boldsymbol{x}\ . \tag{7.30}$$

An analogous consideration of (3.86) yields, in three dimensions,

$$I_{\mathrm{cf},1}=\int_{\Xi^3}[I_2/\sqrt[3]{(I_3)^2}]\mathrm{d}\boldsymbol{\xi}=\int_{X^3}[I_2/(I_3)^{7/6}]\mathrm{d}\boldsymbol{x}\ . \tag{7.31}$$

The quantity (3.89) for $n=3$ defines one more three-dimensional form of the nonconformality functional:

$$\begin{aligned}I_{\mathrm{cf},2}&=\int_{\Xi^3}[I_1/(I_3)^{1/3}]\mathrm{d}\boldsymbol{\xi}\\&=\int_{\Xi^3}\frac{g_{11}+g_{22}+g_{33}}{\sqrt[3]{\det\{g_{ij}\}}}\mathrm{d}\boldsymbol{\xi}\ .\end{aligned} \tag{7.32}$$

Reformulation of $I_{\mathrm{cf},2}$ over the domain X^3 yields

$$I_{\mathrm{cf},2} = \int_{X^3} \frac{1}{\sqrt[6]{\det^5\{g_{ij}\}}} \{g^{11}g^{22} + g^{11}g^{33} + g^{22}g^{33} \\ - [(g^{12})^2 + (g^{13})^2 + (g^{23})^2]\} \mathrm{d}\boldsymbol{x} \ , \tag{7.33}$$

using (7.4) and (2.22).

We can use as the integrand the dimensionally homogeneous quantity:

$$Q_{\mathrm{cf},3} = [I_2/\sqrt[3]{(I_3)^2}]^\alpha = (Q_{\mathrm{cf},1})^\alpha \ , \qquad \alpha > 0 \ , \tag{7.34}$$

which also reaches its minimum value when the three-dimensional coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is conformal. In this case, we can control the form of the Euler–Lagrange equations with the parameter α. The corresponding functional with the quantitative characteristic (7.34) is as follows:

$$I_{\mathrm{cf},3} = \int_{\Xi^3} [I_2/\sqrt[3]{(I_3)^2}]^\alpha \mathrm{d}\boldsymbol{\xi} \\ = \int_{X^3} (I_2)^\alpha/(I_3)^{2\alpha/3+1/2} \mathrm{d}\boldsymbol{x} \ .$$

If $\alpha = 3/2$, we obtain

$$I_{\mathrm{cf},3} = \int_{X^3} (I_2/I_3)^{3/2} \mathrm{d}\boldsymbol{x} \\ = \int_{X^3} (g^{11} + g^{22} + g^{33})^{3/2} \mathrm{d}\boldsymbol{x} \ ,$$

taking into account (3.33). The system of Euler–Lagrange equations (7.15) or (7.20) for this functional has the form

$$\frac{\partial}{\partial x^j}\left(\sqrt{g^{11} + g^{22} + g^{33}} \frac{\partial \xi^i}{\partial x^j}\right) = \frac{\partial}{\partial x^j}\left(\sqrt{\frac{I_2}{I_3}} \frac{\partial \xi^i}{\partial x^j}\right) = 0 \ , \qquad i, j = 1, 2, 3 \ . \tag{7.35}$$

Taking advantage of (2.57) or (7.24), we obtain for the inverted system of (7.35)

$$\frac{\partial}{\partial \xi^j}\left(\sqrt{I_2} g^{ij}\right) = 0 \ , \qquad i, j = 1, 2, 3 \ . \tag{7.36}$$

Also, multiplication of (7.35) by $\partial x^k/\partial \xi^i$ and summation over i yields one more inverted system of (7.35):

$$g^{ij} \frac{\partial^2 x^k}{\partial \xi^i \partial \xi^j} = \sqrt{\frac{I_3}{I_2}} \frac{\partial}{\partial x^k} \sqrt{\frac{I_2}{I_3}} \ , \qquad i, j = 1, 2, 3 \ . \tag{7.37}$$

An analogously simple system of Euler–Lagrange equations is derived for n-dimensional functionals of nonconformality by replacing the local measure (3.88) with

$$Q_{\mathrm{cf},3} = [I_{n-1}/(I_n)^{1-1/n}]^{\alpha} \ , \qquad \alpha > 0 \ .$$

For the functional $I_{\mathrm{cf},3}$ with $\alpha = n/2$, we obtain

$$\begin{aligned} I_{\mathrm{cf},3} &= \int_{\Xi^n} [I_{n-1}/(I_n)^{1-1/n}]^{n/2} \mathrm{d}\boldsymbol{\xi} = \int_{X^n} (I_{n-1}/I_n)^{n/2} \mathrm{d}\boldsymbol{x} \\ &= \int_{X^n} (g^{11} + \cdots + g^{nn})^{n/2} \mathrm{d}\boldsymbol{x} \ . \end{aligned} \tag{7.38}$$

The system of Euler–Lagrange equations for this functional is

$$\frac{\partial}{\partial x^j}\Big((g^{11} + \cdots + g^{nn})^{n/2-1} \frac{\partial \xi^i}{\partial x^j}\Big) = 0 \ , \qquad i, j = 1, \ldots, n \ . \tag{7.39}$$

Multiplying (7.39) by $\partial x^k / \partial \xi^i$ and summing over i, we obtain, in analogy with (7.37), the system with respect to the dependent variables x^i and independent variables ξ^i:

$$g^{ij} \frac{\partial^2 x^k}{\partial \xi^i \partial \xi^j} = H^{-1} \frac{\partial}{\partial x^k} H \ , \qquad i, j, k = 1, \ldots, n \ , \tag{7.40}$$

where

$$H = (g^{11} + \cdots + g^{nn})^{n/2-1} = (I_{n-1}/I_n)^{n/2-1} \ .$$

7.3.2 Dimensionally Heterogeneous Functionals

Smoothness Functionals

The characteristic of local grid concentration is expressed through the invariants by (3.75). In general, this quantity is not dimensionless, and therefore its integration is carried out over the physical domain X^n. The resulting functional,

$$I_{\mathrm{s}} = \int_{X^n} (I_{n-1}/I_n) \mathrm{d}\boldsymbol{x} = \int_{X^n} (g^{11} + \ \cdots + g^{nn}) \mathrm{d}\boldsymbol{x} \ , \tag{7.41}$$

formulated for an arbitrary n-dimensional domain X^n, is called the functional of smoothness. We see that the functional of smoothness (7.41) for $n = 2$ coincides with the functional of conformality (7.30). However, in the three-dimensional case, the functionals (7.31) and (7.41) are different. The Euler–Lagrange equations for the smoothness functional (7.41) comprise a simple system of Laplace equations

$$\frac{\partial}{\partial x^j}\left(\frac{\partial \xi^i}{\partial x^j}\right) = 0 \;, \qquad i, j = 1, \ldots, n \;.$$

The inverted system with respect to the dependent variable $\boldsymbol{x}$ is obtained in the ordinary manner by multiplying (7.41) by $\partial x^k/\partial \xi^i$ and summing over i. As a result, we obtain the n-dimensional inverted Laplace system

$$g^{ij}\frac{\partial \boldsymbol{x}}{\partial \xi^i \partial \xi^j} = 0 \;, \qquad i, j = 1, \ldots, n \;. \tag{7.42}$$

Functionals of Orthogonality

The characteristics (3.71) and (3.72) of the local deviation of the three-dimensional cells from orthogonal cells define two functionals of orthogonality. For the purpose of simplicity of the resulting Euler–Lagrange equations, it is more suitable to integrate (3.71) over Ξ^3 and (3.72) over X^3. So, we obtain the following functionals, which represent some measures of grid nonorthogonality and also can be interpreted as measures of grid skewness:

$$\begin{aligned} I_{o,3} &= \int_{\Xi^3}\Big((g_{12})^2 + (g_{13})^2 + (g_{23})^2\Big)\mathrm{d}\boldsymbol{\xi} \;, \\ I_{o,4} &= \int_{X^3}\Big((g^{12})^2 + (g^{13})^2 + (g^{23})^2\Big)\mathrm{d}\boldsymbol{x} \;. \end{aligned} \tag{7.43}$$

The corresponding Euler–Lagrange equations (7.11) and (7.6) have the form

$$\begin{aligned} &g_{ik}\frac{\partial}{\partial \xi^j}\left(\frac{\partial x^k}{\partial \xi^j}\right) = 0 \;, \\ &g^{ik}\frac{\partial}{\partial x^j}\left(\frac{\partial \xi^k}{\partial x^j}\right) = 0 \;, \quad i, j, k = 1, 2, 3 \;, \qquad i \neq k \;. \end{aligned} \tag{7.44}$$

By applying (2.57) to every equation of the second system of (7.44), a converted system is obtained:

$$g^{ik}\frac{\partial}{\partial \xi^k}J = 0 \;, \qquad k = 1, 2, 3 \;, \qquad i \neq k \;. \tag{7.45}$$

The systems (7.44) and (7.45) derive ill-posed boundary value problems, and therefore the functionals of orthogonality are commonly combined with the functional of smoothness (7.41) to yield well-posed problems of grid generation. In two dimensions, the orthogonality functionals (7.43) are

$$\begin{aligned} I_{o,3} &= \int_{\Xi^2}(g_{12})^2\mathrm{d}\boldsymbol{\xi} \;, \\ I_{o,4} &= \int_{X^2}(g^{12})^2\mathrm{d}\boldsymbol{x} \;. \end{aligned} \tag{7.46}$$

The local departure of a two-dimensional grid from an orthogonal one with a prescribed cell aspect ratio F may be estimated by the measure

$$Q_{o,5} = \Big(\frac{1}{\sqrt{F}}\frac{\partial x^1}{\partial \xi^1} - \sqrt{F}\frac{\partial x^2}{\partial \xi^2}\Big)^2 + \Big(\frac{1}{\sqrt{F}}\frac{\partial x^2}{\partial \xi^1} + \sqrt{F}\frac{\partial x^1}{\partial \xi^2}\Big)^2 , \qquad (7.47)$$

since $Q_{o,5} = 0$ if and only if the grid is orthogonal and $g_{11} = F^2 g_{22}$. From (7.47), we obtain

$$Q_{o,5} = \frac{1}{F}g_{11} + Fg_{22} - 2J.$$

This quantity $Q_{o,5}$ defines one more functional of departure from orthogonality:

$$\begin{aligned} I_{o,5} &= \int_{\Xi^2}\Big(\frac{1}{F}g_{11} + Fg_{22} - 2J\Big)\mathrm{d}\boldsymbol{\xi} \\ &= \int_{\Xi^2}\Big(\frac{1}{F}g_{11} + Fg_{22}\Big)\mathrm{d}\boldsymbol{\xi} - 2S , \qquad (7.48) \end{aligned}$$

where S is the area of the domain X^2, with the following Euler–Lagrange equations:

$$\frac{\partial}{\partial \xi^1}\Big(\frac{1}{F}\frac{\partial x^i}{\partial \xi^1} + F\frac{\partial x^i}{\partial \xi^2}\Big) = 0 , \qquad i = 1, 2 . \qquad (7.49)$$

Analogously, the functional of departure from orthogonality is defined through the elements g^{ii} as

$$I_{o,6} = \int_{X^2}\Big(\frac{1}{F}g^{11} + Fg^{22}\Big)\mathrm{d}\boldsymbol{x} - 2 . \qquad (7.50)$$

7.3.3 Functionals Dependent on Second Derivatives

This subsection reviews a formulation of the functionals in the form (7.2) or (7.3), where the integrands include terms dependent on second derivatives of coordinate transformations.

Functionals of Eccentricity

The eccentricity functionals are derived from the local grid eccentricity measures (3.90) and (3.90). Since $Q_{\epsilon,1}$, from (3.90), is expressed through the first and second derivatives of $\boldsymbol{x}(\boldsymbol{\xi})$ with respect to ξ^i, we will integrate this quantity over Ξ^n. For a similar reason, the relation (3.91) is integrated over X^n. As a result, we obtain the integral characteristics of grid eccentricity in the form

$$I_{\epsilon,1} = \int_{\Xi^n} \sum_{i=1}^{n} \Big(\frac{\partial}{\partial \xi^i} \ln \sqrt{g_{ii}}\Big)^2 \mathrm{d}\boldsymbol{\xi} \ ,$$

$$I_{\epsilon,2} = \int_{X^n} \sum_{i=1}^{n} \Big(\frac{\partial}{\partial x^i} \ln \sqrt{g^{ii}}\Big)^2 \mathrm{d}\boldsymbol{x} \ . \tag{7.51}$$

Unlike the functionals determined by the first derivatives of the varied functions $\boldsymbol{\xi}(\boldsymbol{x})$ or $\boldsymbol{x}(\boldsymbol{\xi})$, the functionals (7.51) include second derivatives. The system of Euler–Lagrange equations (7.11) for grid generation is therefore of fourth order. This makes it possible not only to specify the boundary nodal distribution when generating a grid by solving such a system, but also to specify the directions of the coordinate lines emerging from the boundaries, which is important when one needs to construct smoothly abutting grids in complicated regions, as in the case of applying block grid techniques. However, some questions related to the formulation of the boundary conditions and the correctness of the boundary value problems, and also to the numerical justification for the systems of Euler–Lagrange equations for constructing grids, are still to be resolved.

Functionals of Grid Warping and Grid Torsion

The functionals of grid warping and grid torsion are formulated analogously through the respective local measures (3.92) and (3.93). Like the functionals of eccentricity (7.51), these functionals are dependent on second derivatives, thus generating Euler–Lagrange equations of fourth order.

7.4 Adaptation Functionals

Numerical grids can significantly influence various characteristics of the efficiency of the numerical solution of partial differential equations. One of the most important characteristics is the accuracy of the numerical solution, which is formulated through the error of the numerical calculation. In this matter, the theory of the calculus of variations provides an excellent opportunity to formulate the requirement of a minimal error for a given number of grid points in a straightforward form through the functional of error. The minimization of this functional generates an optimal grid in the sense of accuracy. Thus, the variational approach is a natural tool for generating grids adapted to the physical solution.

The simplest and most logical way of defining the error functional I_{er} seems to be through the integral measure of the local numerical error $\boldsymbol{r} = \boldsymbol{u} - \boldsymbol{u}^h$,

$$I_{\mathrm{er},1} = \int_{\Xi^n} \parallel \boldsymbol{r} \parallel \mathrm{d}\boldsymbol{\xi}, \tag{7.52}$$

or through the integral of the measure of the approximation error $\boldsymbol{T}$,

$$I_{\mathrm{er},2} = \int_{\Xi^n} \parallel \boldsymbol{T} \parallel \mathrm{d}\boldsymbol{\xi} \ . \tag{7.53}$$

However, this logical formulation results in a very cumbersome and high-order system of Euler–Lagrange equations; namely, its order is twice the order of the derivatives in $\boldsymbol{r}$ or $\boldsymbol{T}$. The numerical solution of these Euler–Lagrange equations is a very difficult task, especially in the case of multidimensional space. Thus, the optimal grid can be obtained only at the expense of the efficiency of the grid generation process. Evidently, a more optimal approach to generating adaptive grids through the variational technique lies in formulating simpler error functionals in order to balance the accuracy of the solution against the cost of obtaining the grid.

A common approach aimed at the minimization of the numerical error relies on concentration of grid nodes in the subregions of high truncation error. One version of this approach, reviewed in Chap. 4, was formulated through the equidistribution principle. In fact, this principle is universal, since all adaptive methods aimed at the concentration of grid nodes in the regions of large solution variations are related to the one-dimensional equidistribution principle, which requires the grid spacing to be inversely proportional to a weight function. The equidistribution principle can be formulated in a number of different ways. In this section, a variational version of the equidistribution approach is discussed.

7.4.1 One-Dimensional Functionals

The basic one-dimensional differential model for the equidistribution principle with the weight function w was formulated as the two-point boundary value problem (4.23). In the case in which the weight function w is defined in the interval ξ, and thus does not vary when $x(\xi)$ changes, the problem (4.23) is a boundary value problem for the Euler–Lagrange equation obtained by optimizing the functional

$$I_{\mathrm{eq}} = \int_0^1 w\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2 \mathrm{d}\xi \ . \tag{7.54}$$

The functional (7.54) physically models the energy which arises in a system of nodes x_i connected by springs with stiffness $2w_i$. The equilibrium condition of this system also determines the positions of the grid points x_i defined by a coordinate transformation $x(\xi)$ satisfying (4.23).

There is also a geometric interpretation of the following numerical approximation of the integral (7.54) on a uniform grid $\xi_i = ih, \ h = 1/N$:

$$I^h_{\mathrm{eq}} = N \sum_{i=0}^{N-1} w_{i+1/2}(h_{i+1/2})^2 \ , \tag{7.55}$$

where

$$w_{i+1/2} = [w(\xi_i) + w(\xi_{i+1})]/2\ ,$$
$$h_{i+1/2} = x(\xi_{i+1}) - x(\xi_i)\ .$$

The expression (7.55) describes a hyperellipsoid for each value of I^h_{eq}, if $h_{i+1/2}$ is considered to be the ith coordinate in the $(N-1)$-dimensional Euclidean space, and its minimization means that the hyperplane

$$\sum_{i=1}^{N-1} h_{i+1/2} = b - a\ ,$$

where $b - a$ is the length of the segment x, is an $(N-2)$-dimensional tangent plane for this hyperellipsoid.

Since, in general, the error of the solution in the interval x is described by an expression of the form $\boldsymbol{r}_i = \boldsymbol{C}(h_i)^k \approx \boldsymbol{C}h^k(\partial x/\partial\xi)^k$, the functional (7.54) can be interpreted as the integral error of the second-order approximation of a one-dimensional differential problem. The error functional of the approximation of order k can be represented by the integral

$$I_{\text{eq}} = \int_0^1 w\Big(\frac{\mathrm{d}x}{\mathrm{d}\xi}\Big)^k \mathrm{d}\xi\ , \qquad k > 0\ . \tag{7.56}$$

A geometric interpretation of the functional (7.56) for $k = 4$ is possible if u_x^2 is taken as the weight function. In this case, the value of the functional (7.56) is proportional to the sum of the squares of the areas of the rectangles which border the curve $u = u(x)$ in the (u, x) plane.

Commonly, the weight function w is defined in the physical region, and therefore the variational formulation of the equidistribution method typically utilizes a functional with respect to transformations $\xi(x)$ with specified boundary conditions:

$$I_{\text{eq}} = \int_X w_1(x)\Big(\frac{\mathrm{d}\xi}{\mathrm{d}x}\Big)^k \mathrm{d}x\ , \qquad k > 0\ , \tag{7.57}$$

whose Euler–Lagrange equation is obtained in accordance with (7.6). Thus, the optimal transformation $\xi(x)$ for this functional is the solution to the boundary value problem

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}x}\Big[w_1(x)\Big(\frac{\mathrm{d}\xi}{\mathrm{d}x}\Big)^{k-1}\Big] = 0\ , \qquad a < x < b,\\ &\xi(a) = 0\ , \qquad \xi(b) = 1\ .\end{aligned} \tag{7.58}$$

From (7.58), the following relation follows directly:

$$w_1^{1/(k-1)}\frac{\mathrm{d}\xi}{\mathrm{d}x} = \text{const} \Rightarrow w_1^{1/(1-k)}\frac{\mathrm{d}x}{\mathrm{d}\xi} = \text{const}\ ,$$

which results in small values of $\mathrm{d}x/\mathrm{d}\xi$ when $w_1^{1/(1-k)}$ is large and vice versa. A problem equivalent to (4.23) is obtained from (7.58) when $k = 2,\ w_1(x) = w^{-1}(x),\ w_1 > 0$.

Thus, the equidistribution method considered in Chap. 4 can be interpreted as a variational method for constructing grids by minimizing the functionals (7.56) or (7.57).

7.4.2 Multidimensional Approaches

In this subsection, the variational formulations (7.56) and (7.57) of the one-dimensional equidistribution approach are taken as a starting point for the extension to multiple dimensions.

The basic elements of the functionals (7.56) and (7.57) are the weight functions and the first derivative of the transformation $x(\xi)$ or $\xi(x)$. When generalizing to an n-dimensional region X^n, this derivative can be interpreted as the Jacobian of transformation $\boldsymbol{x}(\boldsymbol{\xi})$ or $\boldsymbol{\xi}(\boldsymbol{x})$, or as the square roots of the values of the diagonal elements of the covariant metric tensor $\{g_{ij}\}$ or the contravariant metric tensor $\{g^{ij}\}$. Thus, in many of the generalizations of the functionals (7.56) and (7.57) that have been proposed for constructing adaptive grids in an n-dimensional domain X^n, the expression $\mathrm{d}x/\mathrm{d}\xi$ is replaced by $J = \det\{\partial x^i/\partial \xi^j\}$, and $\mathrm{d}\xi/\mathrm{d}x$ by $\det\{\partial \xi^i/\partial x^j\} = 1/J$, or combinations of the diagonal elements of the covariant or contravariant metric tensor $\{g_{ij}\}$ and $\{g^{ij}\}$ are used. Since $J = \sqrt{g}$, all these functionals can be formulated through the metric tensors $\{g_{ij}\}$ or $\{g^{ij}\}$. Thus, the Euler–Lagrange equations for these functionals are readily obtained by using (7.15) or (7.16).

Volume-Weighted Functional

For example, the functional defined through the Jacobian $J = \sqrt{g}$, called the volume-weighted functional, has the form

$$I_{\mathrm{VW}} = \int_{X^n} w(\boldsymbol{x}) g^k \mathrm{d}\boldsymbol{x}\ , \qquad k > 0\ . \tag{7.59}$$

The expected result of the minimization of this functional is small values of the Jacobian when $w(x)$ is large and vice versa.

In analogy with the first line of (7.22), we have for arbitrary dimensions

$$\frac{\partial g}{\partial g^{lk}} = -g g_{lk}\ , \qquad l, k = 1, \dots, n\ .$$

Therefore, using (7.15), we obtain a system of Euler–Lagrange equations for the functional (7.59):

$$\frac{\partial}{\partial x^j}\Big(wg^k g_{im}\frac{\partial \xi^m}{\partial x^j}\Big) = \frac{\partial}{\partial x^j}\Big(wg^k\frac{\partial x^j}{\partial \xi^i}\Big) = 0\ , \quad i, j, k, m = 1, \dots, n\ . \tag{7.60}$$

In order to obtain compact equations which include only the derivatives with respect to ξ^i, we use the identity

$$\frac{\partial}{\partial x^j}\Big(g^{-1/2}\frac{\partial x^j}{\partial \xi^i}\Big) \equiv 0\ , \qquad i, j = 1, \dots, n\ ,$$

which is a mere reformulation of (2.48). Therefore, from (7.60), we obtain

$$\begin{aligned}\frac{\partial}{\partial x^j}\Big(wg^k\frac{\partial x^j}{\partial \xi^i}\Big) &= g^{-1/2}\frac{\partial}{\partial x^j}\Big(wg^{k+1/2}\Big)\frac{\partial x^j}{\partial \xi^i}\\ &= g^{-1/2}\frac{\partial}{\partial \xi^i}\Big(wg^{k+1/2}\Big) = 0\ .\end{aligned} \tag{7.61}$$

Tangent-Length-Weighted Functionals

An adaptation functional which uses the diagonal elements of the metric tensor $\{g_{ij}\}$ can be expressed as follows:

$$I_{\mathrm{tw},1} = \int_{\Xi^n}\Big(w(\boldsymbol{\xi})\sum_i g_{ii}\Big)\mathrm{d}\boldsymbol{\xi} = \int_{\Xi^n} w(\boldsymbol{\xi})I_1\mathrm{d}\boldsymbol{\xi}\ . \tag{7.62}$$

A functional aimed at providing an individual grid concentration in each grid direction ξ^i can be formulated through a combination of the edge length characteristics g_{ii} with individually specified weights:

$$I_{\mathrm{tw},2} = \int_{\Xi^n}\Big(\sum_i w^i(\boldsymbol{\xi})g_{ii}\Big)\mathrm{d}\boldsymbol{\xi}\ . \tag{7.63}$$

The weight functions w^i control the grid spacing along each coordinate independently.

The system of Euler–Lagrange equations for the functional (7.63) is of simple elliptic type,

$$\frac{\partial}{\partial \xi^j}\Big(w^i\frac{\partial x^i}{\partial \xi^j}\Big) = 0\ , \qquad i, j = 1, \dots, n\ ,$$

with the index i fixed. The functionals (7.62) and (7.63) influence the grid node distribution in the direction of the coordinate lines.

Normal-Length-Weighted Functionals

Analogous adaptation functionals determined by weighted diagonal elements g^{ii} of the contravariant metric tensor $\{g^{ij}\}$ have the form

$$I_{\mathrm{nw},1} = \int_{X^n} \Big(w(\boldsymbol{x}) \sum_i g^{ii} \Big) \mathrm{d}\boldsymbol{x} \tag{7.64}$$

and

$$I_{\mathrm{nw},2} = \int_{X^n} \Big(\sum_i w_i(\boldsymbol{x}) g^{ii} \Big) \mathrm{d}\boldsymbol{x} \ . \tag{7.65}$$

The functionals (7.64) and (7.65) are also referred to as diffusion functionals. They are formulated for the purpose of distributing the nodes in the direction of the normals to the coordinate surfaces $\xi^i = c$, with clustering of the grid points in the neighborhoods of large values of the weighting functions and rarefying of the nodes in the vicinity of small values of the weights.

The resulting Euler–Lagrange equations for critical functions $\boldsymbol{\xi}(\boldsymbol{x})$ of the functionals (7.64) and (7.65) have the form

$$\frac{\partial}{\partial x^j}\Big(w_i \frac{\partial \xi^i}{\partial x^j} \Big) = 0 \ , \qquad i, j = 1, \ldots, n \ , \tag{7.66}$$

with the index i fixed. If $w^i > 0$, then this system is elliptic. The relation $w^i = w,\ i = 1, \ldots, n$, in this system corresponds to the functional (7.64). In this case, the transformed equations with the dependent and independent variables interchanged are readily obtained by multiplying the system of Euler–Lagrange equations (7.66) by $\partial x^k / \partial \xi^i$ and summing over i. As a result, we obtain

$$g^{ij} \frac{\partial^2 x^k}{\partial \xi^i \partial \xi^j} - \frac{1}{w} \frac{\partial w}{\partial x^k} = 0 \ , \qquad i, j, k = 1, \ldots, n \ . \tag{7.67}$$

Using the relation (2.24), we have

$$\frac{\partial w}{\partial x^k} = \frac{\partial w}{\partial \xi^i} \frac{\partial \xi^i}{\partial x^k} = g^{ij} \frac{\partial w}{\partial \xi^i} \frac{\partial x^k}{\partial \xi^j} \ , \qquad i, j, k = 1, \ldots, n \ .$$

Therefore, we obtain, from the inverted system of Euler–Lagrange equations (7.67),

$$g^{ij} \Big(\frac{\partial^2 x^k}{\partial \xi^i \partial \xi^j} - \frac{1}{w} \frac{\partial w}{\partial \xi^i} \frac{\partial x^k}{\partial \xi^j} \Big) = w g^{ij} \frac{\partial}{\partial \xi^i} \Big(\frac{1}{w} \frac{\partial x^k}{\partial \xi^j} \Big) = 0 \ , \qquad i, j, k = 1, \ldots, n \ .$$

Thus, we obtain another compact form of the Euler–Lagrange equations for the functional (7.64):

$$g^{ij} \frac{\partial}{\partial \xi^i} \Big(\frac{1}{w} \frac{\partial x^k}{\partial \xi^j} \Big) = 0 \ , \qquad i, j, k = 1, \ldots, n \ . \tag{7.68}$$

Metric-Weighted Functionals

The length-weighted functionals permit a natural generalization in the form of metric-weighted functionals:

$$I_{\mathrm{mw},1} = \int_{\Xi^n} w^{ij}(\boldsymbol{\xi}) g_{ij} \mathrm{d}\boldsymbol{\xi} \ , \qquad i, j = 1, \ldots, n \tag{7.69}$$

and

$$I_{\mathrm{mw},2} = \int_{X^n} w_{ij}(\boldsymbol{x}) g^{ij} \mathrm{d}\boldsymbol{x} \ , \qquad i, j = 1, \ldots, n \ . \tag{7.70}$$

The condition of convexity (7.12) will be satisfied for these functionals if the matricies $\{w_{ij}\}$ and $\{w^{ij}\}$, respectively, are positive. Without loss of generality, we can assume in (7.69) and (7.70) that $w_{ij} = w_{ji}$, $w^{ij} = w^{ji}$. The corresponding systems of Euler–Lagrange equations for (7.69) and (7.70) then have the form

$$\begin{aligned} &\frac{\partial}{\partial \xi^j}\Big(w^{ik}\frac{\partial x^k}{\partial \xi^j}\Big) = 0 \ , \\ &\frac{\partial}{\partial x^j}\Big(w_{ik}\frac{\partial \xi^k}{\partial x^j}\Big) = 0 \ , \qquad i, j, k = 1, \ldots, n \ . \end{aligned} \tag{7.71}$$

General Approach

A more general formulation of the adaptation functionals utilizes the weighted elements of the matrix $\{(\partial x^i/\partial \xi^j)^2\}$ or the matrix $\{(\partial \xi^i/\partial x^j)^2\}$. For example,

$$\begin{aligned} I_{\mathrm{ad},6} &= \int_{\Xi^n} \Big[\sum_{i,j} w^{ij}(\boldsymbol{\xi})\Big(\frac{\partial x^i}{\partial \xi^j}\Big)^2\Big] \mathrm{d}\boldsymbol{\xi} \ , \\ I_{\mathrm{ad},7} &= \int_{X^n} \Big[\sum_{i,j} w^{ij}(\boldsymbol{x})\Big(\frac{\partial \xi^i}{\partial x^j}\Big)^2\Big] \mathrm{d}\boldsymbol{x} \ . \end{aligned} \tag{7.72}$$

The corresponding Euler–Lagrange equations are

$$\begin{aligned} &\frac{\partial}{\partial \xi^j}\Big(w^{ij}\frac{\partial x^i}{\partial \xi^j}\Big) = 0 \ , \\ &\frac{\partial}{\partial x^j}\Big(w^{ij}\frac{\partial \xi^i}{\partial x^j}\Big) = 0 \ , \qquad i, j = 1, \ldots, n \ , \end{aligned} \tag{7.73}$$

where the summation is carried out only over j and the index i is fixed.

Nonstationary Functionals

In the construction of adaptive grids for spatial nonstationary elastoplastic and gas-dynamics problems, the adaptation functional characterizing the concentration of grid nodes in the high-gradient region of the flow velocity $\boldsymbol{u} = (u^1, u^2, u^3)$ is defined in terms of the velocity components of the grid nodes $\partial x^i/\partial t$:

$$I_{\mathrm{ad},6} = \int_{X^3} (k \operatorname{div} \overline{\boldsymbol{w}} - \operatorname{div} \overline{\boldsymbol{u}})^2 \mathrm{d}\boldsymbol{x} \ , \tag{7.74}$$

where

$$\overline{\boldsymbol{w}} = (\overline{w}^1, \ldots, \overline{w}^n) \ , \qquad \overline{w}^i = \sum_{j=1}^{3} \frac{\partial x^j}{\partial t} \frac{\partial \xi^i}{\partial x^j},$$
$$\overline{\boldsymbol{u}} = (\overline{u}^1, \ldots, \overline{u}^n) \ , \qquad \overline{u}^i = \sum_{j=1}^{3} \frac{\partial \xi^i}{\partial x^j} u^j \ , \qquad i = 1, 2, 3 \ .$$

Minimization of the functional (7.74) involves equidistribution of the weight function $w(x)$ with respect to the values, to degree k, of the grid cell volumes, generating a grid with small cell volumes in the neighborhood of large values of $w(x)$.

Weight Functions

The weight functions w, w_i, w^i, $w_{i,j}$, and $w^{i,j}$ in the formulations of adaptation functionals considered above are usually taken to be combinations of the moduli of the derivatives of those components u^i of the solution of the physical problem for which these derivatives can take large values. For instance, for the flow of a viscous heat-conducting gas, the function $w(\boldsymbol{x})$ has the form

$$w = \Big(\epsilon + \sum_i |\mathrm{grad} u^i|^{\alpha_j}\Big)^{\beta} + \phi_0(\boldsymbol{x}) \ ,$$

where $\phi_0(\boldsymbol{x})$ is a positive function and ϵ, α_i, and β are positive constants. The weight function is of this form because of the need to construct a grid which is invariant under Galilean transformations and provides node clustering in the region of high gradients of $\boldsymbol{u}$.

7.5 Functionals of Attraction

For some multidimensional problems, there are natural families of lines or vector fields which should be aligned with the grid lines or basic vector fields for reasons of computational efficiency. In gas dynamics, for instance, these are the streamlines (or lines of potential), lines of predominant direction of flow, and a family of the Lagrange coordinates; in plasma theory, they are the preferred vector directions defined by the magnetic field. The solution to viscous transonic flow problems usually contains shock structures which should be aligned with one coordinate direction, while boundary layers should be aligned with the coordinates from the other family; namely, they need to be parallel to a streamwise coordinate. Some problems also have an underlying symmetry which should be matched with the coordinate system.

The alignment of the coordinate lines with natural families of curves of this kind leads to efficiency in the numerical modeling. For example, the use of Lagrange coordinates in problems of fluid motion simplifies the representation of the equations, and makes it possible to localize the moving region and follow the motion of the fluid particles during the numerical solution. The requirement to generate aligned coordinates can be readily realized by variational techniques through suitable functionals of departure. This section presents a formulation of certain functionals of this type.

7.5.1 Lagrangian Coordinates

The condition for a coordinate ξ^i to be Lagrangian in a three-dimensional fluid flow is given by the equation

$$\frac{\partial \xi^i}{\partial t} + \overline{u}^i = 0\ , \qquad i = 1, 2, 3\ , \tag{7.75}$$

where $\overline{u}^i$ is the ith component of the velocity vector in the moving system of coordinates (t, ξ^1, ξ^2, ξ^3), i.e.

$$\overline{u}^i = \sum_{i=1}^{3} u^j \frac{\partial \xi^i}{\partial x^j}\ ,$$

with $u^j,\ j = 1, 2, 3$, representing the jth velocity component in the Cartesian system (t, x^1, x^2, x^3). Since

$$\frac{\partial \xi^i}{\partial t} = -\sum_{j=1}^{3} \frac{\partial \xi^i}{\partial x^j} \frac{\partial x^j}{\partial t}\ ,$$

(7.75) is equivalent to

$$\overline{w}^i - \overline{u}^i = 0\ , \tag{7.76}$$

where $\overline{w}^i$ is the ith component of the grid velocity vector expanded in the tangential vectors $\boldsymbol{x}_{\xi^j},\ j = 1, \dots, n$.

Equation (7.76) can be used to determine the functional of deviation from a Lagrangian coordinate grid:

$$\begin{aligned} I_{\mathrm{L},1} &= \int_{X^3 \times I} w \sum_{i=1}^{3} (\overline{w}^i - \overline{u}^i)^2 \mathrm{d}\boldsymbol{x}\ \mathrm{d}t \\ &= \int_{X^3 \times I} w \sum_{i=1}^{m} \Big(\frac{\partial \xi^i}{\partial t} + \overline{u}^i\Big)^2 \mathrm{d}\boldsymbol{x}\ \mathrm{d}t\ , \end{aligned} \tag{7.77}$$

where I is the range of the variable t. The functional (7.77) is formulated so as to provide attraction of the grid lines to the Lagrangian coordinates.

The Euler–Lagrange equations (7.6) derived from this functional are as follows:

$$\frac{\partial}{\partial t}[w(\overline{w}^i - \overline{u}^i)] - \frac{\partial}{\partial x^j}[w(\overline{w}^i - \overline{u}^i)u^j] = 0 \ , \qquad i, j = 1, 2, 3 \ . \tag{7.78}$$

By applying (2.97), they are transformed into the system

$$\frac{\partial}{\partial t}[Jw(\overline{w}^i - \overline{u}^i)] + \frac{\partial}{\partial \xi^j}[Jw(\overline{w}^i - \overline{u}^i)(\overline{w}^j - \overline{u}^j)] = 0 \ , \quad i, j = 1, 2, 3 \ , \tag{7.79}$$

with respect to the dependent variables $t, \ \xi^1, \ \xi^2, \ \xi^3$.

When all of the coordinates are Lagrangian, the conditions (7.76) with $i = 1, 2, 3$ are equivalent to the system of equations

$$x^i_t(t, \xi^1, \xi^2, \xi^3) - u^i = 0 \ , \qquad i = 1, 2, 3 \ .$$

This relation is used to define a functional that controls the attraction of the generated grid to the Lagrangian grid in the form

$$I_{\mathrm{L},2} = \int_{\Xi^3 \times I} w \sum_{i=1}^{3} \Big(u^i - \frac{\partial x^i}{\partial t}\Big)^2 \mathrm{d}\boldsymbol{\xi}\, \mathrm{d}t \ . \tag{7.80}$$

7.5.2 Attraction to a Vector Field

Alignment can be very useful when there is a natural anisotropy in the physical problem, for example, a dominant flow direction which is expressed by a vector field. The variational approach can be helpful in generating techniques to obtain such alignment. Functionals which take into account the direction of the prescribed vector fields $\boldsymbol{A}_i(x), \ i = 1, 2, 3$, for constructing three-dimensional coordinate transformations are introduced in the form

$$I_{\mathrm{vf},1} = \int_{X^3} \Big(w \sum_{i=1}^{3} (\boldsymbol{A}_i \times \nabla \xi^i)^2\Big) \mathrm{d}\boldsymbol{x} \ , \tag{7.81}$$

where w is the weight function, and $\nabla \xi^i = \operatorname{grad} \xi^i$. In the process of the minimization of this functional, the normals to the surface $\xi^i = c$ tend to become parallel to $\boldsymbol{A}^i$. From (2.29), we have for the integrand of the functional (7.81)

$$
\begin{aligned}
w\sum_{i=1}^{3}(\boldsymbol{A}_i \times \nabla\xi^i)^2 &= w(|\boldsymbol{A}_i|^2 g^{ii} - (\boldsymbol{A}_i \cdot \nabla\xi^i)^2) \\
&= w\Big(|\boldsymbol{A}_i|^2 g^{ii} - A_i^k A_i^p \frac{\partial\xi^i}{\partial x^k}\frac{\partial\xi^i}{\partial x^p}\Big) .
\end{aligned}
$$

From this relation, we readily obtain the Euler–Lagrange equations for the functional (7.81):

$$
\frac{\partial}{\partial x^j}\Big[w\Big(|\boldsymbol{A}_i|^2\frac{\partial\xi^i}{\partial x^j} - A_i^j A_i^p \frac{\partial\xi^i}{\partial x^p}\Big)\Big] , \qquad i, j, p = 1, 2, 3 , \tag{7.82}
$$

with the index i fixed.

Analogously, a functional of alignment of the tangent vectors $\boldsymbol{x}_{\xi^i}$ with the prescribed vector fields $\boldsymbol{A}_i,\ i = 1, 2, 3$, can be defined:

$$
I_{\mathrm{vf},2} = \int_{X^3}\Big(w\sum_{i=1}^{3}(\boldsymbol{A}_i \times \boldsymbol{x}_{\xi^i})^2\Big)\mathrm{d}\boldsymbol{x} . \tag{7.83}
$$

This functional can serve to attract the coordinate lines to the streamlines of the vector fields $\boldsymbol{A}_i,\ i = 1, 2, 3$.

7.5.3 Jacobian-Weighted Functional

A Jacobian-weighted functional represents the deviation of the Jacobian matrix $\{\partial\xi^i/\partial x^j\}$ of the transformation $\boldsymbol{\xi}(x)$ from the prescribed matrix $S(\boldsymbol{x}) = \{S^{ij}(\boldsymbol{x})\}$, $i, j = 1, \dots, n$, via a least-squares fit. In particular, the functional can have the following form:

$$
I_{\mathrm{jw},1}(\boldsymbol{\xi}) = \int_{X^n} G(\boldsymbol{x}, \nabla\xi^i)\mathrm{d}x , \tag{7.84}
$$

with

$$
G(\boldsymbol{x}, \nabla\xi^i) = \sum_{i,j=1}^{n}\Big(\frac{\partial\xi^i}{\partial x^j} - S^{ij}(\boldsymbol{x})\Big)^2 .
$$

In fact, the integrand $G(\boldsymbol{x}, \nabla\xi^i)$ is the square of the Frobenius norm of the matrix

$$
M = \Big\{\frac{\partial\xi^i}{\partial x^j} - S^{ij}\Big\} , \qquad i, j = 1, \dots, n ,
$$

i.e.

$$
G(\boldsymbol{x}, \nabla\xi^i) = \mathrm{tr}\{M^T M\} .
$$

The Euler–Lagrange equations derived from the minimization of the functional I_{jw} have the form

$$\frac{\partial}{\partial x^j}\Big(\frac{\partial \xi^i}{\partial x^j} - S^{ij}\Big) = 0\ , \qquad i, j = 1, \dots, n\ . \tag{7.85}$$

These equations are elliptic and are, in fact, a variant of the Poisson system (6.15) with

$$P^i = \frac{\partial}{\partial x^j} S^{ij}\ , \qquad i, j = 1, \dots, n\ .$$

Multiplying this system by $\partial x^k / \partial \xi^i$ and summing the result over i, we obtain the following transformed equation for the transformed dependent variable $\boldsymbol{x}(\boldsymbol{\xi})$, written in vector form:

$$g^{ij} \frac{\partial^2 \boldsymbol{x}}{\partial \xi^i \xi^j} + \frac{\partial}{\partial x^j}(S^{ij})\boldsymbol{x}_{\xi^i} = 0\ . \tag{7.86}$$

In accordance with (2.71),

$$\frac{\partial}{\partial x^j} S^{ij} = \frac{1}{J}\frac{\partial}{\partial \xi^k}(J\overline{S}^{ik})\ , \qquad i, j, k = 1, \dots, n\ ,$$

where

$$\overline{S}^{ik} = S^{ij}\frac{\partial \xi^k}{\partial x^j}\ , \qquad i, j, k = 1, \dots, n\ .$$

Hence, (7.86) can be written as follows:

$$g^{ij} \frac{\partial^2 \boldsymbol{x}}{\partial \xi^i \xi^j} + \frac{1}{J}\frac{\partial}{\partial \xi^j}(J\overline{S}^{ij})\boldsymbol{x}_{\xi^i} = 0\ . \tag{7.87}$$

Analogously, a Jacobian-weighted functional $I_{\mathrm{jw},2}$, which measures the squared departure of the Jacobi matrix $\{\partial x^i / \partial \xi^j\}$ of the coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ from the prescribed matrix $\{S^{ij}(\boldsymbol{\xi})\}$, can be defined as

$$I_{\mathrm{jw},2} = \int_{\Xi^n} \sum_{i,j=1}^{n} \Big(\frac{\partial x^i}{\partial \xi^j} - S^{ij}\Big)^2 \mathrm{d}\boldsymbol{\xi}\ . \tag{7.88}$$

The Jacobian-weighted functionals $I_{\mathrm{jw},1}$ and $I_{\mathrm{jw},2}$ can also be interpreted as one more form for the functionals of alignment for vector fields. Let a vector field be given by n vectors

$$\boldsymbol{v}_i(\boldsymbol{x})\ , \qquad i = 1, \dots, n\ , \qquad \boldsymbol{v}_i = (v_i^1, \dots, v_i^n)\ .$$

Then, the following form of a functional of grid attraction to the given vector field can be defined:

$$I_{\mathrm{vf},3} = \int_{X^n} \sum_{i=1}^{n} |\nabla \xi^i - \boldsymbol{v}_i|^2 \mathrm{d}\boldsymbol{x} \ . \tag{7.89}$$

This functional is, in fact, the Jacobian-weighted functional (7.85) with

$$S^{ij} = v_i^j \ , \qquad i, j = 1, \ldots, n \ .$$

Note that the functionals of the form (7.84) and (7.88) are more efficient for attracting grid lines to the corresponding vector fields than the functionals of the type (7.89). This is because the former are concerned with attraction to the specified directions only, while with the latter, an attraction to both the directions and the specified lengths is required.

The form of the Jacobian-weighted functional gives a clear guideline for producing nonfolded grids. This guideline is based upon the following global univalence theorem.

Theorem 1 *Let $F : U \to R^n$ be a differentiable mapping, where U is the rectangular region of $R^n : U = \{\boldsymbol{x} : \boldsymbol{x} \in R^n | a_i \le x_i \le b_i\}$. If the Jacobian matrix of F at $\boldsymbol{x}$ is positive for every $\boldsymbol{x} \in U$, then F is globally one-to-one in U.*

Recall that an $n \times n$ real matrix A is positive if every principal minor of A is positive. Thus, in order to obtain one-to-one coordinate transformations, this theorem suggests that one should use only positive matrices as the matrix S.

The minimization of the functional $I_{\mathrm{jw},1}$ generates a transformation $\boldsymbol{\xi}(\boldsymbol{x})$ whose Jacobian matrix may be so close to the matrix S that the matrix $\{\partial\xi^i/\partial x^j\}$ is also positive. Thus, the matrix $\{\partial x^i/\partial\xi^j\}$ is positive as well, and in accordance with the above theorem, the transformation $\boldsymbol{x}(\boldsymbol{\xi}) : \ \Xi^n \to X^n$ is a one-to-one mapping.

7.6 Energy Functionals of Harmonic Function Theory

The theory of harmonic maps is useful for formulating variational grid generation techniques which provide well-posed problems of grid generation.

7.6.1 General Formulation of Harmonic Maps

First, we consider the definition of a harmonic map between two general n-dimensional Riemannian manifolds X^n and Z^n with covariant metric tensors d_{ij} and D_{ij} in some local coordinates $x^i, \ i = 1, \ldots, n$, and $z^i, \ i = 1, \ldots, n$, respectively.

Every $C^1(X^n)$ map $\boldsymbol{z}(\boldsymbol{x}) : X^n \to Z^n$ defines an energy density by the following formula:

$$e(\boldsymbol{z}) = \frac{1}{2} d^{ij}(\boldsymbol{x}) D_{kl}(\boldsymbol{z}) \frac{\partial z^k}{\partial x^i} \frac{\partial z^l}{\partial x^j} \ , \qquad i, j, k, l = 1, \dots, n \ , \tag{7.90}$$

where $\{d^{ij}\}$ is the contravariant metric tensor of X^n, i.e. $d_{ij} d^{jk} = \delta_i^k$. The total energy associated with the mapping $\boldsymbol{z}(\boldsymbol{x})$ is then defined as the integral of (7.90) over the manifold X^n:

$$E(\boldsymbol{z}) = \int_{X^n} e(\boldsymbol{z}) \mathrm{d}X^n \ . \tag{7.91}$$

A transformation $\boldsymbol{z}(\boldsymbol{x})$ of class $C^2(X^n)$ is referred to as a harmonic mapping if it is a critical point of the functional of the total energy (7.91). The Euler–Lagrange equations whose solution minimizes the energy functional (7.91) are given by

$$\frac{1}{\sqrt{d}} \frac{\partial}{\partial x^k} \Big(\sqrt{d} d^{kj} \frac{\partial z^l}{\partial x^j} \Big) + d^{kj} \Gamma^l_{mp} \frac{\partial z^m}{\partial x^k} \frac{\partial z^p}{\partial x^j} = 0 \ , \tag{7.92}$$

where $d = \det(d_{ij})$ and Γ^l_{mp} are Christoffel symbols of the second kind on the manifold Z^n :

$$\Gamma^l_{mp} = \frac{1}{2} D^{lj} \Big(\frac{\partial D_{jm}}{\partial z^p} + \frac{\partial D_{jp}}{\partial z^m} - \frac{\partial D_{mm}}{\partial z^j} \Big) \ . \tag{7.93}$$

The following theorem guarantees the uniqueness of the harmonic mapping.

Theorem 2 *Let X^n, with metric d_{ij}, and Z^n, with metric D_{ij}, be two Riemannian manifolds with boundaries ∂X^n and ∂Z^n, and let $\phi : X^n \to Z^n$ be a diffeomorphism. If the curvature of Z^n is nonpositive and ∂Z^n is convex (with respect to the metric D_{ij}), then there exists a unique harmonic map $\boldsymbol{z}(\boldsymbol{x}) : X^n \to Z^n$ such that $\boldsymbol{z}(\boldsymbol{x})$ is a homotopy equivalent to ϕ. In other words, one can deform $\boldsymbol{z}$ to ϕ by constructing a continuous family of maps $\boldsymbol{g}_t : X^n \to Z^n$, $t \in [0, 1]$, such that $\boldsymbol{g}_0(\boldsymbol{x}) = \phi(\boldsymbol{x})$, $\boldsymbol{g}_1(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})$, and $\boldsymbol{g}_t(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \partial X^n$.*

7.6.2 Application to Grid Generation

In application of the harmonic theory to grid generation, the manifold Z^n is assumed to correspond to the computational domain Ξ^n, with a Euclidean metric $D_{ij} = \delta^i_j$. Since the Euclidean space Ξ^n is flat, i.e. it has zero curvature, and the domain Ξ^n is constructed by the user, both requirements of the above theorem can be satisfied. For the manifold X^n, one uses a set of the points of a physical domain X^n with an introduced Riemannian metric d_{ij}. The functional of the total energy (7.91) then has the form

$$E(\boldsymbol{\xi}) = \frac{1}{2} \int_{X^n} \Big(\sqrt{d} d^{kl} \frac{\partial \xi^i}{\partial x^k} \frac{\partial \xi^i}{\partial x^l} \Big) \mathrm{d}\boldsymbol{x} \ , \qquad i, k, l = 1, \dots, n. \tag{7.94}$$

And for the Euler–Lagrange equations (7.92), we have

$$\frac{1}{\sqrt{d}}\frac{\partial}{\partial x^k}\Big(\sqrt{d}d^{kj}\frac{\partial \xi^i}{\partial x^j}\Big) = 0\ , \qquad i, j, k = 1, \dots, n, \tag{7.95}$$

since, from (7.93), $\Gamma^l_{mp} = 0$. The left-hand part of (7.95) is the Beltrami operator Δ_B, so (7.95) is equivalent to

$$\Delta_B(\boldsymbol{\xi}) = 0\ . \tag{7.96}$$

Equation (7.95), in contrast to (7.92), are linear and of elliptic type, and have a conservative form. Therefore, they satisfy the maximum principle, and the Dirichlet boundary value problem is a well-posed problem for this system of equations, i.e. the above theorem is proved very easily for the functional (7.94).

Equation (7.95) can be reformulated with interchanged dependent and independent variables in the typical manner, by multiplying the system by $\partial x^l/\partial \xi^i$ and summing over i. As a result, we obtain

$$\overline{d}^{km}\frac{\partial^2 x^l}{\partial \xi^k \partial \xi^m} - \frac{1}{\sqrt{d}}\frac{\partial}{\partial x^k}(\sqrt{d}d^{kl}) = 0\ , \qquad k, l, m = 1, \dots, n\ , \tag{7.97}$$

where

$$\overline{d}^{km} = d^{ij}\frac{\partial \xi^k}{\partial x^i}\frac{\partial \xi^m}{\partial x^j}\ , \qquad i, j, k, m = 1, \dots, n,$$

are the elements of the contravariant metric tensor of the Riemannian manifold X^n in the coordinates ξ^i.

7.6.3 Relation to Other Functionals

Some of the functionals given earlier are identical to the functionals of energy (7.91) and (7.94). For example, the smoothness functional (7.41) is the functional of the form (7.94) with the Euclidean metric $d_{ij} = \delta^i_j$ in X^n.

Analogously, the diffusion functional (7.64) can be interpreted as the functional (7.94), with the contravariant metric tensor d_{ij} in X^n satisfying the condition

$$\sqrt{d}d^{ij} = w\delta^i_j\ , \qquad i, j = 1, \dots, n\ .$$

From this relation, we readily obtain

$$d_{ij} = w^{2/(n-2)}\delta^i_j\ , \qquad i, j = 1, \dots, n\ .$$

Thus, $d_{ij} = w^2\delta^i_j$ for $n = 3$. The above formula fails to define the corresponding metric in the two-dimensional domain X^2. However, the formulation of the diffusion functional (7.64) as the energy functional (7.94) can be accomplished by using the Euclidean metric in X^n and the Riemannian metric in Ξ^n for arbitrary n, by setting

$$D_{ij} = w\delta^i_j \ , \qquad i, j = 1, \ldots, n \ .$$

The functional of inhomogeneous diffusion (7.65) can be interpreted in a similar manner by taking

$$d_{ij} = w\delta^i_j \ , \qquad D_{ij} = w_i\delta^i_j \ , \qquad i, j = 1, \ldots, n \ .$$

Similarly, the functionals (7.62), (7.63), and (7.73) can be identified with the functional of energy (7.94).

Other applications of the functionals of energy to generate surface and hypersurface grids will be discussed in Chaps. 8 and 9.

7.7 Combinations of Functionals

The functionals described in Sects. 8.2, 8.3, 8.4, 8.5 and 8.6 are used to control and realize various grid properties. This is carried out by combining these functionals with weights in the form

$$I = \sum_i \lambda_i I_i \ , \qquad i = 1, \ldots, k \ . \tag{7.98}$$

Here, $\lambda_i, \ i = 1, \ldots, k$, are specified parameters which determine the individual contribution of each functional I_i to I. The ranges of the parameters λ_i controlling the relative contributions of the functionals can be defined readily when the functionals I_i are dimensionally homogeneous. However, if they are dimensionally inhomogeneous, then the selection of a suitable value for λ_i presents some difficulties. A common rule for selecting the parameters λ_i involves making each component $\lambda_i I_i$ in (7.98) of a similar scale by using a dimensional analysis.

The most common practice in forming the combination (7.98) uses both the functionals of adaptation to the physical solution and the functionals of grid regularization. The first reason for using such a strategy is connected with the fact that the process of adaptation can excessively distort the form of the grid cells. The distortion can be prevented by functionals which impede cell deformation. These functionals are ones which control grid skewness, smoothness, and conformality. The second reason for using the regularization functionals is connected with the natural requirement for the well-posedness of the grid generation process. This requirement is achieved through the utilization of convex functionals in variational grid generators. The convex functionals are represented by energy-type functionals, producing harmonic maps, and by the functionals of conformality.

The various functionals described above provide broad opportunities to control and realize the required grid properties, though problems still remain; these require more detailed studies of all properties of the functionals. The knowledge of these properties will allow one to utilize the functionals as efficient tools for generating high-quality grids.

7.7.1 Natural Boundary Conditions

In order to achieve the desired result more efficiently when generating grids through the variational approach, one needs to adjust the boundary conditions to the resulting Euler–Lagrange equations. As an illustration, we can consider the process of generating two-dimensional conformal grids with the Laplace equations derived from the functional of conformality (7.30). A conformal grid is not obtained with arbitrary boundary conditions, but only with strictly specified ones.

The natural boundary conditions for the Euler–Lagrange equations yielded by functionals are those for which the boundary contribution to the variation is zero. The natural boundary conditions are derived in the typical way, by writing out the first variation of the functional.

7.8 Comments

A detailed description of the fundamentals and theoretical results of the calculus of variations can be found in the monographs by Gelfand and Fomin (1963) and by Ladygenskaya and Uraltseva (1973).

Liseikin and Yanenko (1977), Danaev et al. (1978), Ghia et al. (1983), Brackbill and Saltzman (1982) and Bell and Shubin (1983) have each used the variational principle for grid adaptation.

The diffusive form of the adaptive functional (7.64) was formulated originally by Danaev et al. (1980) and Winslow (1981). A generalization of this functional to (7.65) with an individual weight function for each direction was realized by Eiseman (1987) and Reed et al. (1988). The most general variational formulation of the modified anisotropic diffusion approach was presented by Hagmeijer (1994). A variational principle for the Jacobian-weighted functional was formulated, studied, and developed by Knupp (1995, 1996) and Knupp et al. (2002). A considarable amount of work in determining and studying the conditions required to guarantee invertibility of coordinate transformations was published by Pathasarathy (1983) and Clement et al. (1996).

The functional measuring the alignment of the two-dimensional grid with a specified vector field was formulated by Giannakopoulos and Engel (1988). The extension of this approach to three dimensions was discussed by Brackbill (1993).

A variational method optimizing cell aspect ratios was presented and analyzed by Mastin (1992). A dimensionally homogeneous functional of two-dimensional grid skewness was proposed by Steinberg and Roache (1986).

The property of eccentricity for the univariate transformation $\boldsymbol{x}(\boldsymbol{\xi})$ was introduced by Sidorov (1966), while a three-dimensional extension was performed by Serezhnikova et al. (1989). A form of smoothness based on the eccentricity term was developed by Winkler et al. (1985).

The variational formulation of grid properties was described by Warsi and Thompson (1990).

The geometric interpretation of the approximation (7.52) was given by Steinberg and Roache (1986).

The introduction of the volume-weighted functional was originally proposed in two dimensions by Yanenko et al. (1977).

The approach of determining functionals which depend on invariants of orthogonal transformations of the metric tensor $\{g_{ij}\}$, to ensure that the problems are well-posed and to obtain more compact formulas for the Euler–Lagrange equations, was proposed by Jacquotte (1987). In his paper, the grids were constructed through functionals obtained by modeling different elastic and plastic properties of a deformed body.

The metric-weighted functional was formulated by Belinsky et al. (1975) for the purpose of generating quasiconformal grids.

The possibility of using harmonic function theory to provide a general framework for developing multidimensional mesh generators was discussed by Dvinsky (1991). The interpretation of the functional of diffusion as a version of the energy functional was presented by Brackbill (1993). A detailed survey of the theory of harmonic mappings was published by Eells and Lenaire (1988).

References

Belinsky, P. P., Godunov, S. K., Ivanov, Yu V, & Yanenko, I. K. (1975). The use of one class of quasiconformal mappings to generate numerical grids in regions with curvilinear boundaries. *Zh. Vychisl. Maths. Math. Fiz.*, *15*, 1499–1511 (Russian).

Bell, J. B., & Shubin, G. R. (1983). An adaptive grid finite-difference method for conservation laws. *Journal of Computational Physics*, *52*, 569–591.

Brackbill, J. U. (1993). An adaptive grid with directional control. *Journal of Computational Physics*, *108*, 38–50.

Brackbill, J. U., & Saltzman, J. (1982). Adaptive zoning for singular problems in two directions. *Journal of Computational Physics*, *46*, 342–368.

Clement, P., Hagmeijer, R., & Sweers, G. (1996). On the invertibility of mapping arising in 2D grid generation problems. *Numerische Mathematik*, *73*, 37–51.

Danaev, N. T., Liseikin, V. D., & Yanenko, N. N. (1978). A method of nonstationary coordinates in gas dynamics. Foreign Technology Division, Wright-Patterson Air Force Base, Ohio, FTD-ID(RS)-1673-78.

Danaev, N. T., Liseikin, V. D., & Yanenko, N. N. (1980). Numerical solution on a moving curvilinear grid of viscous heat-conducting flow about a body of revolution. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *11*(1), 51–61 (Russian).

Dvinsky, A. S. (1991). Adaptive Grid Generation from Harmonic Maps on Riemannian Manifolds. *Journal of Computational Physics*, *95*, 450–476.

Eells, J., & Lenaire, L. (1988). Another report on harmonic maps. *Bulletin of the London Mathematical Society*, *20*(5), 385–524.

Eiseman, P. R. (1987). Adaptive grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 321–376.

Garanzha, V. A. (2000). Barrier variational generation of quasi-isimetric grids. *Computational Mathematics and Mathematical Physics*, *40*, 1617–1637.

Garanzha, V. A., & Kudryavtseva, L. N. (2012). Generation of three-dimensional Delaunay meshes from weakly structured and inconsistent data. *Computational Mathematics and Mathematical Physics*, *52*, 427–447.

Gelfand, I. M., & Fomin, S. V. (1963). *Calculus of variations*. Englewood Cliffs: Prentice-Hall.

Ghia, K. N., Ghia, U., & Shin, C. T. (1983). Adaptive grid generation for flows with local high gradient regions. In K. N. Ghia & U. Ghia (Eds.), *Advances in grid generation* (pp. 35–47). Houston: ASME.

Giannakopoulos, A. E., & Engel, A. J. (1988). Directional control in grid generation. *Journal of Computational Physics*, *74*, 422–439.

Hagmeijer, R. (1994). Grid adaption based on modified anisotropic diffusion equations formulated in the parametric domain. *Journal of Computational Physics*, *115*(1), 169–183.

Jacquotte, O.-P. (1987). A mechanical model for a new grid generation method in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, *66*, 323–338.

Knupp, P. M. (1995). Mesh generation using vector-fields. *Journal of Computational Physics*, *119*, 142–148.

Knupp, P. M. (1996). Jacobian-weighted elliptic grid generation. *SIAM Journal on Scientific Computing*, *17*(6), 1475–1490.

Knupp, P., Margolin, L., & Shashkov, M. (2002). Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, *176*, 93–128.

Ladygenskaya, O. A., & Uraltseva, N. N. (1973). *Linear and quasilinear equations of elliptic type*. Moscow: Nauka (Russian).

Liseikin, V. D., & Yanenko, N. N. (1977). Selection of optimal numerical grids. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *8*(7), 100–104 (Russian).

Mastin, C. W. (1992). Linear variational methods and adaptive grids. *Computers and Mathematics with Applications*, *24*(5/6), 51–56.

Pathasarathy, T. (1983). *On global univalence theorems.*, Lecture notes in mathematics New York: Springer.

Reed, C. W., Hsu, C. C., & Shiau, N. H. (1988). An adaptive grid generation technique for viscous transonic flow problems. AIAA Paper 88-0313.

Serezhnikova, T. I., Sidorov, A. F., & Ushakova, O. V. (1989). On one method of construction of optimal curvilinear grids and its applications. *Soviet Journal of Numerical Analysis and Mathematical Modelling*, *4*(1), 137–155.

Sidorov, A. F. (1966). An algorithm for generating optimal numerical grids. *Trudy MIAN USSR*, *24*, 147–151 (Russian).

Steinberg, S., & Roache, P. J. (1986). Variational grid generation. *Numerical Methods for Partial Differential Equations*, *2*, 71–96.

Warsi, Z. U. A., & Thompson, J. F. (1990). Application of variational methods in the fixed and adaptive grid generation. *Computers and Mathematics with Applications*, *19*(8/9), 31–41.

Winkler, K. H., Mihalas, D., & Norman, M. L. (1985). Adaptive grid methods with asymmetric time-filtering. *Computer Physics Communications*, *36*, 121–140.

Winslow, A. M. (1981). Adaptive mesh zoning by the equipotential method. UCID-19062, Lawrence Livermore National Laboratories.

Yanenko, N. N., Danaev, N. T., & Liseikin, V. D. (1977). A variational method for grid generation. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *8*(4), 157–163 (Russian).

Chapter 8
Curve and Surface Grid Methods

8.1 Introduction

Curvilinear lines and surfaces are common geometrical objects in both structured and unstructured grid generation techniques. Curves appear in grid considerations as the boundary segments of two-dimensional domains or surface patches and as the edges of three-dimensional blocks. Surfaces arise as the boundary segments and/or faces of three-dimensional domains or blocks.

The main goal of grid generation on a curve is to provide boundary data for boundary-fitted grid generators for two-dimensional planar domains and surfaces. Analogously, surface grid generation is needed chiefly to build grids on the boundaries of three-dimensional domains or blocks in order to provide boundary data for volume grid techniques.

In the mapping concept, grid generation on a surface follows the construction of a set of surface patches, specification of a parametrization for every patch, and generation of one-dimensional curve grids on the edges of the surface patches to provide the boundary conditions for the surface grid generator. In fact, for the purpose of simplicity and for maintaining adherence of the surface grid techniques to the physical geometry, the grid is commonly generated in a parametric two-dimensional domain and then mapped onto the original patch of the surface.

Thus, the process of surface grid generation through the mapping approaches may be divided into three steps: forward mapping, grid generation, and backward mapping. The forward mapping is a representation of the background surface patch from a three-dimensional physical domain to a two-dimensional parameter area. Once the forward mapping is complete, the grid is generated in the parameter space and then mapped back into the physical space (backward mapping). A surface patch is formed as curvilinear triangle, trapezoid or a quadrilateral, with three or four boundary segments, respectively. The corresponding parametric domain may also have the shape of a triangle or a quadrilateral with curved boundary edges. The backward transformation from the parametric domain to the patch is defined by specifically adjusted interpolations. The specification depends on variations in surface features.

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_8

The generation of the grid on the parametric domain is derived through the same types of approach – algebraic, differential, and variational – as those which were described for planar domains. However, these approaches are adjusted by including the necessary surface characteristics, expressed in terms of the surface quadratic forms (see Sect. 3.3), to satisfy the required grid properties on the surface.

This chapter gives a review of some advanced techniques of curve and surface grid generation based on parametric mapping approaches.

8.2 Grids on Curves

Methods for generation of grids on curves are the simplest to formulate and analyze. These methods provide the background for the development of surface grid techniques. Some common mapping approaches to grid generation on curves are discussed in this section.

8.2.1 Formulation of Grids on Curves

A curve in n-dimensional space is represented parametrically by a smooth, nonsingular, vector-valued function from a normalized interval [0,1]:

$$\boldsymbol{x}(\varphi) : [0, 1] \to R^n \ , \qquad \boldsymbol{x}(\varphi) = [x^1(\varphi), \dots, x^n(\varphi)] \ . \tag{8.1}$$

Let the curve with the parametrization $\boldsymbol{x}(\varphi)$ be designated by S^{x1}. The transformation (8.1) provides a discrete grid on the curve S^{x1} by mapping the nodes of a uniform grid in the interval [0, 1] into S^{x1} with $\boldsymbol{r}(\varphi)$, i.e. the grid points $\boldsymbol{x}_i$, $i = 0, 1, \dots, N$, are defined as

$$\boldsymbol{x}_i = \boldsymbol{x}(ih) \ , \qquad h = 1/N \ .$$

However, the need to produce a grid with particular desirable properties requires the introduction of a control tool. Such control of the generation of a curve grid is carried out with strongly monotonic and smooth intermediate transformations

$$\varphi(\xi) : [0, 1] \to [0, 1] \ , \tag{8.2}$$

which generate the grid nodes φ_i on the interval [0, 1], where

$$\varphi_i = \varphi(ih) \ , \qquad i = 0, 1, \dots, N \ , \qquad h = 1/N \ .$$

The transformation $\varphi(\xi)$ is chosen in such a way that the composition

$$\boldsymbol{x}[\varphi(\xi)] : [0, 1] \to R^n \ , \tag{8.3}$$

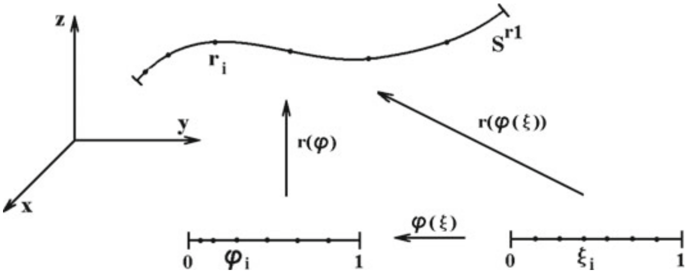


Fig. 8.1 Scheme of generation of a curve grid

which represents a new parametrization of S^{x1}, generates the grid nodes

$$\boldsymbol{x}_i = \boldsymbol{x}[\varphi(ih)] = \boldsymbol{x}(\varphi_i) \ , \qquad i = 0, 1, \ldots, N \ , \qquad h = 1/N \ , \tag{8.4}$$

with the desired properties. Figure 8.1 demonstrates the scheme of the generation of a curve grid.

Thus, the process of grid generation on a curve is turned into the definition of an intermediate transformation $\varphi(\xi)$ so as to provide a suitable parametrization of the curve. One such natural transformation is connected with the scaled arc length parameter ξ which, in analogy with (3.2), is defined by

$$\xi(\varphi) = \frac{1}{c}\int_0^{\varphi} \sqrt{g^{x\varphi}}\mathrm{d}x \ , \qquad c = \int_0^1 \sqrt{g^{x\varphi}}\mathrm{d}x \ , \tag{8.5}$$

where

$$g^{x\varphi} = \boldsymbol{x}_\varphi \cdot \boldsymbol{x}_\varphi = |\boldsymbol{x}_\varphi|^2 \ .$$

The function $\varphi(\xi)$, inverse to $\xi(\varphi)$, is subject to the condition

$$\frac{\mathrm{d}\varphi}{\mathrm{d}\xi} = \frac{c}{\sqrt{g^{x\varphi}}} \ . \tag{8.6}$$

Therefore, we have, for the grid nodes $\varphi_i = \varphi(ih)$ in the interval [0, 1], the relation

$$\frac{\varphi_{i+1} - \varphi_i}{h} \approx \frac{\mathrm{d}\varphi}{\mathrm{d}\xi} = \frac{c}{\sqrt{g^{x\varphi}}} \ , \qquad i = 0, 1, \ldots, N \ , \qquad h = 1/N \ , \tag{8.7}$$

and consequently we obtain, for the grid nodes $\boldsymbol{x}_i$ on the curve S^{x1},

$$|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i| \approx |\boldsymbol{x}_\varphi|(\varphi_{i+1} - \varphi_i) \approx ch \ , \qquad i = 0, 1, \ldots, N \ , \qquad h = 1/N \ . \tag{8.8}$$

Equations (8.6)–(8.8) are examples of the equidistribution principle considered in Sect. 4.3, which is based on a specification of distances between the grid points in accordance with a rule of inverse proportionality to a weight function.

8.2.2 Grid Methods

The main approach to generating one-dimensional grids on curves is based on a specification of the grid step spacing. The approach is realized by direct reparametrization of the curve with suitable univariate intermediate transformations $\varphi(\xi)$, such as the ones considered in Chap. 4, or by the formulation of equations and functionals through the first derivative of the intermediate transformations.

Differential Approach

The simplest differential method for the definition of an intermediate transformation $\varphi(\xi)$ relies on the solution of the initial-value problem in the form of (8.6)

$$\begin{aligned} &\frac{\mathrm{d}\varphi}{\mathrm{d}\xi} = \frac{c}{F(\varphi)} , \qquad 0 < \varphi \leq 1 , \\ &\varphi(0) = 0 , \qquad c = \int_0^1 F(\varphi)\mathrm{d}\varphi , \end{aligned} \tag{8.9}$$

where $F(\varphi)$ is a nonnegative function specified by the user. Differentiation of (8.9) with respect to ξ allows one to eliminate the constant c and obtain the two-point boundary value problem

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\xi}\Big(\frac{\mathrm{d}\varphi}{\mathrm{d}\xi}F(\varphi)\Big) = 0 , \qquad 0 < \varphi < 1 , \\ &\varphi(0) = 0 , \qquad \varphi(1) = 1 . \end{aligned} \tag{8.10}$$

Equations (8.9) and (8.10) represent the formulation of the equidistribution principle of Chap. 4. Taking advantage of (8.6)–(8.8), we see that the solution of (8.9) or (8.10) produces a grid on the curve S^{x1} with a grid spacing inversely proportional to $\sqrt{g^{x\varphi}}F(\varphi)$. Thus, in the weight-concept formulation

$$\frac{|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|}{h} \approx \frac{c_1}{w(\varphi_i)} , \qquad i = 0, \ldots, N-1 ,$$

we obtain

$$F(\varphi) = w(\varphi)\sqrt{g^{x\varphi}} , \tag{8.11}$$

which, in the case $w(\varphi) = 1$, corresponds to the scaled-arc-length parametrization (8.6).

The weight function $w(\varphi)$ is specified by the user in accordance with the requirement to cluster the grid points in the zones of particular interest. It can be defined through the derivatives of the physical quantities or through the measures of the curve features described in Sect. 3.5, in particular, through the metric tensor $g^{x\varphi}$, curvature k, or tension τ. The specification determines the concentration of the curve grid, which becomes larger in the areas of large values of the weight function.

Variational Approach

In accordance with the results of Chap. 7, the differential formulation (8.10) of the equidistribution principle is obtained from the minimization of the functional

$$I = \int_0^1 \frac{1}{F(\varphi)} \left(\frac{\mathrm{d}\xi}{\mathrm{d}\varphi}\right)^2 \mathrm{d}\varphi \ , \tag{8.12}$$

whose Euler–Lagrange equations

$$\frac{\partial}{\partial\varphi}\left(\frac{\mathrm{d}\xi}{\mathrm{d}\varphi}\frac{1}{F(\varphi)}\right) = 0$$

(see Sect. 7.2) are equivalent to (8.10). Taking advantage of (8.11), we obtain an equivalent form of (8.12) through the weight function $w(\varphi)$:

$$\begin{aligned} I = \int_0^1 \frac{1}{w(\varphi)\sqrt{g^{x\varphi}}} \left(\frac{\mathrm{d}\xi}{\mathrm{d}\varphi}\right)^2 \mathrm{d}\varphi &= \int_{S^{x1}} \frac{1}{g^{x\varphi} w(\varphi)} \left(\frac{\mathrm{d}\xi}{\mathrm{d}\varphi}\right)^2 \mathrm{d}S^{x1} \\ &= \int_{S^{x1}} \frac{1}{g^{x\xi} w(\varphi)} \mathrm{d}S^{x1} \ , \end{aligned} \tag{8.13}$$

where

$$g^{x\xi} = \frac{\mathrm{d}\boldsymbol{x}[\varphi(\xi)]}{\mathrm{d}\xi} \cdot \frac{\mathrm{d}\boldsymbol{x}[\varphi(\xi)]}{\mathrm{d}\xi} = \boldsymbol{x}_\varphi \cdot \boldsymbol{x}_\varphi \left(\frac{\mathrm{d}\varphi}{\mathrm{d}\xi}\right)^2 = g^{x\varphi}\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\xi}\right)^2 .$$

In analogy with the differential approach, the weight function $w(\varphi)$ in (8.13) is defined by the values of the solution or its derivatives and/or by the curve quality measures.

Monitor Formulation

The monitor approach for controlling the grid steps on a curve S^{x1} relies on the introduction of a monitor curve which is defined by the values of some vector function $\boldsymbol{f} : X^n \to R^k, \ \boldsymbol{f}(\boldsymbol{x}) = [f^1(\boldsymbol{x}), \dots, f^k(\boldsymbol{x})]$, over the curve, where X^n is a domain containing the curve. The parameter function (8.1) and $\boldsymbol{f}(\boldsymbol{x})$ define a parametrization $\boldsymbol{r}(\varphi)$ of the monitor curve designated by S^{r1} :

$$\boldsymbol{r}(\varphi) : [0, 1] \to R^{n+k} \ , \qquad \boldsymbol{r}(\varphi) = [r^1(\varphi), \dots, r^{n+k}(\varphi)] \ , \tag{8.14}$$

where

$$r^i(\varphi) = x^i(\varphi)\ , \qquad i = 1, \dots, n\ , \qquad r^{n+j}(\varphi) = f^j[\boldsymbol{x}(\varphi)]\ , \qquad j = 1, \dots, k\ .$$

We obtain

$$g^{r\varphi} = g^{x\varphi} + g^{f\varphi}\ , \tag{8.15}$$

where

$$\begin{aligned} &g^{f\varphi} = \frac{\mathrm{d}\boldsymbol{f}[\boldsymbol{x}(\varphi)]}{\mathrm{d}\varphi} \cdot \frac{\mathrm{d}\boldsymbol{f}[\boldsymbol{x}(\varphi)]}{\mathrm{d}\varphi} = \frac{\partial f^j}{\partial x^l}\frac{\partial f^j}{\partial x^m}\frac{\mathrm{d}x^l}{\mathrm{d}\varphi}\frac{\mathrm{d}x^m}{\mathrm{d}\varphi}\ , \\ &j = 1, \dots, k\ , \qquad l, m = 1, \dots, n\ . \end{aligned}$$

In the monitor approach, the grid on the curve S^{x1} is obtained by mapping a uniform grid on S^{r1} with the projection function $P : R^{n+k} \to R^n$ $(P(x^1, \dots, x^{n+k}) = (x^1, \dots, x^n))$. The uniform grid on S^{r1} is derived by means of the arc-length approach, realized with the initial-value problem (8.6), with the two-point boundary value problem (8.10), or with the variational problem (8.12) for $F(\varphi) = \sqrt{g^{r\varphi}}$. As a result, we have for the intermediate transformation $\varphi(\xi)$

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\frac{\mathrm{d}\varphi}{\mathrm{d}\xi}\sqrt{g^{r\varphi}}\right) = 0\ , \qquad 0 < \xi < 1\ , \\ &\varphi(0) = 0\ , \qquad \varphi(1) = 1, \end{aligned} \tag{8.16}$$

with $g^{r\varphi}$ specified by (8.15). The transformation $\boldsymbol{x}[\varphi(\xi)]$ defines the grid on the surface S^{x1}, which coincides with the grid projected from S^{r1}. Since

$$\frac{|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|}{h} \approx \left|\frac{\mathrm{d}x}{\mathrm{d}\varphi}\right|\frac{\mathrm{d}\varphi}{\mathrm{d}\xi} = \frac{\sqrt{g^{x\varphi}}}{\sqrt{g^{r\varphi}}} = \frac{1}{\sqrt{1 + g^{f\varphi}/g^{x\varphi}}}\ ,$$

the monitor approach provides node clustering in the zones of large values of $g^{f\varphi}$ and, consequently, where the derivatives of the function $\boldsymbol{f}(\boldsymbol{x})$ are large.

Note that in accordance with (8.13), the variational formulation of the monitor approach is given by the functional

$$I = \int_{S^{r1}} \frac{1}{g^{r\varphi}} \mathrm{d}S^{r1}\ . \tag{8.17}$$

8.3 Formulation of Surface Grid Methods

It is assumed in this chapter that the surface under consideration lies in the Euclidean space R^3. Without loss of generality, we suggest that the surface, denoted as S^{x2}, is locally represented by a parametrization

$$\boldsymbol{x}(\boldsymbol{s}) : S^2 \to R^3 , \qquad \boldsymbol{x}(\boldsymbol{s}) = [x^1(\boldsymbol{s}), x^2(\boldsymbol{s}), x^3(\boldsymbol{s})] , \qquad \boldsymbol{s} = (s^1, s^2) , \tag{8.18}$$

where S^2 is a two-dimensional parametric domain with the Cartesian coordinates $s^1,\ s^2$, while $\boldsymbol{x}(\boldsymbol{s})$ is a smooth, nondegenerate function.

8.3.1 Mapping Approach

The generation of a grid on the surface S^{x2} is based on the introduction of a standard computational domain Ξ^2 with a reference grid and a one-to-one transformation

$$\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^2 \to S^2 , \qquad \boldsymbol{s}(\boldsymbol{\xi}) = [s^1(\boldsymbol{\xi}), s^2(\boldsymbol{\xi})] , \qquad \boldsymbol{\xi} \in \Xi^2 . \tag{8.19}$$

This mapping (8.19), in fact, generates a grid in the two-dimensional domain S^2. However, a required grid on the surface S^{x2} is defined by mapping some reference grid in Ξ^2 onto S^{x2} by the composite transformation

$$\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})] : \Xi^2 \to S^{x2} , \tag{8.20}$$

or, equivalently, by mapping with $\boldsymbol{x}(\boldsymbol{s})$ the grid generated in S^2 by some suitable transformation $\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^2 \to S^2$ (see Fig. 8.2).

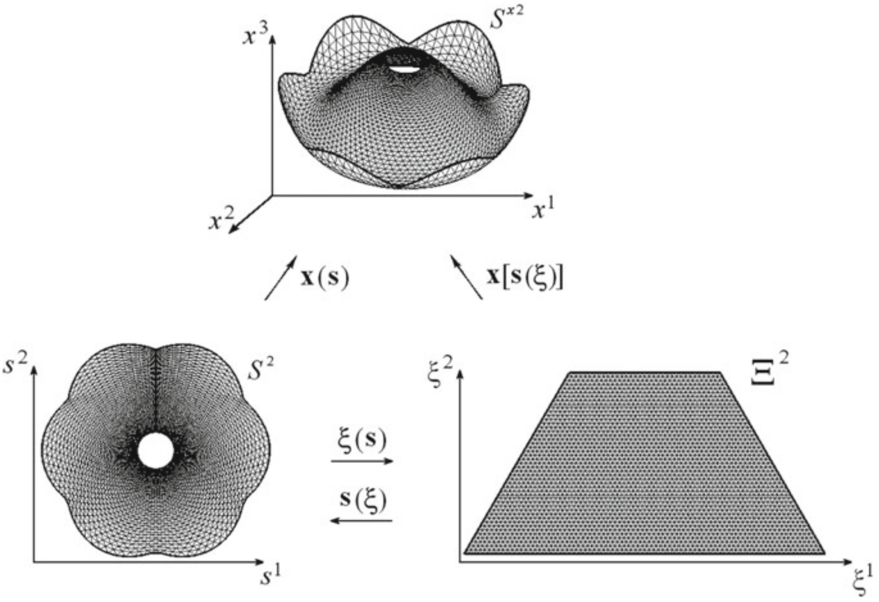


Fig. 8.2 Framework for generation of triangular surface grids

Thus, the problem of the generation of a surface grid is turned into the problem of choosing an appropriate computational domain Ξ^2 with a suitable reference grid and of constructing an adequate transformation between the computational and the parametric two-dimensional domains.

Some techniques for generating two-dimensional coordinate transformations of domains were considered in Chaps. 4–7. However, the direct application of the approaches discussed above to the generation of two-dimensional planar grids may not lead to satisfactory grids on the surface, since the grid on the surface, obtained by the backward mapping $\boldsymbol{x}(\boldsymbol{s})$, may become significantly distorted because of the mapping. The formulation of the proper methods should take into account the geometric features of the surface under consideration and properties of the parametrization $\boldsymbol{x}(\boldsymbol{s})$.

8.3.2 Associated Metric Relations

The features of the surface are described through the first and second fundamental forms (Sect. 3.3), in particular, by the metric elements which are derived from the dot products of the tangential vectors to the coordinate lines.

The scheme of grid generation on the surface S^{x2} implies, in fact, two parametrizations of the surface: the original parametrization $\boldsymbol{x}(\boldsymbol{s})$ with the parametric space S^2 and the final one $\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]$ with the parametric space Ξ^2. The original parametrization is considered to be an input, while the final parametrization $\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]$ is an output of the surface grid generation process. The role of the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$ is to correct the drawbacks of the original mapping $\boldsymbol{x}(\boldsymbol{s})$ by transforming it into $\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]$, which should generate a grid with the properties required by the user.

The covariant metric tensor of the surface in the coordinates $\xi^1,\ \xi^2$, denoted by

$$G^{x\xi} = \{g_{ij}^{x\xi}\}\ , \qquad i, j = 1, 2,$$

is defined by the dot product of the tangent vectors $\boldsymbol{x}_{\xi^i} = \partial \boldsymbol{x}(\boldsymbol{s})/\partial \xi^i(\boldsymbol{\xi}),\ i = 1, 2$, i.e.

$$g_{ij}^{x\xi} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j}\ , \qquad i, j = 1, 2\ .$$

Analogously, the elements of the covariant metric tensor

$$G^{xs} = \{g_{ij}^{xs}\}\ , \qquad i, j = 1, 2\ ,$$

in the coordinates $s^i,\ i = 1, 2$, are expressed in the following form:

$$g_{ij}^{xs} = \frac{\partial \boldsymbol{x}}{\partial s^i} \cdot \frac{\partial \boldsymbol{x}}{\partial s^j}\ , \qquad i, j = 1, 2\ .$$

It is clear that

$$\begin{aligned} g^{x\xi}_{ij} &= g^{xs}_{mk} \frac{\partial s^m}{\partial \xi^i} \frac{\partial s^k}{\partial \xi^j} \ , \\ g^{xs}_{ij} &= g^{x\xi}_{mk} \frac{\partial \xi^m}{\partial s^i} \frac{\partial \xi^k}{\partial s^j} \ , \qquad i, j, k, m = 1, 2 \ , \end{aligned} \tag{8.21}$$

using the convention of summation over repeated indices.

The contravariant metric tensor of the surface in the coordinates ξ^i, $i = 1, 2$, denoted by

$$G_{\xi x} = \{g^{ij}_{\xi x}\} \ , \qquad i, j = 1, 2 \ ,$$

and in the coordinates s^i, $i = 1, 2$, written as

$$G_{sx} = \{g^{ij}_{sx}\} \ , \qquad i, j = 1, 2 \ ,$$

is the matrix inverse to $G^{x\xi}$ and G^{xs}, respectively. Thus,

$$\begin{aligned} g^{ij}_{\xi x} &= (-1)^{i+j} g^{x\xi}_{3-i\ 3-j}/g^{x\xi} \ , \qquad & g^{x\xi}_{ij} &= (-1)^{i+j} g^{x\xi} g^{3-i,3-j}_{\xi x} \ , \\ g^{ij}_{sx} &= (-1)^{i+j} g^{xs}_{3-i\ 3-j}/g^{xs} \ , \qquad & g^{xs}_{ij} &= (-1)^{i+j} g^{xs} g^{3-i,3-j}_{sx} \ , \end{aligned} \tag{8.22}$$

where $i, j = 1, 2$, and

$$g^{x\xi} = \det G^{x\xi} \ , \qquad g^{xs} = \det G^{xs} \ .$$

Note that here, on the right-hand side of every relation in (8.22), the summation convention is not applied over i and j.

Similarly to (8.21), the elements of the contravariant metric tensor in the coordinates ξ^i and s^i, $i = 1, 2$, are connected by the relations

$$\begin{aligned} g^{ij}_{\xi x} &= g^{mk}_{sx} \frac{\partial \xi^i}{\partial s^m} \frac{\partial \xi^j}{\partial s^k} \ , \\ g^{ij}_{sx} &= g^{mk}_{\xi x} \frac{\partial s^i}{\partial \xi^m} \frac{\partial s^j}{\partial \xi^k} \ , \qquad i, j, k, m = 1, 2 \ . \end{aligned} \tag{8.23}$$

These relations and (8.21) readily yield

$$\begin{aligned} g^{ii}_{sr} &= g^{mk}_{\xi x} g^{s\xi}_{mk} \ , \qquad i, j, k, m = 1, 2 \ , \\ g^{x\xi} &= g^{xs} g^{s\xi} \ , \end{aligned} \tag{8.24}$$

where $g^{s\xi}_{mk},\ k, m = 1, 2$, are the elements of the covariant metric tensor of S^2 in the coordinates ξ^i, i.e. $g^{s\xi}_{mk} = \boldsymbol{s}_{\xi^m} \cdot \boldsymbol{s}_{\xi^k}$, while $g^{s\xi} = \det\{g^{s\xi}_{ij}\} = \det^2\{\partial s^i/\partial \xi^j\}$.

Now we proceed to the description of some advanced grid generation techniques for the generation of grids on the surface S^{x2}.

8.4 Beltramian System

It is desirable to develop methods of surface grid generation which are invariant of the parametrizations $\boldsymbol{x}(\boldsymbol{s}) : S^2 \to S^{x2}$. One such surface grid generation system is obtained from the Beltrami second-order differential operator.

8.4.1 Beltramian Operator

The Beltrami operator Δ_B is defined as

$$\Delta_B[f] = \frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^j}(\sqrt{g^{xs}}g^{mj}_{sr}\frac{\partial}{\partial s^m}f)\ , \qquad j, m = 1, 2\ . \tag{8.25}$$

When S^{x2} is a plane and the coordinate system $s^1,\ s^2$ is orthonormal, i.e. $g^{xs}_{ij} = g^{ij}_{sx} = \delta^i_j$, then (8.25) is the Laplace operator. Thus, the operator (8.25) is a generalization of the Laplacian on a surface.

The Beltrami operator does not depend on the parametrization of the surface. For instance, let $u^i,\ i = 1, 2$, be another parametrization. Taking into account the general relation (2.57), we obtain, for arbitrary smooth functions A^1 and A^2,

$$\frac{\partial A^j}{\partial s^j} = \frac{1}{\sqrt{g^{su}}}\frac{\partial}{\partial u^k}\Bigl(\sqrt{g^{su}}A^m\frac{\partial u^k}{\partial s^m}\Bigr)\ , \qquad i, j, k, m = 1, 2\ , \tag{8.26}$$

where

$$\sqrt{g^{su}} = \det\Bigl\{\frac{\partial s^i}{\partial u^m}\Bigr\}\ , \qquad i, m = 1, 2\ .$$

Assuming

$$A^j = \sqrt{g^{xs}}g^{mj}_{sr}\frac{\partial}{\partial s^m}f\ , \qquad j = 1, 2\ ,$$

we have, from (8.25),

$$\frac{1}{\sqrt{g^{xs}}}\frac{\partial A^j}{\partial s^j} = \Delta_B[f]\ , \qquad j = 1, 2\ .$$

Therefore, (8.26) yields

$$\Delta_B[f] = \frac{1}{\sqrt{g^{ru}}}\frac{\partial}{\partial u^j}\left(\sqrt{g^{ru}}g^{mp}_{sr}\frac{\partial u^j}{\partial s^p}\frac{\partial u^k}{\partial s^m}\frac{\partial f}{\partial u^k}\right), \qquad j,k,m,p = 1,2\ ,$$

where $g^{ru} = g^{xs}g^{su}$. Thus, using (8.23), we obtain

$$\frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^j}\left(\sqrt{g^{xs}}g^{mj}_{sr}\frac{\partial}{\partial s^m}f\right) = \frac{1}{\sqrt{g^{ru}}}\frac{\partial}{\partial u^j}\left(\sqrt{g^{ru}}g^{jk}_{ur}\frac{\partial f}{\partial u^k}\right), \qquad j,k,m = 1,2\ .$$

So, the value of the operator Δ_B at a function f does not depend on any parametrization of the surface S^{x2}.

8.4.2 Surface Grid System

A surface grid system can be formed, in analogy with the Laplace system (6.5), by the Beltrami equations with respect to the components $\xi^i(\boldsymbol{s})$ of the inverse mapping of the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$:

$$\Delta_B[\xi^i] = 0\ , \qquad i = 1,2\ ,$$

i.e. taking advantage of (8.25),

$$\frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^j}\left(\sqrt{g^{xs}}g^{mj}_{sr}\frac{\partial \xi^i}{\partial s^m}\right) = 0\ , \qquad i,j,m = 1,2\ . \tag{8.27}$$

The system (8.27) is a generalization of the two-dimensional Laplace system (6.5) applied to the generation of planar grids. As in the case of the Laplace equations, the solution to a Dirichlet boundary value problem for (8.27) satisfies the conditions necessary for efficient surface grid generation. In particular, if the computational domain Ξ^2 is chosen to be convex, then the values of the function $\boldsymbol{\xi}(\boldsymbol{s})$ satisfying (8.27) lie in Ξ^2 if $\boldsymbol{\xi}(\boldsymbol{s})$ maps the boundary of S^2 onto the boundary of Ξ^2. Moreover, the transformation $\boldsymbol{\xi}(\boldsymbol{s})$ is a one-to-one transformation if it is homeomorphic on the boundary. This is the main justification for the formulation of (8.27) with respect to the inverse transformation $\boldsymbol{\xi}(\boldsymbol{s})$.

In order to generate a grid on S^2, the system (8.27) is inverted to interchange its dependent and independent variables. This is done in the typical manner, by multiplying the system by $\partial s^l/\partial \xi^i$ and summing over i. Thus, we obtain

$$\begin{aligned}&\frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^j}\left(\sqrt{g^{xs}}g^{mj}_{sr}\frac{\partial \xi^i}{\partial s^m}\right)\frac{\partial s^l}{\partial \xi^i}\\ &= -g^{mj}_{sr}\frac{\partial \xi^i}{\partial s^m}\frac{\partial \xi^k}{\partial s^j}\frac{\partial^2 s^l}{\partial \xi^i \partial \xi^k} + \frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^j}(\sqrt{g^{xs}}g^{lj}_{sr}) = 0\ , \quad i,j,k,l,m = 1,2\ .\end{aligned}$$

So, taking into account (8.23), the system (8.27) is transformed into the following nonlinear system with respect to the dependent variables $s^i(\boldsymbol{\xi})$:

$$g^{ij}_{\xi x}\frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} = \frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^i}(\sqrt{g^{xs}} g^{il}_{sr}) \ , \qquad i,j,l = 1,2 \ . \tag{8.28}$$

We shall refer to these equations as the inverted Beltrami equations.

The right-hand part of (8.28) is, in fact, the value of the Beltrami operator applied to the function s^l; thus, the system (8.28) can be written out as follows:

$$g^{ij}_{\xi x}\frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} = \Delta_B[s^l] \ , \qquad i,j,l = 1,2 \ .$$

Taking advantage of (8.22), the system (8.28) is transformed into a system whose coefficients are determined by the elements of the metric tensors $\{g^{x\xi}_{ij}\}$ and $\{g^{xs}_{ij}\}$:

$$B^{x\xi}_2[s^l] = (-1)^l \frac{g^{x\xi}}{\sqrt{g^{xs}}}\left(\frac{\partial}{\partial s^2}\frac{g^{xs}_{3-l\ 1}}{\sqrt{g^{xs}}} - \frac{\partial}{\partial s^1}\frac{g^{xs}_{3-l\ 2}}{\sqrt{g^{xs}}}\right) \ , \qquad l = 1,2 \ , \tag{8.29}$$

where $B^{x\xi}_2$ is an operator defined as follows:

$$B^{x\xi}_2[y] = g^{x\xi}_{22}\frac{\partial^2 y}{\partial \xi^1 \partial \xi^1} - 2g^{x\xi}_{12}\frac{\partial^2 y}{\partial \xi^1 \partial \xi^2} + g^{x\xi}_{11}\frac{\partial^2 y}{\partial \xi^2 \partial \xi^2} \ .$$

In particular, let the surface S^{x2} be a monitor surface formed by the values of a scalar height function $u(\boldsymbol{s})$, and consequently be represented by the parametrization

$$\boldsymbol{x}(\boldsymbol{s}) = [s^1, s^2, u(\boldsymbol{s})] \ .$$

Then, we obtain

$$g^{xs}_{ij} = \delta^i_j + u_{s^i}u_{s^j} \ , \qquad g^{ij}_{sx} = (-1)^{i+j}\frac{\delta^i_j + u_{s^{3-i}}u_{s^{3-j}}}{1 + (u_{s^1})^2 + (u_{s^2})^2} \ , \qquad i,j = 1,2 \ ,$$

$$g^{xs} = 1 + (u_{s^1})^2 + (u_{s^2})^2 \ , \qquad g^{x\xi} = J^2 g^{xs} \ , \qquad J = \det\{\partial s^i/\partial \xi^j\}$$

without summing over i and j, and correspondingly,

$$\Delta_B[s^l] = \left(\frac{(-1)^{j+l}}{\sqrt{1 + (u_{s^1})^2 + (u_{s^2})^2}}\right)\frac{\partial}{\partial s^j}\left(\frac{\delta^j_l + u_{s^{3-j}}u_{s^{3-l}}}{\sqrt{1 + (u_{s^1})^2 + (u_{s^2})^2}}\right) \ , \qquad j,l = 1,2 \ ,$$

with l fixed. Thus, the system (8.29) has, in this case, the form

$$B_2^{x\xi}[s^l] = (-1)^{k+l} J^2 \sqrt{1 + (u_{s^1})^2 + (u_{s^2})^2}\, \frac{\partial}{\partial s^k} \left(\frac{\delta_l^k + u_{s^{3-k}} u_{s^{3-l}}}{\sqrt{1 + (u_{s^1})^2 + (u_{s^2})^2}} \right) ,$$

$$i, j, k, l = 1, 2 \ , \qquad l \text{ fixed} \ .$$

One more equivalent form of the system (9.29) in this case is as follows:

$$B_2^{x\xi}[s^l] + B_2^{x\xi}[u_1] \frac{\partial u}{\partial s^l} = 0 \ , \qquad l = 1, 2 \ ,$$

where $u_1(\boldsymbol{\xi}) = u[\boldsymbol{s}(\boldsymbol{\xi})]$.

The right-hand sides of the systems (8.28) and (8.29) are defined through the metric elements of the original surface parametrization in the coordinates $s^i, \ i = 1, 2$, and do not depend on the transformation $\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^2 \to S^2$; therefore, they can be computed in advance or at a previous step of an iterative solution procedure.

The values of the numerical solution of a boundary value problem for (8.28) or (8.29) on a reference grid in Ξ^2 define a grid in the parametric domain S^2. The final grid on the surface S^{x2}, generated through the inverted Beltrami system (8.28), is obtained by mapping the above grid with the original parametric transformation $\boldsymbol{x}(\boldsymbol{s})$.

8.5 Interpretations of the Beltramian System

Equations (8.27) are a generalization of the two-dimensional Laplace system on a surface. In this section, we give some interpretations and justifications of the systems (8.27)–(8.29) concerned with grid generation.

8.5.1 Variational Formulation

As was shown in Sect. 7.2, the Laplace system (6.5) for generating grids in domains can be obtained from the minimization of the functional of smoothness

$$I_{\mathrm{S}} = \int\limits_{X^n} \Big(\sum_{i=1}^{n} g^{ii} \Big) \mathrm{d}\boldsymbol{x} \ , \qquad g^{ij} = \sum_{m=1}^{n} \frac{\partial \xi^i}{\partial x^m} \frac{\partial \xi^j}{\partial x^m} \ . \tag{8.30}$$

A similar functional, whose Euler–Lagrange equations are equivalent to (8.27), can be formulated for the Beltrami equations as well. This functional has the form of (8.30), with g^{ii} and X^n replaced by $g^{ii}_{\xi x}$ and S^{x2}, respectively:

$$I_{\mathrm{S}} = \int\limits_{S^{x2}} \Big(\sum_{i=1}^{2} g^{ii}_{\xi r}\Big) \mathrm{d}S^{x2} = \int\limits_{S^{x2}} \Big(\mathrm{tr}\ G_{\xi x}\Big) \mathrm{d}S^{x2} \ . \tag{8.31}$$

In order to write out the Euler–Lagrange equations for this functional, we consider the integration over S^2. We obtain, using (8.23) and the relation $\mathrm{d}S^{x2} = \sqrt{g^{xs}}\mathrm{d}\boldsymbol{s}$,

$$I_{\mathrm{S}} = \int\limits_{S^2} \sqrt{g^{xs}} g^{mk}_{sr} \frac{\partial \xi^i}{\partial s^m} \frac{\partial \xi^i}{\partial s^k} \mathrm{d}\boldsymbol{s} \ , \qquad i,k,m = 1,2 \ . \tag{8.32}$$

The functional (8.32) is formulated on a set of smooth functions $\xi^i(\boldsymbol{s})$, and the terms $\sqrt{g^{xs}} g^{mk}_{sr}$ of its integrand are defined through the metric elements of the original parametrization $\boldsymbol{x}(\boldsymbol{s})$, and therefore do not depend on these functions. So, the Euler–Lagrange equations derived from the functional (8.32) have the form of (7.6), namely,

$$\frac{\partial}{\partial s^j}\Big(\sqrt{g^{xs}} g^{mj}_{sr} \frac{\partial \xi^i}{\partial s^m}\Big) = 0 \ , \qquad i,j,m = 1,2 \ , \tag{8.33}$$

and they are equivalent to (8.27). This variational formulation allows one to generate surface grids through a variational method. For this purpose, the functional (8.31) written out with respect to the integral over the domain Ξ^2,

$$I_{\mathrm{S}} = \int\limits_{\Xi^2} \frac{1}{\sqrt{g^{x\xi}}} (g^{x\xi}_{11} + g^{x\xi}_{22}) \mathrm{d}\boldsymbol{\xi} \ ,$$

is used.

8.5.2 Harmonic-Mapping Interpretation

The integral (8.31), and consequently (8.32), according to the accepted terminology in differential geometry, is the total energy associated with the function $\boldsymbol{\xi}(\boldsymbol{s}) : S^2 \to \Xi^2$ which represents a mapping between the manifold S^{x2} with the metric tensor g^{xs}_{ij} and the computational domain Ξ^2 with the Cartesian coordinates ξ^i. A function which is a critical point of the energy functional is called a harmonic function. It follows from the theory of harmonic functions on manifolds that if there is a diffeomorphism $\boldsymbol{f}(\boldsymbol{s}) : S^2 \to \Xi^2$ and the boundary of the manifold Ξ^2 is convex, while its curvature is nonpositive, then a harmonic function coinciding with $\boldsymbol{f}$ on the boundary of the manifold S^{x2} is also a diffeomorphism between S^2 and Ξ^2. In the case under consideration, the coordinates of the manifold Ξ^2 are Cartesian, and therefore its curvature is nonpositive. So, if the boundary of the computational domain Ξ^2 is convex (e.g. Ξ^2 is a rectangle) and the surface S^{x2} is diffeomorphic to Ξ^2, then the mapping $\boldsymbol{\xi}(\boldsymbol{s})$ that minimizes the functional of smoothness (8.32) is a one-to-one transformation and the grid obtained by the proposed variational method is therefore nondegenerate.

8.5.3 Formulation Through Invariants

We note that, in accordance with (3.31), the trace tr $G_{\xi x}$ of the contravariant 2 × 2 metric tensor can be expressed through the invariants $I_1,\ I_2$ of the orthogonal transforms of the covariant metric tensor $G^{x\xi}$, namely,

$$\mathrm{tr}\ G_{\xi x} = \frac{I_1}{I_2}.$$

Therefore, the functional of smoothness (8.31) can also be expressed through these invariants:

$$I_{\mathrm{S}} = \int\limits_{S^{x2}} \Big(\frac{I_1}{I_2}\Big) \mathrm{d}S^{x2}\ . \qquad (8.34)$$

The invariant I_2 is the Jacobian of the matrix $G^{x\xi}$ and it equals the area squared of the parallelogram formed by the tangent vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$. The invariant I_1 is $q_{11}^{x\xi} + q_{22}^{x\xi}$ and means the sum of the lengths squared of the sides of the parallelogram. So,

$$\frac{I_1}{I_2} = \frac{g_{11}^{x\xi} + g_{22}^{x\xi}}{g^{x\xi}}\ . \qquad (8.35)$$

It is obvious that

$$g^{x\xi} = g_{11}^{x\xi}(d_2)^2, \qquad g^{x\xi} = g_{22}^{x\xi}(d_1)^2\ ,$$

where $d_i,\ i = 1, 2$, is the distance between the vertex of the vector $\boldsymbol{x}_{\xi^i}$ and the other vector $\boldsymbol{x}_{\xi^j},\ j = 3 - i$. Therefore, from (8.35),

$$\frac{I_1}{I_2} = \frac{1}{(d_1)^2} + \frac{1}{(d_2)^2}\ . \qquad (8.36)$$

The quantity d_i is connected with the distance l_i between the grid lines $\xi^i = c$ and $\xi^i = c + h$ by the relation

$$l_i = d_i h + O(h)^2\ .$$

So, from (8.36),

$$\frac{I_1}{I_2} = \sum_{i=1}^{2} (h/l_i)^2 + O(h) \; .$$

The quantity $(h/l_i)^2$ increases as the grid nodes cluster in the direction normal to the coordinate $\xi^i = \text{const}$, and therefore it can be considered as some measure of the grid concentration in this direction and, consequently, the functional (8.34) defines an integral measure of the surface grid clustering in all directions. Hence, the problem of the minimization of the smoothness functional (8.31) can be interpreted as a problem of finding a grid with uniform clustering on the surface S^{x2}. So, the Beltrami equations (8.27) tend to generate a uniform grid on the surface in the same manner as the system of the Laplace equations does in a domain.

8.5.4 Formulation Through the Surface Christoffel Symbols

Equivalent forms of the surface system of equations (8.27) and (8.28) can be obtained by a consideration of the formulas of Gauss. These formulas represent the derivatives of the tangential vectors $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$ through the basis $(\boldsymbol{x}_{\xi^1},\ \boldsymbol{x}_{\xi^2},\ \boldsymbol{n})$, where $\boldsymbol{n}$ is a unit normal to the surface.

Surface Gauss Equations

In accordance with (2.6), we can write

$$\boldsymbol{x}_{\xi^i\xi^j} = a^{lm}(\boldsymbol{x}_{\xi^i\xi^j} \cdot \boldsymbol{a}_l)\boldsymbol{a}_m \; , \qquad i, j = 1, 2 \; , \qquad l, m = 1, 2, 3 \; , \tag{8.37}$$

where $\boldsymbol{a}_1 = \boldsymbol{x}_{\xi^1}$, $\boldsymbol{a}_2 = \boldsymbol{x}_{\xi^2}$, $\boldsymbol{a}_3 = \boldsymbol{n}$, and $\{a^{lm}\}$ is the matrix inverse to $\{a_{kp}\} = \{\boldsymbol{a}_k \cdot \boldsymbol{a}_p\}$. We readily obtain, for the elements of the matrix $\{a_{ij}\}$,

$$a_{ij} = g^{x\xi}_{ij} \; , \qquad a_{i3} = a_{3i} = 0 \; , \qquad i, j = 1, 2 \; , \qquad a_{33} = 1 \; .$$

Therefore,

$$a^{ij} = g^{ij}_{\xi x} \; , \qquad a^{i3} = a^{3i} = 0 \; , \qquad a^{33} = 1 \; , \qquad i, j = 1, 2 \; . \tag{8.38}$$

Thus, (8.37) results in

$$\boldsymbol{x}_{\xi^i\xi^j} = g^{km}_{\xi x}(\boldsymbol{x}_{\xi^i\xi^j} \cdot \boldsymbol{r}_{\xi m})\boldsymbol{x}_{\xi k} + (\boldsymbol{x}_{\xi^i\xi^j} \cdot \boldsymbol{n})\boldsymbol{n} \; , \qquad i, j, k, m = 1, 2 \; . \tag{8.39}$$

The quantities $\boldsymbol{x}_{\xi^i\xi^j} \cdot \boldsymbol{x}_{\xi m}$ in (8.39) are the surface Christoffel symbols of the first kind, denoted by $[ij, m]$. These quantities coincide with the space Christoffel symbols (2.40) for the indices $i, j, k = 1, 2$, and, in the same manner as (2.45), they are subject to the relations

$$[ij,k] = \frac{1}{2}\Big(\frac{\partial g^{x\xi}_{ik}}{\partial \xi^j} + \frac{\partial g^{x\xi}_{jk}}{\partial \xi^i} - \frac{\partial g^{x\xi}_{ij}}{\partial \xi^k}\Big) , \qquad i,j,k = 1,2 .$$

The surface Christoffel symbols of the second kind, denoted by Υ^k_{ij}, are defined in analogy with (2.43) by

$$\Upsilon^k_{ij} = g^{km}_{\xi x}[ij,m] = g^{km}_{\xi x}\boldsymbol{x}_{\xi^i\xi^j}\cdot\boldsymbol{x}_{\xi^m} , \qquad i,j,k,m = 1,2 . \qquad (8.40)$$

The quantities $\boldsymbol{x}_{\xi^i\xi^j}\cdot\boldsymbol{n}$ in (8.39), designated as b_{ij}, i.e.

$$b_{ij} = \boldsymbol{x}_{\xi^i\xi^j}\cdot\boldsymbol{n} , \qquad i,j = 1,2, \qquad (8.41)$$

are called the coefficients of the second fundamental form. With these designations, we have for (8.39)

$$\boldsymbol{x}_{\xi^i\xi^j} = \Upsilon^k_{ij}\boldsymbol{x}_{\xi^k} + b_{ij}\boldsymbol{n} . \qquad (8.42)$$

The relations (8.42) are referred to as the surface Gauss identities.

Weingarten Equation

Some other important relations are concerned with the first derivatives of the unit normal $\boldsymbol{n}$. Since $\boldsymbol{n}_{\xi^i}\cdot\boldsymbol{n} = 0,\ i = 1,2$, the vectors $\boldsymbol{n}_{\xi^i},\ i = 1,2$, are orthogonal to $\boldsymbol{n}$, and hence can be expanded in the tangential vectors $\boldsymbol{x}_{\xi^i},\ i = 1,2$. Taking advantage of (2.6) and (8.38), we obtain

$$\boldsymbol{n}_{\xi^i} = g^{lm}_{\xi x}(\boldsymbol{n}_{\xi^i}\cdot\boldsymbol{x}_{\xi^l})\boldsymbol{x}_{\xi^m} , \qquad i,l,m = 1,2 . \qquad (8.43)$$

Since

$$\boldsymbol{n}_{\xi^i}\cdot\boldsymbol{x}_{\xi^l} = (\boldsymbol{n}\cdot\boldsymbol{x}_{\xi^l})_{\xi^i} - \boldsymbol{n}\cdot\boldsymbol{r}_{\xi^l\xi^i} = -b_{li} ,$$

(8.43) has the form

$$\boldsymbol{n}_{\xi^i} = -g^{lm}_{\xi x}b_{li}\boldsymbol{x}_{\xi^m} , \qquad i,l,m = 1,2 . \qquad (8.44)$$

The relations (8.44) are called the Weingarten equations.

Mean Curvature

The value of the Beltrami operator over the position vector is connected with the mean curvature. The quantity known as the mean surface curvature is expressed in accordance with (3.23) by

$$K_{\mathrm{m}} = \frac{1}{2}g^{ij}_{\xi x}b_{ij} , \qquad i,j = 1,2 . \qquad (8.45)$$

The mean surface curvature does not depend on the surface parametrization. In fact, if $s^i,\ i = 1,2$, are new surface coordinates, then

$$\boldsymbol{x}_{\xi^i\xi^j}=\frac{\partial s^k}{\partial \xi^i}\frac{\partial s^m}{\partial \xi^j}\boldsymbol{x}_{s^k s^m}+\frac{\partial^2 s^m}{\partial \xi^i \partial \xi^j}\boldsymbol{x}_{s^m}\ ,$$

and hence

$$b_{ij}=\frac{\partial s^k}{\partial \xi^i}\frac{\partial s^m}{\partial \xi^j}\boldsymbol{x}_{s^k s^m}\cdot \boldsymbol{n}\ ,\qquad i,j,k,m=1,2\ .$$

As the contravariant tensor $\{g^{im}_{\xi x}\}$ is transformed into the coordinates s^i in accordance with the relations (8.23), we now readily see that the mean curvature is an invariant of the surface parametrizations. Therefore, it can be defined from the original parametrization $\boldsymbol{x}(\boldsymbol{s})$.

Recall that in geometry, the first and second fundamental forms in the coordinates $\xi^1,\ \xi^2$ are, respectively,

$$\mathrm{I}=g^{x\xi}_{ij}\mathrm{d}\xi^i\mathrm{d}\xi^j\ ,\qquad \mathrm{II}=b_{ij}\mathrm{d}\xi^i\mathrm{d}\xi^j\ ,\qquad i,j=1,2\ .$$

The first fundamental form, derived from the surface metric, describes the inner geometry of the surface, while the second form deals with the outer geometry, since it is derived from the specification of the surface immersion into R^3. Obviously, the second fundamental form gives an expression for the local departure of the surface from its tangent plane.

Relation Between Beltrami's Equation and Christoffel Symbols

We assume that the basic surface vectors $(\boldsymbol{x}_{\xi^1},\ \boldsymbol{x}_{\xi^2},\ \boldsymbol{n})$ compose a right-handed triad. This leads to

$$\boldsymbol{n}=\frac{1}{\sqrt{g^{x\xi}}}(\boldsymbol{x}_{\xi^1}\times \boldsymbol{r}_{\xi^2})\ . \tag{8.46}$$

Note that the tangent vectors $\boldsymbol{x}_{\xi^i},\ i=1,2$, are neither normalized nor orthogonal to each other, while $\boldsymbol{n}$ is normalized and orthogonal to both $\boldsymbol{x}_{\xi^1}$ and $\boldsymbol{x}_{\xi^2}$. Using the general vector identity

$$\boldsymbol{u}\times(\boldsymbol{v}\times\boldsymbol{w})=(\boldsymbol{u}\cdot\boldsymbol{w})\boldsymbol{v}-(\boldsymbol{u}\cdot\boldsymbol{v})\boldsymbol{w}\ ,$$

we obtain, taking into account (8.46),

$$\boldsymbol{x}_{\xi^i}\times\boldsymbol{n}=\frac{1}{\sqrt{g^{x\xi}}}(g^{x\xi}_{i2}\boldsymbol{x}_{\xi^1}-g^{x\xi}_{i1}\boldsymbol{x}_{\xi^2})\ .$$

With (8.22), this results in the relation

$$(-1)^{i+1}\boldsymbol{x}_{\xi^{3-i}}\times\boldsymbol{n}=\sqrt{g^{x\xi}}g^{ij}_{\xi x}\boldsymbol{x}_{\xi^j}\ ,\qquad i,j=1,2\ , \tag{8.47}$$

with i fixed. The application of (8.47) to the Beltramian operator yields

$$\begin{aligned}\Delta_B[\boldsymbol{x}] &= \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial \xi^j}(\sqrt{g^{x\xi}}g^{ij}_{\xi x}\boldsymbol{x}_{\xi^i}) \\ &= (-1)^{j+1}\frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial \xi^j}(\boldsymbol{x}_{\xi^{3-j}} \times \boldsymbol{n}) \\ &= (-1)^{j+1}\frac{1}{\sqrt{g^{x\xi}}}(\boldsymbol{x}_{\xi^{3-j}} \times \boldsymbol{n}_{\xi^j}) \ , \qquad i, j = 1, 2 \ . \end{aligned} \tag{8.48}$$

Taking advantage of the Weingarten equations (8.44) and (8.46), we have

$$\begin{aligned}(-1)^{j+1}\frac{1}{\sqrt{g^{x\xi}}}(\boldsymbol{x}_{\xi^{3-j}} \times \boldsymbol{n}_{\xi^j}) &= (-1)^{j}\frac{1}{\sqrt{g^{x\xi}}}g^{lm}_{\xi x}b_{lj}(\boldsymbol{x}_{\xi^{3-j}} \times \boldsymbol{x}_{\xi^m}) \\ &= g^{lj}_{\xi x}b_{lj}\boldsymbol{n} \ , \qquad j, l, m = 1, 2 \ . \end{aligned}$$

Thus, (8.48) also has the form

$$\Delta_B[\boldsymbol{x}] = b_{ij}g^{ij}_{\xi x}\boldsymbol{n} = 2K_{\mathrm{m}}\boldsymbol{n} \ , \qquad i, j = 1, 2 \ , \tag{8.49}$$

where the quantity K_{m}, defined by (8.45), is the mean curvature of the surface. Equation (8.49) means that the surface position vector $\boldsymbol{x}(\boldsymbol{\xi})$ is transformed by the Beltrami operator to the vector $\Delta_B[\boldsymbol{x}]$, which is orthogonal to the surface. Now, expanding the differentiation in $\Delta_B[\boldsymbol{x}]$ and using the expression for $\Delta_B[\xi^i]$, we obtain one more form of $\Delta_B[\boldsymbol{x}]$:

$$\Delta_B[\boldsymbol{x}] = g^{ij}_{\xi x}\boldsymbol{x}_{\xi^i\xi^j} + (\Delta_B[\xi^i])\boldsymbol{x}_{\xi^i} \ , \qquad i, j = 1, 2 \ . \tag{8.50}$$

Equating the right-hand sides of (8.49) and (8.50), we have the identity

$$g^{ij}_{\xi x}\boldsymbol{x}_{\xi^i\xi^j} + (\Delta_B[\xi^i])\boldsymbol{r}_{\xi^i} = 2K_{\mathrm{m}}\boldsymbol{n} \ , \qquad i, j = 1, 2 \ . \tag{8.51}$$

Thus, if the surface coordinate system ξ^1, ξ^2 is obtained by the solution of (8.27), then, from (8.51), we obtain

$$g^{ij}_{\xi x}\boldsymbol{x}_{\xi^i\xi^j} = 2K_{\mathrm{m}}\boldsymbol{n} \ , \qquad i, j = 1, 2 \ .$$

We obtain one more identity by multiplying (8.42) by $g^{ij}_{\xi r}$:

$$g^{ij}_{\xi x}\boldsymbol{x}_{\xi^i\xi^j} = g^{ij}_{\xi x}\Upsilon^k_{ij}\boldsymbol{x}_{\xi^k} + 2K_{\mathrm{m}}\boldsymbol{n} \ , \qquad i, j = 1, 2 \ .$$

Comparing this identity with (8.51), we have the identity

$$\Delta_B[\xi^i] = -g^{kl}_{\xi x}\Upsilon^i_{kl} \ , \qquad i, k, l = 1, 2 \ , \tag{8.52}$$

i.e. the value obtained by applying the Beltrami operator to the function $\xi^i(\boldsymbol{s})$ is defined through the surface Christoffel symbols and the surface metric elements. Note that the surface identity (8.52) is a reformulation of the identity (6.31), valid for domains.

Taking advantage of (8.52), the Beltrami system (8.27) can be written in the following equivalent form as

$$g^{kl}_{\xi x}\Upsilon^i_{kl} = 0\,, \qquad i,k,l = 1,2\,, \tag{8.53}$$

or, using (8.40), as

$$g^{kl}_{\xi x}g^{ij}_{\xi x}[kl,j] \equiv g^{kl}_{\xi x}g^{ij}_{\xi x}\boldsymbol{x}_{\xi^k\xi^l}\cdot\boldsymbol{x}_{\xi^j} = 0\,, \qquad i,j,k,l = 1,2\,. \tag{8.54}$$

Multiplying (8.54) by $g^{x\xi}_{im}$ and summing over i yields

$$g^{kl}_{\xi x}\boldsymbol{x}_{\xi^k\xi^l}\cdot\boldsymbol{x}_{\xi^m} = 0\,, \qquad k,l,m = 1,2\,. \tag{8.55}$$

Recall that the first and second Christoffel symbols in (8.53) and (8.54) are defined in terms of the coordinates ξ^i.

In the particular case in which the surface S^{x2} is a monitor surface defined by the values of a height function $u(\boldsymbol{s})$ over the domain S^2, i.e.

$$\boldsymbol{x}(\boldsymbol{s}) = [s^1, s^2, u(\boldsymbol{s})]\,,$$

multiplication of (8.55) by $g^{x\xi}(\partial\xi^m/\partial s^i)$ and summation over m produces the following system of inverted surface Beltrami equations:

$$B_2^{x\xi}[s^i] + B_2^{x\xi}[u]\frac{\partial u}{\partial s^i} = 0\,, \qquad i = 1,2\,, \tag{8.56}$$

where

$$B_2^{x\xi}[y] \equiv g^{x\xi}g^{ij}_{\xi x}\frac{\partial^2 y}{\partial\xi^i\partial\xi^j} \equiv g^{x\xi}_{22}\frac{\partial^2 y}{\partial\xi^1\partial\xi^1} - 2g^{x\xi}_{12}\frac{\partial^2 y}{\partial\xi^1\partial\xi^2} + g^{x\xi}_{11}\frac{\partial^2 y}{\partial\xi^2\partial\xi^2}\,,\ i,j = 1,2\,.$$

Another form of the inverted Beltrami system can be obtained from the elliptic system (8.28) for generating surface grids. Namely, applying (8.28)–(8.52) with the identification $\xi^i = s^i,\ i = 1,2$, we obtain

$$B_2^{x\xi}[s^l] + g^{x\xi}g^{km}_{sx}\Upsilon^l_{km} = 0\,, \qquad i,j,k,l,m = 1,2\,, \tag{8.57}$$

where the Υ^l_{km} are the second surface Christoffel symbols in the coordinates s^i. An equivalent form of (8.56) is also obtained by utilizing (8.40) for $\xi^i = s^i,\ i = 1,2$:

$$B_2^{x\xi}[s^l] + g^{x\xi} g_{sx}^{km} g_{sx}^{lp} [km, p] = 0\ , \qquad i, j, k, l, m, p = 1, 2\ ,$$

where $[km, p] = \boldsymbol{x}_{s^k s^m} \cdot \boldsymbol{x}_{s^p}$.

In the particular case in which the surface S^{x2} is a monitor surface defined by the values of a height function $u(\boldsymbol{s})$ over the domain S^2, i.e.

$$\boldsymbol{x}(\boldsymbol{s}) = [s^1, s^2, u(\boldsymbol{s})]\ ,$$

the Beltrami equations (8.57) have the form

$$g_{\xi x}^{ij} \frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} + g_{sx}^{km} g_{sx}^{lp} \frac{\partial^2 u}{\partial s^k \partial s^m} \frac{\partial u}{\partial s^p} = 0\ , \qquad i, j, k, l, m, p = 1, 2\ ,$$

where

$$g_{sx}^{km} = (-1)^{k+m} \frac{\delta_m^k + \dfrac{\partial u}{\partial s^{3-k}} \dfrac{\partial u}{\partial s^{3-m}}}{1 + \left(\dfrac{\partial u}{\partial s^1}\right)^2 + \left(\dfrac{\partial u}{\partial s^2}\right)^2}\ , \qquad k, m = 1, 2\ ;$$

here, the indices k and m are fixed.

8.6 Control of Surface Grids

8.6.1 Control Functions

One approach to controlling the generation of a surface grid is to add forcing terms to the Beltrami operator in analogy with the Poisson system, i.e. to extend the system (8.27) to the following one:

$$\Delta_B[\xi^i] = P^i, \qquad i = 1, 2\ . \tag{8.58}$$

The system inverse to (8.58) with interchanged dependent and independent variables is obtained through the same procedure as was applied to produce (8.28) and has the form

$$g_{\xi x}^{ij} \frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} + \frac{\partial s^l}{\partial \xi^i} P^i = \Delta_B[s^l], \qquad i, j, l = 1, 2\ , \tag{8.59}$$

or with the application of the operator $B_2^{x\xi}$ from (8.29),

$$B_2^{x\xi}[s^l] + g^{x\xi} \frac{\partial s^l}{\partial \xi^i} P^i = g^{x\xi} \Delta_B[s^l], \qquad i, l = 1, 2\ .$$

The control functions $P^i,\ i=1,2$, can be chosen in the same manner as was done in Sect. 6.2. For example, for the generation of grids that are nearly orthogonal at the boundaries, we may apply the approach of Sect. 6.2, determining the values of P^i on the boundary curves and then propagating them into the interior.

For this purpose, we have to find the values of $\Delta_B[\xi^i]$ on the boundary curves.

8.6.2 Monitor Approach

Another approach to controlling the generation of a grid on a surface relies on the concept of the monitor surface defined by the values of some vector function $\boldsymbol{f} = [f^1(\boldsymbol{s}), \dots, f^k(\boldsymbol{s})]$ over the surface S^{x2}. The monitor surface, denoted by S^{r2}, lies in R^{3+k} and can be represented through the parametric coordinates $s^1,\ s^2$ by the equation

$$\boldsymbol{r}(\boldsymbol{s}) = [x^1(\boldsymbol{s}), x^2(\boldsymbol{s}), x^3(\boldsymbol{s}), f^1(\boldsymbol{s}), \dots, f^k(\boldsymbol{s})]\ , \qquad \boldsymbol{s} = (s^1, s^2).$$

The covariant (contravariant) metric elements of the monitor surface S^{r2} in the coordinates v^1, v^2 are denoted as g^{rv}_{ij} (g^{ij}_{vr}). In this approach, the Beltrami equations on the surface S^{r2} are used to generate the grid on the parametric domain S^2. Mapping this grid on S^{x2} with the transformation $\boldsymbol{x}(\boldsymbol{s}) : S^2 \to S^{x2}$ produces a grid on the surface S^{x2} dependent on the control function $\boldsymbol{f}$.

The equations for generating the grid on the parametric domain S^2 are obtained in the same manner as the Eqs. (8.27) and have, with respect to the dependent coordinates s^i and the independent coordinates ξ^i, a form similar to (8.28):

$$g^{ij}_{\xi r} \frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} = \Delta_B[s^l], \qquad i, j, l = 1, 2\ , \tag{8.60}$$

where $g^{ij}_{\xi r},\ i, j = 1, 2$, are the elements of the contravariant tensor of the surface S^{r2} in the coordinates $\xi^1,\ \xi^2$, defined through the elements of the covariant tensor

$$g^{r\xi}_{ij} = \frac{\partial \boldsymbol{r}}{\partial \xi^i} \cdot \frac{\partial \boldsymbol{r}}{\partial \xi^j} = g^{x\xi}_{ij} + \frac{\partial \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^i} \cdot \frac{\partial \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^j}\ , \qquad i, j = 1, 2\ ,$$

by the formula

$$g^{ij}_{\xi r} = (-1)^{i+j} g^{r\xi}_{3-i3-j} / g^{r\xi}\ , \qquad g^{r\xi} = \det\{g^{r\xi}_{ij}\}\ ,$$

with fixed i and j. The Beltrami operator in the right-hand part of (8.60) is expressed as

$$\Delta_B = \frac{1}{\sqrt{g^{rs}}} \frac{\partial}{\partial s^j} \left(\sqrt{g^{rs}} g^{ij}_{sr} \frac{\partial}{\partial s^i} \right)\ , \quad i, j = 1, 2$$

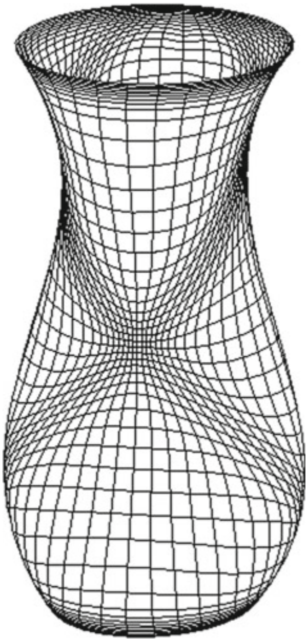


Fig. 8.3 Example of a quadrangular adaptive surface grid

through the metric elements in the coordinates $s^1,\ s^2$ of the monitor surface S^{r2}.

The same considerations as in Sect. 8.8 show that the grid on the monitor surface S^{r2} obtained through (8.60) tends to become uniform, and its projection onto the surface S^{x2} produces a grid with a node concentration in the regions of large variations of the control function $\boldsymbol{f}$. One example of such an adaptive grid is illustrated by Fig. 8.3.

8.6.3 Control Through Variational Methods

The variational approaches of Chap. 7 can be successfully applied to surface grid generation. What is needed is the formulation of the corresponding surface-grid quality measures. Chapter 3 gives a detailed description of domain-grid characteristics in terms of the space metric elements and the space Christoffel symbols. The quantities expressing the local grid properties are readily reformulated for grids on surfaces, using for this purpose the surface metric elements and surface Christoffel symbols. The integration of these quantities provides functionals which reflect the integral measures of the respective grid qualities. The grid on the surface is then generated by using the standard scheme of Chap. 7, by optimizing the grid characteristics with the minimization of a combination of functionals to obtain a grid with certain desired properties. In this subsection, we describe some surface functionals which represent popular geometric quality measures of surface grids. For generality, the functionals are formulated with weights which are determined by derivatives of the solution quantities or by measures of the quality features of the surface.

Functionals Dependent on Invariants

In analogy with Chap. 7 there are three basic surface functionals determined by the invariants I_1 and I_2 of the metric tensor $\{g_{ij}^{x\xi}\}$ in the coordinates ξ^i. The first is the length-weighted functional

$$I_{\mathrm{lw}} = \int_{\Xi^2} w I_1 \mathrm{d}\boldsymbol{\xi} = \int_{\Xi^2} w(g_{11}^{x\xi} + g_{22}^{x\xi})\mathrm{d}\boldsymbol{\xi} \ . \tag{8.61}$$

The second is the area-weighted functional

$$I_{\mathrm{a}} = \int_{S^2} w(I_2)^m \mathrm{d}\boldsymbol{s} = \int_{S^2} w(g^{x\xi})^m \mathrm{d}\boldsymbol{s} \ . \tag{8.62}$$

And the third is the density-weighted functional

$$I_{\mathrm{d}} = \int_{\Xi^2} w(I_1/\sqrt{I_2})\mathrm{d}\boldsymbol{\xi} = \int_{S^2} w\sqrt{g^{xs}}(g_{\xi x}^{11} + g_{\xi x}^{22})\mathrm{d}\boldsymbol{s} \ . \tag{8.63}$$

The length-weighted functional controls the lengths of the cell edges, while the functional (8.62) regulates the areas of the surface grid cells. The control of the grid density may be carried out through the functional (8.63). Note that the functional (8.63) with $w = 1$ is, in fact, the smoothness functional (8.31).

Weight Skewness and Orthogonality Functionals

Analogously, functionals can be formulated which measure the surface grid skewness and deviation from orthogonality. Thus, in accordance with (7.27), the functional for the weighted surface grid skewness may be either of the following forms:

$$\begin{aligned} I_{\mathrm{sk},1} &= \int_{\Xi^2} w \frac{(g_{12}^{x\xi})^2}{g_{11}^{x\xi} g_{22}^{x\xi}} \mathrm{d}\boldsymbol{\xi} \ , \\ I_{\mathrm{sk},2} &= \int_{\Xi^2} w \frac{(g_{\xi x}^{12})^2}{g_{\xi x}^{11} g_{\xi x}^{22}} \mathrm{d}\boldsymbol{\xi} \ . \end{aligned} \tag{8.64}$$

Similarly to (7.29) and (7.46), the orthogonality functionals can be expressed as follows:

$$\begin{aligned} I_{\mathrm{O},1} &= \int_{\Xi^2} w\Big(\frac{g_{11}^{x\xi} g_{22}^{x\xi}}{g^{x\xi}}\Big) \mathrm{d}\boldsymbol{\xi} \ , \\ I_{\mathrm{O},2} &= \int_{\Xi^2} w(g_{12}^{x\xi})^2 \mathrm{d}\boldsymbol{\xi} \ , \\ I_{\mathrm{O},3} &= \int_{S^{x2}} w(g^{12})^2 \mathrm{d}S^{x2} = \int_{S^2} w\sqrt{g^{xs}}(g^{12})^2 \mathrm{d}\boldsymbol{s} \ . \end{aligned} \tag{8.65}$$

Weight Functions

Commonly, the weight functions for surface grid generation are formulated through the derivatives of the solution quantities and through the features of the surface. Surface grid generation also requires adjustments of the measures of the grid cells to the curvature of the surface. This can be carried out by applying the curvature measures as weight functions.

A surface S^{x2} has two curvatures: the mean curvature

$$K_{\mathrm{m}} = \frac{1}{2} g_{sx}^{ij} b_{ij} \ , \qquad i, j = 1, 2 \ , \tag{8.66}$$

and the Gaussian curvature

$$K_{\mathrm{G}} = \frac{1}{g^{xs}}(b_{11}b_{22} - (b_{12})^2) \ , \tag{8.67}$$

where

$$b_{ij} = \frac{1}{\sqrt{g^{xs}}} \boldsymbol{x}_{s^i s^j} \cdot (\boldsymbol{x}_{s^1} \times \boldsymbol{x}_{s^2}) \ , \qquad i, j = 1, 2 \ .$$

The corresponding weights can be formulated in the form

$$w = (K_{\mathrm{m}})^2 \qquad \text{or} \qquad w = (K_{\mathrm{G}})^2 \ . \tag{8.68}$$

An expression for the quantity $b_{11}b_{22} - (b_{12})^2$ can be obtained through derivatives of the elements of the metric tensor and the coefficients of the second fundamental form of the surface. This is accomplished by using the expansion (8.39) with the substitution of ξ^i for s^i, which results in the following relation:

$$\begin{aligned} &\boldsymbol{x}_{s^1 s^1} \cdot \boldsymbol{x}_{s^2 s^2} - \boldsymbol{x}_{s^1 s^2} \cdot \boldsymbol{x}_{s^1 s^2} = g_{ij}^{xs}(\Upsilon_{11}^i \Upsilon_{22}^j - \Upsilon_{12}^i \Upsilon_{12}^j) + b_{11}b_{22} - (b_{12})^2 \ , \\ &i, j = 1, 2 \ . \end{aligned} \tag{8.69}$$

The left-hand part of (8.69) equals

$$\boldsymbol{x}_{s^1 s^1} \cdot \boldsymbol{x}_{s^2 s^2} - \boldsymbol{x}_{s^1 s^2} \cdot \boldsymbol{x}_{s^1 s^2} = \frac{\partial}{\partial s^1}(\boldsymbol{x}_{s^2 s^2} \cdot \boldsymbol{x}_{s^1}) - \frac{\partial}{\partial s^2}(\boldsymbol{x}_{s^1 s^2} \cdot \boldsymbol{x}_{s^1}) \ .$$

Since

$$\begin{aligned}\boldsymbol{x}_{s^2s^2}\cdot\boldsymbol{x}_{s^1} &= [22,1] = \frac{1}{2}\Big(2\frac{\partial g^{xs}_{12}}{\partial s^2} - \frac{\partial g^{xs}_{22}}{\partial s^1}\Big)\ ,\\ \boldsymbol{x}_{s^1s^2}\cdot\boldsymbol{x}_{s^1} &= [12,1] = \frac{1}{2}\frac{\partial g^{xs}_{11}}{\partial s^2}\ ,\end{aligned}$$

we obtain, from (8.69),

$$\boldsymbol{x}_{s^1s^1}\cdot\boldsymbol{x}_{s^2s^2} - \boldsymbol{x}_{s^1s^2}\cdot\boldsymbol{x}_{s^1s^2} = \frac{1}{2}\Big(2\frac{\partial^2 g^{xs}_{12}}{\partial s^1\partial s^2} - \frac{\partial^2 g^{xs}_{11}}{\partial s^2\partial s^2} - \frac{\partial^2 g^{xs}_{22}}{\partial s^1\partial s^1}\Big)\ .$$

Therefore, (8.67) results in

$$K_{\mathrm{G}} = \frac{1}{g^{xs}}\Big[\frac{1}{2}\Big(2\frac{\partial^2 g^{xs}_{12}}{\partial s^1\partial s^2} - \frac{\partial^2 g^{xs}_{11}}{\partial s^2\partial s^2} - \frac{\partial^2 g^{xs}_{22}}{\partial s^1\partial s^1}\Big) - g^{xs}_{ij}(\Upsilon^i_{11}\Upsilon^j_{22} - \Upsilon^i_{12}\Upsilon^j_{12})\Big]\ ,$$
$$i,j=1,2\ . \qquad (8.70)$$

8.6.4 Orthogonal Grid Generation

Orthogonal elliptic coordinate systems on a surface are formulated in the same standard way as domains. Namely, an appropriate identity is chosen, which is then transformed to the orthogonal system by substituting zero for the nondiagonal metric elements.

For example, using the Beltrami operator

$$\Delta_B\xi^i = \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial\xi^j}\Big(\sqrt{g^{x\xi}}g^{ij}_{\xi r}\Big)\ , \qquad i,j=1,2\ ,$$

we obtain

$$\Delta_B\xi^i\frac{\partial x^l}{\partial\xi^i} = -g^{ij}_{\xi x}\frac{\partial^2 s^l}{\partial\xi^i\partial\xi^j} + \Delta_B s^l\ , \qquad i,j,l=1,2\ .$$

Thus, we have the identity

$$\frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial\xi^j}\Big(\sqrt{g^{x\xi}}g^{ij}_{\xi r}\Big)\frac{\partial s^l}{\partial\xi^i} + g^{ij}_{\xi r}\frac{\partial^2 s^l}{\partial\xi^i\partial\xi^j} = \Delta_B s^l\ ,$$

which implies

$$\frac{1}{\sqrt{g^{x\xi}}}\Big[\frac{\partial}{\partial\xi^j}\Big(\sqrt{g^{x\xi}}g^{ij}_{\xi r}\frac{\partial s^l}{\partial\xi^i}\Big)\Big] = \Delta_B s^l\ , \qquad i,j,l=1,2\ .$$

Substituting in these equations the condition of orthogonality $g^{12}_{\xi r} = 0$ for the term $g^{12}_{\xi x}$, we obtain an elliptic system for generating orthogonal or nearly orthogonal grids on the surface S^{x2} :

$$\frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial \xi^j}\left(\sqrt{g^{x\xi}} g^{jj}_{\xi x}\frac{\partial s^l}{\partial \xi^j}\right) = \Delta_B s^l \ , \tag{8.71}$$

where

$$g^{x\xi} = g^{x\xi}_{11}\, g^{x\xi}_{22} \ , \qquad g^{11}_{\xi x} = 1/g^{x\xi}_{11} \ , \qquad g^{22}_{\xi r} = 1/g^{x\xi}_{22} \ .$$

Thus, we have, from (8.71),

$$\frac{1}{\sqrt{g^{x\xi}_{11}\, g^{x\xi}_{22}}}\left[\frac{\partial}{\partial \xi^1}\left(\sqrt{g^{x\xi}_{22}/g^{x\xi}_{11}}\frac{\partial s^l}{\partial \xi^1}\right) + \frac{\partial}{\partial \xi^2}\left(\sqrt{g^{x\xi}_{11}/g^{x\xi}_{22}}\frac{\partial s^l}{\partial \xi^2}\right)\right] = \Delta_B s^l \ ,$$

$$l = 1, 2 \ . \tag{8.72}$$

The system (8.72) is, in fact, a generalization of the planar system (6.88).

8.7 Hyperbolic Method

Although the elliptic methods described above can provide satisfactory grids for most applications, there are situations when it is more convenient to use hyperbolic methods, in particular, when the four boundaries of the surface grid need not be specified and constructed prior to the generation of the interior grid. Such a situation, for example, occurs in the generation of grids for intersecting geometric components where the surface grid is generated hyperbolically by marching away from the intersection curve. For the overset grid approach, it is frequently the case that the location of some boundary components is not restricted. Also, domain decomposition is simplified under the overset grid approach, and the grid generation time with a hyperbolic technique is relatively fast, since only one boundary needs to be specified.

The hyperbolic method of surface grid generation involves marching a grid away from an initial boundary curve by a user-specified distance. This is achieved through the numerical solution of a set of hyperbolic partial differential equations. Desirable grid attributes such as grid point clustering and orthogonality control are naturally achieved. The grid points obtained are projected onto the underlying surface after each marching step.

8.7.1 Hyperbolic Governing Equations

Let ξ^1 and ξ^2 be the coordinates of the surface, where ξ^1 runs along some initial boundary curve and ξ^2 is the marching direction away from the curve on the surface. Also, let $\boldsymbol{n} = (n^1, n^2, n^3)$ be the local unit normal, which is assumed to be computable anywhere on the surface. The constraints of orthogonality of the families of grid lines and specified mesh cell area are

$$\begin{aligned} &\boldsymbol{x}_{\xi^1} \cdot \boldsymbol{x}_{\xi^2} = 0 \ , \\ &\boldsymbol{n} \cdot (\boldsymbol{x}_{\xi^1} \times \boldsymbol{x}_{\xi^2}) = \triangle S \ , \end{aligned} \tag{8.73}$$

where $\triangle S$ is a user-specified surface mesh cell area. A third equation, needed to close the system, is provided by requiring that the marching direction of the grid be orthogonal to the surface normal at the local grid point, i.e. the marching direction is along the tangent plane of the underlying surface at this point. This gives

$$\boldsymbol{n} \cdot \boldsymbol{x}_{\xi^l} = 0 \ . \tag{8.74}$$

A unit vector in the marching direction ξ^2 can be obtained from the cross product of $\boldsymbol{n}$ with a unit vector in the initial curve direction ξ^1.

Equations (8.73) and (8.74), in the usual variables x, y, z and ξ, η, can be written as

$$\begin{aligned} &x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta = 0 \ , \\ &n^1(y_\xi z_\eta - z_\xi y_\eta) + n^2(z_\xi x_\eta - x_\xi z_\eta) + n^3(x_\xi y_\eta - y_\xi x_\eta) = \triangle S \ , \\ &n^1 x_\eta + n^2 y_\eta + n^3 z_\eta = 0 \ . \end{aligned} \tag{8.75}$$

These equations form a hyperbolic system for marching in the η direction.

Equations (8.75) are written in terms of the physical coordinates instead of the parametric coordinates. In order to preserve the specified surface shape, the physical coordinates are repeatedly projected onto the surface in the course of the iteration.

8.8 Comments

A number of algorithms for generation of curve grids were discussed by Eiseman (1987) and Knupp and Steinberg (1993).

The use of Beltrami's equations to generate surface grids was proposed by Warsi (1982), in analogy with the widely utilized Laplace grid generator of Crowley (1962) and Winslow (1967). Warsi (1990) has also justified these equations by using some fundamental results of differential geometry. Theoretical analyses of the relation of Beltrami's equations to the equations of Gauss and Weingarten were given by

Warsi (1982, 1990) and Garon and Camarero (1983). An implementation of the Beltrami operator to derive a fourth-order surface elliptic system was performed by Ronzheimer et al. (1994), while Spekreijse et al. (1995) applied this operator with algebraic techniques to generate surface grids with orthogonality at the edges.

A surface grid generation scheme that uses a quasi-two-dimensional elliptic system, obtained by projecting the inverted three-dimensional Laplace system, to generate grids on smooth surfaces analytically specified by the equation $z = f(x, y)$ was proposed by Thomas (1982). The method was extended and updated by Takagi et al. (1985) and Warsi (1986) for arbitrary curved surfaces using a parametric surface representation. An adaptive surface grid technique based on control functions and parametric specifications was also considered by Lee and Loellbach (1989).

Some robust blending functions for algebraic surface grid generation were proposed by Soni (1985) using the normalized arc lengths of the physical edges of the surface patches. These functions and various techniques for the surface patch parametrization were discussed by Samareh-Abolhassani and Stewart (1994) for the purpose of the development of a surface grid software system.

The hyperbolic approach based on grid orthogonality was extended to surfaces by Steger (1991). An analogous technique for generating overset surface grids was presented by Chan and Buning (1995).

Liseikin (1991, 1992, 2004, 2007) used an elliptic system derived from a variational principle to produce n-dimensional harmonic coordinate transformations which, in a particular two-dimensional case, generate both uniform and adaptive grids on surfaces. Harmonic mapping was also used by Arina and Casella (1991) to derive a surface elliptic system. The conformal mapping technique for generating surface grids presented in this chapter was formulated by Khamayseh and Mastin (1996).

Variational approaches for generating grids on surfaces were described by Saltzman (1986), Liseikin (1991), and Steinberg and Roache (1986). A variational adaptive technique for deriving a surface grid approaching orthogonality was developed by Desbois and Jacquotte (1991). A variational approach was also given by Castilio (1991) for the control of spacing, cell area, orthogonality, and quality measures. Several grid generation anomalies which appear while implementing some surface variational techniques were discovered by Steinberg and Roache (1986).

An optimization approach to surface grid generation which aimed to maximize grid smoothness and orthogonality was discussed by Pearce (1990). Some techniques for clustering the grid points in regions of larger curvature were considered by Weilmuenster et al. (1991).

References

Arina, R., & Casella, M. (1991). A harmonic grid generation technique for surfaces and three-dimensional regions. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 935–946). New York: North-Holland.

Castilio, J. E. (1991). The discrete grid generation method on curves and surfaces. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 915–924). New York: North-Holland.

Chan, W. M., & Buning, P. G. (1995). Surface grid generation methods for overset grids. *Computer and Fluids*, *24*(5), 509–522.

Crowley, W. P. (1962). An equipotential zoner on a quadrilateral mesh. Memo, Lawrence Livermore National Lab, 5 July 1962.

Desbois, F., & Jacquotte, O.-P. (1991). Surface mesh generation and optimization. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 131–142). New York: North-Holland.

Eiseman, P. R. (1987). Adaptive grid generation. *Computer Methods in Applied Mechanics and Engineering*, *64*, 321–376.

Garon, A., & Camarero, R. (1983). Generation of surface-fitted coordinate grids. In K. N. Ghia & U. Ghia (Eds.), *Advances in Grid Generation* (pp. 117–122). Houston, TX: ASME.

Khamayseh, A., & Mastin, C. W. (1996). Computational conformal mapping for surface grid generation. *Journal of Computational Physics*, *123*, 394–401.

Knupp, P. M., & Steinberg, S. (1993). *Fundamentals of Grid Generation*. Boca Raton: CRC Press.

Lee, K. D., & Loellbach, J. M. (1989). Geometry-adaptive surface grid generation using a parametric projection. *Journal of Aircraft*, *2*, 162–167.

Liseikin, V. D. (1991): On generation of regular grids on n-dimensional surfaces. *Journal of Computational Mathematics and Mathematical Physics*. *31*, 1670–1689 (1991) (Russian). [English transl.: *USSR Computational Mathematics and Mathematical Physics*. *31*(11), 47–57].

Liseikin, V. D. (1992). On a variational method of generating adaptive grids on n-dimensional surfaces. *Soviet Mathematics - Doklady*, *44*(1), 149–152.

Liseikin, V. D. (2004). *A Computational Differential Geometry Approach to Grid Generation*. Berlin: Springer.

Liseikin, V. D. (2007). *A computational differential geometry approach to grid generation* (2nd edition). Berlin: Springer.

Pearce, D. (1990). Optimized grid generation with geometry definition encoupled. AIAA Paper 90-0332.

Ronzheimer, A., Brodersen, O., Rudnik, R., & Findling, A. (1994). A new interactive tool for the management of grid generation processes around arbitrary configurations. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulation and Related Fields* (p. 441). Swansea: Pineridge.

Saltzman, J. S. (1986). Variational methods for generating meshes on surfaces in three dimensions. *Journal of Computational Physics*, *63*, 1–19.

Samareh-Abolhassani, J., & Stewart, J. E. (1994). Surface grid generation in a parameter space. *Journal of Computational Physics*, *113*, 112–121.

Soni, B. K. (1985). Two and three dimensional grid generation for internal flow applications of computational fluid dynamics. AIAA Paper 85-1526.

Spekreijse, S. P., Nijhuis, G. H., & Boerstoel, J. W. (1995). Elliptic surface grid generation on minimal and parametrized surfaces. In: *Proceedings of the Surface Modeling, Grid Generation and Related Issues in Computational Fluid Dynamics Workshop. NASA Conference Publication* (Vol. 3291, p. 617). Cleveland: NASA Lewis Research Center.

Steger, J. L. (1991). Grid generation with hyperbolic partial differential equations for application to complex configurations. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 871–886). New York: North-Holland.

Steinberg, S., & Roache, P. J. (1986). Variational grid generation. *Numerical Methods for Partial Differential Equations*, *2*, 71–96.

Takagi, T., Miki, K., Chen, B. C. J., & Sha, W. T. (1985). Numerical generation of boundary-fitted curvilinear coordinate systems for arbitrarily curved surfaces. *Journal of Computational Physics*, *58*, 69–79.

Thomas, P. D. (1982). Composite three-dimensional grids generated by elliptic systems. *AIAA Journal*, *20*(9), 1195–1202.

Warsi, Z. U. A. (1982). Basic differential models for coordinate generation. In J. F. Thompson (Ed.), *Numerical Grid Generation* (pp. 41–78). New York: North-Holland.

Warsi, Z. U. A. (1986). Numerical grid generation in arbitrary surfaces through a second-order differential-geometric model. *Journal of Computational Physics*, *64*, 82–96.

Warsi, Z. U. A. (1990). Theoretical foundation of the equations for the generation of surface coordinates. *AIAA Journal*, *28*(6), 1140–1142.

Weilmuenster, K. J., Smith, R. E., & Everton, E. L. (1991). Gridding strategies and associated results for winged entry vechicles. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 217–228). New York: North-Holland.

Winslow, A. M. (1967). Equipotential zoning of two-dimensional meshes. *Journal of Computational Physics*, *1*, 149–172.

Chapter 9
Comprehensive Method

9.1 Introduction

Many physical phenomena involve the rapid formation, propagation, and disintegration of small-scale structures. Examples include shock waves in compressible flows; shear layers in laminar and turbulent flows; phase boundaries in nonequilibrium and boundary and interior layers; tearing layers and magnetic reconnection regions in magnetically confined plasmas; and combustion and detonation fronts. A promising tool for dealing with the numerical problems related to these structures is adaptive grid generation technology. With increasing complexity of the physical problem, there is an increased need for more reliable, robust, and fully-automated grid generation codes which enable one to generate suitable meshes in a uniform "black box" mode, with negligible human interaction (Liseikin 1999). The development of such grid systems is a challenging problem of modern computational physics and applied mathematics.

The method advocated in this chapter is a further development of that presented in Liseikin (1999, 2007). It is based on a mapping from a simple structured or unstructured grid in the logical domain to a curvilinear grid with the desired properties in the physical domain. At the heart of the method are Beltrami and diffusion equations with control metrics and weight functions whose specification governs the grid properties. The equations for the evolving grid can be incorporated into a single implicit time step in which the grid and the physical solution evolve together. The main purpose of this chapter is to illustrate the choice of these quantities to generate adaptive grids for a wide variety of computational physics problems.

Recent new results in the field of grid generation by means of the mapping approach have largely been related to the application of harmonic function theory to adaptive grid generation. The suggestion to use harmonic functions for generating adaptive grids was made by Dvinsky (1991). Adaptive grids can be generated by mapping the reference grid into the domain with a coordinate transformation which is inverse to a harmonic vector function (in terms of Riemannian manifolds). Adaptation is performed by a specified adaptive metric in the domain which converts it

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_9

into a Riemannian manifold. Each harmonic function minimizes some functional of the total energy, and hence it can be found through the numerical solution either of a variational problem or of a boundary value problem for a system of Euler–Lagrange equations.

The basic comprehensive method discussed in this chapter is one version of the harmonic-function approach for generating both adaptive and fixed grids. The method relies on a variational technique for the generation of grids on hypersurfaces with the help of an energy functional with respect to control metrics. Specifically, the adaptive grid with node clustering in the zones of large values of a vector function is obtained as a projection of a quasiuniform grid from a monitor surface generated as a surface of the function values over the physical space (Liseikin 1991). The vector function can be the physical solution or a combination of its components or derivatives, or it can be any other quantity that suitably monitors the behavior of the solution. A generalization of this approach for generating grids with other properties was performed by Liseikin (2007). The method developed allows the designer to merge the two tasks of surface grid generation and volume grid generation into one task while developing a comprehensive grid generation code. Since the grid generation is based on harmonic coordinate transformations that are able to generate unfolded grids in regions with complex geometry, the method can also relieve an array of bottlenecks of codes by reducing the number of blocks required for partitioning a complicated physical domain. The functions representing the monitor surface and control metrics are easily determined, thus providing efficient and straightforwardly controlled grid adaptation of various types. Thus, the method is free from the drawbacks of the elliptic method based on Poisson equations, and its numerical implementation should provide a uniform environment for the generation of fixed and adaptive grids in arbitrary regions. This gives grounds to expect that the method will be relevant to a large number of application areas.

For the purpose of commonality, a general approach based on differential and variational methods for the generation of quasiuniform grids on arbitrary hypersurfaces is considered. The variational method of generating quasiuniform grids is grounded on the minimization of a generalized functional of grid smoothness on hypersurfaces, which was introduced for the Euclidian control metric by Brackbill and Saltzman (1982). The energy functional with respect to the metric of a monitor surface is the functional of smoothness which is defined for a general hypersurface through the invariants of its metric tensor in the grid coordinates and has a clear geometric interpretation of a measure of grid nonuniformity (Liseikin 1993). In fact, the grid in this method is derived from a coordinate transformation that is inverse to the solution of a system of Beltrami equations which are the Euler–Lagrange equations of the functional. A two-dimensional Laplace system using the logical coordinates as dependent variables was proposed by Crowley (1962) and Winslow (1967). A rather geometric approach to analyzing the qualitative behavior near boundary segments of the coordinate lines obtained through the inverted Laplace equations was described for this purpose in the monograph by Thompson et al. (1985). A more detailed analysis of grid behavior near the boundary of both surfaces and two- and three-dimensional domains was performed by Liseikin in (2007).

This chapter describes the formulation and some properties of the method and discusses its relation to other techniques, in particular, to the approaches using harmonic functions and Beltrami equations.

9.2 Hypersurface Geometry and Grid Formulation

In order to formulate a unified grid model for all physical geometries, we shall consider curves, surfaces, and domains of arbitrary dimensions in the same manner, as n-dimensional hypersurfaces lying in the $(n+k)$-dimensional space R^{n+k}, $n \geq 1$, $k \geq 0$, though in practical applications, the dimension n equals 1, 2, and 3. Each hypersurface under consideration is supposed to be represented by a smooth parametrization of rank n at all points $\boldsymbol{s} \in S^n$:

$$\boldsymbol{x}(\boldsymbol{s}) : S^n \to R^{n+k} \ , \ \boldsymbol{x} = (x^1(\boldsymbol{s}), \dots, x^{n+k}(\boldsymbol{s})) \ , \ \boldsymbol{s} = (s^1, \cdots, s^n) \ , \tag{9.1}$$

where S^n is some n-dimensional parametric domain in R^n (an interval if $n = 1$) with Cartesian coordinates s^i, $i = 1, \dots, n$. The hypersurface represented by (9.1) is designated by S^{xn}. Note, when $k = 0$, then S^{xn} is a domain $X^n \subset \mathbf{R}^n$ which itself can naturally be considered as a parametric domain S^n for X^n with $\mathbf{x}(\mathbf{s})$ being the identical, i.e., $\mathbf{x}(\mathbf{s}) \equiv \mathbf{s}$ (Fig. 9.1), or another parametrization.

This section gives a natural multidimensional generalization of the notions and relations considered for n-dimensional domains, curves, and two-dimensional surfaces in Chaps. 2, 3, and 8.

9.2.1 Hypersurface Grid Formulation

In order to generate a numerical grid by a mapping approach on an arbitrary hypersurface S^{xn}, represented by a parametrization $\boldsymbol{x}(\boldsymbol{s}) : S^n \to R^{n+k}$, both an n-dimensional computational domain $\Xi^n \subset R^n$ and an intermediate invertible, smooth transformation

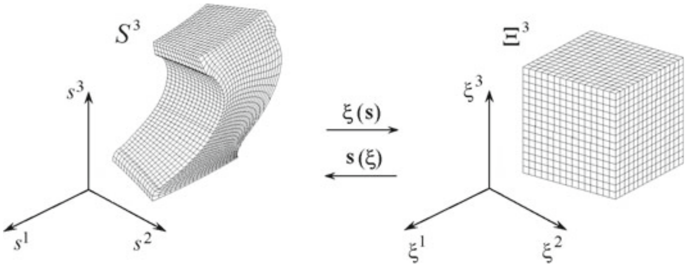


Fig. 9.1 Scheme for generating hexahedral grids

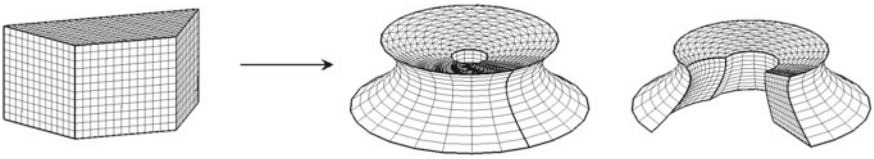


Fig. 9.2 Scheme for generating prismatic grids

$$s(\boldsymbol{\xi}) : \Xi^n \to S^n\ , \quad s(\boldsymbol{\xi}) = [s^1(\boldsymbol{\xi}), \ldots, s^n(\boldsymbol{\xi})]\ , \quad \boldsymbol{\xi} = (\xi^1, \ldots \xi^n)\ , \quad \boldsymbol{\xi} \in \Xi^n\ , \tag{9.2}$$

between a suitable computational (logical) domain Ξ^n and the parametrization domain S^n are determined. Then, the numerical grid on the hypersurface S^{xn} is built by mapping the nodes of some reference grid, specified in the computational domain Ξ^n, onto the hypersurface S^{xn} with the aid of the composition of the transformations $\boldsymbol{x}(\boldsymbol{s})$ and $\boldsymbol{s}(\boldsymbol{\xi})$, i.e., with

$$\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})] : \Xi^n \to R^{n+k}\ , \tag{9.3}$$

(see Fig. 9.1 for the case when S^{xn} coincides with S^n, $n = 3$). The transformation (9.3) defines a new coordinate system $\xi^1, \ldots \xi^n$, hereafter referred to as the grid coordinate system of the hypersurface S^{xn}.

The original parametrization $\boldsymbol{x}(\boldsymbol{s})$ also generates a grid on S^{xn} by mapping some reference grid in S^n. However, this grid may be unsatisfactory. The role of the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$ is to generate a suitable grid in the parametrization domain S^n, which is mapped with $\boldsymbol{x}(\boldsymbol{s})$ on S^{xn} thus forming the grid on S^{xn} with the necessary properties.

Of course, in contrast to the parametric domain S^n, the computational domain Ξ^n should have a more simple geometry of its boundary, allowing one to specify the reference grid easily, for instance, analytically. The reference grid may be rectangular (see Fig. 9.1) or have another configuration, say, tetrahedral or prismatic (see Fig. 9.2) when $n = 3$, or unstructured, which results in an unstructured grid in S^{xn}.

9.2.2 Monitor Hypersurfaces

An example of a hypersurface, suitable for the purpose of adaptive grid generation, is represented by a monitor hypersurface formed by values of some, in general, vector function over the physical domain or surface. This vector function can be a solution to the problem of interest, a combination of its components or derivatives, or any other variable vector quantity that suitably monitors the features of the physical solution or of the geometry of the physical domain or surface which significantly affect the accuracy of the calculations. The vector functions provide an efficient opportunity to control the grid quality, in particular, the concentration of grid nodes in the zones of large variations of such a function (see Fig. 9.3, left-hand). The parametrization of the

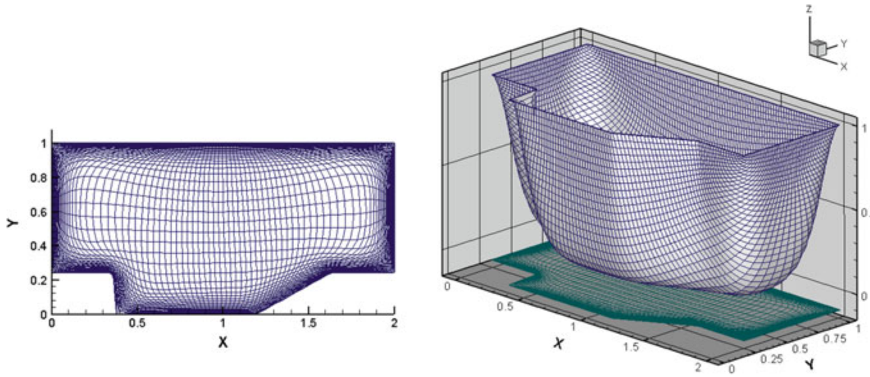


Fig. 9.3 Scheme of grid adaptation by the use of a monitor hypersurface

monitor hypersurface is established very simply. For example, in one case, important for the generation of adaptive grids in a physical domain $X^n \subset R^n$, the monitor hypersurface designated further by S^{rn} is defined as an n-dimensional hypersurface formed as the graph of some vector function

$$\boldsymbol{f}(\boldsymbol{x}) : X^n \to R^l, \quad \boldsymbol{x} = (x^1, \ldots, x^n) \ , \quad \boldsymbol{f} = [f^1(\boldsymbol{x}), \ldots, f^l(\boldsymbol{x})] \ ,$$

over X^n. This monitor hypersurface S^{rn} lies in the $(n+l)$-dimensional space R^{n+l} (see Fig. 9.3, right-hand, for $n = 2, l = 1$). It is apparent that the parametric domain S^n can coincide with X^n and, consequently, the parametric mapping designated further by $\boldsymbol{r}(\boldsymbol{s}) : S^n \to R^{n+l}$ of S^{rn} can be defined as

$$\boldsymbol{r}(\boldsymbol{s}) = [\boldsymbol{s}, \boldsymbol{f}(\boldsymbol{s})] = [s^1, \ldots, s^n, f^1(\boldsymbol{s}), \ldots, f^l(\boldsymbol{s})] \ , \qquad S^n = X^n \ , \qquad \boldsymbol{s} = \boldsymbol{x} \ . \tag{9.4}$$

If the monitor hypersurface S^{xn} is formed by the values of the function $\boldsymbol{f}(\boldsymbol{x})$ over an n-dimensional hypersurface S^{xn} lying in the space R^{n+k} and represented by the parametrization (9.1) from an n-dimensional domain S^n in the space R^{n+k}, then the monitor hypersurface S^{rn} lies in the space R^{n+k+l} and it can be described by a parametrization from S^n in the form

$$\boldsymbol{r}(\boldsymbol{s}) : S^n \to R^{n+k+l} \ , \qquad \boldsymbol{r}(\boldsymbol{s}) = \{\boldsymbol{x}(\boldsymbol{s}), \boldsymbol{f}[\boldsymbol{x}(\boldsymbol{s})]\} \ . \tag{9.5}$$

In particular, a one-dimensional monitor surface S^{r1} (curve) over a curve S^{x1} lying in R^n, $n = 1, 2, 3$, and represented by

$$\boldsymbol{x}(\varphi) : [a, b] \to R^n \ ,$$

can be defined by the parametrization

$$\boldsymbol{r}(s) : [a, b] \to R^{n+l} \ , \qquad \boldsymbol{r}(s) = \{\boldsymbol{x}(s), \boldsymbol{f}[\boldsymbol{x}(s)]\} \ , \qquad s = \varphi \ . \tag{9.6}$$

9.2.3 Metric Tensors

As for ordinary two-dimensional surfaces lying in R^3, the interior features of the hypersurface S^{xn} are defined by the elements of the covariant metric tensor which in the coordinates v^i are denoted as g_{ij}^{xv} and computed by the dot products of the vectors $\boldsymbol{x}_{v^i} = \partial \boldsymbol{x}/\partial v^i$ that are tangent to the corresponding coordinate lines in S^{xn}, i.e.,

$$g_{ij}^{xv} = \boldsymbol{x}_{v^i} \cdot \boldsymbol{x}_{v^j} \ , \qquad i, j = 1, \dots, n \ . \tag{9.7}$$

The determinant of the covariant metric tensor $\{g_{ij}^{xv}\}$ is denoted by g^{xv}.

The contravariant elements of the hypersurface S^{xn} in the coordinates v^i are denoted by g_{vx}^{ij}. The contravariant metric tensor $\{g_{vx}^{ij}\}$ is the inverse of the matrix $\{g_{ij}^{xv}\}$.

Similar to (9.7), we shall designate by g_{ij}^{rv} (g_{vr}^{ij}) the covariant (contravariant) metric elements of the monitor hypersurface S^{rn} in the parametric coordinates $v^1, \dots, v^n$, i.e.,

$$g_{ij}^{rv} = \boldsymbol{r}_{v^i} \cdot \boldsymbol{r}_{v^j} \ , \ i, j = 1, \dots, n \ . \tag{9.8}$$

Thus, in the case of the parametrization (9.4) of the monitor hypersurface S^{rn} over the domain S^n, we obtain

$$g_{ij}^{rs} = \delta_j^i + \boldsymbol{f}_{s^i} \cdot \boldsymbol{f}_{s^j} \ , \qquad i, j = 1, \dots, n \ . \tag{9.9}$$

If $\boldsymbol{f}$ is a scalar function f, then the determinant g^{rs} of the covariant metric tensor $\{g_{ij}^{rs}\}$, whose elements are defined by (9.9), is readily computed:

$$g^{rs} = 1 + \frac{\partial f}{\partial s^i}\frac{\partial f}{\partial s^i} = 1 + (\nabla f)^2 \ , \qquad i = 1, \dots, n \ .$$

The parametrization (9.5) of the monitor hypersurface S^{rn} determined by the values of $\boldsymbol{f}$ over the surface S^{xn} yields

$$g_{ij}^{rs} = g_{ij}^{xs} + \frac{\partial \boldsymbol{f}[\boldsymbol{x}(\boldsymbol{s})]}{\partial s^i} \cdot \frac{\partial \boldsymbol{f}[\boldsymbol{x}(\boldsymbol{s})]}{\partial s^j} \ , \qquad i, j = 1, \dots, n \ , \tag{9.10}$$

where $\{g_{ij}^{xs}\}$ is the covariant metric tensor of the hypersurface S^{xn} in the coordinates s^i, $i = 1, \dots, n$. In the case of a scalar monitor function f, we obtain for (9.10)

$$g^{rs} = g^{xs}\Big(1 + g_{sx}^{ij}\frac{\partial f}{\partial s^i}\frac{\partial f}{\partial s^j}\Big) \ , \qquad i, j = 1, \dots, n \ , \tag{9.11}$$

where $g^{xs} = \det\{g_{ij}^{xs}\}$, $\{g_{sx}^{ij}\}$ is the inverse of $\{g_{ij}^{xs}\}$, and by $\partial f/\partial s^i$, $\partial f[\boldsymbol{x}(\boldsymbol{s})]/\partial s^i$ is meant here. In particular, for $n = 2$,

$$g^{rs} = g^{xs} + g^{xs}_{11}\Big(\frac{\partial f}{\partial s^2}\Big)^2 - 2g^{xs}_{12}\frac{\partial f}{\partial s^1}\frac{\partial f}{\partial s^2} + g^{xs}_{22}\Big(\frac{\partial f}{\partial s^1}\Big)^2 .$$

As a reminder, repeated indices in formulas here and further mean summation over them.

9.2.4 Relations Between Metric Elements

The mapping $\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]$ which generates a grid on the hypersurface S^{xn} determines a new coordinate system $\xi^i,\ i = 1, \dots, n$, on S^{xn} and it also defines the values of the covariant metric tensor $\{g^{x\xi}_{ij}\}$ in the coordinates ξ^i, whose elements are the scalar products of the vectors $\boldsymbol{x}_{\xi^i} = \partial \boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]/\partial \xi^i,\ i = 1, \dots, n$, i.e.,

$$g^{x\xi}_{ij} = \boldsymbol{x}_{\xi^i} \cdot \boldsymbol{x}_{\xi^j} , \qquad i, j = 1, \dots, n .$$

The covariant metric tensor $\{g^{x\xi}_{ij}\}$ of the hypersurface S^{xn} in the grid coordinates $\xi^1, \dots, \xi^n$, represented by (9.3), has n invariants $I_1, \dots, I_n$, whose geometrical meaning is described through the geometrical measures of the edges, faces, etc., of the n-dimensional parallelepiped formed by the tangent vectors $\boldsymbol{x}_{\xi^i},\ i = 1, \dots, n$. In analogy with Chap. 3, the invariants of the metric tensor $\{g^{x\xi}_{ij}\}$ of the hypersurface S^{xn} can be used to formulate some quality properties of the grid on S^{xn}.

The elements of the covariant tensor of S^{xn} in the coordinates s^i and ξ^i are connected by the following relations:

$$\begin{aligned} g^{x\xi}_{ij} &= g^{xs}_{ml}\frac{\partial s^m}{\partial \xi^i}\frac{\partial s^l}{\partial \xi^j} , \\ g^{xs}_{ij} &= g^{x\xi}_{ml}\frac{\partial \xi^m}{\partial s^i}\frac{\partial \xi^l}{\partial s^j} , \qquad i, j, l, m = 1, \dots, n , \end{aligned} \tag{9.12}$$

where $\partial \xi^m/\partial s^i,\ i, m = 1, \dots, n$, is the first derivative with respect to s^i of the mth component $\xi^m(\boldsymbol{s})$ of the mapping

$$\boldsymbol{\xi}(\boldsymbol{s}) : S^n \to \Xi^n , \quad \boldsymbol{\xi}(\boldsymbol{s}) = [\xi^1(\boldsymbol{s}), \dots, \xi^n(\boldsymbol{s})] , \quad \boldsymbol{s} = (s^1, \dots, s^n) , \quad \boldsymbol{s} \in S^n ,$$

which is inverse to the intermediate mapping

$$\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^n \to S^n , \qquad \boldsymbol{s}(\boldsymbol{\xi}) = [s^1(\boldsymbol{\xi}), \dots, s^n(\boldsymbol{\xi})] .$$

In particular, for the covariant metric elements $g^{r\xi}_{ij}$ of a monitor hypersurface S^{rn} over a domain S^n in the coordinates $\xi^i,\ i = 1, \dots, n$, formulas (9.9) and (9.12) give

$$\begin{aligned} g^{r\xi}_{ij} &= g^{rs}_{ml}\frac{\partial s^m}{\partial \xi^i}\frac{\partial s^l}{\partial \xi^j} = \frac{\partial s^l}{\partial \xi^i}\frac{\partial s^l}{\partial \xi^j} + \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^i}\cdot\frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^j} \\ &= g^{s\xi}_{ij} + \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^i}\cdot\frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^j}\,, \qquad i,j,l,m = 1,\dots,n\,, \end{aligned} \tag{9.13}$$

where

$$g^{s\xi}_{ij} = \frac{\partial \boldsymbol{s}}{\partial \xi^i}\cdot\frac{\partial \boldsymbol{s}}{\partial \xi^j} = \frac{\partial s^l}{\partial \xi^i}\frac{\partial s^l}{\partial \xi^j}\,, \qquad i,j,l = 1,\dots,n\,,$$

is the (ij)th covariant metric element of the domain S^n in the coordinates ξ^i, $i = 1,\dots,n$.

Similarly, for the metric elements $g^{r\xi}_{ij}$ of a monitor hypersurface S^{rn} over S^{xn} in the coordinates ξ^i, $i = 1,\dots,n$, we have, from (9.10) and (9.12),

$$g^{r\xi}_{ij} = g^{x\xi}_{ij} + \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^i}\cdot\frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^j}\,, \qquad i,j = 1,\dots,n\,. \tag{9.14}$$

The intermediate mapping $\boldsymbol{s}(\boldsymbol{\xi})$ and parametric transformation $\boldsymbol{x}(\boldsymbol{s}) : S^n \to S^{xn}$ serve to generate a suitable grid on the hypersurface S^{xn}.

The contravariant metric tensor $\{g^{ij}_{\xi x}\}$ of the hypersurface S^{xn} in the grid coordinates ξ^i, $i = 1,\dots,n$, is the inverse to the covariant metric tensor $\{g^{x\xi}_{ij}\}$, that is,

$$g^{x\xi}_{ij}\, g^{jl}_{\xi x} = \delta^i_l\,, \qquad i,j,l = 1,\dots,n\,.$$

In analogy with the relations (8.23) for two-dimensional surfaces, the elements of the contravariant tensor of S^{xn} in the coordinates s^i and ξ^i are connected as follows:

$$\begin{aligned} g^{ij}_{\xi x} &= g^{ml}_{sx}\frac{\partial \xi^i}{\partial s^m}\frac{\partial \xi^j}{\partial s^l}\,, \\ g^{ij}_{sx} &= g^{ml}_{\xi x}\frac{\partial s^i}{\partial \xi^m}\frac{\partial s^j}{\partial \xi^l}\,, \qquad i,j,l,m = 1,\dots,n\,. \end{aligned} \tag{9.15}$$

Similarly to (8.24), we also have the relations

$$\begin{aligned} g^{ii}_{sx} &= g^{mk}_{\xi x}\, g^{s\xi}_{mk}\,, \qquad i,k,m = 1,\dots,n\,, \\ g^{x\xi} &= J^2 g^{xs}\,, \end{aligned}$$

where

$$g^{x\xi} = \det\{g^{x\xi}_{ij}\}\,, \qquad J = \det\left\{\frac{\partial s^i}{\partial \xi^j}\right\}\,, \quad i,j = 1,\dots,n\,.$$

9.2.5 Christoffel Symbols

The quantities

$$[ij,l] = \boldsymbol{x}_{s^i s^j} \cdot \boldsymbol{x}_{s^l}\ , \qquad i,j,l = 1,\dots,n\ , \tag{9.16}$$

where

$$\boldsymbol{x}_{s^i s^j} = \frac{\partial^2 \boldsymbol{x}(\boldsymbol{s})}{\partial s^i \partial s^j}\ , \qquad \boldsymbol{x}_{s^l} = \frac{\partial \boldsymbol{x}(\boldsymbol{s})}{\partial s^l}\ , \qquad i,j,l = 1,\dots,n\ ,$$

are the Christoffel symbols of the first kind of the hypersurface S^{xn} in the parametric coordinates s^i. These symbols are, in fact, the space Christoffel symbols and, therefore, are subject to the relations (2.45).

In analogy with (8.40), the hypersurface Christoffel symbols of the second kind in the coordinates s^i are defined by the relation

$$\Upsilon^l_{ij} = g^{lm}_{sx}[ij,m]\ , \qquad i,j,l,m = 1,\dots,n\ . \tag{9.17}$$

From (2.45),

$$\begin{aligned}\Upsilon^j_{ji} &= g^{jm}_{sx}[ji,m] = \frac{1}{2} g^{jm}_{sx}\Big(\frac{\partial g^{xs}_{jm}}{\partial s^i} + \frac{\partial g^{xs}_{im}}{\partial s^j} - \frac{\partial g^{xs}_{ij}}{\partial s^m}\Big)\\ &= \frac{1}{2} g^{jm}_{sx}\frac{\partial g^{xs}_{jm}}{\partial s^i}\ , \qquad i,j,m = 1,\dots,n\ ,\end{aligned}$$

and in accordance with the formula (2.47) for differentiation of the Jacobian, we have an analog of the identity (2.46) in the form

$$\begin{aligned}\frac{\partial}{\partial s^i}\sqrt{g^{xs}} &= \frac{1}{2}\sqrt{g^{xs}}\, g^{jm}_{sx}\frac{\partial g^{xs}_{jm}}{\partial s^i}\\ &= \sqrt{g^{xs}}\,\Upsilon^j_{ji}\ , \qquad i,j,m = 1,\dots,n\ .\end{aligned} \tag{9.18}$$

Now we determine the role of the Christoffel symbols in the expansion of the derivatives of the tangent vectors $\boldsymbol{x}_{s^i}$. Let $\boldsymbol{d}_{ij}$ be the vector defined by the relation

$$\boldsymbol{d}_{ij} = \boldsymbol{x}_{s^i s^j} - \Upsilon^l_{ij}\boldsymbol{x}_{s^l}\ , \qquad i,j,l = 1,\dots,n\ .$$

We have

$$\boldsymbol{d}_{ij} \cdot \boldsymbol{x}_{s^m} = [ij,m] - \Upsilon^l_{ij} g^{xs}_{lm} = 0\ , \qquad i,j,l,m = 1,\dots,n\ ,$$

from (9.17). Thus, we find that in the following expansion of the vectors $\boldsymbol{x}_{s^i s^j}$,

$$\boldsymbol{x}_{s^i s^j} = \Upsilon^l_{ij}\boldsymbol{x}_{s^l} + \boldsymbol{d}_{ij}\ , \qquad i,j,l = 1,\dots,n\ , \tag{9.19}$$

the vectors $\boldsymbol{d}_{ij}$ lie in the k-dimensional hypersurface which is orthogonal to the tangent n-dimensional hypersurface defined by the tangent vectors $\boldsymbol{x}_{s^i}$. Note that if some vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ from R^{n+k} comprise an orthonormal basis for this k-dimensional hypersurface, i.e.,

$$\begin{aligned}&\boldsymbol{v}_m \cdot \boldsymbol{x}_{s^j} = 0\,, \quad m = 1, \ldots, k\,, \quad j = 1, \ldots, n\,,\\ &\boldsymbol{v}_m \cdot \boldsymbol{v}_p = \delta^m_p\,,\ m, p = 1, \ldots, k\,,\end{aligned}$$

then, in accordance with (2.6), we find that

$$\boldsymbol{x}_{s^i s^j} = \Upsilon^l_{ij} \boldsymbol{x}_{s^l} + (\boldsymbol{x}_{s^i s^j} \cdot \boldsymbol{v}_m)\boldsymbol{v}_m\,, \qquad i, j, l = 1, \ldots, n\,, \qquad m = 1, \ldots, k\,.$$

Thus, we obtain, from (9.19),

$$\boldsymbol{d}_{ij} = (\boldsymbol{x}_{s^i s^j} \cdot \boldsymbol{v}_m)\boldsymbol{v}_m\,, \qquad i, j = 1, \ldots, n\,, \qquad m = 1, \ldots, k\,,$$

and so (9.19) is a generalization of (2.36) and (8.42) with the identification $\xi^i = s^i$.

9.3 Functional of Smoothness

One of the ways of finding the intermediate coordinate transformations $\boldsymbol{s}(\boldsymbol{\xi})$ between the computational and parametric domains required to generate grids on hypersurfaces is to use variational methods. In accordance with the grid methods for domains and two-dimensional surfaces considered above, the most appropriate functional for this purpose is that of smoothness, since it generates a system of Beltrami equations possessing the unique properties desired for grid generation.

9.3.1 Formulation of the Functional

Similarly to (8.31), the expression for the functional of grid smoothness on the hypersurface S^{xn} with the parametrization (9.1) is represented as

$$I_{\mathrm{S}} = \int\limits_{S^{xn}} \Big(\sum_{i=1}^{n} g^{ii}_{\xi x}\Big) \mathrm{d}S^{xn} = \int\limits_{S^{xn}} \mathrm{tr}\{g^{ij}_{\xi x}\} \mathrm{d}S^{xn}\,. \tag{9.20}$$

This functional is defined on the set of invertible functions $\boldsymbol{\xi}(\boldsymbol{s}) \in C^2(S^n)$. Since

$$\mathrm{d}S^{xn} = \sqrt{g^{xs}}\mathrm{d}\boldsymbol{s} = \sqrt{g^{x\xi}}\mathrm{d}\boldsymbol{\xi}\,,$$

we obtain, from (9.15) and (9.20),

$$I_{\mathrm{S}} = \int\limits_{S^n} \sqrt{g^{xs}} \mathrm{tr}\{g^{ij}_{\xi x}\} \mathrm{d}\boldsymbol{s} = \int\limits_{S^n} \sqrt{g^{xs}} g^{kl}_{sx} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^l} \mathrm{d}\boldsymbol{s} \ , \qquad i, k, l = 1, \dots, n \ . \quad (9.21)$$

Thus, for $n = 1, 2$, and 3, we have

$$I_{\mathrm{S}} = \begin{cases} \int\limits_{S^1} \sqrt{g^{xs}} g^{11}_{\xi x} \mathrm{d}s \ , & n = 1 \ , \\ \int\limits_{S^2} \sqrt{g^{xs}} (g^{11}_{\xi x} + g^{22}_{\xi x}) \mathrm{d}s^1 \mathrm{d}s^2 \ , & n = 2 \ , \\ \int\limits_{S^3} \sqrt{g^{xs}} (g^{11}_{\xi x} + g^{22}_{\xi x} + g^{33}_{\xi x}) \mathrm{d}s^1 \mathrm{d}s^2 \mathrm{d}s^3 \ , & n = 3 \ , \end{cases} \quad (9.22)$$

with the corresponding contravariant metric elements $g^{ij}_{\xi x}$ and determinants g^{xs} for each $n = 1, 2, 3$.

Analogously, we obtain the formulation of the inverted smoothness functional in terms of the covariant metric elements $g^{x\xi}_{ij}$:

$$I_{IS} = \int\limits_{\Xi^n} \sqrt{g^{x\xi}} \mathrm{tr}\{g^{ij}_{\xi x}\} \mathrm{d}\boldsymbol{\xi} \ . \quad (9.23)$$

Note that the functional I_{IS}, in this formulation, is defined on the set of invertible transformations $\boldsymbol{s}(\boldsymbol{\xi}) \in C^2(\Xi^n)$. Using suitable formulas for the elements of inverse matrices $\{g^{ij}_{\xi x}\}$, we have for $n = 1, 2$, and 3

$$I_{IS} = \begin{cases} \int\limits_{\Xi^1} \frac{1}{\sqrt{g^{x\xi}}} \mathrm{d}\xi \ , & n = 1 \ , \\ \int\limits_{\Xi^2} \frac{1}{\sqrt{g^{x\xi}}} (g^{x\xi}_{11} + g^{x\xi}_{22}) \mathrm{d}\xi^1 \mathrm{d}\xi^2 \ , & n = 2 \ , \\ \int\limits_{\Xi^3} \frac{1}{\sqrt{g^{x\xi}}} [g^{x\xi}_{11} g^{x\xi}_{22} + g^{x\xi}_{11} g^{x\xi}_{33} + g^{x\xi}_{22} g^{x\xi}_{33} & \\ \qquad - (g^{x\xi}_{12})^2 - (g^{x\xi}_{13})^2 - (g^{x\xi}_{23})^2] \mathrm{d}\xi^1 \mathrm{d}\xi^2 \mathrm{d}\xi^3 \ , & n = 3 \ . \end{cases} \quad (9.24)$$

When the hypersurface S^{xn} is an n-dimensional region S^n, the functional (9.20) is the very functional of grid smoothness on S^n,

$$I_{\mathrm{S}} = \int\limits_{S^n} \Big(\sum_{i=1}^{n} g^{ii}_{\xi s} \Big) \mathrm{d}\boldsymbol{s} \ , \qquad g^{ij}_{\xi s} = \sum_{m=1}^{n} \frac{\partial \xi^i}{\partial s^m} \frac{\partial \xi^j}{\partial s^m} \ , \qquad i, j = 1, \dots, n \ , \quad (9.25)$$

described in Chap. 7 for $n = 2$ and $n = 3$. Therefore, the functional (9.20), being the generalization of (9.25), is called the functional of grid smoothness on the hypersurface S^{xn}. For the inverted smoothness functional (9.25), we have for $n = 1,\ 2,\ 3,$ from (9.24),

$$I_{IS} = \begin{cases} \displaystyle\int_{\Xi^1} \frac{1}{\sqrt{g^{s\xi}}} \mathrm{d}\xi \ , & n = 1 \ , \\ \displaystyle\int_{\Xi^2} \frac{1}{\sqrt{g^{s\xi}}} (g_{11}^{s\xi} + g_{22}^{s\xi}) \mathrm{d}\xi^1 \mathrm{d}\xi^2 \ , & n = 2 \ , \\ \displaystyle\int_{\Xi^3} \frac{1}{\sqrt{g^{s\xi}}} [g_{11}^{s\xi} g_{22}^{s\xi} + g_{11}^{s\xi} g_{33}^{s\xi} + g_{22}^{s\xi} g_{33}^{s\xi} & \\ \qquad -(g_{12}^{s\xi})^2 - (g_{13}^{s\xi})^2 - (g_{23}^{s\xi})^2] \mathrm{d}\xi^1 \mathrm{d}\xi^2 \mathrm{d}\xi^3 \ , & n = 3 \ , \end{cases} \tag{9.26}$$

where

$$g_{ij}^{s\xi} = \frac{\partial \boldsymbol{s}}{\partial \xi^i} \cdot \frac{\partial \boldsymbol{s}}{\partial \xi^j} \ , \qquad i, j = 1, \dots, n \ , \quad g^{s\xi} = \det\{g_{ij}^{s\xi}\} \ . \tag{9.27}$$

Furthermore, it will be shown that such a generalization of the functional (9.25) to n-dimensional hypersurfaces preserves all salient features of grids obtained by applying the smoothness functional on domains.

9.3.2 Geometric Interpretation

This subsection describes a geometric meaning of the smoothness functional which justifies to some extent its expression (9.20) for the generation of quasiuniform grids on hypersurfaces, in particular, on monitor hypersurfaces S^{rn} over domains or surfaces specified by the parametrizations (9.4) and (9.5) and, consequently, adaptive grids in domains and on surfaces. The explanation of the geometric interpretation of the functional generally follows the considerations presented in Sect. 3.7 for domain grid generation and in Sect. 8.5 for two-dimensional surface grid generation.

First, note that the trace of the contravariant n-dimensional metric tensor $\{g_{\xi x}^{ij}\}$ can be expressed through the invariants I_{n-1} and I_n of the orthogonal transforms of the covariant tensor $\{g_{ij}^{x\xi}\}$, namely,

$$\mathrm{tr}\,\{g_{\xi x}^{ij}\} = \frac{I_{n-1}}{I_n}.$$

Therefore, the functional of smoothness (9.20) can also be expressed through these invariants:

$$I_{\mathrm{S}} = \int_{S^{xn}} \Big(\frac{I_{n-1}}{I_n}\Big) \mathrm{d}S^{xn} \ . \tag{9.28}$$

Now, for the purpose of simplicity, we restrict our consideration to three dimensions. The functional (9.28) then has the form

$$I_{\mathrm{S}} = \int\limits_{S^{x3}} \Big(\frac{I_2}{I_3}\Big) \mathrm{d}S^{x3} \ .$$

In three dimensions, the invariant I_3 is the Jacobian of the matrix $\{g_{ij}^{x\xi}\}$, and it represents the volume V^3 of the three-dimensional parallelepiped P^3 formed by the basic tangent vectors $\boldsymbol{x}_i,\ i = 1, 2, 3$. The invariant I_2 of the matrix $\{g_{ij}^{x\xi}\}$ is the sum of its principal minors of order 2. Every principal minor of order 2 equals the Jacobian of the two-dimensional matrix A^2 obtained from $\{g_{ij}^{x\xi}\}$ by crossing out a row and a column which intersect on the diagonal. Therefore, each element of the matrix A^2 is a dot product of two tangential vectors of the basis $\boldsymbol{x}_i,\ i = 1, 2, 3$, and, consequently, the Jacobian of A^2 equals the square of the area of the parallelogram formed by these two vectors. So, the invariants I_2, I_3 can be expressed as

$$I_2 = \sum_{m=1}^{3} \Big(V_m^2\Big)^2 \ , \qquad I_3 = \Big(V^3\Big)^2 \ ,$$

where V_m^2 is the area of the boundary segment of the parallelepiped P^3 formed by the vectors $\boldsymbol{x}_i,\ i = 1, 2, 3$, except for $\boldsymbol{x}_m$, and V^3 is the volume of P^3. Therefore,

$$\frac{I_2}{I_3} = \sum_{m=1}^{3} \Big(V_m^2\Big)^2 / \Big(V^3\Big)^2 \ . \tag{9.29}$$

It is obvious that

$$V^3 = d_m V_m^2 \ , \qquad m = 1, 2, 3 \ ,$$

where d_m is the distance between the vertex of the vector $\boldsymbol{x}_m$ and the plane spanned by the vectors $\boldsymbol{x}_i,\ i \neq m$. Hence, from (9.29),

$$\frac{I_2}{I_3} = \sum_{m=1}^{3} (1/d^m)^2 \ . \tag{9.30}$$

Now let us consider two grid surfaces $\xi^m = c$ and $\xi^m = c + h$ obtained by mapping a uniform rectangular grid with a step size h in the computational domain Ξ^3 onto the hypersurface S^{x3}. The distance l_m between a node on the coordinate surface $\xi^m = c$ and the nearest node on the surface $\xi^m = c + h$ equals $d_m h + O(h)^2$. Therefore, (9.30) is equivalent to

$$\frac{I_2}{I_3} = \sum_{m=1}^{3} (h/l_m)^2 + O(h) \ .$$

The quantity $(h/l_m)^2$ increases as the grid nodes cluster in the direction normal to the surface $\xi^m = c$, and therefore it can be considered as some measure of the grid concentration in this direction; consequently, the functional (9.28) for $n = 3$ defines an integral measure of the grid clustering in all directions. Therefore, as in the case of two-dimensional surfaces considered in Chap. 8, the problem of minimizing the functional of smoothness (9.20) for $n = 3$ can be interpreted as a problem of finding a grid with a minimum of nonuniform clustering, namely a quasiuniform grid on the surface S^{x3}. Analogous interpretations are valid for arbitrary dimensions, i.e., the integrand

$$\sigma(\mathbf{s}) = g^{km}_{\xi s} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^m}, \quad i, k, m = 1, \dots, n \tag{9.31}$$

in the functional (9.20) can be considered to be a relative measure of the grid nonuniformity on the hypersurface S^{xn} represented by (9.1). Since the grid in a domain S^n, and consequently on S^{xn}, obtained by minimizing the inverted energy functional in the metric of a monitor hypersurface S^{rn} is, in fact, the grid obtained by projecting a quasiuniform grid on S^{rn}, and therefore it is clustering in the zones of large variations of the monitor function $\boldsymbol{f}(\boldsymbol{s})$. So, measure (9.31) with respect to the metric (9.10) of the monitor surface S^{rn} over S^{xn}, i.e., when $g^{km}_{\xi s}$ is replaced by $g^{km}_{\xi r}$, can also be considered as a measure of departure of the grid in the physical geometry S^{xn} from an adaptive grid with node clustering in the zones of large variation of the function $\mathbf{f}(\mathbf{s})$.

Notice that, for $n = 2$ and a scalar monitor function $f(s^1,\ s^2)$, the inverted functional of smoothness in the metric of the monitor surface over S^2 is

$$\begin{aligned} I_{IS}[\boldsymbol{s}] = \frac{1}{2} \int\limits_{\Xi^2} \frac{1}{J\sqrt{1 + |\mathrm{grad} f(\boldsymbol{s})|^2}} \Big\{ \Big[\Big(\frac{\partial s^1}{\partial \xi^1}\Big)^2 + \Big(\frac{\partial s^1}{\partial \xi^2}\Big)^2 \Big] \Big[1 + \Big(\frac{\partial f(\boldsymbol{s})}{\partial s^1}\Big)^2 \Big] \\ + \Big[\Big(\frac{\partial s^2}{\partial \xi^1}\Big)^2 + \Big(\frac{\partial s^2}{\partial \xi^2}\Big)^2 \Big] \Big[1 + \Big(\frac{\partial f(\boldsymbol{s})}{\partial s^2}\Big)^2 \Big] \\ + 2 \Big[\frac{\partial s^1}{\partial \xi^1} \frac{\partial s^2}{\partial \xi^1} + \frac{\partial s^1}{\partial \xi^2} \frac{\partial s^2}{\partial \xi^2} \Big] \frac{\partial f(\boldsymbol{s})}{\partial s^1} \frac{\partial f(\boldsymbol{s})}{\partial s^2} \Big\} \, \mathrm{d}\xi^1 \mathrm{d}\xi^2 \,. \end{aligned} \tag{9.32}$$

Note that the expressions

$$\bar{g}^{ij}_{s\xi} = \frac{\partial s^i}{\partial \xi^k} \frac{\partial s^j}{\partial \xi^k} \,, \quad i, j, k = 1, \dots, n \,,$$

are the contravariant elements of a domain Ξ^n in coordinates $s^1, \dots, s^n$, and therefore functional (9.32) can be written in the following form:

$$I_{IS}[\boldsymbol{s}] = \frac{1}{2} \int\limits_{\Xi^2} \frac{1}{J\sqrt{1 + |\mathrm{grad} f(\boldsymbol{s})|^2}} \sum_{i,j=1}^{n} \bar{g}^{ij}_{s\xi} g^{s}_{ij} \mathrm{d}\boldsymbol{\xi}. \tag{9.33}$$

Functionals (9.20) and (9.23) in the metric of a monitor surface were originally published in Liseikin (1991). These functionals demonstrate a simple and efficient way for combining derivatives of monitor functions and intermediate transformations $\boldsymbol{s}(\boldsymbol{\xi})$ for generating adaptive grids with node clustering in the zones of large variations of the monitor functions.

Variable interpretations of the inverted functional (9.23) in the metric of a monitor surface were published in Liseikin (1993), in which the method for generating adaptive grids based on the minimization of the inverted energy functional in the metric of a monitor surface was named as a projection method, since the adaptive grid in a domain S^n is, in fact, the projection of a quasiuniform grid on a monitor surface S^{rn} over S^n. Later, this method was referred to in Ivanenko (1997) without consulting with its deviser as a method of harmonic adaptation. The authors of papers Garanzha (2000), Ivanenko (1997), Ivanenko and Charakch'yan (1988) made considerable contributions in the development of numerical algorithms (barrier methods) for solving the minimization problems for functionals (9.32) and (9.26) for $n = 2$ and $n = 3$ in the metric of a monitor surface. These algorithms were applied for generating adaptive grids with node clustering in the zones of shock waves and solving two-dimensional gas dynamics problems on such grids (see, for example, Azarenok and Ivanenko 2001).

The interpretations of the smoothness functional considered above justifies, to some extent, its potential to generate adaptive grids in a domain S^n in the metric of a monitor surface or on a surface S^{xn} by the minimization of the inverted functional of smoothness (see Fig. 9.4 for $n = 2$).

9.3.3 Euler–Lagrange Equations

The substitution of the parametric domain S^n for the integration hypersurface S^{xn} in (9.20) yields the smoothness functional in the following equivalent form with integration over S^n:

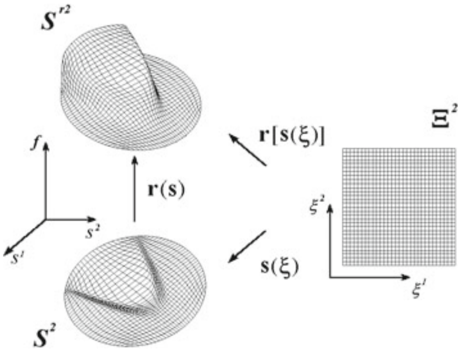


Fig. 9.4 Illustration for grid adaptation in a domain S^2

$$I_{\mathrm{S}} = \int\limits_{S^n} \sqrt{g^{xs}}\, g^{ml}_{sx} \frac{\partial \xi^i}{\partial s^m} \frac{\partial \xi^i}{\partial s^l} \mathrm{d}\boldsymbol{s} \ , \qquad i, m, l = 1, \ldots, n \ . \tag{9.34}$$

The quantities g^{xs} and g^{ml}_{sx} in (9.34) are defined through the specified parametrization $\boldsymbol{x}(\boldsymbol{s}) : S^n \to S^{xn}$, and therefore they remain unchanged when the functions $\xi^i(\boldsymbol{s})$ are varied. So, the Euler–Lagrange equations derived from the functional of smoothness (9.34) are readily obtained and, in accordance with (7.6), have the following divergent form:

$$\frac{\partial}{\partial s^m}\left(\sqrt{g^{xs}}\, g^{ml}_{sx} \frac{\partial \xi^i}{\partial s^l}\right) = 0 \ , \qquad i, m, l = 1, \ldots, n \ . \tag{9.35}$$

If S^{xn} is an n-dimensional domain S^n, then the system (9.35) is equivalent to the system of Laplace equations

$$\nabla^2 \xi^i \equiv \frac{\partial}{\partial s^j}\left(\frac{\partial \xi^i}{\partial s^j}\right) = 0 \ , \qquad i, j, = 1, \ldots, n \ ,$$

introduced by Crowley (1962) and Winslow (1967) for the generation of fixed grids in two-dimensional domains. Therefore, the method for generating grids on hypersurfaces S^{xn} by solving a boundary value problem for (9.35) derived from the functional of smoothness can also be considered as an extension of the Crowley–Winslow approach.

We considered in Chap. 6 the technique for generating adaptive grids in S^n which is based on the numerical solution of the Poisson system

$$\frac{\partial}{\partial s^j}\left(\frac{\partial \xi^i}{\partial s^j}\right) = P^i \ , \qquad i, j, = 1, \ldots, n \ , \tag{9.36}$$

where the P^i are the control functions. In the case of a monitor hypersurface S^{rn} represented by (9.4) or (9.5), the system (9.35) with the identification $g^{xs} = g^{rs}$ and $g^{ml}_{sx} = g^{ml}_{sr}$ can also be interpreted as a system of elliptic equations with a control function. The control function is the monitor mapping $\boldsymbol{f}(\boldsymbol{s})$ whose values over the physical domain or surface form the monitor hypersurface S^{rn}. The influence of the control function $\boldsymbol{f}(\boldsymbol{s})$ is realized through the magnitudes $\boldsymbol{f}_{s^i} \cdot \boldsymbol{f}_{s^j}$ in the terms g^{rs} and g^{ml}_{sr}. These terms are determined by the tensor elements g^{rs}_{ij} in the form (9.9) or (9.10) which define the covariant metric tensor of the hypersurface S^{rn} in the coordinates s^i represented by the parametrization (9.4) or (9.5). The system (9.35), in contrast to that of (9.36), has a divergent form and its solution is a harmonic function.

Notice that the Dirichlet problem for both the Eq. (9.35) with respect to the metric of a monitor hypersurface S^{rn} and (9.36) is well posed. Let us consider a twice differentiable function $\boldsymbol{\xi}_0(\boldsymbol{s}) : S^n \to \Xi^n$, $\boldsymbol{\xi}_0(\boldsymbol{s}) = (\xi^1_0(\boldsymbol{s}), \ldots, \xi^n_0(\boldsymbol{s}))$ that is a one-to-one transformation at the boundary points but that itself transforms two different interior points of S^n into one point in Ξ^n, that is, $\boldsymbol{\xi}_0(\boldsymbol{s})$ is not invertible. Then, assuming in (9.36)

$$P^i = \frac{\partial}{\partial s^j}\left(\frac{\partial \xi_0^i}{\partial s^j}\right) , \qquad i, j, = 1, \dots, n ,$$

we obtain that $\boldsymbol{\xi}_0(\boldsymbol{s})$ is the solution to (9.36), however, the inverted Poisson system with such P^i does not have a solution.

9.3.4 Equivalent Forms

In analogy with two-dimensional space, the left-hand part of (9.35) multiplied by $1/\sqrt{g^{xs}}$ is the value of the Beltrami operator

$$\Delta_B = \frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^m}(\sqrt{g^{xs}}\, g^{ml}_{sx}\, \frac{\partial}{\partial s^l}) , \qquad l, m = 1, \dots, n, \tag{9.37}$$

applied to the function $\xi^i(x)$. Similarly, as in Chap. 8, it is proved that this operator is invariant of parametrizations for arbitrary twice-differentiable function $\boldsymbol{g}(\boldsymbol{s})$ and another coordinate system v^i, $i = 1, \dots, n$, in S^{xn}, i.e.,

$$\frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^m}(\sqrt{g^{xs}}\, g^{ml}_{sx}\, \frac{\partial}{\partial s^l}\boldsymbol{g}(\boldsymbol{s}))(\boldsymbol{v}) = \frac{1}{\sqrt{g^{xv}}}\frac{\partial}{\partial v^m}(\sqrt{g^{xv}}\, g^{ml}_{vx}\, \frac{\partial}{\partial v^l}\boldsymbol{g}(\boldsymbol{s}(\boldsymbol{v}))) ,$$
$$l, m = 1, \dots, n .$$

In particular, we get

$$\Delta_B[\xi^i] = \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial \xi^m}(\sqrt{g^{x\xi}}\, g^{mi}_{\xi x}) , \qquad i, m = 1, \dots, n ,$$

thus we find that the system of Euler–Lagrange equations (9.35) for the generation of quasiuniform grids on hypersurfaces is equivalent to the following system:

$$\Delta_B[\xi^i] \equiv \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial \xi^j}(\sqrt{g^{x\xi}}\, g^{jl}_{\xi x}) = 0 , \qquad i, j, l = 1, \dots, n. \tag{9.38}$$

Now we obtain other forms of (9.38). For this purpose, we first compute the value of $\Delta_B[\boldsymbol{x}]$, where the operator Δ_B is defined by (9.37). Expanding the differentiation in $\Delta_B[\boldsymbol{x}]$, we have

$$\Delta_B[\boldsymbol{x}] = \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial \xi^j}(\sqrt{g^{x\xi}}\, g^{jl}_{\xi x}\boldsymbol{x}_{\xi^l}) = g^{jl}_{\xi x}\boldsymbol{x}_{\xi^j \xi^l} + \Delta_B[\xi^l]\boldsymbol{x}_{\xi^l} ,$$
$$j, l = 1, \dots, n , \tag{9.39}$$

where

$$\boldsymbol{x}_{\xi^i\xi^j} = \frac{\partial^2 \boldsymbol{x}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^i \partial \xi^j}\ , \qquad \boldsymbol{x}_{\xi^l} = \frac{\partial \boldsymbol{x}(\boldsymbol{\xi})}{\partial \xi^l}\ , \qquad i,j,l = 1,\dots,n\ .$$

Using the expansion (9.19) with the assumption $\xi^i = s^i,\ i = 1,\dots,n$, we obtain

$$\boldsymbol{x}_{\xi^i\xi^j} = \Upsilon^l_{ij}\boldsymbol{x}_{\xi^l} + \boldsymbol{d}_{ij}\ , \qquad i,j,l = 1,\dots,n\ ,$$

where

$$\Upsilon^l_{ij} = g^{lm}_{\xi x}[ij,m]\ , \qquad [ij,m] = \boldsymbol{x}_{\xi^i\xi^j}\cdot\boldsymbol{x}_{\xi^m}\ , \qquad i,j,l,m = 1,\dots,n\ , \tag{9.40}$$

and the $\boldsymbol{d}_{ij}$ are the vectors orthogonal to the tangent n-dimensional hyperplane defined by the vectors $\boldsymbol{x}_{\xi^i},\ i = 1,\dots,n$. From the above expansion of $\boldsymbol{x}_{\xi^i\xi^j}$, we have

$$g^{jl}_{\xi x}\boldsymbol{x}_{\xi^j\xi^l} = g^{jl}_{\xi x}\Upsilon^m_{jl}\boldsymbol{x}_{\xi^m} + g^{jl}_{\xi x}\boldsymbol{d}_{jl}\ , \qquad j,l,m = 1,\dots,n\ .$$

Substitution of these identities in (9.39) yields

$$\Delta_B[\boldsymbol{x}] = (g^{jl}_{\xi x}\Upsilon^m_{jl} + \Delta_B[\xi^m])\boldsymbol{x}_{\xi^m} + g^{jl}_{\xi x}\boldsymbol{d}_{jl}\ , \qquad j,l,m = 1,\dots,n\ . \tag{9.41}$$

Now we show that the vector $\Delta_B[\boldsymbol{x}]$, as well as the vectors $\boldsymbol{d}_{ij}$, lies in the k-dimensional hyperplane which is orthogonal to the tangent hyperplane, i.e.,

$$\Delta_B[\boldsymbol{x}]\cdot\boldsymbol{x}_{\xi^i} = 0 \qquad \text{for all} \qquad i = 1,\dots,n\ .$$

Indeed, we have

$$\begin{aligned}\Delta_B[\boldsymbol{x}]\cdot\boldsymbol{x}_{\xi^i} &= \frac{1}{\sqrt{g^{x\xi}}}\Big(\frac{\partial}{\partial\xi^j}\sqrt{g^{x\xi}}g^{jm}_{\xi x}\boldsymbol{x}_{\xi^m}\Big)\cdot\boldsymbol{x}_{\xi^i}\\ &= \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial\xi^j}(\sqrt{g^{x\xi}}g^{jm}_{\xi x}\boldsymbol{x}_{\xi^m}\cdot\boldsymbol{x}_{\xi^i}) - g^{jm}_{\xi x}\boldsymbol{x}_{\xi^m}\cdot\boldsymbol{x}_{\xi^i\xi^j}\\ &= \frac{1}{\sqrt{g^{x\xi}}}\frac{\partial}{\partial\xi^i}\sqrt{g^{x\xi}} - \Upsilon^j_{ji}, \qquad i,j,m = 1,\dots,n\ .\end{aligned} \tag{9.42}$$

Now, using the identity (9.18) (valid for arbitrary parametrization) in the coordinates ξ^i, we obtain

$$\Delta_B[\boldsymbol{x}]\cdot\boldsymbol{x}_{\xi^i} = 0\ , \qquad i = 1,\dots,n\ ,$$

from (9.42). Therefore, the coefficients before $\boldsymbol{x}_{\xi^m}$ in (9.41) are equal to zero, i.e., we have the identity

$$\Delta_B[\xi^m] = -g^{jl}_{\xi x}\Upsilon^m_{jl}\ , \qquad j,l,m = 1,\dots,n\ . \tag{9.43}$$

Thus, (9.41) becomes

$$\Delta_B[\boldsymbol{x}] = g^{jl}_{\xi x}\boldsymbol{d}^{jl} \ , \qquad j, l = 1, \dots, n \ . \tag{9.44}$$

Note that (9.44) is an extension of (6.32) and (8.49) to general hypersurfaces.

From (9.40), the identity (9.43) also has the form

$$\begin{aligned} \Delta_B[\xi^m] = -g^{jl}_{\xi x}\, g^{mi}_{\xi x}[jl, i] = \ & -g^{jl}_{\xi x}\, g^{mi}_{\xi x}(\boldsymbol{x}_{\xi^j\xi^l} \cdot \boldsymbol{x}_{\xi^i}) \ , \\ & i, j, l, m = 1, \dots, n \ . \end{aligned} \tag{9.45}$$

Analogously, we have, assuming $s^i = \xi^i, \ i = 1, \dots, n$, in (9.45),

$$\Delta_B[s^m] = -g^{jl}_{sx}\, g^{mi}_{sx}(\boldsymbol{x}_{s^j s^l} \cdot \boldsymbol{x}_{s^i}) \ , \qquad i, j, l, m = 1, \dots, n \ . \tag{9.46}$$

The identities (9.43) and (9.45) represent an extension of the identities (8.52) valid for two-dimensional surfaces.

Since the Beltrami system (9.38) is equivalent to (9.35), we obtain, from (9.45), one more system of equations:

$$g^{jl}_{\xi x}\, g^{mi}_{\xi x}[jl, i] = 0 \ , \qquad i, j, l, m = 1, \dots, n \ ,$$

which is equivalent to the Euler–Lagrange equations (9.35). Multiplication of this system by $g^{x\xi}_{mp}$ and summation over m yields one more equivalent system:

$$g^{jl}_{\xi x}[jl, p] = 0 \ , \qquad j, l, p = 1, \dots, n \ ,$$

namely,

$$g^{jl}_{\xi x}\frac{\partial^2 \boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^j \partial \xi^l} \cdot \frac{\partial \boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^p} = 0 \ , \qquad j, l, p = 1, \dots, n \ . \tag{9.47}$$

In particular, if S^{rn} is a monitor surface S^{xn} over a domain S^n identified with a physical domain X^n and represented by (9.4), then the system (9.47) is

$$g^{jl}_{\xi r}\left\{\frac{\partial^2 \boldsymbol{s}}{\partial \xi^j \partial \xi^l} \cdot \frac{\partial \boldsymbol{s}}{\partial \xi^p} + \frac{\partial^2 \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^j \partial \xi^l} \cdot \frac{\partial \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^p}\right\} = 0 \ , \quad j, l, p = 1, \dots, n \ . \tag{9.48}$$

This system of equations can be used to generate adaptive grids on S^{xn} and nearly orthogonal grids in the vicinity of boundary segments of a domain X^n by specifying the monitor hypersurface with a suitable choice of a monitor function $\boldsymbol{f}(\boldsymbol{x})$.

9.3.5 Inverted Beltrami Equations

The components $s^i(\boldsymbol{\xi})$ of the function $\boldsymbol{s}(\boldsymbol{\xi})$, which is the inverse of $\boldsymbol{\xi}(\boldsymbol{s})$, are to satisfy the inverse system of (9.35). This system is obtained first by multiplying each ith component of system (9.35) by the derivative $\partial s^m/\partial \xi^i$ and then by summing the result over i. This operation produces the following system:

$$\begin{aligned}&\frac{\partial}{\partial s^p}\Big(\sqrt{g^{xs}}\, g^{pl}_{sx}\, \frac{\partial \xi^i}{\partial s^l}\Big)\frac{\partial s^m}{\partial \xi^i}\\ &= \frac{\partial}{\partial s^p}\Big(\sqrt{g^{xs}}\, g^{pl}_{sx}\, \frac{\partial \xi^i}{\partial s^l}\frac{\partial s^m}{\partial \xi^i}\Big) - \sqrt{g^{xs}}\, g^{pl}_{sx}\, \frac{\partial \xi^i}{\partial s^l}\frac{\partial \xi^j}{\partial s^p}\frac{\partial^2 s^m}{\partial \xi^i \partial \xi^j} = 0\,,\end{aligned}$$

$$i, j, l, m, p = 1, \dots, n\,.$$

By multiplying this system of equations by $1/\sqrt{g^{xs}}$ and taking into account (9.15) and the relation

$$\frac{\partial \xi^i}{\partial s^l}\frac{\partial s^m}{\partial \xi^i} = \delta^l_m\,, \qquad i, l, m = 1, \dots, n\,,$$

the system of the inverse equations with s^i as dependent and ξ^i as independent variables has the form

$$g^{ip}_{\xi x}\frac{\partial^2 s^m}{\partial \xi^i \partial \xi^p} = \frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^l}\Big(\sqrt{g^{xs}}\, g^{lm}_{sx}\Big)\,, \qquad i, l, m, p = 1, \dots, n\,. \tag{9.49}$$

Notice that in contrast to Beltrami equations (9.35) the inverse equations (9.49) are not divergent for $n \geq 2$. However, for $n = 1$, the inverse Beltrami equation is also equivalent to the equation in a divergent form:

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\Big(\sqrt{g^{xs}}\frac{\mathrm{d}s}{\mathrm{d}\xi}\Big) = 0\,. \tag{9.50}$$

System (9.49) is a system of quasilinear equations and, an important point for the creation of iterative numerical algorithms, its right-hand part is defined only by the tensor elements g^{xs}_{ij} of the hypersurface S^{xn} and, therefore, remains unchanged when the function $\boldsymbol{s}(\boldsymbol{\xi})$ is varied. Moreover, each ith equation of the right-hand part of the system (9.49) is the value of the Beltrami operator applied to the function s^i. Thus, the system (9.49) can be written in the following equivalent form:

$$g^{ij}_{\xi x}\frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} = \Delta_B[s^l]\,, \qquad i, j, l = 1, \dots, n\,. \tag{9.51}$$

As in (6.18), introducing an operator

$$B_n^{x\xi}[y] \equiv g^{x\xi} g_{\xi x}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j} \ , \qquad i, j = 1, \dots, n \ , \tag{9.52}$$

gives the system (9.51) another form

$$B_n^{x\xi}[s^l] = g^{x\xi} \Delta_B[s^l] \ , \qquad l = 1, \dots, n \ . \tag{9.53}$$

The coefficients $g^{x\xi} g_{\xi x}^{ij}$ of the operator $B_n^{x\xi}$ for $n > 1$ are expressed by the elements of the covariant metric tensor $\{g_{ij}^{x\xi}\}$, in particular, for $n = 2$,

$$g^{x\xi} g_{\xi x}^{ij} = (-1)^{i+j} g_{3-i\,3-j}^{x\xi}, \quad i, j \text{ fixed},$$

and for $n = 3$,

$$g^{x\xi} g_{\xi x}^{ij} = g_{i+1\,j+1}^{x\xi} g_{i+2\,j+2}^{x\xi} - g_{i+1\,j+2}^{x\xi} g_{i+2\,j+1}^{x\xi}, \quad i, j = 1, 2, 3, \quad i, j \text{ fixed},$$

with the convention for the latter formula that any index, say, k, is identified with $k \pm 3$, so, for instance, $g_{45}^{x\xi} = g_{12}^{x\xi}$. Thus, for $n = 1$, 2, and 3, we have

$$\begin{aligned}
&B_1^{x\xi}[y] = \frac{\partial^2 y}{\partial \xi \partial \xi}, \\
&B_2^{x\xi}[y] = g_{22}^{x\xi} \frac{\partial^2 y}{\partial \xi^1 \partial \xi^2} - 2 g_{12}^{x\xi} \frac{\partial^2 y}{\partial \xi^1 \partial \xi^2} + g_{11}^{x\xi} \frac{\partial^2 y}{\partial \xi^2 \partial \xi^2} \ , \\
&B_3^{x\xi}[y] = [g_{22}^{x\xi} g_{33}^{x\xi} - (g_{23}^{x\xi})^2] \frac{\partial^2 y}{\partial \xi^1 \partial \xi^1} + [g_{11}^{x\xi} g_{33}^{x\xi} - (g_{13}^{x\xi})^2] \frac{\partial^2 y}{\partial \xi^2 \partial \xi^2} \\
&+ [g_{11}^{x\xi} g_{22}^{x\xi} - (g_{12}^{x\xi})^2] \frac{\partial^2 y}{\partial \xi^3 \partial \xi^3} + 2[g_{23}^{x\xi} g_{13}^{x\xi} - g_{12}^{x\xi} g_{33}^{x\xi}] \frac{\partial^2 y}{\partial \xi^1 \partial \xi^2} \\
&+ 2[g_{12}^{x\xi} g_{23}^{x\xi} - g_{22}^{x\xi} g_{13}^{x\xi}] \frac{\partial^2 y}{\partial \xi^1 \partial \xi^3} + 2[g_{13}^{x\xi} g_{12}^{x\xi} - g_{23}^{x\xi} g_{11}^{x\xi}] \frac{\partial^2 y}{\partial \xi^2 \partial \xi^3}.
\end{aligned} \tag{9.54}$$

Also, taking advantage of (9.46), we obtain, from (9.53), one more system of equations equivalent to (9.49):

$$B_n^{x\xi}[s^l] = -g^{x\xi} g_{sx}^{jm} \ g_{sx}^{li}[jm, i] \ , \qquad i, j, l, m = 1, \dots, n \ , \tag{9.55}$$

where $[jm, i] = \boldsymbol{x}_{s^j s^m} \cdot \boldsymbol{x}_{s^i}$. Thus, (9.55) is a generalization of (8.57).

If S^{xn} is a monitor surface S^{rn} over S^n, represented by the parametrization (9.4), then multiplying (9.48) by $\partial \xi^p / \partial s^i$ yields

$$g_{\xi r}^{jl} \Big(\frac{\partial^2 s^i}{\partial \xi^j \partial \xi^l} + \frac{\partial^2 \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^j \partial \xi^l} \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s})}{\partial s^i} \Big) = 0 \ , \qquad i, j, l = 1, \dots, n \ , \tag{9.56}$$

or introducing an operator $B_n^{r\xi}$, similar to (9.52),

$$B_n^{r\xi}[y] = g^{r\xi} g_{\xi r}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j} , \qquad i, j = 1, \ldots, n , \tag{9.57}$$

the system (9.56) is expressed as

$$B_n^{r\xi}[s^i] + B_n^{r\xi}[f] \cdot \frac{\partial \boldsymbol{f}}{\partial s^i} = 0 , \qquad i = 1, \ldots, n . \tag{9.58}$$

9.4 Role of the Mean Curvature

This section shows how the mean curvature, which is one of the abstract geometric characteristics reviewed in Chap. 3, is involved in grid generation technologies.

9.4.1 Mean Curvature and Inverted Beltrami Grid Equations

General Formulas

In the case of a regular surface $S^{xn} \subset R^{n+1}$ defined by the parametrization

$$\mathbf{x}(\mathbf{s}) : S^n \to R^{n+1} , \quad \mathbf{x}(\mathbf{s}) = [x^1(\mathbf{s}), \ldots, x^{n+1}(\mathbf{s})] ,$$

with its natural metric elements g_{ij}^{xs} specified in the coordinates $s^1, \ldots, s^n$ as

$$g_{ij}^{xs} = \mathbf{x}_{s^i} \cdot \mathbf{x}_{s^j} , \quad i, j = 1, \ldots, n ,$$

we have, from (9.37) and (9.43), assuming $\xi^i = s^i$,

$$\Delta_B[s^i] = -g_{sx}^{kj} \Upsilon_{kj}^i = -g_{sx}^{kj} g_{sx}^{im} (\mathbf{x}_{s^k s^j} \cdot \mathbf{x}_{s^m}) , \quad i, j, k, m = 1, \ldots, n . \tag{9.59}$$

Now let us remind ourselves that the quantity

$$K_m = \frac{1}{n} g_{sx}^{kj} \mathbf{x}_{s^k s^j} \cdot \mathbf{n} , \quad j, k = 1, \ldots, n , \tag{9.60}$$

where $\mathbf{n}$ is an $(n+1)$-dimensional unit normal vector to S^{xn} in R^{n+1}, is the mean curvature of the surface S^{xn} with respect to this normal $\mathbf{n}$.

Note the mean curvature with respect to the same normal is an invariant of parametrizations of S^{xn}.

It appears that the mean curvature of S^{xn} is connected with the parametrization $\mathbf{x}(\mathbf{s}) : S^n \to S^{xn} \subset R^{n+1}$ by the following relation:

$$\Delta_B[\mathbf{x}] = nK_m\mathbf{n} \ , \tag{9.61}$$

where $\mathbf{n}$ is the same vector used in (9.60). Indeed, we have

$$\Delta_B[\mathbf{x}] = \frac{1}{\sqrt{g^{xs}}}\frac{\partial}{\partial s^j}(\sqrt{g^{xs}}g^{ji}_{\mathbf{s}}\mathbf{x}_{s^i}) = g^{ji}_{sx}\mathbf{x}_{s^j s^i} + \Delta_B[s^k]\mathbf{x}_{s^k} \ , \qquad i,j,k = 1,\dots,n \ , \tag{9.62}$$

and applying the relation (9.59) to the last item of (9.62), we obtain

$$\Delta_B[\mathbf{x}] = g^{ji}_{sx}[\mathbf{x}_{s^j s^i} - g^{mk}_{sx}(\mathbf{x}_{s^j s^i}\cdot\mathbf{x}_{s^m})\mathbf{x}_{s^k}] \ , \quad i,j,k,m = 1,\dots,n \ . \tag{9.63}$$

Now, taking into account equation (2.6), yielding that the expansion of the vector $\mathbf{x}_{s^i s^j}$ in the basis $(\mathbf{x}_{s^1},\dots,\mathbf{x}_{s^n},\mathbf{n})$ is expressed as

$$\mathbf{x}_{s^i s^j} = g^{mk}_{sx}(\mathbf{x}_{s^i s^j}\cdot\mathbf{x}_{s^m})\mathbf{x}_{s^k} + (\mathbf{x}_{s^i s^j}\cdot\mathbf{n})\mathbf{n} \ ,$$

we find, applying this expansion to (9.63) and using (9.60),

$$\Delta_B[\mathbf{x}] = g^{ji}_{sx}(\mathbf{x}_{s^j s^i}\cdot\mathbf{n})\mathbf{n} = nK_m\mathbf{n} \ , \quad i,j = 1,\dots,n \ ,$$

i.e., Eq. (9.61) is valid. Consequently, from the relation (9.62), we obtain

$$g^{ji}_{sx}\mathbf{x}_{s^j s^i} + \Delta_B[s^k]\mathbf{x}_{s^k} = nK_m\mathbf{n} \ , \quad i,j,k = 1,\dots,n \ . \tag{9.64}$$

This formula is valid in arbitrary coordinates, in particular, in the grid coordinates $\xi^1,\dots,\xi^n$ satisfying the Euler–Lagrange equations (9.35), and consequently the Beltrami equations (9.38).

Application to a Monitor Surface over a Domain

Let the parametric transformation $\mathbf{r}(\mathbf{s})$ for a monitor surface S^{rn} be specified as $\mathbf{r}(\mathbf{s}) = [\mathbf{s}, f(\mathbf{s})]$ with a scalar-valued monitor function $f(\mathbf{s})$. Then, $S^{rn} \subset R^{n+1}$ is a monitor surface over S^n, whose covariant metric elements in the coordinates $s^1,\dots,s^n$ are computed as

$$g^{rs}_{ij} = \mathbf{r}_{s^i}\cdot\mathbf{r}_{s^j} = \delta^i_j + f_{s^i}f_{s^j} \ , \quad i,j = 1,\dots,n \ , \tag{9.65}$$

and therefore

$$g^{rs} = \det\{g^{rs}_{ij}\} = 1 + f_{s^m}f_{s^m}, \quad m = 1,\dots,n \ .$$

Consequently, we have for the contravariant metric elements g^{ij}_{sr} the following formula:

$$g^{ij}_{sr} = \delta^i_j - \frac{1}{g^{rs}}f_{s^i}f_{s^j} \ , \quad i,j = 1,\dots,n \ . \tag{9.66}$$

Thus, we find

$$g^{im}_{sr} f_{s^m} = \Big(\delta^i_m - \frac{1}{g^{rs}} f_{s^i} f_{s^m}\Big) f_{s^m} = \Big(1 - \frac{1}{g^{rs}} f_{s^m} f_{s^m}\Big) f_{s^i} = \frac{1}{g^{rs}} f_{s^i} \ , \quad i, m = 1, \dots, n \ . \tag{9.67}$$

So, using (9.43), with the identification $s^i = \xi^i$, we conclude that in the metric (9.65)

$$\Delta_B[s^i] = -g^{kj}_{sr} g^{im}_{sr} f_{s^k s^j} f_{s^m} = -\frac{1}{g^{rs}} g^{kj}_{sr} f_{s^k s^j} f_{s^i} \ , \ i, j, k, m = 1, \dots, n \ . \tag{9.68}$$

For the parametrization $\mathbf{r}(\mathbf{s}) = [\mathbf{s}, f(\mathbf{s})]$, we find

$$\mathbf{r}_{s^i} = (\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{n-i}, f_{s^i}) \ , \quad i = 1, \dots, n \ ,$$

so it is obvious that for the unit normal $\mathbf{n}$, we can take

$$\mathbf{n} = \frac{1}{\sqrt{g^{rs}}}(-f_{s^1}, \dots, -f_{s^n}, 1) \ . \tag{9.69}$$

For the above expressions for $\mathbf{r}(\mathbf{s})$ and $\mathbf{n}$,

$$\mathbf{r}_{s^k s^j} \cdot \mathbf{n} = \frac{1}{\sqrt{g^{rs}}} f_{s^k s^j} \ , \quad j, k = 1, \dots, n \ ,$$

so, in accordance with (9.60), the mean curvature of this monitor surface S^{rn} with respect to the normal (9.69) is computed by the following formula:

$$K_m = \frac{1}{n\sqrt{g^{rs}}} g^{kj}_{sr} f_{s^k s^j} \ , \quad j, k = 1, \dots, n \ . \tag{9.70}$$

Thus, in the metric (9.65), Eq. (9.68) have the following form:

$$\Delta_B[s^i] = -\frac{n}{\sqrt{g^{rs}}} K_m f_{s^i} \ , \quad i = 1, \dots, n \ . \tag{9.71}$$

Consequently, the inverted Beltrami grid equations (9.51) in application to a domain $X^n = S^n$ with a scalar monitor function $f(\mathbf{x})$ are expressed through the mean curvature of S^{rn} with respect to the normal (9.69) as follows:

$$g^{ij}_{\xi r} \frac{\partial^2 s^k}{\partial \xi^i \partial \xi^j} = -\frac{n}{\sqrt{g^{rs}}} K_m f_{s^k} \ , \quad i, j, k = 1, \dots, n \ . \tag{9.72}$$

So, if $K_m = 0$, i.e., the monitor surface S^{rn} is a minimal surface, then Eq. (9.72) have a simple form (with the zero right-hand part). Some results related to the construction of minimal n-dimensional surfaces, providing such equations, can be found in the monograph by Fomenko and Thi (1991).

Notice that another form for the expression of the mean curvature of the monitor surface with the monitor metric (9.65) can be computed using the $(n+1)$th component of (9.61). Thus, we have, with respect to the normal (9.69),

$$K_m = \frac{1}{n}\sqrt{g^{rs}}\Delta_B[f]\ . \tag{9.73}$$

Availing us of (9.67), we find, in the case of the parametrization $\mathbf{r}(\mathbf{s}) = [\mathbf{s}, f(\mathbf{s})]$,

$$\Delta_B[f] = \frac{1}{\sqrt{g^{rs}}}\frac{\partial}{\partial s^j}\Big(\sqrt{g^{rs}}g^{jk}_{sr}\frac{\partial f}{\partial s^k}\Big) = \frac{1}{\sqrt{g^{rs}}}\frac{\partial}{\partial s^j}\Big(\frac{1}{\sqrt{g^{rs}}}\frac{\partial f}{\partial s^j}\Big)\ ,\ \ j,k = 1,\dots,n\ .$$

Thus, from (9.73),

$$K_m = \frac{1}{n}\frac{\partial}{\partial s^j}\Big(\frac{1}{\sqrt{g^{rs}}}\frac{\partial f}{\partial s^j}\Big)\ ,\quad j = 1,\dots,n\ , \tag{9.74}$$

with respect to the normal (9.69).

Notice that there is a formula of the mean curvature for the n-dimensional hypersurface in R^{n+1} defined by the equation $\phi(\boldsymbol{s}) = 0$, $\boldsymbol{s} = (s^1,\dots,s^{n+1})$, namely,

$$K_m = -\frac{1}{n}\frac{\partial}{\partial s^j}\Big(\frac{\phi_{s^j}}{\sqrt{\nabla(\phi)}}\Big)\ ,\quad j = 1,\dots,n+1\ , \tag{9.75}$$

where $\nabla(\phi) = \phi_{s^i}\phi_{s^i}$, $i = 1,\dots,n+1$. Since the points of the monitor surface $S^{rn} \subset R^{n+1}$ in the coordinates $s^1,\dots,s^n, s^{n+1}$ can be found from the solution of the equation

$$s^{n+1} - f(s^1,\dots,s^n) = 0\ ,$$

the formula (9.74) is a particular case of (9.75) for

$$\varphi(s^1,\dots,s^n,s^{n+1}) \equiv s^{n+1} - f(s^1,\dots,s^n)\ .$$

9.4.2 Mean Curvature and Rate of Grid Clustering

Fundamental Formula

It is well-known that when S^{xn} is a domain with the Euclidean metric tensor, then the operator of Beltrami in this metric is the Laplace operator and the spacing between $(n-1)$-dimensional grid hypersurfaces $\xi^i = \text{const}$ in S^{xn} related to the solution of the Laplace equations, for both $n = 2$ and $n = 3$, increases near a convex boundary segment and, conversely, the spacing decreases when the boundary segment is con-

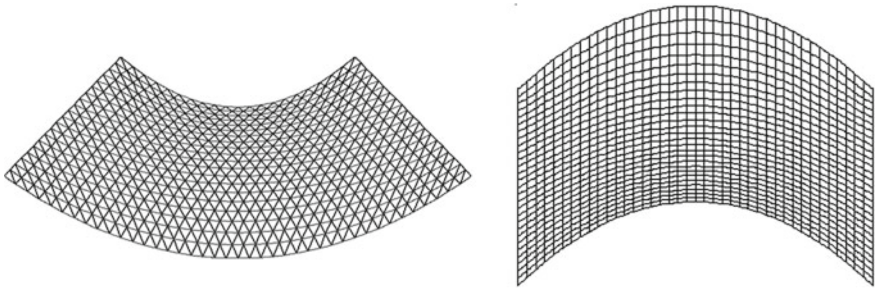


Fig. 9.5 Grid rarefaction (clustering) near convex (concave) boundary segments

cave (see Fig. 9.5 for $n = 2$). It is shown further that similar facts are also valid for the grid hypersurfaces related to the solution of the Beltrami equations in arbitrary n-dimensional regular surfaces S^{xn}.

Throughout this subsection, we assume i_0, $1 \le i_0 \le n$ as a fixed index, i.e., the summation in the formulas with i_0 repeated is not carried out over this index.

Relative Spacing Between Coordinate Surfaces

Let $s^1, \dots, s^n$ be an arbitrary local coordinate system of an n-dimensional regular surface S^{xn} represented by (9.1). We first consider in the surface S^{xn} a family of the coordinate $(n-1)$-dimensional hypersurfaces $s^{i_0} = const$. We can readily find that the vector $\mathbf{n}_{i_0}$ lying in the tangent n-dimensional plane to S^{xn}, and which is expressed in the form

$$\mathbf{n}_{i_0} = \frac{1}{\sqrt{g_{sx}^{i_0 i_0}}} g_{sx}^{i_0 j} \mathbf{x}_{s^j} \ , \quad j = 1, \dots, n \ , \tag{9.76}$$

is a unit normal to the coordinate hypersurface $s^{i_0} = c_0$. Here, g_{sx}^{ij} is the (ij)th element of the contravariant metric tensor of S^{xn} in the coordinates $s^1, \dots, s^n$, and the values of g_{sx}^{ij} and $\mathbf{x}_{s^j}$ are considered at the points of S^n for which $s^{i_0} = c_0$. Indeed,

$$\mathbf{n}_{i_0} \cdot \mathbf{x}_{s^k} = \frac{1}{\sqrt{g_{sx}^{i_0 i_0}}} g_{sx}^{i_0 j} \mathbf{x}_{s^j} \cdot \mathbf{x}_{s^k} = \frac{1}{\sqrt{g_{sx}^{i_0 i_0}}} g_{sx}^{i_0 j} g_{jk}^{xs} = \frac{1}{\sqrt{g_{sx}^{i_0 i_0}}} \delta_k^{i_0} \ ,$$
$$j, k = 1, \dots, n \ ,$$

i.e., $\mathbf{n}_{i_0}$ is orthogonal to the coordinate surface $s^{i_0} = c_0$ in S^{xn}. Furthermore,

$$\begin{aligned} \mathbf{n}_{i_0} \cdot \mathbf{n}_{i_0} &= \frac{1}{\sqrt{g_{sx}^{i_0 i_0}}} g_{sx}^{i_0 j} \mathbf{x}_{s^j} \cdot \frac{1}{\sqrt{g_{sx}^{i_0 i_0}}} g_{sx}^{i_0 k} \mathbf{x}_{s^k} \\ &= \frac{1}{g_{sx}^{i_0 i_0}} g_{sx}^{i_0 j} g_{sx}^{i_0 k} g_{jk}^{xs} = 1 \ , \quad j, k = 1, \dots, n \ , \end{aligned}$$

i.e., $\mathbf{n}_{i_0}$ is a unit vector.

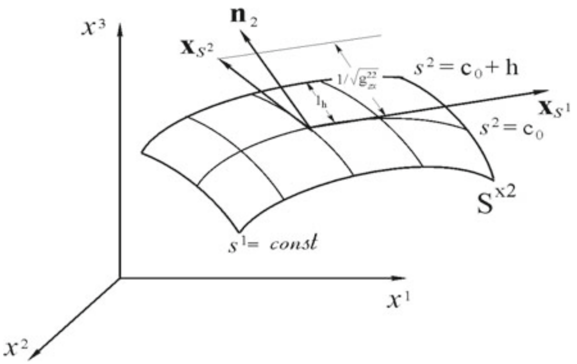


Fig. 9.6 Spacing of the coordinate lines $s^2 = const$ on the regular surface

Let us denote by l_h the distance between a point on the hypersurface $s^{i_0} = c_0$ and the nearest point on the hypersurface $s^{i_0} = c_0 + h$ in S^{xn}. We have

$$\begin{aligned} l_h &= (\mathbf{n}_{i_0} \cdot \mathbf{x}_{s^{i_0}})h + O(h^2) = h\frac{1}{\sqrt{g^{i_0 i_0}_{sx}}} g^{i_0 j}_{sx} \mathbf{x}_{s^j} \cdot \mathbf{x}_{s^{i_0}} + O(h^2) \\ &= h\frac{1}{\sqrt{g^{i_0 i_0}_{sx}}} g^{i_0 j}_{sx} g^{xs}_{j i_0} + O(h^2) = h\frac{1}{\sqrt{g^{i_0 i_0}_{sx}}} + O(h^2) \;, \quad j = 1, \dots, n \;. \end{aligned}$$

So, the quantity $1/\sqrt{g^{i_0 i_0}_{sx}}$ with i_0 fixed reflects the relative spacing between the corresponding points on the coordinate hypersurfaces $s^{i_0} = c_0 + h$ and $s^i = c_0$ on S^{xn} (see Fig. 9.6 for $n = 2$).

Rate of Change of the Relative Spacing

The vector $\mathbf{n}_{i_0}$ is orthogonal to the coordinate hypersurface $s^{i_0} = c_0$ in S^{xn}, and therefore the derivative of $1/\sqrt{g^{i_0 i_0}_{sx}}$ in the $\mathbf{n}_{i_0}$ direction is the rate of change, designated as v_{i_0} of the relative spacing between the coordinate hypersurfaces $s^{i_0} = const$. Using (9.76), we obtain

$$\begin{aligned} v_{i_0} &= \frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_{i_0}}\Bigl(\frac{1}{\sqrt{g^{i_0 i_0}_{sx}}}\Bigr) = \frac{1}{\sqrt{g^{i_0 i_0}_{sx}}} g^{i_0 j}_{sx} \frac{\partial}{\partial s^j}\Bigl(\frac{1}{\sqrt{g^{i_0 i_0}_{sx}}}\Bigr) \\ &= \frac{1}{\sqrt{g^{i_0 i_0}_{sx}}} \nabla\Bigl(s^{i_0}, \frac{1}{\sqrt{g^{i_0 i_0}_{sx}}}\Bigr) \;, \quad j = 1, \dots, n \;, \end{aligned} \tag{9.77}$$

where $\nabla(\ ,\ )$ is the Beltrami mixed differential parameter, i.e.,

$$\nabla(f, g) = g^{ij}_{sx} f_{s^i} g_{s^j} \;, \quad i, j = 1, \dots, n \;.$$

On the other hand,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_{i_0}}\Big(\frac{1}{\sqrt{g^{i_0i_0}_{sx}}}\Big) &= -\frac{1}{2\sqrt{(g^{i_0i_0}_{sx})^3}}\frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_{i_0}}g^{i_0i_0}_{sx} \\ &= -\frac{1}{2(g^{i_0i_0}_{sx})^2}g^{i_0j}_{sx}\frac{\partial}{\partial s^j}g^{i_0i_0}_{sx}\,, \quad j=1,\dots,n\,.\end{aligned} \tag{9.78}$$

Availing us of the relations

$$\begin{aligned}\frac{\partial g^{ij}_{sx}}{\partial s^k} &= g^{lj}_{sx}g^{xs}_{ml}\frac{\partial g^{im}_{sx}}{\partial s^k} = -g^{lj}_{sx}g^{im}_{sx}\frac{\partial g^{xs}_{ml}}{\partial s^k} \\ &= -g^{lj}_{sx}g^{im}_{sx}([mk,l]+[lk,m]) = -g^{im}_{sx}\Upsilon^j_{km} - g^{lj}_{sx}\Upsilon^i_{lk}\,, \\ &\quad i,j,k,l,m=1,\dots,n\,.\end{aligned} \tag{9.79}$$

in Eq. (9.78) yields

$$v_{i_0} = \frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_{i_0}}\Big(\frac{1}{\sqrt{g^{i_0i_0}_{sx}}}\Big) = \frac{1}{(g^{i_0i_0}_{sx})^2}g^{i_0l}_{sx}g^{i_0j}_{sx}\Upsilon^{i_0}_{lj}\,, \quad j,l=1,\dots,n\,. \tag{9.80}$$

Note that this equation is valid for an arbitrary coordinate system $s^1,\dots,s^n$.

Relations to the Mean Curvature

In order to connect the rate of change of the relative spacing of the coordinate hypersurfaces with geometrical characteristics, we need to consider the following general situation in the theory of matrices. Let $\{a_{ij}\}$, $i,j=1,\dots,n$, $n\geq 2$, be a nondegenerate symmetric matrix of rank n and $\{a^{ij}\}$, $i,j=1,\dots,n$, be its inverse matrix. Let $a^{i_0i_0}\neq 0$ for some fixed index i_0, $1\leq i_0\leq n$. Define a matrix $\{b^{ij}\}$ where

$$b^{ij} = \frac{1}{a^{i_0i_0}}(a^{i_0i_0}a^{ij} - a^{i_0i}a^{i_0j})\,, \quad i,j=1,\dots,n\,. \tag{9.81}$$

Let $\{a^{i_0}_{ij}\}$ and $\{b^{ij}_{i_0}\}$ be the $(n-1)\times(n-1)$ matrices obtained by deleting the i_0th row and i_0th column of the matrices $\{a_{ij}\}$ and $\{b^{ij}\}$, respectively.

Theorem 3 *The matrix $\{b^{ij}_{i_0}\}$ is the inverse of $\{a^{i_0}_{ij}\}$.*

Proof It is sufficient to show that

$$a^{i_0}_{ij}b^{jl}_{i_0} = \delta^i_l\,, \quad i,j,l=1,\dots,n\,, \text{ and } i,j,l\neq i_0\,. \tag{9.82}$$

Here and from this point on, the entries $i=1,\dots,n$, and $i\neq k$ mean $i=1,\dots,k-1,k+1,\dots,n$.

From (9.81), we readily see

$$b^{i_0k} = b^{ki_0} = 0\,, \quad k=1,\dots,n\,,$$

and therefore

$$a_{ij}^{i_0} b_{i_0}^{jl} = a_{im} b^{ml} \ , \quad i, j, l, m = 1, \dots, n \ , \ \text{and}\ i, j, l \neq i_0 \ .$$

Since (9.81)

$$\begin{aligned} a_{im} b^{ml} &= a_{im} a^{ml} - \frac{a_{im}}{a^{i_0 i_0}} a^{i_0 m} a^{i_0 l} \\ &= \delta_l^i - \frac{\delta_{i_0}^i}{a^{i_0 i_0}} a^{i_0 l} = \delta_l^i \ , \quad i, l, m = 1, \dots, n \ , \ \text{and}\ i, l \neq i_0 \ , \end{aligned}$$

i.e., (9.82) is valid. This proves the theorem. □

Now we apply the matrix consideration given above to the matrix $\{g_{ij}^{xs}\}$ which is the covariant metric tensor of the regular surface S^{xn} in the coordinates $s^1, \dots, s^n$.

Designate by $\{g_{ij}^{i_0}\}$ the matrix obtained by deleting the i_0th row and i_0th column of $\{g_{ij}^{xs}\}$. It is clear that $\{g_{ij}^{i_0}\}$ is the covariant metric tensor of the coordinate hypersurface $s^{i_0} = c_0$ in the coordinates $s^1, \dots, s^{i_0-1}, s^{i_0+1}, \dots, s^n$. The matrix which is inverse to $\{g_{ij}^{i_0}\}$ is the contravariant metric tensor of this coordinate hypersurface $s^{i_0} = c_0$ in the same coordinates $s^1, \dots, s^{i_0-1}, s^{i_0+1}, \dots, s^n$. Let this matrix be designated by $\{g_{i_0}^{ij}\}$. Since

$$g_{sx}^{i_0 i_0} = \det\{g_{ij}^{i_0}\}/g^{xs} \neq 0 \ ,$$

where $g^{xs} = \det\{g_{ij}^{xs}\}$, we find, availing us of Theorem 3,

$$g_{i_0}^{ij} = \frac{1}{g_{sx}^{i_0 i_0}} (g_{sx}^{i_0 i_0} g_{sx}^{ij} - g_{sx}^{i_0 i} g_{sx}^{i_0 j}) \ , \quad i, j = 1, \dots, n \ , \ \text{and}\ i, j \neq i_0 \ .$$

Therefore,

$$g_{sx}^{i_0 i} g_{sx}^{i_0 j} = g_{sx}^{i_0 i_0} (g_{sx}^{ij} - g_{i_0}^{ij}) \ , \quad i, j = 1, \dots, n \ , \ \text{and}\ i, j \neq i_0 \ . \tag{9.83}$$

Since

$$\begin{aligned} g_{sx}^{i_0 l} g_{sx}^{i_0 j} \Upsilon_{lj}^{i_0} = g_{sx}^{i_0 k} g_{sx}^{i_0 p} \Upsilon_{kp}^{i_0} + 2 g_{sx}^{i_0 i_0} g_{sx}^{i_0 j} \Upsilon_{i_0 j}^{i_0} - g_{sx}^{i_0 i_0} g_{sx}^{i_0 i_0} \Upsilon_{i_0 i_0}^{i_0} \ , \\ j, k, l, p = 1, \dots, n \ , \ \text{and}\ k, p \neq i_0 \ , \end{aligned}$$

we obtain, using (9.83),

$$\begin{aligned} & g_{sx}^{i_0 l} g_{sx}^{i_0 j} \Upsilon_{lj}^{i_0} = g_{sx}^{i_0 i_0} (g_{sx}^{kp} - g_{i_0}^{kp}) \Upsilon_{kp}^{i_0} + 2 g_{sx}^{i_0 i_0} g_{sx}^{i_0 j} \Upsilon_{i_0 j}^{i_0} - g_{sx}^{i_0 i_0} g_{sx}^{i_0 i_0} \Upsilon_{i_0 i_0}^{i_0} \\ & = g_{sx}^{i_0 i_0} g_{sx}^{lj} \Upsilon_{lj}^{i_0} - g_{sx}^{i_0 i_0} g_{i_0}^{kp} \Upsilon_{kp}^{i_0} \ , \\ & j, k, l, p = 1, \dots, n \ , \ \text{and}\ k, p \neq i_0 \ . \end{aligned} \tag{9.84}$$

Now we note that, from (9.17),

$$\Upsilon_{kp}^{i_0} = g_{sx}^{i_0 l} [\mathbf{x}_{s^k s^p} \cdot \mathbf{x}_{s^l}] \ , \quad k, l, p = 1, \dots, n \ ,$$

so, since (9.76),

$$\Upsilon^{i_0}_{kp} = \sqrt{g^{i_0 i_0}_{sx}}\mathbf{x}_{s^k s^p} \cdot \mathbf{n}_{i^0} \ , \quad k, p = 1, \ldots, n \ ,$$

where $\mathbf{n}_{i^0}$ is the unit normal to the coordinate hypersurface $s^{i_0} = c_0$ in S^{xn}. Thus,

$$\Upsilon^{i_0}_{kp} = \sqrt{g^{i_0 i_0}_{sx}} b_{kp} \ , \quad k, p = 1, \ldots, n \ , \text{ and } k, p \neq i_0 \ , \tag{9.85}$$

where

$$b_{kp} = \mathbf{x}_{s^k s^p} \cdot \mathbf{n}_{i^0} \ , \quad k, p = 1, \ldots, n \ , \text{ and } k, p \neq i_0 \ ,$$

is the (kp)th element of the second fundamental form of the coordinate hypersurface $s^{i_0} = c_0$ in S^{xn}. Therefore, using (9.60) and (9.85), we find

$$\begin{gathered} g^{kp}_{i_0} \Upsilon^{i_0}_{kp} = \sqrt{g^{i_0 i_0}_{sx}} g^{kp}_{i_0} b_{kp} = (n-1)\sqrt{g^{i_0 i_0}_{sx}} K_m(s^{i_0}) \ , \\ k, p = 1, \ldots, n \ , \text{ and } k, p \neq i_0 \ , \end{gathered} \tag{9.86}$$

where $K_m(s^{i_0})$ is the mean curvature of the coordinate hypersurface $s^{i_0} = c_0$ in S^{xn} with respect to the normal $\mathbf{n}_{i_0}$. Furthermore, from (9.43), with the identification $\xi^i = s^i$, we obtain

$$g^{lj}_{sx} \Upsilon^{i_0}_{lj} = -\Delta_B[s^{i_0}] \ , \quad j, l = 1, \ldots, n \ ,$$

where Δ_B is the operator of Beltrami in the metric of S^{xn}. Substituting this equation and (9.86) in (9.84) gives

$$g^{i_0 l}_{sx} g^{i_0 j}_{sx} \Upsilon^{i_0}_{lj} = -g^{i_0 i_0}_{sx} \Delta_B[s^{i_0}] - (n-1)(g^{i_0 i_0}_{sx})^{3/2} K_m(s^{i_0}) \ , \quad j, l = 1, \ldots, n \ .$$

Therefore, using (9.80), we conclude that the rate of change of the relative spacing of the coordinate hypersurfaces $s^{i_0} = const$ is expressed through the mean curvature and the Beltrami second differential parameter as follows:

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_{i_0}}\left(\frac{1}{\sqrt{g^{i_0 i_0}_{sx}}}\right) = -\frac{1}{g^{i_0 i_0}_{sx}} \Delta_B[s^{i_0}] - \frac{n-1}{\sqrt{g^{i_0 i_0}_{sx}}} K_m(s^{i_0}) \ . \tag{9.87}$$

The application of (9.77) to this equation gives the following relation between the mean curvature of the coordinate hypersurface $s^{i_0} = c_0$ and the Beltrami mixed and second differential parameters:

$$(n-1)K_m(s^{i_0}) + \nabla\left(s^{i_0}, \frac{1}{\sqrt{g^{i_0 i_0}_{sx}}}\right) + \frac{\Delta_B[s^{i_0}]}{\sqrt{g^{i_0 i_0}_{sx}}} = 0 \ . \tag{9.88}$$

Note that the Eqs. (9.87) and (9.88) are readily rewritten for the case of the monitor surface S^{rn}, namely, it suffices to substitute sx for sr in these equations. In particular, when S^{rn} is the monitor surface over S^n with a scalar-valued monitor function $f(\mathbf{s})$, i.e., S^{rn} is represented by

$$\mathbf{r}(\mathbf{s}) : S^n \to R^{n+1} , \quad \mathbf{r}(\mathbf{s}) = [\mathbf{s}, f(\mathbf{s})] ,$$

then, availing us of (9.71), we obtain that the Eq. (9.87) has the following specific form:

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_{i_0}}\Big(\frac{1}{\sqrt{g_{sr}^{i_0 i_0}}}\Big) = \frac{n}{\sqrt{g^{rs} g_{sr}^{i_0 i_0}}} K_m(S^{rn}) f_{s^{i_0}} - \frac{n-1}{\sqrt{g_{sr}^{i_0 i_0}}} K_m(s^{i_0}) \tag{9.89}$$

where $g^{rs} = 1 + f_{s^i} f_{s^i}$, $i = 1, \dots, n$, $K_m(S^{rn})$ is the mean curvature of S^{rn} in R^{n+1} with respect to the unit normal (9.69).

Two-Dimensional Case

Let S^{x2} be an arbitrary two-dimensional regular surface. Since for the coordinate line $s^{i_0} = c_0$ in S^{x2}

$$K_m(s^{i_0}) = \sigma_{i_0} , \tag{9.90}$$

where σ_{i_0} is the geodesic curvature of the curve $s^{i_0} = c_0$ in S^{xn}, the equations derived from (9.87) in the two-dimensional case have the form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_1}\Big(\frac{1}{\sqrt{g_{sx}^{11}}}\Big) &= -\frac{1}{g_{sx}^{11}} \Delta_B^x[s^1] - \frac{\sigma_1}{\sqrt{g_{sx}^{11}}} , \\ \frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_2}\Big(\frac{1}{\sqrt{g_{sx}^{22}}}\Big) &= -\frac{1}{g_{sx}^{22}} \Delta_B^x[s^2] - \frac{\sigma_2}{\sqrt{g_{sx}^{22}}} . \end{aligned} \tag{9.91}$$

Meanwhile Eq. (9.88) in the two-dimensional case yields

$$\begin{aligned} \sigma_1 + \nabla(s^1, 1/\sqrt{g_{sx}^{11}}) + \Delta_B^x[s^1]/\sqrt{g_{sx}^{11}} = 0 , \\ \sigma_2 + \nabla(s^2, 1/\sqrt{g_{sx}^{22}}) + \Delta_B^x[s^2]/\sqrt{g_{sx}^{22}} = 0 . \end{aligned} \tag{9.92}$$

Basic Relation to Grid Coordinates

Let us apply formulas (9.87) and (9.91) to the grid coordinates $\xi^1, \dots, \xi^n$ in S^{xn} obtained by the composition of the parametrization (9.2) and intermediate transformation (9.3).

Basic Theorem

We designate by v_p the rate of change of the relative spacing between the grid hypersurfaces $\xi^p = const$ in S^{xn}, i.e., analogously to (9.80),

$$v_p = \frac{\mathrm{d}}{\mathrm{d}\mathbf{n}_p}\Big(\frac{1}{\sqrt{g_{\xi x}^{pp}}}\Big) = \frac{1}{(g_{\xi x}^{pp})^2} g_{\xi x}^{pl} g_{\xi x}^{pj} \Upsilon_{lj}^p , \quad j, l, p = 1, \dots, n , \quad p \text{ fixed} ; \tag{9.93}$$

here, Υ^p_{lj} is the Christoffel symbol of the second rank of S^{xn} in the grid coordinates $\xi^1, \ldots, \xi^n$, and $\mathbf{n}_p$ is the normal to the grid hypersurface $\xi^p = const$, namely, similar to (9.76),

$$\mathbf{n}_p = \frac{1}{\sqrt{g^{pp}_{\xi x}}} g^{pj}_{\xi x} \frac{\partial}{\partial \xi^j} \mathbf{x}[\mathbf{s}(\boldsymbol{\xi})] \ , \quad j, p = 1, \ldots, n \ , \quad p \text{ fixed} \ . \tag{9.94}$$

We see that if $v_p < 0 (v_p > 0)$, then the nodes of the coordinate grid cluster (rarefy) in the $\mathbf{n}_p$ direction, i.e., v_p is a measure of change of grid spacing. We also call it a measure of grid clustering. The formulas (9.87) and (9.91), rewritten in the grid coordinates, result in the following:

Theorem 4 *Let* $\mathbf{x}(\mathbf{s})$ *in (9.2) and* $\mathbf{s}(\boldsymbol{\xi})$ *in (9.3) be nondegenerate transformations of the class* $C^2[S^n]$ *and* $C^2[\Xi^n]$*, respectively. Then,*

$$v_p = -\frac{1}{g^{pp}_{\xi x}} \Delta_B[\xi^p] - \frac{n-1}{\sqrt{g^{pp}_{\xi x}}} K_m(\xi^p) \ , \quad p = 1, \ldots, n \ , \quad p \textit{ fixed} \ , \tag{9.95}$$

where Δ_B *is the operator of Beltrami defined by (9.37) in the metric of* S^{xn}*; the function* $\xi^p(\mathbf{s})$ *is the pth component of the transformation* $\boldsymbol{\xi}(\mathbf{s}) : S^n \to \Xi^n$ *inverse to the intermediate transformation (9.2);* $K_m(\xi^p)$ *is the geodesic curvature of the curve* $\xi^p = c_0$ *in* S^{x2} *when* $n = 2$*, while* $K_m(\xi^p)$*, when* $n > 2$*, is the mean curvature of the hypersurface* $\xi^p = c_0$ *in* S^{xn}*.*

Remarks

We assume here that the logical domain Ξ^n is a rectangular n-dimensional parallelepiped $0 \le \xi^i \le l_i$, $i = 1, \ldots, n$, and the $(n-1)$-dimensional boundary plane $\xi^i = 0$ or $\xi^i = l_i$ is mapped onto some $(n-1)$-dimensional segment of the boundary of S^{xn}.

Formula (9.95) demonstrates how the measure v_p of grid clustering near a boundary hypersurface $\xi^p = c_0$ in S^{xn} depends on its mean curvature. In particular, when the grid coordinate function $\xi^p(\mathbf{s})$, $p = 1, \ldots, n$, is subject to the corresponding pth equation of the grid system

$$\Delta_B[\xi^i] = 0 \ , \quad i = 1, \ldots, n \ , \tag{9.96}$$

in the original metric of the physical geometry S^{xn}, then (9.95) yields

$$v_p = -\frac{n-1}{\sqrt{g^{pp}_{\xi x}}} K_m(\xi^p) \ , \quad p = 1, \ldots, n \ , \quad p \text{ fixed.} \tag{9.97}$$

So, the sign of v_p is determined by the sign of $K_m(\xi^p)$. Note that the Eq. (9.96) proposed for two-dimensional domains by Winslow (1967) and for surfaces by Warsi (1981) are the most popular for the generation of fixed grids in domains and on two-dimensional surfaces.

If S^{xn} is a domain S^n with the Euclidean metric, then the Eq. (9.96) are the Laplace equations

$$\nabla^2[\xi^i] \equiv \frac{\partial}{\partial s^j}\Big(\frac{\partial \xi^i}{\partial s^j}\Big) = 0\,, \quad i, j = 1 \ldots, n\,. \tag{9.98}$$

In the monographs of Thompson et al. (1985) and Liseikin (1999), it was proved that the nodes of the coordinate grid obtained in S^n by the solution of the equations inverse to (9.98) on a rectangular computational domain $\Xi^n : 0 \le \xi^i \le l_i,\ i = 1, \ldots, n$, cluster near concave boundary segments of S^n and rarefy near its convex boundary segments (see Figs. 6.6 and 6.7 for $n = 3$). However, formula (9.97) yields more strong conclusions for n-dimensional domains S^n when $n \ge 3$. In order to formulate these results, we first note that the unit normal $\mathbf{n}_p$, specified by (9.94), in this case, is as follows:

$$\mathbf{n}_p = \frac{1}{\sqrt{g^{pp}_{\xi s}}} g^{pj}_{\xi s} \mathbf{s}_{\xi^j}\,, \quad j, p = 1, \ldots, n\,, \quad p \text{ fixed}\,,$$

where

$$g^{pj}_{\xi s} = \frac{\partial \boldsymbol{\xi}}{\partial s^p} \cdot \frac{\partial \boldsymbol{\xi}}{\partial s^j}\,, \quad j, p = 1, \ldots, n\,.$$

Since

$$\mathbf{n}_p \cdot \mathbf{s}_{\xi^p} = \frac{1}{\sqrt{g^{pp}_{\xi s}}} > 0\,, \quad p = 1, \ldots, n\,, \quad p \text{ fixed}\,,$$

the unit normal $\mathbf{n}_p$ is directed to the interior of S^n at the points of the boundary hypersurface $\xi^p = 0$. Contrastingly, at the points of the hypersurface $\xi^p = l_p$, it is directed to the exterior of S^n. Therefore, the inequality $v_p > 0$ ($v_p < 0$) at the points of the boundary hypersurface $\xi^p = 0$ means that the grid nodes cluster (rarefy) near it. Contrastingly for the hypersurface $\xi^p = l_p$ in S^n, the inequality $v_p > 0$ ($v_p < 0$) means rarefaction (clustering) of grid nodes near it. Thus, the nodes of the grid produced by the equations inverted to (9.98) will also cluster (rarefy) near the boundary $\xi^p = l_p$ if this segment is not concave (convex) but rather has the negative (positive) mean curvature, for example, it is a saddle surface (see Fig. 9.7).

Formula (9.97) also yields a new result in the theory of surface grid generation. Namely, the nodes of the coordinate grid on the surface S^{x2} generated through the equations inverse to (9.96) cluster (rarefy) near concave (convex) segments of the boundary of S^{x2}. Figures 9.8 and 9.9 of the surface grid generated through the solution of equations (9.96) for $n = 2$ demonstrate node clustering near its concave boundary segments. The right-hand part of Fig. 9.8 illustrates the grid in a parametric domain S^2.

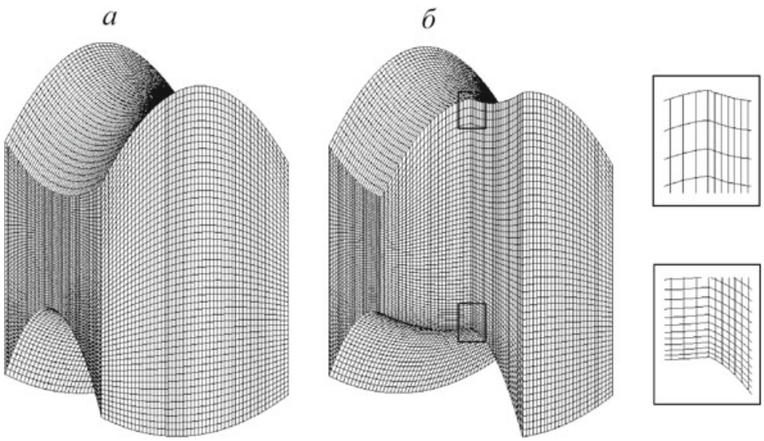


Fig. 9.7 Three-dimensional domain with a hexahedral numerical grid generated through the inverted system of Laplace equations. The grid is clustering near a concave boundary segment and rarefying near a convex boundary segment

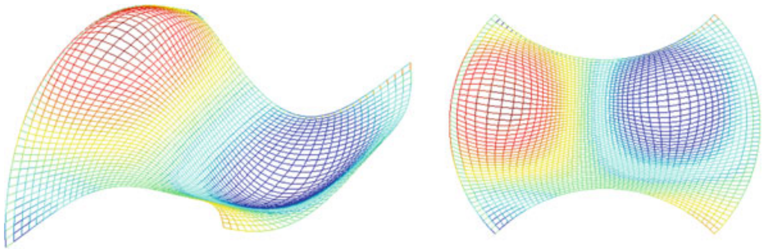


Fig. 9.8 A surface grid with node clustering near concave boundary segments (*left*). The grid on the parametric domain S^2 (*right*)

9.4.3 Diffusion Functional

One of the generalizations of the smoothness functional (9.21) is represented by a diffusive functional:

$$I_D[\xi] = \int\limits_{S^n} w(\mathbf{s}) g_{sx}^{jk} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^j} d\mathbf{s} \equiv \int\limits_{S^n} w(\mathbf{s}) g_{\xi x}^{ii} d\mathbf{s} \equiv \int\limits_{\Xi^n} w[\mathbf{s}(\boldsymbol{\xi})] J g_{\xi x}^{ii} d\boldsymbol{\xi}, \qquad (9.99)$$
$$i, j, k = 1, \dots, n,$$

where $J = \det\{\partial s^i / \partial s^j\}$.

Analogously to the inverted functional of grid smoothness in the form (9.23), the diffusion functional (9.99) yields the following inverted diffusion functional $I_{ID}[\mathbf{s}]$ with respect to the intermediate transformations $\mathbf{s}(\boldsymbol{\xi})$:

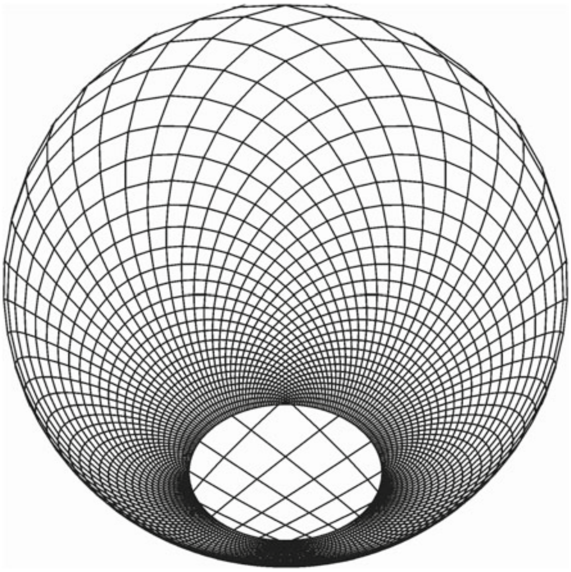


Fig. 9.9 A grid with node clustering near a boundary segment with negative curvature

$$I_{ID}[\mathbf{s}] = \int\limits_{\Xi^n} w(\mathbf{s}(\boldsymbol{\xi}))\mathrm{tr}\{g^{ij}_{\xi x}\}\mathrm{d}\boldsymbol{\xi} \ . \qquad (9.100)$$

Thus, for $n = 1, 2, 3$, we have

$$I_{ID}[s] = \begin{cases} \int\limits_{\Xi^1} \dfrac{w[s(\xi)]}{J g^{xs}}\mathrm{d}\xi \ , & n = 1 \ , \\ \int\limits_{\Xi^2} \dfrac{w[\mathbf{s}(\boldsymbol{\xi})]}{J g^{xs}}(g^{s\xi}_{11} + g^{s\xi}_{22})\mathrm{d}\xi^1\mathrm{d}\xi^2 \ , & n = 2 \ , \\ \int\limits_{\Xi^3} \dfrac{w[\mathbf{s}(\boldsymbol{\xi})]}{J g^{xs}}[g^{s\xi}_{11}g^{s\xi}_{22} + g^{s\xi}_{11}g^{s\xi}_{33} + g^{s\xi}_{22}g^{s\xi}_{33} & \\ \quad -(g^{s\xi}_{12})^2 - (g^{s\xi}_{13})^2 - (g^{s\xi}_{23})^2]\mathrm{d}\xi^1\mathrm{d}\xi^2\mathrm{d}\xi^3 \ , & n = 3 \ , \end{cases} \qquad (9.101)$$

The expression of functional (9.99) prompts one to determine what the monitor metric should be to provide the generation of the numerical grid with a required property. For this purpose, the metric is to have such a form so that the integrand of the functional (9.99)

$$\sigma(\mathbf{s}) = w(\mathbf{s})g^{jk}_{sx}\frac{\partial \xi^i}{\partial s^j}\frac{\partial \xi^i}{\partial s^k}, \quad i, j, k = 1, \dots, n \qquad (9.102)$$

describes a measure of departure of the grid from the necessary grid at the point $\mathbf{s} \in S^n$. If such a metric is found, then it can be expected that the minimization of the functional (9.99) will produce the grid with the required property. In particular, formula (9.102) with respect to the metric of the monitor surface S^{rn} over S^{xn} can be considered as a measure of departure of a grid from the reference grid with node clustering in the zones of large variations of the function $\mathbf{f}(\mathbf{s})$.

The Euler–Lagrange equations for the diffusion functional (9.99) have the following form:

$$\frac{\partial}{\partial s^m}\Big(w(\mathbf{s})\, g^{ml}_{sx}\, \frac{\partial \xi^i}{\partial s^l}\Big) = 0\ , \qquad i, m, l = 1, \dots, n\ , \tag{9.103}$$

in particular, when $S^{xn} \equiv S^n$, then these equations are the diffusion equations

$$\frac{\partial}{\partial s^m}\Big(w(\mathbf{s})\frac{\partial \xi^i}{\partial s^m}\Big) = 0\ , \qquad i, m = 1, \dots, n\ , \tag{9.104}$$

described in Sect. 6.

9.4.4 Dimensionless Functionals

This section formulates some dimensionally homogeneous functionals through the invariants of the metric tensor $\{g^{x\xi}_{ij}\}$ which, in analogy with the smoothness functional, measure global nonuniformity of hypersurface grids.

As was demonstrated in Sect. 9.3.2, the quantity I_{n-1}/I_n represents a measure of the local clustering of a hypersurface grid. Integration of this measure over the hypersurface S^{xn} derives the smoothness functional whose minimization tends to yield a uniform grid on S^{xn}. The smoothness functional possesses the spectacular properties reviewed in Sects. 9.3.2 and 9.3.3. In the particular case when $n = 2$, this functional is also dimensionless. However, it is not dimensionless in three-dimensions, which may be important for the generation of spatial adaptive grids. Nevertheless, using the invariants of the covariant metric tensor $\{g^{x\xi}_{ij}\}$, we can formulate dimensionless functionals measuring grid nonuniformity for arbitrary $n > 1$.

For this purpose, we, in analogy with (7.34), consider a dimensionless measure of the local departure of a hypersurfaces grid, in particular of a monitor surface grid, from a conformal one:

$$Q_{\mathrm{cf},3} = [I_{n-1}/(I_n)^{1-1/n}]^{\alpha}\ , \qquad \alpha > 0\ , \qquad n > 1\ .$$

As was mentioned in Sect. 7.3.1, the dimensionless functionals are formulated by integrating local dimensionless measures over the unit cube Ξ^n. Thus, we obtain one functional, using the quantity $Q_{\mathrm{cf},3}$,

$$I_{\mathrm{cf},3} = \int\limits_{\Xi^n} [I_{n-1}/(I_n)^{1-1/n}]^{\alpha} \mathrm{d}\boldsymbol{\xi}$$
$$= \int\limits_{S^{xn}} \frac{1}{\sqrt{I_n}} [I_{n-1}/(I_n)^{1-1/n}]^{\alpha} \mathrm{d}S^{xn} \ . \tag{9.105}$$

Functional (9.105) can be interpreted as an integral measure of the departure from conformality. When $(1 - 1/n)\alpha = 1/2$, this functional is independent of the parametrization of S^{xn}.

Another dimensionless functional measuring the departure from conformality is defined, in analogy with (7.32), through the first and nth invariants:

$$I_{\mathrm{cf},2} = \int\limits_{\Xi^n} [I_1/(I_n)^{1/n}]^{\alpha} \mathrm{d}\boldsymbol{\xi}$$
$$= \int\limits_{S^{xn}} \frac{1}{\sqrt{I_n}} [I_1/(I_n)^{1/n}]^{\alpha} \mathrm{d}S^{xn} \ , \quad \alpha > 0 \ , \quad n > 1 \ . \tag{9.106}$$

This functional is independent on parametrizations of S^{xn} if $\alpha = n/2$. The parameter α in (9.105) and (9.106) can be used to control the form of the functionals. In particular, assuming $\alpha = n/2$ in (9.105), we obtain a dimensionless functional which is defined through the measure of the local grid clustering $I_{n-1}/I_n = g^{ii}_{\xi x}$, $i = 1, \ldots, n$:

$$I_{\mathrm{cf},3} = \int\limits_{S^{xn}} (I_{n-1}/I_n)^{n/2} \mathrm{d}S^{xn} \ . \tag{9.107}$$

Note that functional (9.107) coincides with the smoothness functional for $n = 2$. For $n = 3$, we obtain, using (9.15),

$$I_{\mathrm{cf},3} = \int\limits_{S^{r3}} \left(\frac{I_2}{I_3}\right)^{3/2} \mathrm{d}S^{r3}$$
$$= \int\limits_{S^3} \sqrt{g^{xs}} \left(g^{kl}_{sx} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^l}\right)^{3/2} \mathrm{d}\boldsymbol{s} \ , \quad i, j, k, l = 1, 2, 3 \ . \tag{9.108}$$

Thus, the Euler–Lagrange equations for this functional have the form

$$\frac{\partial}{\partial s^j} \left(\sqrt{g^{xs}} \sqrt{\frac{I_2}{I_3}} g^{kj}_{sx} \frac{\partial \xi^i}{\partial s^k}\right) = 0 \ , \qquad i, j, k = 1, 2, 3 \ , \tag{9.109}$$

while the corresponding inverted equations are represented as

$$g^{ij}_{\xi x}\frac{\partial^2 s^l}{\partial\xi^i\partial\xi^j}=\frac{1}{\sqrt{g^{xs}}}\sqrt{\frac{I_3}{I_2}}\frac{\partial}{\partial s^k}\left(\sqrt{g^{xs}}\sqrt{\frac{I_2}{I_3}}g^{lk}_{sx}\right)\,,\quad i,j,k,l=1,2,3\,,\qquad (9.110)$$

which are a generalization of (9.49) with $n = 3$ for three-dimensional hypersurfaces. In the same manner, there can be written Euler–Lagrange equations for the functional $I_{\mathrm{cf},3}$ for arbitrary $n > 1$.

Analogously, by assuming $\alpha = n/[2(n-1)]$, we find a simpler form of the dimensionless functional (9.106):

$$I_{\mathrm{cf},2}=\int\limits_{S^{xn}}(I_1/I_n)^{n/[2(n-1)]}\mathrm{d}S^{xn}\,. \qquad (9.111)$$

This functional coincides with the smoothness functional for $n = 2$, while for $n = 3$, it has the form

$$I_{\mathrm{cf},2}=\int\limits_{S^{r3}}(I_1/I_3)^{3/4}\mathrm{d}S^{r3}\,. \qquad (9.112)$$

Substituting in formulas (9.105)–(9.112), S^{xn} for S^{rn} yields the corresponding dimensionless functionals for generating adaptive grids on S^{xn}.

9.5 Formulation of Comprehensive Grid Generator

Sections 9.2 and 9.4 give a schematic description of a mapping approach in which an adaptive grid in the physical geometry S^{xn} represented by (9.1) is generated as a smooth mapping subject to the inverted Beltrami equation with respect to the metric of a monitor hypersurface S^{rn} over S^{xn}. This metric provides the generation of grids with node clustering in the zones of large variations of a specified function $\boldsymbol{f}(\boldsymbol{x})$. However, the application of such a metric does not provide the generation of grids with other very important properties, in particular, grids with node clustering according to given function values, grid alignment with given vector fields, and combinations thereof. So, it is necessary to generalize the metrics (9.9) and (9.10) of the monitor hypersurface S^{rn} in order to generate adaptive grids for a more wide variety of applied problems.

For the purpose of providing more efficient control of grid generation in the hypersurface S^{xn}, we introduce the notion of a monitor manifold designated as M^n over S^{xn}. The points of the manifold M^n are the points of S^{xn}, while its metric may differ from the metric of the given hypersurface. We shall refer to this more general metric as the control metric.

This section reviews mathematical models for robust grid generators based on energy and diffusion functionals and corresponding differential Beltrami and diffusion equations, with respect to the control metrics.

9.5.1 Formulation of Control Metrics

The control metric serves to control the properties of grids in the physical geometry S^{xn}. It is evident that the metric should take into account the quantities requiring grid adaptation: geometric characteristics of S^{xn}, physical variables and their derivatives, specified vector fields, error of the numerical solution applied, etc.

For the purpose of better handling grid control, it is reasonable to restrict the whole set of the control metrics to a basic subset which, however, can be adequate for realizing the necessary grid properties. Also, these basic control metrics are to be described by simple formulas which allow one to establish readily the relations between them and the grid characteristics. Furthermore, one of the natural ways to satisfy balanced grid properties, each of which is realized by an individual control metric, is to combine these metrics linearly. Therefore, the basic metric tensors formulated should be subject to the operation of summation in the sense that the sum of the two control metrics from the subset is also the metric (nonsingular tensor). Note, in general, that the sum of two metrics may not be a metric, since the sum of two nonsingular matrices may be a singular matrix. In addition, a mathematical formulation of the control metric should be simple and comprehensive, so that a robust grid code could be developed for automatic generation of grids with required properties. It turns out that defining a suitable control metric is the key to success in mesh adaptation.

This subsection describes an approach for formulating such basic control metrics.

Let the covariant (contravariant) elements of the control metric in arbitrary coordinates $v^1, \ldots, v^n$ of the monitor manifold M^n over S^{xn} be designated as $g^{\mathbf{v}}_{ij}\,(g^{ij}_{\mathbf{v}})$.

Notice that the matrices $\{g^{\mathbf{v}}_{ij}\}$ and $\{g^{ij}_{\mathbf{v}}\}$, referred to as covariant and contravariant metric tensors in the coordinates $v^1, \ldots, v^n$, are to be nondegenerate, symmetric, and mutually inverse, i.e.,

$$g^{\mathbf{v}}_{ij} g^{jk}_{\mathbf{v}} = \delta^i_k, \quad i, j, k = 1, \ldots, n \ . \tag{9.113}$$

Besides this covariant, metric elements in new coordinates $w^1, \ldots, w^n$ and old coordinates $v^1, \ldots, v^n$ are connected by the following relations:

$$g^{\mathbf{v}}_{ij} = g^{\mathbf{w}}_{kp} \frac{\partial w^k}{\partial v^i} \frac{\partial w^p}{\partial v^j}, \quad i, j, k, p = 1, \ldots, n \ . \tag{9.114}$$

In this case, the contravariant metric elements will be subject to the relations

$$g^{ij}_{\mathbf{v}} = g^{kp}_{\mathbf{w}} \frac{\partial v^i}{\partial w^k} \frac{\partial v^j}{\partial w^p}, \quad i, j, k, p = 1, \ldots, n \ . \tag{9.115}$$

Thus, for formulating a control metric on S^{xn}, it is sufficient to define covariant metric elements or contravariant metric elements in one coordinate system. In other coordinates, they are defined through the formulas (9.114) and (9.115).

The most general and simple formulation of covariant metric elements g^s_{ij} of the imposed control metric on S^{xn} in the parametric coordinates $s^1, \dots, s^n$, is given through a set of covariant tensors of the first rank

$$\mathbf{F}^k(\mathbf{s}) = [F^k_1(\mathbf{s}) \ , \dots, \ F^k_n(\mathbf{s})] \ , \quad k = 1, \dots, l \ ,$$

by the following formula:

$$g^{\mathbf{s}}_{ij} = z(\mathbf{s})g^{xs}_{ij} + F^k_i(\mathbf{s})F^k_j(\mathbf{s}) \ , \ i, j = 1, \dots, n \ , \quad k = 1, \dots, l \ , \tag{9.116}$$

where g^{xs}_{ij} is the metric of S^{xn}, and $z(\mathbf{s}) \geq 0$ is a weight function specifying the contribution of this metric to the control metric $g^{\mathbf{s}}_{ij}$. When $S^{xn} \equiv S^n$, then, from (9.116),

$$g^{\mathbf{s}}_{ij} = z(\mathbf{s})\delta^i_j + F^k_i(\mathbf{s})F^k_j(\mathbf{s}) \ , \ i, j = 1, \dots, n \ , \quad k = 1, \dots, l \ , \tag{9.117}$$

and consequently the covariant metric elements (9.10) and (9.9) have the form (9.116) and (9.117), respectively, if $z(s) = 1 \ , \ F^k_i = \partial f^k/\partial s^i$.

Of course, it is assumed that the function $z(\mathbf{s})$ and the vectors $\mathbf{F}^k(\mathbf{s}), \ k = 1, \dots, l$, in (9.116) and (9.117) are subject to the restriction

$$g^{\mathbf{s}} = \det\{g^{\mathbf{s}}_{ij}\} > 0,$$

at each point $\mathbf{s} \in S^n$.

Notice that if we introduce in R^{n+k+l} vectors $\mathbf{w}_i(\mathbf{s}), \ i = 1, \dots, n$ by the formula

$$\mathbf{w}_i(\mathbf{s}) = [\sqrt{z(\mathbf{s})}\frac{\partial \mathbf{x}}{\partial s^i}, F^1_i, \dots, F^l_i], \quad i = 1, \dots, n,$$

where $\mathbf{x}(\mathbf{s})$ is the parametrization (9.1) of the physical geometry S^{xn}, then it is obvious that

$$g^{\mathbf{s}}_{ij} = \mathbf{w}_i \cdot \mathbf{w}_j, \quad i, j = 1, \dots, n.$$

So, for nonsingularity of the control metric tensor (9.116), the vectors $\mathbf{w}_i(\mathbf{s}), \ i = 1, \dots, n$, must be independent. In particular, the vectors $\mathbf{w}_i(\mathbf{s}), \ i = 1, \dots, n$, will be independent if $z(\mathbf{s}) > 0$ at each point $\mathbf{s} \in S^n$, since the tangent vectors $\partial \mathbf{x}/\partial s^i, \ i = 1, \dots, n$, of S^{xn} are independent.

It is evident that the linear combination of two metric tensors of the form (9.116) with corresponding nonnegative coefficients $\varepsilon_1(\mathbf{s})$ and $\varepsilon_2(\mathbf{s})$ is the matrix of the same form (9.116), and it is nonsingular (metric tensor) if $\varepsilon_1(\mathbf{s}) + \varepsilon_2(\mathbf{s}) > 0$ at each point of $\boldsymbol{s} \in S^n$.

Sometimes, instead of the covariant metric components $g^{\mathbf{s}}_{ij}$, it is convenient, in particular in order to define the measure of grid departure from a required grid, to formulate the contravariant components $g^{ij}_{\mathbf{s}}$ of the control metric in the parametric coordinates $s^1, \dots, s^n$, for example, in the form (9.116), namely as

$$g^{ij}_{\mathbf{s}} = \epsilon(\mathbf{s})g^{ij}_{sx} + B^i_k B^j_k , \quad i, j = 1, \ldots, n \ , \ k = 1, \ldots, l \ , \tag{9.118}$$

where $\epsilon(\mathbf{s}) \geq 0$, g^{ij}_{sx} are the contravariant metric elements of S^{xn}, while $B^i_k, i = 1, \ldots, n$, are the components of a contravariant vector $\mathbf{B}_k = (B^1_k, \ldots, B^n_k)$, $k = 1, \ldots, l$. In particular, if $S^{xn} \equiv S^n$, (9.118) becomes

$$g^{ij}_{\mathbf{s}} = \epsilon(\mathbf{s})\delta^i_j + B^i_k B^j_k , \quad i, j = 1, \ldots, n \ , \ k = 1, \ldots, l \ , \tag{9.119}$$

9.5.2 Energy and Diffusion Functionals

Similar to (9.34), the energy functional I_E with respect to the control metric g^s_{ij} in the parametric coordinates $s^1, \ldots, s^n$ is as follows:

$$I_E[\boldsymbol{\xi}] = \int\limits_{S^n} \sqrt{g^s} g^{ml}_s \frac{\partial \xi^i}{\partial s^m} \frac{\partial \xi^i}{\partial s^l} \mathrm{d}\boldsymbol{s} \ , \qquad i, l, m = 1, \ldots, n \ , \tag{9.120}$$

where $g^s = \det\{g^s_{ij}\}$.

Assuming in (9.120), $\sqrt{g^s} = w(\boldsymbol{s})$, where $w(\boldsymbol{s}) > 0$ is a weight function, produces the functional of diffusion I_D with respect to the control metric g^s_{ij}:

$$I_D[\boldsymbol{\xi}] = \int\limits_{S^n} w(\boldsymbol{s}) g^{ml}_s \frac{\partial \xi^i}{\partial s^m} \frac{\partial \xi^i}{\partial s^l} \mathrm{d}\boldsymbol{s} \ , \qquad i, l, m = 1, \ldots, n \ . \tag{9.121}$$

The inverse of the function $\boldsymbol{\xi}(\boldsymbol{s}) : S^n \to \Xi^n$, which is a critical point of the functionals (9.120) or (9.121), determines the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^n \to S^n$ applied to specify the grid nodes on the parametric domain S^n by mapping the nodes of a reference grid in Ξ^n with $\boldsymbol{s}(\boldsymbol{\xi})$. Consequently, the grid nodes on the physical geometry S^{xn} are obtained by mapping the nodes in S^n with parametrization (9.1): $\boldsymbol{x}(\boldsymbol{s}) : S^n \to S^{xn}$. Equally, the nodes in S^{xn} are obtained by mapping the nodes of a reference grid in Ξ^n with the transformation

$$\boldsymbol{x}(\boldsymbol{s}(\boldsymbol{\xi})) : \Xi^n \to S^{xn} \subset R^{n+k} \ . \tag{9.122}$$

Notice that the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$ is typically obtained by minimizing the inverse functionals which, similar to (9.23) and (9.100), have, respectively, the following forms:

$$I_{IE}[\boldsymbol{s}] = \int\limits_{\Xi^n} \sqrt{g^{\xi}} g^{ii}_{\xi} \ \mathrm{d}\boldsymbol{\xi} \ , \qquad i = 1, \ldots, n \ , \tag{9.123}$$

where $g^{\xi} = \det\{g^{\xi}_{ij}\}$,

$$I_{ID}[\boldsymbol{\xi}] = \int\limits_{\Xi^n} w(\boldsymbol{s}) g^{ii}_{\boldsymbol{\xi}} \, \mathrm{d}\boldsymbol{\xi} \ , \qquad i = 1, \dots, n \ . \tag{9.124}$$

The novel feature of the functionals (9.120) and (9.121) is the recognition that the control metric g^s_{ij} can be freely chosen to give the grid on S^{xn} desired properties. The incorporation of the weight function w provides additional control over the redistribution of grid nodes in selected regions of S^{xn}. The job of constructing the grid then rests entirely on the choice of the control metric and the weight function.

The functionals (9.120) and (9.121) give some guess as to how to specify the control metric g^s_{ij} in order to obtain the grid with a desired property. Namely, the metric g^s_{ij} should be specified so that the quantity

$$m(\boldsymbol{s}) = w(\boldsymbol{s}) g^{jl}_{s} \frac{\partial \xi^i}{\partial s^j} \frac{\partial \xi^i}{\partial s^l} \ , \qquad i, j, l = 1, \dots, n \ , \tag{9.125}$$

reflects some measure of departure from the desired grid in S^{xn}. In this case, minimization of the functionals with this metric gives the intermediate transformations $\boldsymbol{s}(\boldsymbol{\xi})$ which are able to produce desirable grids. In particular, in accordance with Sect. 10.3.2, for the metric $g^s_{ij} = g^{rs}_{ij}$ of a monitor hypersurface S^{rn} over S^{xn}, the quantity (9.125) with $w(\boldsymbol{s}) = \sqrt{g^s}$ is a measure of departure from a uniform grid on S^{rn}, and consequently from an adaptive grid on S^{xn}.

Notice that the dimensionless functionals considered in Sect. 9.4.4 are readily formulated for the hypersurface S^{xn} with the introduced control metric g^s_{ij} through the invariants of the control metric tensor $\{g^{\xi}_{ij}\}$ in the grid coordinates $\xi^1, \dots, \xi^n$. However, these functionals are the same for the metrics g^s_{ij} and $w(\boldsymbol{s}) g^s_{ij}$, and therefore, the functionals should also include weight functions, similar to the way in which it was done for formulating the diffusion functional (9.121).

9.5.3 Beltrami and Diffusion Equations

Similar to (9.35), the intermediate transformation $\mathbf{s}(\xi)$ for generating grids in S^n and, consequently, on S^{xn} is determined as the inverse of the transformation

$$\boldsymbol{\xi}(\mathbf{s}) : S^n \to \Xi^n \ , \quad \boldsymbol{\xi}(\mathbf{s}) = [\xi^1(\mathbf{s}), \dots, \xi^n(\mathbf{s})]$$

which is a solution to the following Dirichlet boundary value problem for the Euler-Lagrange equations derived from the functional of energy (9.120):

$$\begin{aligned}&\frac{\partial}{\partial s^j}\Big(\sqrt{g^{\mathrm{s}}}g_{\mathrm{s}}^{jk}\frac{\partial \boldsymbol{\xi}}{\partial s^k}\Big)=0\ ,\quad j,k=1,\dots,n\ ,\\ &\boldsymbol{\xi}(\boldsymbol{s})|_{\partial S^n}=\boldsymbol{\varphi}(\mathbf{s}):\partial S^n\to\partial\Xi^n\ ,\quad \boldsymbol{\varphi}(\mathbf{s})=[\varphi^1(\mathbf{s}),\dots,\varphi^n(\mathbf{s})]\ ,\end{aligned}$$

where g_{s}^{jk} are the contravariant components of the control metric in the parametric coordinates $s^1,\dots,s^n$, ∂S^n and $\partial\Xi^n$ are the boundaries of S^n and Ξ^n, respectively, while $\varphi(\mathbf{s})$ is a one-to-one continuous transformation between the boundaries of S^n and Ξ^n. This boundary value problem has the following form for the components $\xi^i(\mathbf{s})$:

$$\begin{aligned}&\frac{\partial}{\partial s^j}\Big(\sqrt{g^{\mathrm{s}}}g_{\mathrm{s}}^{jk}\frac{\partial \xi^i}{\partial s^k}\Big)=0\ ,\quad i,j,k=1,\dots,n\ ,\\ &\xi^i(\boldsymbol{s})|_{\partial S^n}=\varphi^i(\mathbf{s})\ ,\quad i=1,\dots,n\ .\end{aligned}\tag{9.126}$$

In the theory of Riemannian manifolds, the equations in (9.126) are called generalized Laplace equations or Beltrami equations of second order. We shall hereafter call them Beltrami equations.

The functions $\xi^1(\boldsymbol{s}),\dots,\xi^n(\boldsymbol{s})$ satisfying (9.126) form a curvilinear coordinate system in S^n and on S^{xn}. These curvilinear coordinates are further referred to as the grid coordinates.

It is easily shown that an arbitrary one-to-one twice differentiable transformation $\boldsymbol{\psi}(\boldsymbol{s}):S^n\to\Xi^n, \boldsymbol{\psi}(\boldsymbol{s})=(\psi^1(\boldsymbol{s})\ ,\dots,\psi^n(\boldsymbol{s}))$, is a solution to the Dirichlet problem (9.126) with respect to the control metric g_{ij}^s specified in the parametric coordinates $s^1,\dots,s^n$, by

$$g_{ij}^s=\frac{\partial\boldsymbol{\psi}}{\partial s^i}\cdot\frac{\partial\boldsymbol{\psi}}{\partial s^i}\ ,\quad i,j=1,\dots,n\ ,\tag{9.127}$$

and with the following boundary conditions:

$$\boldsymbol{\xi}(\boldsymbol{s})|_{\partial S^n}=\boldsymbol{\psi}(\boldsymbol{s})\ ,\quad i=1,\dots,n\ .$$

Indeed, it is readily obtained, from (9.127), that

$$\sqrt{g^s}=\det\Big\{\frac{\partial\psi^l}{\partial s^k}\Big\}\ ,\quad g_s^{jk}=\frac{\partial s^j}{\partial\psi^m}\frac{\partial s^k}{\partial\psi^m}\ ,\quad j,k,l,m=1,\dots,n\ ,$$

so the ith equation in (9.126) with $\partial\xi^i/\partial s^k=\partial\psi^i/\partial s^k$ is, in fact, the identity of the form (2.48)

$$\frac{\partial}{\partial s^j}\Big(\det\Big\{\frac{\partial\psi^l}{\partial s^k}\Big\}\frac{\partial s^j}{\partial\psi^i}\Big)\equiv 0\ ,\quad i,j=1,\dots,n\ ,$$

i.e., the functions $\psi^i(\boldsymbol{s}),\ i=1,\dots,n$, are the solutions to the Dirichlet problem (9.126) for the Beltrami equations with respect to the metric (9.127).

The substitution of $w(\mathbf{s})$ for $\sqrt{g^{\mathrm{s}}}$ in the system of equations in (9.126) yields the Dirichlet problem for more general diffusion equations

$$\begin{aligned}&\frac{\partial}{\partial s^j}\left(w(\mathbf{s})g_{\mathbf{s}}^{jk}\frac{\partial \xi^i}{\partial s^k}\right)=0\ , \quad i,j,k=1,\dots,n\ ,\\ &\xi^i(\mathbf{s})|_{\partial S^n}=\varphi^i(\mathbf{s})\ , \quad i=1,\dots,n\ .\end{aligned} \tag{9.128}$$

Here, $w(\mathbf{s})>0$ is a weight function aimed at increasing or decreasing the effect of the control metric in the necessary zones of S^{xn}. The equations in (9.128) are, in fact, the Euler–Lagrange equations for the diffusion functional (9.121), so they will be referred to as the diffusion equations.

The diffusion equations in (9.128) are equivalent to the Beltrami equations in (9.126) if $w(\mathbf{s})=\sqrt{g^{\mathbf{s}}}$, $g^{\mathbf{s}}=\det\{g_{ij}^{\mathbf{s}}\}$. Moreover, for $n\neq 2$, they are always equivalent to the Beltrami equations, with respect to the metric

$$g_{ij}=(g^{\mathbf{s}})^{\frac{1}{2-n}}[w(\mathbf{s})]^{\frac{2}{n-2}}g_{ij}^{\mathbf{s}}, \quad i,j=1,\dots,n\ , \tag{9.129}$$

regardless of the weight function $w(\mathbf{s})>0$. Indeed, for this metric,

$$\begin{aligned}&g=\det\{g_{ij}\}=(g^{\mathbf{s}})^{\frac{n}{2-n}}[w(\mathbf{s})]^{\frac{2n}{n-2}}g^{\mathbf{s}}=(g^{\mathbf{s}})^{\frac{2}{2-n}}[w(\mathbf{s})]^{\frac{2n}{n-2}}\ ,\\ &g^{ij}=(g^{\mathbf{s}})^{\frac{1}{n-2}}[w(\mathbf{s})]^{\frac{2}{2-n}}g_{\mathbf{s}}^{ij}, \quad i,j=1,\dots,n\ ,\end{aligned}$$

so

$$\sqrt{g}g^{ij}=w(\mathbf{s})g_{\mathbf{s}}^{ij}, \quad i,j=1,\dots,n\ ,$$

and therefore

$$\frac{\partial}{\partial s^j}\left(w(s)g_{\mathbf{s}}^{jk}\frac{\partial \xi^i}{\partial s^k}\right)\equiv\frac{\partial}{\partial s^j}\left(\sqrt{g}g^{jk}\frac{\partial \xi^i}{\partial s^k}\right)\ , \quad i,j,k=1,\dots,n\ ,$$

i.e., the system in (9.128) is the system of Beltrami equations with respect to the metric (9.129).

Though the Beltrami equations in (9.126) are comprehensive, i.e., for an arbitrary nondegenerate twice differentiable intermediate transformation $\mathbf{s}(\xi)$ there exists a control metric such that the transformation is the inverse of the solution of these equations, the form in (9.128) of the diffusion equations with the weight function appears to be more convenient, especially for $n=2$, for realizing the necessary requirements for the grid properties in different zones of S^{xn}. However, one should remember that for another coordinate system $v^1,\dots,v^n$, the equations in (9.128) become equivalent to

$$\frac{\partial}{\partial v^j}\left(w[\mathbf{s}(\mathbf{v})]g_{\mathbf{v}}^{jk}\frac{\partial \xi^i}{\partial v^k}\right)=0\ , \quad i,j,k=1,\dots n\ ,$$

where $J=\det\left\{\partial s^i/\partial v^j\right\}$. This very system in the parametric coordinates $v^1,\dots,v^n$ should be solved in order to obtain the same grid computed by the solution of the boundary value (9.128) in the parametric coordinates $s^1,\dots,s^n$.

9.5.4 Inverted Beltrami and Diffusion Equations

The numerical grid on the hypersurface S^{xn} is built by mapping a reference grid in the computational domain Ξ^n with the coordinate transformation $\boldsymbol{x}[\boldsymbol{s}(\boldsymbol{\xi})] : \Xi^n \to S^{xn}$. Practically, the grid is first defined in the parametric domain S^n by mapping the grid in Ξ^n on S^n with the use of the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$. Then, the grid in S^n is mapped on S^{xn} by the parametrization $\boldsymbol{x}(\boldsymbol{s})$, thus forming a grid on S^{xn}. Note that, in order to find the grid nodes in S^n through the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$, the inverse of which is a solution of the problem (9.126) or (9.128), there is no necessity to compute the transformation $\mathbf{s}(\boldsymbol{\xi})$ at all points $\boldsymbol{\xi} \in \Xi^n$. It is sufficient to solve numerically the boundary value problem obtained by interchanging dependent and independent variables in (9.126) or (9.128), i.e., considering the variables $\xi^1, \dots, \xi^n$ as independent while considering the variables $s^1, \dots, s^n$ of the parametric domain S^n as dependent. This change of dependent and independent variables yields, from (9.126) and (9.128), nonlinear elliptic equations with respect to the intermediate function $\mathbf{s}(\boldsymbol{\xi})$. We shall refer to these transformed equations as inverted equations.

The inverted problem with respect to the components $s^i(\boldsymbol{\xi})$, $i = 1, \dots, n$, of the intermediate transformation $\mathbf{s}(\xi)$ should be solved on the reference grid in Ξ^n. The values of this numerical solution

$$\mathbf{s}(\boldsymbol{\xi}) = [s^1(\boldsymbol{\xi}), \dots, s^n(\boldsymbol{\xi})]$$

at the points of the reference grid determine grid nodes in S^n, and consequently on S^{xn}, by mapping them through the parametrization $\mathbf{x}(\mathbf{s}) : S^n \to S^{xn}$.

Similar to (9.53), it is readily shown, from the equations in (9.126), that the general inverted Beltrami equations with respect to the control metric g^s_{ij} are as follows:

$$B^{\xi}_n[s^l] = g^{\xi} \Delta_B[s^l] \ , \quad l = 1, \dots n \ , \tag{9.130}$$

where

$$\begin{aligned} B^{\xi}_n[s^l] &= g^{\xi} g^{ij}_{\xi} \frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} \ , \quad i, j, l = 1, \dots n \ , \\ g^{\xi} &= \det\{g^{\xi}_{ij}\} = g^s (J)^2 = 1/\det\{g^{ij}_{\xi}\} \ , \\ \Delta_B[s^l] &= \frac{1}{\sqrt{g^s}} \frac{\partial}{\partial s^j} (\sqrt{g^s} g^{lj}_s) \ , \quad i, j, l = 1, \dots n \ , \end{aligned}$$

$J = \det\{\partial s^i / \partial \xi^j\}$, $g^{\xi}_{ij} (g^{ij}_{\xi})$ are the covariant (contravariant) elements of the control metric in the grid coordinates $\xi^1, \dots, \xi^n$, i.e.,

$$g^{\xi}_{ij} = g^s_{kl} \frac{\partial s^k}{\partial \xi^i} \frac{\partial s^l}{\partial \xi^j} \ , \quad g^{ij}_{\xi} = g^{kl}_s \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^l} \ , \quad i, j, k, l = 1, \dots n \ . \tag{9.131}$$

Notice that the operator B_n^{ξ} is, in fact, the operator $B_n^{x\xi}$ (see (9.52) and (9.54)) if $g_{ij}^{\xi} = g_{ij}^{x\xi}$, while it is the operator $B_n^{r\xi}$ (see (9.57)) if $g_{ij}^{\xi} = g_{ij}^{r\xi}$, and the operator B_n in the formula (6.18) if $g_{ij}^{\xi} = g_{ij}$.

Consequently, the boundary value problem (9.126) is transformed into the following inverted problem with respect to the components $s^i(\boldsymbol{s})$ of the intermediate function $\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^n \to S^n$:

$$\begin{aligned} &B_n^{\xi}[s^i] = (J)^2\sqrt{g^s}\frac{\partial}{\partial \xi^k}(\sqrt{g^s}g_s^{ij})\frac{\partial \xi^k}{\partial s^j} \ , \quad i, j, k = 1, \ldots n \ , \\ &s^i(\boldsymbol{\xi})|_{\partial \Xi^n} = \psi^i(\boldsymbol{\xi}) \ , \quad i = 1, \ldots, n \ , \end{aligned} \tag{9.132}$$

where $\boldsymbol{\psi}(\boldsymbol{\xi}) : \partial \Xi^n \to \partial S^n$, $\boldsymbol{\psi}(\boldsymbol{\xi}) = [\psi^1(\boldsymbol{\xi}), \ldots, \psi^n(\boldsymbol{\xi})]$, is the inverse of $\boldsymbol{\varphi}(\mathbf{s}) : \partial S^n \to \partial \Xi^n$.

In the one-dimensional case, the equation in (9.132) is equivalent, as for (9.50), to the following equation in a divergent form

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\sqrt{g^s}\frac{\mathrm{d}s}{\mathrm{d}\xi}\right) = 0 \ . \tag{9.133}$$

Similar to (9.132), the Dirichlet boundary value problem obtained from (9.128) for the inverted diffusion equations is written out:

$$\begin{aligned} &B_n^{\xi}[s^i] = \frac{g^s(J)^2}{w(s)}\frac{\partial}{\partial \xi^k}(w(s)g_s^{ij})\frac{\partial \xi^k}{\partial s^j} \ , \quad i, j, k = 1, \ldots n \ , \\ &s^i(\boldsymbol{\xi})|_{\partial \Xi^n} = \psi^i(\boldsymbol{\xi}) \ , \quad i = 1, \ldots, n \ . \end{aligned} \tag{9.134}$$

Substitution in (9.133) $w(s)$ for $\sqrt{g^s}$ gives a divergent form of the inverted diffusion equation in (9.134) for $n = 1$.

The inverted diffusion equations in (9.134) are also transformed into the following divergent form:

$$\frac{\partial}{\partial \xi^j}\{Jw[\mathbf{s}(\boldsymbol{\xi})]g_{\xi}^{ij}\} = 0, \quad i, j = 1, \ldots, n, \tag{9.135}$$

obtained by applying the identity (2.48) to the equations in (9.128). Substituting in (9.135) $\sqrt{g^s}$ for $w(\boldsymbol{s})$ gives a divergent form for the inverted Beltrami equations, namely,

$$\frac{\partial}{\partial \xi^j}(\sqrt{g^{\xi}}g_{\xi}^{ij}) = 0, \quad i, j = 1, \ldots, n \ . \tag{9.136}$$

Another formulation of a grid model through the Beltrami equations with respect to the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$,

$$\begin{aligned} &\frac{\partial}{\partial \xi^j}\left(\sqrt{g}g^{jk}\frac{\partial s^i}{\partial \xi^k}\right) = 0 \ , \quad i, j, k = 1, \ldots, n \ , \\ &s^i(\boldsymbol{\xi})|_{\Xi^n} = \psi^i(\boldsymbol{\xi}) \ , \end{aligned} \tag{9.137}$$

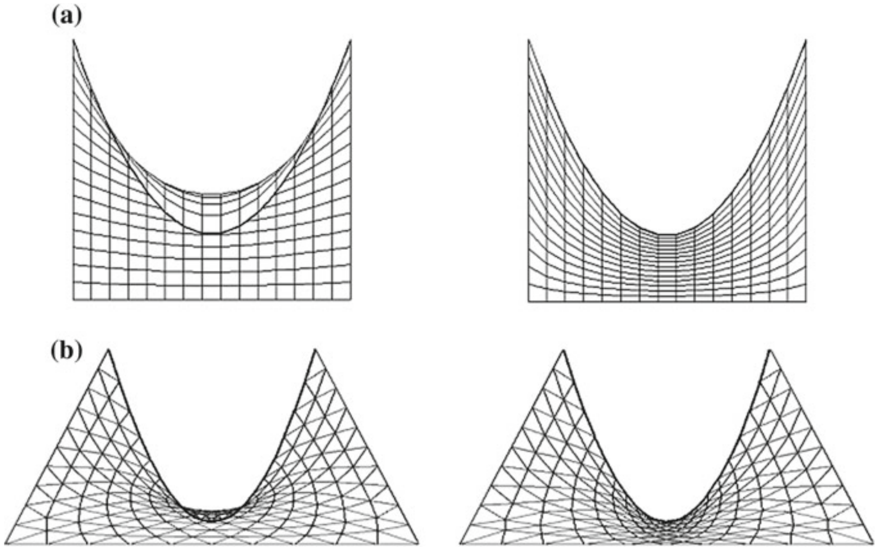


Fig. 9.10 Quadrilateral (**a**) and triangular (**b**) grids in concave domains generated by the solution of equations (9.137) (*left*) and by the solution of equations in (9.132) (*right*), both with respect to the Euclidean metric

where g^{jk} are the contravariant metric components of a monitor manifold over Ξ^n, $g = \det\{g_{jk}\} = 1/\det\{g^{jk}\}$, was proposed by Godunov and Prokopov (1967), and Ryskin and Leal (1983). The equations in (9.137) seem to be more natural than the nonlinear inverted Beltrami equations in (9.132) obtained from the Beltrami equations in (9.126) for the implementation into numerical codes, since they are linear and of divergent form with respect to the intermediate transformation $\mathbf{s}(\xi)$. However, such a divergent model, owing to the maximum principle, does not guarantee that all grid points will be inside of the physical geometry S^{xn} when the parametric domain S^n is not convex (see Fig. 9.10, left-hand); moreover, the grid cells may be folded. Providing grid nondegeneracy through the solution of equations in (9.137) depends on devising a suitable metric in Ξ^n, which hasn't been done so far in a general form.

The considerations mentioned are such that it is more reasonable to make the formulation of grid systems through the Beltrami equations with respect to the function $\boldsymbol{\xi}(\mathbf{s}) : S^n \to \Xi^n$, the inverse of which yields the intermediate transformation $\mathbf{s}(\boldsymbol{\xi}) : \Xi^n \to S^n$.

9.5.5 Specification of Individual Control Metrics

Control Metric for Generating Field-Aligned Grids

The contravariant metric tensor in the form (9.118) allows one to control the angle between the normal to a grid coordinate hypersurface in S^{xn} and a specified vector field, in particular, to generate vector field-aligned grids. As a tensor of the first rank in the formula (9.118), one may take either the same or a transformed vector field.

The need for a vector field-aligned coordinate system can be appreciated in the case $S^{xn} = S^n$ by considering the heat equation in a magnetized plasma, written in the grid coordinates ξ^i,

$$\frac{\partial T}{\partial t} = \nabla \cdot (\chi \cdot \nabla T) = \frac{1}{J}\frac{\partial}{\partial \xi^i}\Big(J\chi : \nabla\xi^i \nabla\xi^j \frac{\partial T}{\partial \xi^j}\Big), \quad i, j = 1, 2, 3. \tag{9.138}$$

By far, the largest component of the anisotropic thermal conductivity tensor χ equals $\chi_{\|}\mathbf{bb}$, where $\chi_{\|}$ is parallel conductivity and $\mathbf{b}$ is the unit vector along the magnetic field $\mathbf{B} = (B^1, \dots, B^n)$. If $\mathbf{b} \cdot \nabla\xi^i$ is of order unity for all grid coordinates ξ^i, then the much smaller transverse terms in (9.138) involve the difference between large terms, resulting in a loss of numerical accuracy. If it vanishes, or nearly so, for all but one coordinate, this inaccuracy can be avoided. Similar considerations hold for other manifestations of magnetic anisotropy.

For simple magnetic fields, such as those in the core region of the tokamak, it is possible to define a flux coordinate ψ labeling the magnetic surfaces, satisfying $\mathbf{B} \cdot \nabla\psi = 0$ exactly. In more complicated cases, such as nonaxisymmetric magnetic fields with multiple islands and regions of stochasticity, this is not possible. For example, the condition of orthogonality between a vector field $\mathbf{B} = (B^1, \dots, B^n)$ specified at the points of a domain S^n and a normal to the coordinate hypersurface $\xi^1 = const$ can be described as an equation for a quadratic form

$$(\mathbf{B} \cdot \mathrm{grad}\xi^1)^2 \equiv B^i B^j \frac{\partial \xi^1}{\partial s^i}\frac{\partial \xi^1}{\partial s^j} = 0, \quad i, j = 1, \dots, n,$$

with a degenerate matrix $\{B^i B^j\}$ $(\det\{B^i B^j\} = 0)$. This quadratic form as a measure of grid departure from field-alignment was used in Glasser and Tang (2004) for generating nearly field-aligned grids in the domain S^n through the minimization of the functional

$$L = \int_{S^n} B^i B^j \frac{\partial \xi^1}{\partial s^i}\frac{\partial \xi^1}{\partial s^j} d\mathbf{s}\,, \quad i, j = 1, \dots, n. \tag{9.139}$$

The integrand in the functional of energy (9.120) is formulated as the sum of quadratic forms

$$g^{jk}_{\mathbf{s}} \frac{\partial \xi^i}{\partial s^k}\frac{\partial \xi^i}{\partial s^j}, \quad i, j, k = 1, \dots, n, \quad i \text{ fixed},$$

multiplied by $\sqrt{g^{\mathrm{s}}}$, but contrary to (9.139), with a nondegenerate matrix $\{g^{jk}_{\mathbf{s}}\}$. The condition of non-degeneracy is indispensable for obtaining unfolded grids through the minimization of the functional (9.120), and consequently through the solution of boundary value problem (9.132). The factor $\sqrt{g^{\mathrm{s}}}$ provides the independence of grids of parametrizations. Therefore, in order to get grids which are both nearly field-aligned and unfolded, we have to change slightly the matrix $\{B^iB^j\}$ in the functional (9.139) to make it nondegenerate. The matrix $\{g^{ij}_{\mathbf{s}}\}$ whose elements are specified in the form (9.119) is nondegenerate for an arbitrary $\epsilon(\mathbf{s}) > 0$; in addition, this matrix is close to the matrix $\{B^iB^j\}$ when both $\epsilon(\mathbf{s})$ and $\mathbf{B}_k,\ k = 2, \dots, l$ are small and $\mathbf{B}_1 = \mathbf{B}$. Assuming this matrix

$$g^{ij}_{\boldsymbol{s}} = \epsilon(\boldsymbol{s})\delta^i_j + B^iB^j + B^i_mB^j_m\ , \quad i, j = 1, \dots, n, \quad m = 2, \dots, l\ , \tag{9.140}$$

as a contravariant metric tensor of a monitor manifold M^n over S^n yields that the functional of energy (9.120) is as follows:

$$\begin{aligned} I[\xi] = \int\limits_{S^n} \frac{1}{\sqrt{g_{\mathbf{s}}}}\Big[\epsilon(\mathbf{s})\frac{\partial \xi^i}{\partial s^m}\frac{\partial \xi^i}{\partial s^m} + B^j_kB^p_k\frac{\partial \xi^i}{\partial s^j}\frac{\partial \xi^i}{\partial s^p}\Big]d\mathbf{s}\ ,\\ i, j, m, p = 1, \dots, n, \quad k = 1, \dots, l\ , \end{aligned} \tag{9.141}$$

where $g_{\mathbf{s}} = det\{g^{ij}_{\mathbf{s}}\} = 1/g^{\mathrm{s}}$. In particular, for $l = 2$,

$$g_{\mathbf{s}} = \epsilon^{n-2}(\mathbf{s})\big\{[\epsilon(\mathbf{s}) + |\mathbf{B}_1|^2][\epsilon(\mathbf{s}) + |\mathbf{B}_2|^2] - (\mathbf{B}_1 \cdot \mathbf{B}_2)^2\big\}, \quad l = 2.$$

Consequently, equations in (9.132) aimed at the generation of grids providing that the angle between $\mathbf{B}$ and a normal to a coordinate hypersurface is close to $\pi/2$ have the form

$$g^{km}_{\xi}\frac{\partial^2 s^i}{\partial \xi^k \partial \xi^m} = \sqrt{g_{\mathbf{s}}}\frac{\partial}{\partial s^j}\Big(\frac{\epsilon(\mathbf{s})\delta^i_j + B^i_aB^j_a}{\sqrt{g_{\mathbf{s}}}}\Big), \quad i, j, k, m = 1, \dots, n, \quad a = 1, \dots, l. \tag{9.142}$$

This contravariant metric tensor of a monitor manifold over a domain S^n yields, in accordance with (9.125), the following measure of grid nonalignment with the vector field $\boldsymbol{B}$

$$\begin{aligned} m(\boldsymbol{s}) = w(\boldsymbol{s})[\epsilon(\boldsymbol{s})\delta^j_k + B^jB^k + B^j_mB^k_m]\frac{\partial \xi^i}{\partial s^j}\frac{\partial \xi^i}{\partial s^k}\ ,\\ i, j, k = 1, \dots, n, \quad m = 2, \dots, l\ . \end{aligned}$$

Figure 9.11 demonstrates a grid (right-hand) with one family of grid lines orthogonal to a vector field (left-hand) by solving Eq. (9.142) for $n = 2$ and $l = 2$.

Similarly, one can define a control metric and equations for generating grids with the angle that is close to 0.

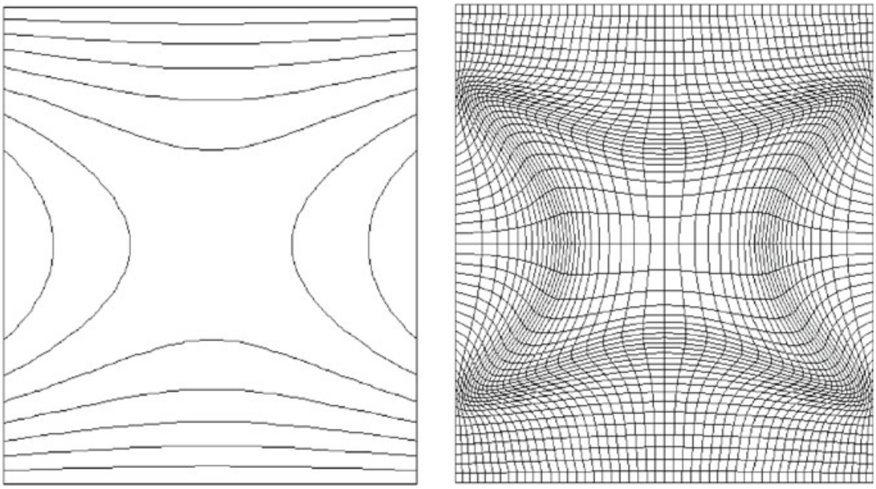


Fig. 9.11 Isocontours of a vector field (*left*), quadrangular field-aligned grid (*right*)

While this approach to grid alignment with the magnetic field was motivated by the work of Brackbill (1993), there is a substantial difference. Whereas Brackbill has a variational principle similar to (9.120) or (9.121), his integrand is designed to make a coordinate gradient *parallel* to a specified vector field; the approach described above is designed to make a coordinate gradient *perpendicular* to a specified vector field.

Control Metric for Generating Grids Adapting to the Values of a Function

For generating a numerical grid on a physical geometry S^{xn} with node clustering in the zones of large values of a function $|v(\boldsymbol{s})|$, the measure of departure from the necessary grid in S^{xn} can be expressed in the form

$$m(\boldsymbol{s}) = Z[v](\boldsymbol{s}) g_{sx}^{km} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^m}, \quad i, j, k, m = 1, \dots, n, \tag{9.143}$$

where $Z[v] > 0$ is a positive operator such that $Z[v](\boldsymbol{s})$ is large (small) where $|v(\boldsymbol{s})|$ is small (large), for example,

$$Z[v](\boldsymbol{s}) = \frac{1}{c_1 |v(\boldsymbol{s})| + c_2}, \quad c_1 > 0\,, \quad c_2 > 0\,. \tag{9.144}$$

So, the contravariant elements of the control metric are as follows:

$$g_s^{ij} = Z[v](\boldsymbol{s}) g_{sx}^{ij}, \quad i, j = 1, \dots, n. \tag{9.145}$$

Consequently, the covariant elements of the control metric are written out as

$$g^s_{ij} = \frac{1}{Z[v](\boldsymbol{s})} g^{xs}_{ij} , \quad i, j = 1, \ldots, n , \tag{9.146}$$

and therefore $g^s = g^{xs}(Z[v])^{-n/2}$. So, the Beltrami equations in (9.126) with respect to the metric (9.146) are expressed in the form

$$\frac{\partial}{\partial s^j}\Big((Z[v])^{(2-n)/2}\sqrt{g^{xs}} g^{jk}_{sx} \frac{\partial \xi^i}{\partial s^k}\Big) = 0 , \quad i, j = 1, \ldots, n , \tag{9.147}$$

while the inverted Beltrami equations with respect to the metric (9.146) are as follows:

$$B^{x\xi}_n[s^k] = J^2\sqrt{g^{xs}}(Z[v])^{(n-2)/2}\frac{\partial}{\partial s^j}((Z[v])^{(2-n)/2}\sqrt{g^{xs}} g^{jk}_{xs}) , \quad j, k = 1, \ldots, n , \tag{9.148}$$

where

$$B^{x\xi}_n[y] = g^{x\xi} g^{ij}_{\xi x} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j} , \quad i, j = 1, \ldots, n .$$

In particular, if S^{xn} coincides with S^n, i.e., $g^{xs}_{ij} = \delta^i_j$, and consequently $g^{xs} = 1$, $g^{ij}_{sx} = \delta^i_j$,

$$g^s_{ij} = \frac{1}{Z[v](\boldsymbol{s})} \delta^i_j , \quad i, j = 1, \ldots, n , \tag{9.149}$$

then the Beltrami equations (9.147) are as follows:

$$\frac{\partial}{\partial s^j}\Big((Z[v])^{(2-n)/2} \frac{\partial \xi^i}{\partial s^j}\Big) = 0 , \quad i, j = 1, \ldots, n . \tag{9.150}$$

This system of equations is equivalent to the system of Poisson equations

$$\frac{\partial^2 \xi^i}{\partial s^j \partial s^j} = P^i , \quad i = 1, \ldots, n ,$$

where

$$P^i = \frac{n-2}{2}\frac{1}{Z[v]}\frac{\partial Z[v](\boldsymbol{s})}{\partial s^j}\frac{\partial \xi^i}{\partial s^j} = \frac{n-2}{2}\frac{1}{Z[v]}\frac{\partial Z[v](\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^m} g^{im}_{\xi x} , \quad i, j, m = 1, \ldots, n .$$

Consequently, the inverted Beltrami equations in the control metric (9.149) are as follows:

$$B^{x\xi}_n[s^k] = \frac{2-n}{2} J^2 \frac{1}{Z[v]} \frac{\partial}{\partial s^k} Z[v](\boldsymbol{s}) , \quad k = 1, \ldots, n . \tag{9.151}$$

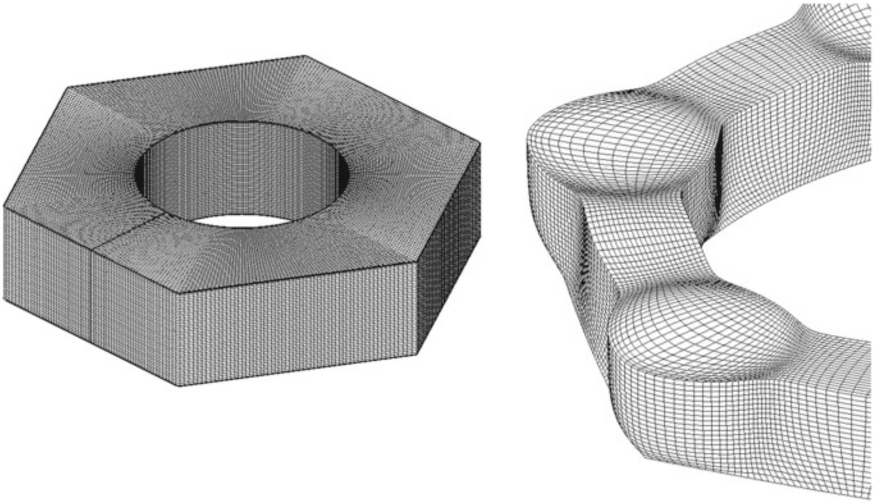


Fig. 9.12 Three-dimensional domain (*left*) and a fragment of an adaptive grid (*right*) in the interior of the domain

Figure 9.12 illustrates an adaptive three-dimensional grid obtained by the numerical solution of equations (9.151). The grid nodes cluster in the zones of large values of a function $v(\boldsymbol{s})$.

The diffusion equations in (9.128) with respect to the metric (9.146) are expressed as

$$\frac{\partial}{\partial s^j}\Big(w(\boldsymbol{s})Z[v]g^{jk}_{sx}\frac{\partial \xi^i}{\partial s^k}\Big) = 0 \;, \quad i, j = 1, \dots, n \;. \tag{9.152}$$

So, the inverted diffusion equations in this metric, necessary to generate adaptive grids, are

$$B^{x\xi}_n[s^k] = \frac{J^2 g^{xs}}{w(\boldsymbol{s})Z[v]}\frac{\partial}{\partial s^j}(w(\boldsymbol{s})Z[v]g^{jk}_{xs}) \;, \quad j, k = 1, \dots, n \;. \tag{9.153}$$

In particular, when $S^{xn} = S^n$, $w(\boldsymbol{s}) = 1$, then these equations are

$$B^{s\xi}_n[s^k] = J^2\frac{1}{Z[v]}\frac{\partial}{\partial s^k}Z[v](\boldsymbol{s}) \;, \quad k = 1, \dots, n \;, \tag{9.154}$$

where

$$B_n^{s\xi}[y] = g^{s\xi} g_{\xi s}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j} \ , \quad i, j = 1, \dots, n \ ,$$
$$g^{s\xi} = \det\{g_{ij}^{s\xi}\} = J^2 \ , \quad g_{ij}^{s\xi} = \frac{\partial \boldsymbol{s}}{\partial \xi^i} \cdot \frac{\partial \boldsymbol{s}}{\partial \xi^j} = \frac{\partial s^k}{\partial \xi^i} \frac{\partial s^k}{\partial \xi^j} \ , \quad i, j, k = 1, \dots, n \ ,$$
$$g_{\xi s}^{ij} = \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^k} \ , \quad i, j, k = 1, \dots, n \ . \tag{9.155}$$

Equations (9.152)–(9.154) for $n = 2$ are more efficient for adaptation than inverted Beltrami equations (9.148) and (9.151), which for $n = 2$ do not depend on the operator $Z[v]$.

The control metric (9.149) in Eq. (9.154) allows one to generate grids with node clustering near the surface in S^{xn} specified by the formula $\phi(\boldsymbol{s}) = 0$. Such grid clustering is provided by assuming in (9.149) $Z[v](\boldsymbol{s}) = v(\phi(\boldsymbol{s}))$, for a function $v(t)$, subject to the following restrictions $v(t) > 0$, $v(t_1) > v(t_2)$ if $t_1 > t_2$. If a set A in S^{xn} is defined by a discrete number of points $\boldsymbol{s}_1, \dots \boldsymbol{s}_N$, then $Z[v](\boldsymbol{s})$ can be identified with $v(\rho(\boldsymbol{s}, \ A))$, where

$$\rho(\boldsymbol{s}, \ A) = \min_{i=1,\dots,N} |\boldsymbol{s} - \boldsymbol{s}_i|.$$

Figure 9.13 for $v(t) = t^2 + 10^{-5}$ illustrates such a grid with node clustering near the boundary of a carotis.

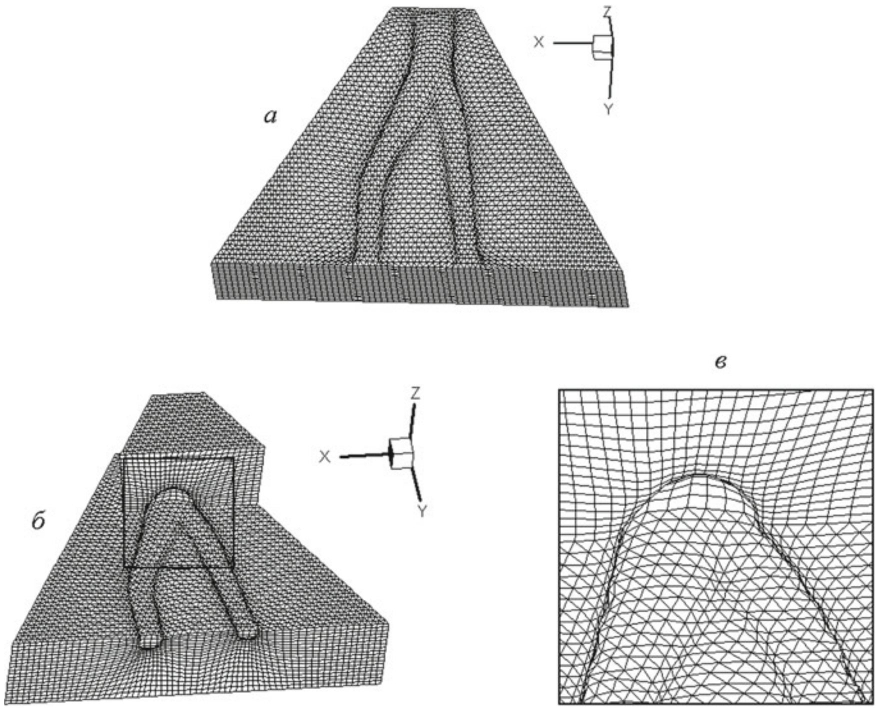


Fig. 9.13 Prismatic grid with node clustering near the boundary of a carotis

General Diagonal Metric

For a more general control metric with a diagonal tensor in the parametric coordinates $s^1, \dots, s^n$,

$$g^{\mathbf{s}}_{ij} = v^k(\mathbf{s})\delta^k_i\delta^k_j \ , \quad i, j, k = 1, \dots, n \ , \quad v^k(\mathbf{s}) > 0 \ , \tag{9.156}$$

i.e.,

$$g^{\mathbf{s}}_{ij} = \begin{pmatrix} v^1(\mathbf{s}) & 0 & \dots & 0 \\ 0 & v^2(\mathbf{s}) \; 0 \dots & & 0 \\ \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots \\ 0 & \dots\dots & 0 & v^n(\mathbf{s}) \end{pmatrix} ,$$

we get

$$g^{\mathbf{s}} = \prod_{k=1}^{n} v^k(\mathbf{s}) \ , \quad g^{ij}_{\mathbf{s}} = \frac{1}{v^k(\mathbf{s})}\delta^k_i\delta^k_j \ , \quad i, j, k = 1, \dots, n \ .$$

So, the Beltrami equations in (9.126) with respect to metric (9.156), are, in fact, the following equations:

$$\frac{\partial}{\partial s^j}\Big(\frac{\sqrt{g^{\mathbf{s}}}}{v^j(\mathbf{s})}\frac{\partial \xi^i}{\partial s^j}\Big) = 0 \ , \quad i, j = 1, \dots, n \ . \tag{9.157}$$

In particular, if $n = 2$, Eq. (9.157) are as follows:

$$\frac{\partial}{\partial s^1}\Big(F(\mathbf{s})\frac{\partial \xi^i}{\partial s^1}\Big) + \frac{\partial}{\partial s^2}\Big(\frac{1}{F(\mathbf{s})}\frac{\partial \xi^i}{\partial s^2}\Big) = 0 \, , \quad i = 1, 2 \, , \tag{9.158}$$

where

$$F(\mathbf{s}) = \sqrt{v^2(\mathbf{s})/v^1(\mathbf{s})} \ .$$

The inverted Beltrami grid equations with respect to metric (9.156) are also equivalent to the following equations:

$$g^{\xi} g^{ij}_{\xi} \frac{\partial}{\partial \xi^i}\Big(\frac{v_k(\mathbf{s})}{\sqrt{g^{\mathbf{s}}}}\frac{\partial s^k}{\partial \xi^j}\Big) = 0 \ , \quad i, j, k = 1, \dots, n \ , \ k \text{ fixed.} \tag{9.159}$$

Control Metrics for Generating Grids Adapting to the Gradient of a Function

The contravariant metric tensor (9.145) can also be used for providing node clustering in the zones of large variations of a function $\boldsymbol{f}(\boldsymbol{s}) = (f^1(\boldsymbol{s}), \dots, f^l(\boldsymbol{s}))$, introducing for this purpose an operator $Z[\text{grad } \boldsymbol{f}]$ such that $Z[\text{grad } \boldsymbol{f}](\boldsymbol{s})$ is large where $|\text{grad } \boldsymbol{f}|(\boldsymbol{s})$ is small, and vice versa. In particular, for grid clustering, we may assume

$$Z[\text{grad } \boldsymbol{f}](\boldsymbol{s}) = \frac{1}{1 + c_1 \mid \text{grad } \boldsymbol{f}(\boldsymbol{s}) \mid^{\alpha}} \ , \quad c_1 > 0 \ , \quad \alpha > 0 \ . \tag{9.160}$$

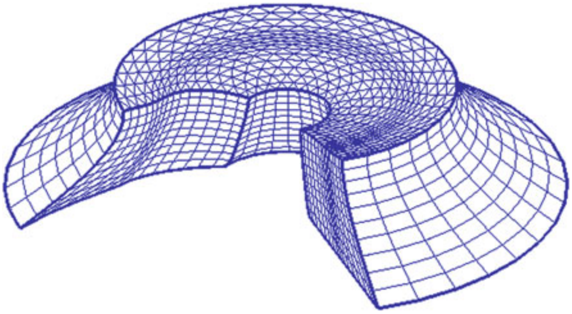


Fig. 9.14 Three-dimensional prismatic adaptive grid

Figure 9.14 illustrates an adaptive prismatic grid with node clustering in such zones.

One more control metric for generating grids with node clustering in the zones of the large variation of the function $\boldsymbol{f}(\boldsymbol{s})$ is of the form

$$g^{s}_{ij} = \varepsilon(\boldsymbol{s})g^{xs}_{ij} + \varepsilon_k(\boldsymbol{s})\frac{\partial f^k}{\partial s^i}\frac{\partial f^k}{\partial s^j} , \quad i, j = 1, \dots, n , \quad k = 1, \dots, l , \tag{9.161}$$

where $\varepsilon(\boldsymbol{s}) \geq 0$, $\varepsilon_k(\boldsymbol{s}) \geq 0$, $k = 1, \dots, l$. This control metric is a generalization of the metric of a monitor hypersurface (see formulas (9.9) and (9.10)).

9.5.6 Control Metrics for Generating Grids with Balanced Properties

For computing numerical grids that are field-aligned and adaptive to the values of one function and/or to the variation of another function, a natural way for defining a control metric consists in combining the corresponding metrics, i.e., the covariant elements of the balanced control metric are to have a form

$$g^{s}_{ij}(\boldsymbol{s}) = \varepsilon_1(\boldsymbol{s})g^{al}_{ij} + \varepsilon_2(\boldsymbol{s})g^{adg}_{ij} + \varepsilon_3(\boldsymbol{s})g^{adv}_{ij}, \quad i, j = 1, \dots, n , \tag{9.162}$$

where $\epsilon_k(\boldsymbol{s}) \geq 0$, $k = 1, 2, 3$ are the weight functions specifying the contribution of the covariant elements g^{al}_{ij}, g^{adg}_{ij}, and g^{adv}_{ij}. The marks *al*, *adg* and *adv* in this formula mean that the corresponding metric elements are chosen for grid alignment, adaptation to gradients of a function, and adaptation to the values of the same of another function, respectively. It is evident that (9.162) will be a covariant metric tensor if

$$\varepsilon_1(\boldsymbol{s}) + \varepsilon_2(\boldsymbol{s}) + \varepsilon_3(\boldsymbol{s}) > 0 , \quad \boldsymbol{s} \in S^n .$$

Analogously, there is written out a formula for the contravariant components g^{ij}_s of the balanced control metric

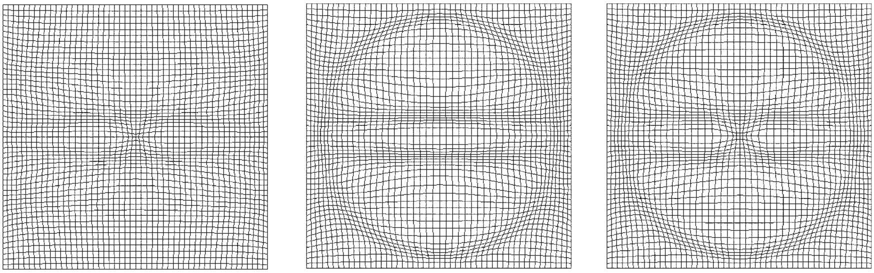


Fig. 9.15 Examples of balanced numerical grids

$$g_s^{ij}(\boldsymbol{s}) = w_1(\boldsymbol{s})g_{al}^{ij} + w_2(\boldsymbol{s})g_{adg}^{ij} + w_3(\boldsymbol{s})g_{adv}^{ij}\,, \quad i, j = 1, \dots, n\,. \tag{9.163}$$

For computing the balanced numerical grids that are field-aligned and adaptive to the values of one function and to the variations of another function, formulas (9.118) and (9.163) were used for the contravariant metric elements of the control metric, written in the following form:

$$g_s^{ij} = (1-\alpha)g_{al}^{ij} + \alpha\Big((1-\beta)g_{adg}^{ij} + \beta g_{adv}^{ij}\Big), \quad i, j = 1, \dots, n, \tag{9.164}$$

$$\begin{aligned} &g_{al}^{ij} = \delta_j^i \varepsilon(\boldsymbol{s}) + B^i B^j, \quad g_{adv}^{ij} = \delta_j^i f_1(\varphi_1), \\ &g_{adg}^{ij} = \delta_j^i - \frac{1}{1 + |grad\ f_2(\varphi_2)|^2}\frac{\partial f_2(\varphi_2)}{\partial s^i}\frac{\partial f_2(\varphi_2)}{\partial s^j}. \end{aligned}$$

Some two-dimensional balanced grids in a square domain X^2 are shown in Fig. 9.15. These grids were generated through the solution of equations (9.153) for $n = 2$ by a finite-difference algorithm. The left-hand picture of the figure demonstrates the grid aligned to a vector-field $\boldsymbol{B} = (B^1,\ B^2)$ and adapted to the values of a function $\varphi_1(\boldsymbol{s})$. The center picture demonstrates the grid aligned to the same vector-field and adapted to the gradients of a function $f_2[\varphi_2(\boldsymbol{s})]$. The right-hand picture demonstrates the grid aligned to the same vector-field and adapted to the values of one function and the gradients of another. These grids were generated with the help of the functions and parameters in (9.164) for $n = 2$ specified as follows:

$$\begin{aligned} &\boldsymbol{B} = \Big(-\frac{\partial\psi(\boldsymbol{s})}{\partial s^2}\,, \quad \frac{\partial\psi(\boldsymbol{s})}{\partial s^1}\Big), \\ &f_1(\varphi_1) = \Big(\frac{0.6}{0.6 + \varphi_1}\Big)^3, \quad \varphi_1(\boldsymbol{s}) = \Big(\frac{0.01}{0.01 + R^2}\Big)^5, \\ &f_2(\varphi_2) = 0.05\tanh\Big(\frac{\varphi_2}{0.03}\Big), \quad \varphi_2(\boldsymbol{s}) = R^2 - 0.2, \\ &R^2 = (s^1 - 0.5)^2 + (s^2 - 0.5)^2, \\ &\psi(\boldsymbol{s}) = v(s^2)(1 - v(s^2))[(s^1 - 0.5)^2 + 2(v(s^2) - 0.5)^2], \\ &v(s^2) = 0.5\Big[1 + \tanh\Big(\frac{s^2 - 0.5}{0.2}\Big)\Big], \end{aligned}$$

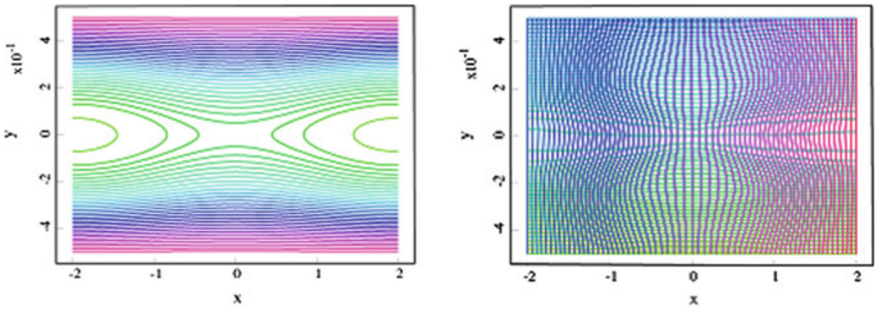


Fig. 9.16 Magnetic vector field flux (*left*) and a balanced grid (*right*)

$$
\begin{aligned}
&1)\ \ \alpha = \varepsilon(\boldsymbol{s}) = \left(\frac{0.3}{0.3 + |\boldsymbol{B}|^2}\right)^6, \quad \beta = 1,\\
&2)\ \ \alpha = \varepsilon(\boldsymbol{s}) = \left(\frac{0.3}{0.3 + |\boldsymbol{B}|^2}\right)^8, \quad \beta = 0,\\
&3)\ \ \alpha = \varepsilon(\boldsymbol{s}) = \left(\frac{0.3}{0.3 + |\boldsymbol{B}|^2}\right)^5, \quad \beta = exp(-(f_1)^2/0.1).
\end{aligned}
$$

There may be other effective ways for combining the corresponding tensor components; in particular, for generating grids that are field-aligned and adaptive to the values of a function $v(\boldsymbol{s})$, adequate results are demonstrated with the help of the following formula for the contravariant control metric elements:

$$
g_s^{ij}(\boldsymbol{s}) = Z[v](\boldsymbol{s}) g_{al}^{ij}\,, \quad i, j = 1, \ldots, n\,. \tag{9.165}
$$

This metric provides a compromise between alignment and adaptation. Figure 9.16 illustrates both the integral lines of a two-dimensional magnetic field (left-hand) and such a balanced grid aligned to the magnetic field and adapted to the numerical error (right-hand) via the metric (9.165).

9.6 Comments

Functionals of nonconformality for generating grids on hypersurfaces were formulated, in analogy to the functional (8.33) or (8.37), by Liseikin (1991, 1999).

The methods based on the utilization of control metrics in the energy and diffusion functionals were formulated and justified by Liseikin (1991, 1992, 1993, 1999). All properties and interpretations of the functionals and the corresponding Beltrami and diffusion equations described here were studied and published by Liseikin (1991, 1992, 1993, 1996, 1999, 2004, 2005, 2007). The formulas of the general control metrics were proposed by Liseikin (2004, 2007). An application of the geometry of manifolds and the control metrics for generating adaptive grids was performed by Liseikin (2004, 2005, 2007).

Implementations of the inverted Beltrami and diffusion equations with respect to the control metric in numerical codes for generating adaptive grids in domains and on surfaces were performed by Kupin and Liseikin (1994), Liseikin and Petrenko (1994), Liseikin (2004), Shokin et al. (2005), Glasser et al. (2006), Liseikin (2007), Liseikin et al. (2007).

The measure (9.143) for generating adaptive grids in domains was introduced by Danaev et al. (1980) and Winslow (1981). The generation of grids through such a metric is helpful for numerically solving problems with strong anisotropy, and problems of magnetized plasmas, in particular.

References

Azarenok, B. N., & Ivanenko, S. A. (2001). Application of moving adaptive grids for simulation of supersonic gas flow. *Computational Fluid Dynamics Journal Japanese*, *10*(3), 400–404.

Brackbill, J. U. (1993). An adaptive grid with directional control. *Journal of Computational Physics*, *108*, 38–50.

Brackbill, J. U., & Saltzman, J. (1982). Adaptive zoning for singular problems in two directions. *Journal of Computational Physics*, *46*, 342–368.

Charakch'yan, A. A., & Ivanenko, S. A. (1997). A variational form of the Winslow grid generator. *Journal of Computational Physics*, *136*, 385–398.

Crowley, W. P. (1962). An equipotential zoner on a quadrilateral mesh. Memo, Lawrence Livermore National Lab., 5 July 1962.

Danaev, N. T., Liseikin, V. D., & Yanenko, N. N. (1980). Numerical solution on a moving curvilinear grid of viscous heat-conducting flow about a body of revolution. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *11*(1), 51–61. (Russian).

Dvinsky, A. S. (1991). Adaptive grid generation from harmonic maps on Riemannian manifolds. *Journal of Computational Physics*, *95*, 450–476.

Fletcher, C. A. J. (1997). *Computational techniques for fluid dynamics 1: Fundamental and general techniques*. Berlin: Springer.

Fomenko, A. T., & Thi, D. T. (1991). *Minimal surfaces, stratified multivarifolds, and the Plateau problem*. New York: AMS.

Garanzha, V. A. (2000). Barrier variational generation of quasi-isometric grids. *Computational Mathematics and Mathematical Physics*, *40*, 1617–1637.

Glasser, A. H., & Tang, X. Z. (2004). The SEL macroscopic modeling code. *Computer Physics Communications*, *164*, 237–243.

Glasser, A. H., Liseikin, V. D., Shokin, Ju I, Vaseva, I. A., & Likhanova, Ju V. (2006). *Grid generation with the use of Beltrami and diffusion equations*. Novosibirsk: Nauka. (Russian).

Godunov, S. K., & Prokopov, G. P. (1967). Calculation of conformal mappings in the construction of numerical grids. *Journal of Computational Mathematics and Mathematical Physics*, *7*, 1031–1059. (Russian).

Ivanenko, S. A. (1997). Adaptive-harmonic grids. M., CC RAS.

Ivanenko, S. A., & Charakch'yan, A. A. (1988). An algorithm for constructing curvilinear grids consisting of convex quadrangles. *Soviet Mathematics Doklady*, *36*(1), 51.

Kovenya, V. M., Tarnavskii, G. A., & Chernyi, S. G. (1990). *Application of a splitting method to fluid problems*. Novosibirsk: Nauka. (in Russian).

Kupin, E. P., & Liseikin, V. D. (1994). A method of projection for generating multidimensional adaptive grids. *Journal of Computational Technologies*, *3*(8), 189–198. Novosibirsk (Russian).

Langtangen, H. P. (2003). *Computational partial differential equations: Numerical methods and diffpack programming*. Berlin: Springer.

Liseikin, V. D. (1991). On generation of regular grids on n-dimensional surfaces. *Journal Computational Mathematics and Mathematical Physics*, *31*, 1670–1689 (Russian). [English transl.: USSR *Computational Mathematics and Mathematical Physics*, *31*(11), 47–57 (1991)].

Liseikin, V. D. (1992). On a variational method of generating adaptive grids on n-dimensional surfaces. *Soviet Mathematics Doklady*, *44*(1), 149–152.

Liseikin, V. D. (1993). On some interpretations of a smoothness functional used in constructing regular and adaptive grids. *Russian Journal of Numerical Analysis and Mathematical Modelling*, *8*(6), 507–518.

Liseikin, V. D. (1996). Adaptive grid generation on the basis of smoothness functional. In B.K. Soni, J.F. Thompson, J. Hauser & P.R. Eiseman (Eds.) *Numerical grid generation in CFD* (vol. 2, pp. 1131–1140). Mississippi State University.

Liseikin, V. D. (1999). *Grid generation methods*. Berlin: Springer.

Liseikin, V. D. (2004). *A computational differential geometry approach to grid generation*. Berlin: Springer.

Liseikin, V. D. (2005). On a universal monitor metric for numerical grid generation. *Doklady Mathematics*, *71*(1), 15–19.

Liseikin, V. D. (2007). *A computational differential geometry approach to grid generation* (2nd ed.). Berlin: Springer.

Liseikin, V. D., Likhanova, Ju. V, & Shokin, Ju. I. (2007). *Numerical grids and coordinate transformations for solution of singular perturbation problems*. Novosibirsk: Nauka. (Russian).

Liseikin, V. D., & Petrenko, V. E. (1994). On analytical and numerical investigations of a projection method for the generation of adaptive grids. *Journal of Computational Technologies*, *3*(9), 108–120. Novosibirsk (Russian).

Ryskin, G., & Leal, L. G. (1983). Orthogonal mapping. *Journal of Computational Physics*, *50*, 71–100.

Shokin, Ju I, Liseikin, V. D., Lebedev, A. S., Danaev, N. T., & Kitaeva, I. A. (2005). *Method of Riemannian geometry for generating numerical grids*. Novosibirsk: Nauka. (Russian).

Thompson, J. F., Warsi, Z. U. A., & Mastin, C. W. (1985). *Numerical grid generation, foundations and applications*. New York: North-Holland.

Warsi, Z. U. A. (1981). *Tensors and differential geometry applied to analytic and numerical coordinate generation*. MSSU-EIRS-81-1, Aerospace Engineering, Mississippi State University.

Winslow, A. M. (1967). Equipotential zoning of two-dimensional meshes. *Journal of Computational Physics*, *1*, 149–172.

Winslow, A. M. (1981). Adaptive mesh zoning by the equipotential method. UCID-19062, Lawrence Livermore National Laboratories.

Yanenko, N. N. (1971). *The method of fractional steps: The solution of problems of mathematical physics in several variables*. New York: Springer.

Chapter 10
Numerical Implementations of Comprehensive Grid Generators

10.1 One-Dimensional Equation

Here, we consider a curve S^{x1} specified by parametrization from a normalized parametric interval $S^1 = [0, 1]$

$$\mathbf{x}(s) : [0, 1] \to R^n \ , \quad \mathbf{x} = (x^1, \dots, x^n) \ . \tag{10.1}$$

For generating a grid on the curve S^{x1}, we first define a grid on the parametric interval [0, 1] with the use of the one-dimensional inverted Beltrami equation in a divergent form. Then, the grid nodes on the parametric interval [0, 1] are mapped with the parametric transformations $\boldsymbol{x}(s)$ on S^{x1}, thus forming a grid on the curve S^{x1}.

The grid nodes in [0, 1] are computed by numerically solving the Dirichlet boundary value problem with respect to an intermediate transformation

$$s(\xi) : [0, 1] \to [0, 1]$$

for Eq. (9.133), i.e.

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\xi}\Big(\sqrt{g^s}\frac{\mathrm{d}s}{\mathrm{d}\xi}\Big) = 0 \ , \quad 0 < \xi < 1 \ , \\ &s(0) = 0 \ , \quad s(1) = 1 \ , \end{aligned} \tag{10.2}$$

where g^s is the determinant of a control covariant metric g^s_{11} over the curve S^{x1}, in particular, specified in the form (9.116) for $n = 1$, i.e.

$$\begin{aligned} g^s_{11} &= z(s)g^{xs}_{11} + F^m(s)F^m(s) \ , \quad m = 1, \dots, l \ , \\ g^{xs}_{11} &= \frac{\partial \mathbf{x}}{\partial s} \cdot \frac{\partial \mathbf{x}}{\partial s}. \end{aligned}$$

It is evident that in the one-dimensional case, $g^s = g^s_{11}$, $g^{11}_s = 1/g^s_{11}$.

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_10

The metric g^s_{11} can also be the metric of a monitor curve S^{r1} prescribed by a monitor function for controlling grid properties

$$\mathbf{f}(\mathbf{x}) : G^n \to R^l \ , \quad \mathbf{f} = (f^1, \dots, f^l) \ ,$$

where G^n is a domain in R^n containing S^{x1}. As a result, the monitor curve S^{r1} over S^{x1} is parametrized by

$$\mathbf{r}(s) : [0, 1] \to R^{n+l} \ , \quad \mathbf{r}(s) = (\mathbf{x}(s), \mathbf{f}[\mathbf{x}(s)]) \ ,$$

and consequently

$$g^s_{11} = g^{rs}_{11} = \mathbf{r}_s \cdot \mathbf{r}_s = \mathbf{x}_s \cdot \mathbf{x}_s + \mathbf{f}_s \cdot \mathbf{f}_s = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} + \frac{\mathrm{d}\mathbf{f}[\mathbf{x}(s)]}{\mathrm{d}s} \cdot \frac{\mathrm{d}\mathbf{f}[\mathbf{x}(s)]}{\mathrm{d}s} \ .$$

10.1.1 Numerical Algorithm

The grid nodes $\mathbf{x}_j$, $j = 0, 1, \dots, N$, on S^{x1} are defined by the relation

$$\mathbf{x}_j = \mathbf{x}(s(jh)) \ , \quad j = 0, 1, \dots, N \ , \quad h = 1/N \ ,$$

or by

$$\mathbf{x}_j = \mathbf{x}(s_j) \ , \quad j = 0, 1, \dots, N \ , \quad h = 1/N \ ;$$

here, s_j, $j = 0, 1, \dots, N$, is a difference function obtained by the numerical solution on a uniform grid $\xi_j = jh$, $j = 0, 1, \dots, N$, of the Dirichlet problem (10.2).

Iterative Scheme

The nonlinear problem (10.2) is solved through an iterative process which is engendered by the numerical solution of the following parabolic problem with respect to a function $s(\xi, t)$:

$$\begin{aligned} &\frac{\partial s}{\partial t} - \frac{\partial}{\partial \xi}\Big(\sqrt{g^s}\frac{\partial s}{\partial \xi}\Big) = 0 \ , \quad 0 \le \xi \le 1 \ , \quad 0 \le t \le T \ , \\ &s(0, t) = 0 \ , \quad s(1, t) = 1 \ , \quad s(\xi, 0) = s_0(\xi) \ . \end{aligned} \tag{10.3}$$

The problem (10.3) is approximated on the uniform grid $(ih, k\tau)$ with respect to s^k_i, $i = 0, 1, \dots, N$, $k = 0, 1, \dots,$ by the following natural stencil:

$$\begin{aligned} &\frac{s^{k+1}_i - s^k_i}{\tau} = \frac{1}{h^2}[v^k_{i+1/2}(s^{k+1}_{i+1} - s^{k+1}_i) - v^k_{i-1/2}(s^{k+1}_i - s^{k+1}_{i-1})] \ , \\ &s^k_0 = 0 \ , \quad s^k_N = 1 \ , \quad s^0_i = s_0(ih) \ , \quad h = 1/N \ , \end{aligned} \tag{10.4}$$

where

$$v_{i+1/2}^k = \frac{1}{2}\left(\sqrt{g^s(s_i^k)} + \sqrt{g^s(s_{i+1}^k)}\right) , \quad i = 0, 1, \dots, N-1. \tag{10.5}$$

The scheme (10.4) is implicit. Its solution is obtained from the algorithm which is elucidated by the application to the following well-known difference reference problem:

$$\begin{aligned} &A_i^{k+1} s_{i-1}^{k+1} - C_i^{k+1} s_i^{k+1} + B_i^{k+1} s_{i+1}^{k+1} = -F_i^k , \quad i = 1, 2, \dots, N-1 , \\ &s_0^{k+1} = a , \quad s_N^{k+1} = b . \end{aligned} \tag{10.6}$$

The solution to (10.6) is found through the following recursive formulas:

$$s_i^{k+1} = \alpha_{i+1}^{k+1} s_{i+1}^{k+1} + \beta_{i+1}^{k+1} , \quad i = 1, \dots, N-1 , \quad s_N^{k+1} = b , \tag{10.7}$$

where

$$\begin{aligned} \alpha_{i+1}^{k+1} &= \frac{B_i^{k+1}}{C_i^{k+1} - \alpha_i^{k+1} A_i^{k+1}} , \quad i = 1, \dots, N-1 , \quad \alpha_1^{k+1} = 0 , \\ \beta_{i+1}^{k+1} &= \frac{A_i^{k+1}\beta_i^{k+1} + F_i^k}{C_i^{k+1} - \alpha_i^{k+1} A_i^{k+1}} , \quad i = 1, \dots, N-1 , \quad \beta_1^{k+1} = a . \end{aligned} \tag{10.8}$$

Thus, assuming in (10.6) $a = 0$, $b = 1$, and

$$\begin{aligned} &A_i^{k+1} = v_{i-1/2}^k , \quad B_i^{k+1} = v_{i+1/2}^k , \quad C_i^{k+1} = v_{i-1/2}^k + v_{i+1/2}^k + \theta , \\ &F_i^k = \theta s_i^k , \quad \theta = h^2/\tau , \quad i = 1, \dots, N-1 , \end{aligned} \tag{10.9}$$

we obtain a solution of (10.4) at a step $k+1$ if it is known at the previous step k. Note that the values of the initial function s_i^0, $i = 0, 1, \dots, N$, are specified by the user. Naturally, it may be assumed that

$$s_i^0 = ih , \quad i = 0, \dots, N , \quad h = 1/N .$$

As an approximate numerical solution of (10.3), the solution s_i^k, $i = 0, 1, \dots, N$, of (10.4) at a step number k is taken if

$$\max_{0 \le i \le N} \frac{|s_i^{k+1} - s_i^k|}{\tau} \le \varepsilon , \tag{10.10}$$

for some sufficiently small $\varepsilon > 0$.

Step–by–Step Algorithm

The algorithm described above is presented here in a step-by-step manner.

Step 1.
Define an initial grid distribution of the parametric interval [0,1] by introducing a monotone difference function s_i^0, $i = 0, \dots, N$, such that $s_0^0 = 0$, $s_N^0 = 1$.
Step 2.
Compute the difference function $v_{i+1/2}^0$, $i = 0, \dots, N-1$, by formula (10.5).
Step 3.
Compute the difference functions A_i^1, B_i^1, C_i^1, F_i^0, $i = 1, \dots, N-1$, by formulas in (10.9).
Step 4.
Compute the coefficients α_i^1 and β_i^1, $i = 1, \dots, N$, by formulas in (10.8) with $a = 0$.
Step 5.
Compute the difference solution s_i^1, $i = 0, \dots, N$, of the first step through the formula (10.7), taking into account $s_0^1 = 0$, $s_N^1 = b = 1$.
Step 6.
Return to step 2 assuming $s_0^0 = s_i^1$, $i = 0, \dots, N$, where s_i^1 is the solution obtained at step 5.

Continue until the tolerance requirement (10.10) is observed.

Step 7.
Map the final nodes s_i^k, $i = 0, \dots, N$, satisfying (10.10), with the parametrization $\boldsymbol{x}(s)$ on S^{x1}.

The algorithm described is readily reformulated for the numerical solution of the inverted diffusion equation in a divergent form, namely, by substituting $w(\mathbf{s})$ for $\sqrt{g^{\mathbf{s}}}$ in (10.2), (10.3), and (10.5).

10.2 Multidimensional Finite Difference Algorithms

In this section, we apply one version of the multidimensional algorithm of fractional steps proposed by Yanenko (1971) for the numerical solution of the inverted n-dimensional ($n \geq 2$) Beltrami and diffusion equations. Other versions of this algorithm that can be readily implemented for solving the resulting multidimensional grid equations, in particular, the popular ADI (alternating direction implicit) method are reviewed by Kovenya et al. (1990), Fletcher (1997), and Langtangen (2003).

10.2.1 Parabolic Simulation

We rewrite the boundary value problems (9.132) and (9.134) in the following general form:

$$\begin{aligned} &B_n^{\xi}[s^i] = R^i[\mathbf{s}], \quad i = 1, \dots, n, \\ &s^i(\boldsymbol{\xi}) = \psi^i(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \partial \Xi^n, \end{aligned} \tag{10.11}$$

where

$$B^{\boldsymbol{\xi}}_n[s^l] = g^{\boldsymbol{\xi}} g^{ij}_{\boldsymbol{\xi}} \frac{\partial^2 s^l}{\partial \xi^i \partial \xi^j} \ , \quad i, j, l = 1, \ldots n \ ,$$
$$g^{\boldsymbol{\xi}} = \det\{g^{\boldsymbol{\xi}}_{ij}\} = (J)^2 g^s = 1/\det\{g^{ij}_{\boldsymbol{\xi}}\} \ .$$

For inverted Beltrami equations (9.132) in a general control metric $g^{\mathbf{s}}_{ij}$, we have in (10.11)

$$R^i[\boldsymbol{s}] = (J)^2 \sqrt{g^{\mathbf{s}}} \frac{\partial}{\partial \xi^j} (\sqrt{g^{\mathbf{s}}} g^{im}_{\mathbf{s}}) \frac{\partial \xi^j}{\partial s^m} \ , \quad i, j, m = 1, \ldots, n \ . \tag{10.12}$$

Notice that for the metric (9.9) of a monitor surface over a domain S^n, i.e.

$$g^{\mathbf{s}}_{ij} = g^{rs}_{ij} = \delta^i_j + \frac{\partial \boldsymbol{f}(\boldsymbol{s})}{\partial s^i} \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s})}{\partial s^j} \ , \quad i, j = 1, \ldots, n \ ,$$

in accordance with (9.56), formula (10.12) also has the following form:

$$R^i[\boldsymbol{s}] = -B^{r\boldsymbol{\xi}}_n[\boldsymbol{f}] \cdot \frac{\partial \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^j} \frac{\partial \xi^j}{\partial s^i} \ , \quad i, j = 1, \ldots, n \ , \tag{10.13}$$

where

$$B^{\boldsymbol{\xi}}_n[y] = B^{r\xi}_n[y] = g^{r\xi} g^{ij}_{\xi r} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j} \ , \quad i, j, l = 1, \ldots n \ ,$$
$$g^{r\xi} = \det\{g^{r\xi}_{ij}\} = (J)^2 g^{rs} = 1/\det\{g^{ij}_{\xi r}\} \ ,$$
$$g^{r\xi}_{ij} = g^{rs}_{kl} \frac{\partial s^k}{\partial \xi^i} \frac{\partial s^l}{\partial \xi^j} = g^{s\xi}_{ij} + \frac{\partial \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^i} \cdot \frac{\partial \boldsymbol{f}[\boldsymbol{s}(\boldsymbol{\xi})]}{\partial \xi^j} \ , \quad i, j, k, l = 1, \ldots, n \ .$$

For the general inverted diffusion equations (9.134), we have in (10.11)

$$R^i[\boldsymbol{s}] = \frac{g^{\mathbf{s}}(J)^2}{w(s)} \frac{\partial}{\partial \xi^k} (w(\boldsymbol{s}) g^{ij}_{\mathbf{s}}) \frac{\partial \xi^k}{\partial s^j} \ , \quad i, j, k = 1, \ldots, n \ , \tag{10.14}$$

in particular, for (13.49), i.e. when $w(\boldsymbol{s}) = 1$, $g^{ij}_s = Z[v](\boldsymbol{s})\delta^i_j$,

$$R^i = \frac{J^2}{Z[v](\boldsymbol{s})} \frac{\partial}{\partial s^i} Z[v](\boldsymbol{s}) \ , \qquad i = 1, \ldots, n \ , \tag{10.15}$$

$$B^{\boldsymbol{\xi}}_n[y] = B^{s\xi}_n[y] = g^{s\xi} g^{ij}_{\xi s} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j} \ , \quad g^{s\xi} = \det\{g^{s\xi}_{ij}\} = J^2 \ , \quad i, j = 1, \ldots, n \ ,$$
$$g^{s\xi}_{ij} = \frac{\partial \boldsymbol{s}}{\partial \xi^i} \cdot \frac{\partial \boldsymbol{s}}{\partial \xi^j} = \frac{\partial s^k}{\partial \xi^i} \frac{\partial s^k}{\partial \xi^j} \ , \qquad g^{ij}_{\xi s} = \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^k} \ , \qquad i, j, k = 1, \ldots, n \ . \tag{10.16}$$

Solutions to the non-linear boundary value problem (10.11) may be found in the following way. First, the problem is replaced by a nonstationary boundary value problem with respect to the components $s^i(\boldsymbol{\xi}, t)$, $i = 1, \ldots, n$, of the vector function $\boldsymbol{s}(\boldsymbol{\xi}, t) : \Xi^n \times [0, T] \to S^n$:

$$\begin{aligned}
&\frac{\partial s^i}{\partial t} = (J)^p \Big\{ B^{\xi}_n[s^i] - R^i[\boldsymbol{s}] \Big\}, \quad i, j, m = 1, \ldots, n \ , \\
&s^i(\boldsymbol{\xi}, t) = \psi^i(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \partial \Xi^n, \quad t \geq 0 \ , \\
&s^i(\boldsymbol{\xi}, 0) = s^i_0(\boldsymbol{\xi}) \ , \quad \boldsymbol{\xi} \in \Xi^n \ ,
\end{aligned} \tag{10.17}$$

where $J = \det\{\partial s^i / \partial \xi^j\}$, $p \geq 0$, $s^i_0(\boldsymbol{\xi})$ is the i-th component of the initial transformation

$$\mathbf{s}_0(\boldsymbol{\xi}) : \Xi^n \to S^n, \quad \mathbf{s}_0(\boldsymbol{\xi}) = [s^1_0(\boldsymbol{\xi}), \ldots, s^n_0(\boldsymbol{\xi})]$$

specified by the user. Then, for an approximate solution $\boldsymbol{s}(\boldsymbol{\xi})$ of (10.11), there can be taken the solution $\boldsymbol{s}(\boldsymbol{\xi},\ t)$ of (10.17) for some sufficiently large t.

If the elements g^{s}_{ij} are known at all points of S^n, then in (10.17), $p = 0$, and for (10.12), (10.13), (10.14), and (10.15), we can assume, respectively,

$$R^i[\boldsymbol{s}] = (J)^2 \sqrt{g^{\mathrm{s}}} \frac{\partial}{\partial s^m} (\sqrt{g^{\mathrm{s}}} g^{im}_{\mathrm{s}}) \ , \quad i, m = 1, \ldots, n \ , \tag{10.18}$$

$$R^i[\boldsymbol{s}] = -B^{r\xi}_n[\boldsymbol{f}(\boldsymbol{s})] \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s})}{\partial s^i} \ , \quad i = 1, \ldots, n \ , \tag{10.19}$$

$$R^i[\boldsymbol{s}] = \frac{g^{\mathrm{s}}(J)^2}{w(\boldsymbol{s})} \frac{\partial}{\partial s^j} (w(\boldsymbol{s}) g^{ij}_{\mathrm{s}}) \ , \quad i, j = 1, \ldots, n \ , \tag{10.20}$$

$$R^i = \frac{J^2}{Z[v](\boldsymbol{s})} \frac{\partial}{\partial s^i} Z[v](\boldsymbol{s}) \ , \qquad i = 1, \ldots, n \ . \tag{10.21}$$

When B^{ξ}_n is an elliptic operator, the solution to the problem (10.17) relaxes to the solution of (10.11) as $t \to \infty$.

The factor $(J)^p$, $p \geq 1$ in (10.17) is introduced in the case when the control metric g^{s}_{ij} is not known in advance, but is found numerically, for instance, if it is dependent on the solution of the physical problem for which the numerical grid is generated. This factor allows one to rule out the Jacobian J being a denominator after replacing in $R^i[\boldsymbol{s}]$ the derivatives $\partial \xi^i / \partial s^j$ with the derivatives $\partial s^k / \partial \xi^m$. In particular, in the case of the metric of a monitor hypersurface S^{rn} over S^n (see (9.13) and (9.14)), i.e. when $g^{\mathrm{s}}_{ij} = g^{r\xi}_{ij}$, it is sufficient to assume $p = 1$. Namely, for $n = 2$, we have, from (10.13) and (2.4),

$$\begin{aligned} JR^i[\boldsymbol{s}] &= -JB_2^{r\xi}[\boldsymbol{f}(\boldsymbol{s})] \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^m} \frac{\partial \xi^m}{\partial s^i} \\ &= -(-1)^{i+m} B_2^{r\xi}[\boldsymbol{f}(\boldsymbol{s})] \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^m} \frac{\partial s^{3-i}}{\partial \xi^{3-m}} \ , \quad i, m = 1, 2 \ , \end{aligned} \tag{10.22}$$

while for $n = 3$, we obtain, from (10.13) and (2.5),

$$\begin{aligned} JR^i[\boldsymbol{s}] &= -JB_3^{r\xi}[\boldsymbol{f}(\boldsymbol{s})] \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^m} \frac{\partial \xi^m}{\partial s^i} \\ &= -B_3^{r\xi}[\boldsymbol{f}(\boldsymbol{s})] \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^m} \Big(\frac{\partial s^{i+1}}{\partial \xi^{m+1}} \frac{\partial s^{i+2}}{\partial \xi^{m+2}} - \frac{\partial s^{i+1}}{\partial \xi^{m+2}} \frac{\partial s^{i+2}}{\partial \xi^{m+1}} \Big) \ , \\ &\quad i, m = 1, 2, 3 \ . \end{aligned} \tag{10.23}$$

With such incorporation of $(J)^p$, one can produce a final nondegenerate grid even if the initial and intermediate grids may be singular. Note that the numerical implementations of the inverted energy and diffusion functionals cannot eliminate the Jacobian being the denominator.

The boundary value problem (10.17) is usually solved through alternative direction implicit methods, in particular, through the method of fractional steps.

10.2.2 Two-Dimensional Equations

In this section, a finite-difference numerical algorithm for generating grids in two-dimensional domains and surfaces is described.

Boundary Value Problem

Let us first discuss the grid algorithm for a two-dimensional domain S^2. We shall use, for the logical domain Ξ^2, the unit square: $\Xi^2 = \{0 \le \xi^1, \xi^2 \le 1\}$. Let the transformation $\mathbf{s}(\boldsymbol{\xi})$ for generating a grid in S^2 be specified on the boundary of Ξ^2, i.e. there is a map

$$\boldsymbol{\varphi}(\boldsymbol{\xi}) : \partial \Xi^2 \to \partial S^2 \ , \quad \boldsymbol{\varphi} = (\varphi^1, \varphi^2) \tag{10.24}$$

which is continuous on $\partial \Xi^2$. Note that the one-dimensional transformation on any segment of $\partial \Xi^2$ can be computed by the algorithm described in Sect. 11.1. We consider here the generation of a grid in S^2 by the numerical solution of the Dirichlet problem (10.11) for the most general system of inverted Beltrami equations in a control metric g^s_{ij} for $n = 2$ written in a vector form

$$\begin{aligned} &B_2^{\xi}[\boldsymbol{s}] = \boldsymbol{R}[\boldsymbol{s}] \ , \\ &\boldsymbol{s}(\boldsymbol{\xi}) \Big|_{\partial \Xi^2} = \boldsymbol{\varphi}(\boldsymbol{\xi}), \quad i = 1, 2 \ , \end{aligned} \tag{10.25}$$

where

$$
\begin{aligned}
&\boldsymbol{s}(\boldsymbol{\xi}) = (s^1(\boldsymbol{\xi})\ ,\ s^2(\boldsymbol{\xi}))\ , \quad \boldsymbol{\varphi}(\boldsymbol{\xi}) = (\varphi^1(\boldsymbol{\xi}),\ \varphi^2(\boldsymbol{\xi}))\ , \quad \boldsymbol{R}[\boldsymbol{s}] = (R^1[\boldsymbol{s}]\ ,\ R^2[\boldsymbol{s}])\ ,\\
&B_2^{\xi}[\boldsymbol{s}] = g_{22}^{\xi}\frac{\partial^2 \boldsymbol{s}}{\partial\xi^1\partial\xi^1} - 2g_{12}^{\xi}\frac{\partial^2 \boldsymbol{s}}{\partial\xi^1\partial\xi^2} + g_{11}^{\xi}\frac{\partial^2 \boldsymbol{s}}{\partial\xi^2\partial\xi^2}\ ,\\
&R^1[\boldsymbol{s}] = J\sqrt{g^{\mathrm{s}}}\Big[\frac{\partial}{\partial\xi^1}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{11})\frac{\partial s^2}{\partial\xi^2} - \frac{\partial}{\partial\xi^1}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{12})\frac{\partial s^1}{\partial\xi^2}\\
&\qquad + \frac{\partial}{\partial\xi^2}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{12})\frac{\partial s^1}{\partial\xi^1} - \frac{\partial}{\partial\xi^2}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{11})\frac{\partial s^2}{\partial\xi^1}\Big]\ ,\\
&R^2[\boldsymbol{s}] = J\sqrt{g^{\mathrm{s}}}\Big[\frac{\partial}{\partial\xi^1}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{11})\frac{\partial s^2}{\partial\xi^2} - \frac{\partial}{\partial\xi^1}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{12})\frac{\partial s^1}{\partial\xi^2}\\
&\qquad + \frac{\partial}{\partial\xi^2}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{12})\frac{\partial s^1}{\partial\xi^1} - \frac{\partial}{\partial\xi^2}(\sqrt{g^{\mathrm{s}}}g_{\mathbf{s}}^{11})\frac{\partial s^2}{\partial\xi^1}\Big]\ .
\end{aligned}
\tag{10.26}
$$

Parabolic Equations

The nonlinear boundary value problem (10.25) is solved by an iterative process. For this purpose, in accordance with (10.17), the problem (10.25) is replaced by the following boundary value parabolic problem with respect to the function $\boldsymbol{s}(\xi^1, \xi^2, t) = (s^1(\xi^1, \xi^2, t),\ s^2(\xi^1, \xi^2, t))$:

$$
\begin{aligned}
&\frac{\partial \boldsymbol{s}}{\partial t} = (J)^p\Big\{B_2^{\xi}[\boldsymbol{s}] - \boldsymbol{R}(\boldsymbol{s})\Big\}\ ,\\
&\boldsymbol{s}(\boldsymbol{\xi}, t) = \boldsymbol{\varphi}(\boldsymbol{\xi})\ , \quad \boldsymbol{\xi} \in \partial\Xi^2\ , \quad t \geq 0\ ,\\
&\boldsymbol{s}(\boldsymbol{\xi}, 0) = \boldsymbol{s}_0(\boldsymbol{\xi})\ , \quad \boldsymbol{\xi} \in \Xi^2\ ,
\end{aligned}
\tag{10.27}
$$

where $\boldsymbol{s}_0(\boldsymbol{\xi})$ is an initial transformation

$$
\mathbf{s}_0(\boldsymbol{\xi}) : \Xi^2 \to S^2, \quad \mathbf{s}_0(\boldsymbol{\xi}) = [s_0^1(\boldsymbol{\xi}), s_0^2(\boldsymbol{\xi})],
$$

specified by the user.

The solution $\mathbf{s}(\boldsymbol{\xi}, t)$ satisfying (10.27) aspires to the solution to (10.25) when $t \to \infty$. Therefore, an approximate solution of (10.25) is obtained from the solution to (10.27) computed for some sufficiently large value $t = T_0$.

Initial Transformation

The initial transformation for (10.27)

$$
\mathbf{s}(\boldsymbol{\xi}, 0) = \mathbf{s}_0(\boldsymbol{\xi}) : \Xi^2 \to S^2.
$$

can be found by propagating the values of $\boldsymbol{\varphi}(\boldsymbol{\xi}) = [\varphi^1(\boldsymbol{\xi}), \varphi^2(\boldsymbol{\xi})]$ from the boundary points into the interior of the domain Ξ^2, for example, if Ξ^2 is the unit square through the formula of the Lagrange two-dimensional transfinite interpolation described in Chap. 5. This formula has the following recursive form for the components $s^i(\boldsymbol{\xi}, 0)$ of the mapping $\mathbf{s}_0(\boldsymbol{\xi})$:

$$\begin{aligned} F^i(\xi^1,\xi^2) &= \alpha^i_{01}(\xi^1)\varphi^i(0,\xi^2)+\alpha^i_{11}(\xi^1)\varphi^i(1,\xi^2)\ , \\ s^i(\xi^1,\xi^2,0) &= F^i(\xi^1,\xi^2)+\alpha^i_{02}(\xi^2)[\varphi^i(\xi^1,0)-F^i(\xi^1,0)] \\ &\quad +\alpha^i_{12}(\xi^2)[\varphi^i(\xi^1,1)-F^i(\xi^1,1)]\ , \quad i=1,2, \quad i \text{ fixed}\ , \end{aligned} \tag{10.28}$$

where the functions $\alpha^i_{kj}(s), 0 \le s \le 1$, (referred to as blending functions) are subject to the following restrictions:

$$\alpha^i_{0j}(0)=\alpha^i_{1j}(1)=1\ , \quad \alpha^i_{0j}(1)=\alpha^i_{1j}(0)=0\ . \tag{10.29}$$

In particular, for the simplest expressions of the blending functions

$$\alpha^i_{0j}(s)=1-s\ , \quad \alpha^i_{1j}(s)=s\ ,$$

satisfying (10.29), we find, from (10.28),

$$\begin{aligned} F^i(\xi^1,\xi^2) &= (1-\xi^1)\varphi^i(0,\xi^2)+\xi^1\varphi^i(1,\xi^2)\ , \\ s^i(\xi^1,\xi^2,0) &= F^i(\xi^1,\xi^2)+(1-\xi^2)[\varphi^i(\xi^1,0)-F^i(\xi^1,0)] \\ &\quad +\xi^2[\varphi^i(\xi^1,1)-F^i(\xi^1,1)]\ , \quad i=1,2\ . \end{aligned} \tag{10.30}$$

Iterative Algorithm for Generating Quadrilateral Grids

The problem (10.27) is approximated on the rectangular grid $(ih_1, jh_2, k\tau)$, $h_1 = 1/N_1$, $h_2 = 1/N_2$, in the logical domain $\Xi^2 \times [0,T]$, where Ξ^2 is a rectangle (Fig. 10.1 (*left-hand*)), by the scheme

$$\begin{aligned} \frac{\boldsymbol{s}^{k+1/2}-\boldsymbol{s}^k}{\tau/2} &= J^p(\boldsymbol{s}^k)\Big\{g^{\xi}_{22}[\boldsymbol{s}^k]L^h_{11}[\boldsymbol{s}^{k+1/2}]+g^{\xi}_{11}[\boldsymbol{s}^k]L^h_{22}[\boldsymbol{s}^k] \\ &\quad -2g^{\xi}_{12}[\boldsymbol{s}^k]L^h_{12}[\boldsymbol{s}^k]\Big\}-J^p(\boldsymbol{s}^k)\boldsymbol{R}[\boldsymbol{s}^k], \\ \frac{\boldsymbol{s}^{k+1}-\boldsymbol{s}^{k+1/2}}{\tau/2} &= J^p(\boldsymbol{s}^k)\Big\{g^{\xi}_{11}[\boldsymbol{s}^k]L^h_{22}[\boldsymbol{s}^{k+1}]-g^{\xi}_{11}[\boldsymbol{s}^k]L^h_{22}[\boldsymbol{s}^k]\Big\}\ , \end{aligned} \tag{10.31}$$

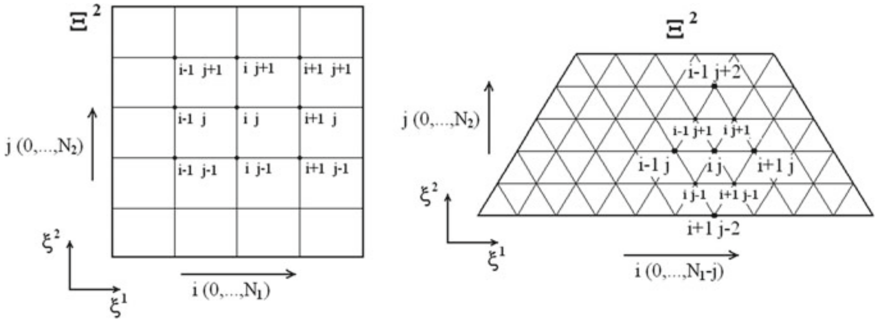


Fig. 10.1 Two-dimensional quadrilateral and triangular stencils for finite differences

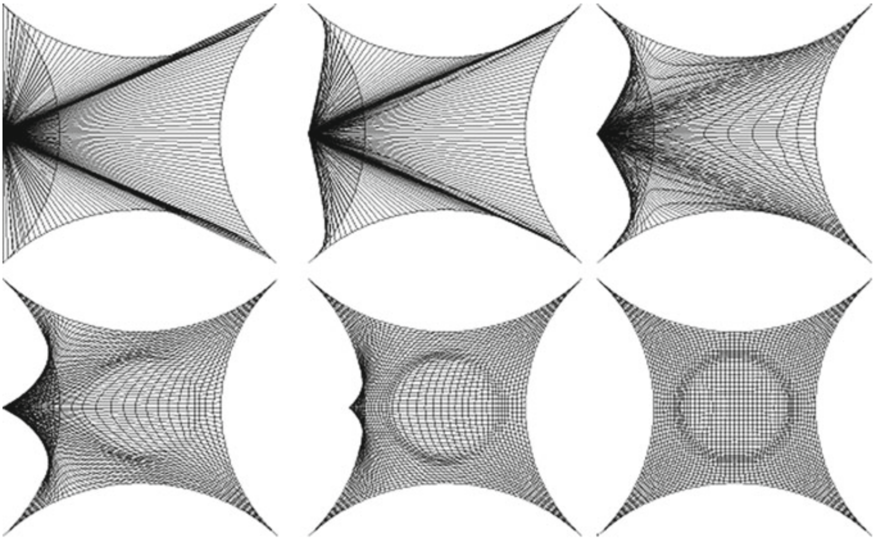


Fig. 10.2 Stages of the iterative generation of a quadrilateral grid with the use of a singular initial grid

where $\boldsymbol{s}^{k+\alpha} = \boldsymbol{s}(\boldsymbol{\xi}, (k+\alpha\tau)),\ k = 0, 1, 2, \dots,\ \alpha = 0,\ 1/2,\ 1,\ L^h_{ij}$ is a finite-difference operator approximating the operator $\partial^2/(\partial\xi^i\partial\xi^j)$, by the central differences. The derivatives in J, g^{ξ}_{ij}, and $\boldsymbol{R}$ are also approximated by the central differences. The initial transformation $\boldsymbol{s}(\boldsymbol{\xi}, 0) = \boldsymbol{s}_0(\boldsymbol{\xi})$ is found through the formulas of transfinite interpolations.

The solution to (10.31) at each step k and $k+1/2$ is obtained in the same way as it was described in Sect. 11.1.

An approximate solution to (10.27) is the solution $\mathbf{s}^k_{ij}$ at a step k such that

$$\max_{0\le i\le N_1, 0\le j\le N_2} \frac{1}{\tau}|\boldsymbol{s}^{k+1}_{ij} - \boldsymbol{s}^k_{ij}| \le \varepsilon\ , \tag{10.32}$$

for some sufficiently small $\varepsilon > 0$.

Figure 10.2 demonstrates some steps of the grid generation in a two-dimensional domain by the solution of the inverted Beltrami equations with the iterative algorithm described. The initial grid is singular (all its interior nodes merge into one node lying outside of the domain).

Generation of Triangular Grids

The numerical algorithm described above for generating quadrilateral grids is naturally applied to the generation of triangular grids when the logical domain is a symmetric trapezoid (Fig. 10.1 (*right-hand*)).

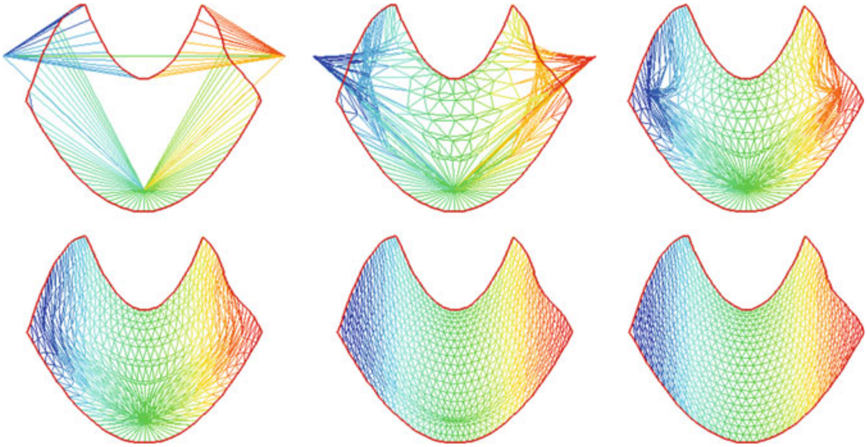


Fig. 10.3 Stages for generating a triangular grid by using a singular initial grid

An example of a triangular grid generated by such an algorithm is exhibited by Fig. 10.3. As it is in Fig. 10.2, the initial grid is singular. All its interior points are placed into three points, two of which lie outside of the domain.

Algorithm for Generating Grids on Two-Dimensional Surfaces

In the same way as for domains, grids are generated in a two-dimensional surface S^{x2} represented as

$$\mathbf{x}(\mathbf{s}) : S^2 \to R^3 , \quad \mathbf{x} = (x^1, x^2, x^3) , \quad \mathbf{s} = (s^1, s^2) , \qquad (10.33)$$

by solving the boundary value problem for the inverted two-dimensional diffusion equations as well as for the corresponding inverted Beltrami equations with respect to a monitor metric $g^{\mathbf{s}}_{ij}$ over S^{x2}.

Similarly to the case of a two-dimensional domain, we can choose a rectangle or trapezoid for the logical domain Ξ^2. We can also assume that the boundary transformation

$$\boldsymbol{\varphi}(\boldsymbol{\xi}) : \partial\Xi^2 \to \partial S^2 , \quad \boldsymbol{\varphi} = (\varphi^1, \varphi^2) ,$$

which is continuous on $\partial\Xi^2$, has been specified on the boundary grid points of $\partial\Xi^2$, for example, by computing it through the algorithm described in Sect. 10.1.

The grid on S^{x2} is obtained by mapping, with $\mathbf{x}(\mathbf{s})$, the grid nodes computed in S^2 through the numerical solution of the Dirichlet problem with respect to $\mathbf{s}(\boldsymbol{\xi})$ for the inverted grid equations.

Figure 10.4 illustrates a surface triangular adaptive grid generated by the algorithm.

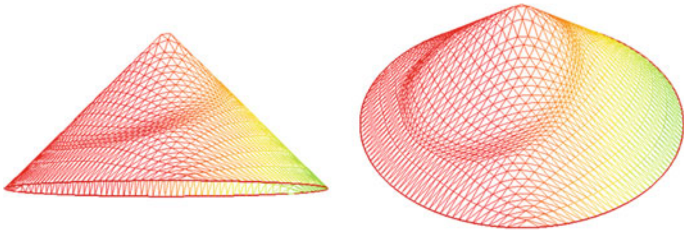


Fig. 10.4 A triangular adaptive grid on a conical surface

10.2.3 Three–Dimensional Problem

For generating grids in a three-dimensional domain $S^3 \subset R^3$ with the use of the inverted Beltrami or diffusion equations in a control metric $g^{\mathbf{s}}_{ij}$, $i, j = 1, 2, 3$, we consider boundary value problem (10.11) for $n = 3$ written in a vector form

$$\begin{aligned} &B_3^{\xi}[\boldsymbol{s}] = \boldsymbol{R}[\boldsymbol{s}]\ , \\ &\boldsymbol{s}(\boldsymbol{\xi})\ \Big|_{\partial \Xi^3} = \boldsymbol{\varphi}(\boldsymbol{\xi})\ , \end{aligned} \tag{10.34}$$

where

$$\begin{aligned} &\boldsymbol{s}(\boldsymbol{\xi}) = (s^1(\boldsymbol{\xi})\ ,\ s^2(\boldsymbol{\xi}\ ,\ s^3(\boldsymbol{\xi}))\ , \quad \boldsymbol{\varphi}(\boldsymbol{\xi}) = [\varphi^1(\boldsymbol{\xi}), \varphi^2(\boldsymbol{\xi}), \varphi^3(\boldsymbol{\xi})]\ , \\ &\boldsymbol{R}(\boldsymbol{s}) = (R^1(\boldsymbol{s})\ ,\ R^2(\boldsymbol{s})\ ,\ R^3(\boldsymbol{s}))\ , \end{aligned}$$

$$\begin{aligned} B_3^{\xi}[v] &= g^{\xi} g_{\xi}^{ij} \frac{\partial^2 \boldsymbol{s}}{\partial \xi^i \partial \xi^j} \\ &= [g^{\xi}_{22} g^{\xi}_{33} - (g^{\xi}_{23})^2] \frac{\partial^2 \boldsymbol{s}}{\partial \xi^1 \partial \xi^1} + 2[g^{\xi}_{23} g^{\xi}_{13} - g^{\xi}_{12} g^{\xi}_{33}] \frac{\partial^2 \boldsymbol{s}}{\partial \xi^1 \partial \xi^2} \\ &\quad + 2[g^{\xi}_{12} g^{\xi}_{23} - g^{\xi}_{22} g^{\xi}_{13}] \frac{\partial^2 \boldsymbol{s}}{\partial \xi^1 \partial \xi^3} + [g^{\xi}_{11} g^{\xi}_{33} - (g^{\xi}_{13})^2] \frac{\partial^2 \boldsymbol{s}}{\partial \xi^2 \partial \xi^2} \\ &\quad + 2[g^{\xi}_{12} g^{\xi}_{13} - g^{\xi}_{11} g^{\xi}_{23}] \frac{\partial^2 \boldsymbol{s}}{\partial \xi^2 \partial \xi^3} + [g^{\xi}_{11} g^{\xi}_{22} - (g^{\xi}_{12})^2] \frac{\partial^2 \boldsymbol{s}}{\partial \xi^3 \partial \xi^3}\ , \\ R^i[\boldsymbol{s}] &= J\sqrt{g^{\mathrm{s}}} \frac{\partial}{\partial \xi^j} (\sqrt{g^{\mathrm{s}}} g^{im}_{\mathbf{s}}) \Big(\frac{\partial s^{m+1}}{\partial \xi^{j+1}} \frac{\partial s^{m+2}}{\partial \xi^{j+2}} - \frac{\partial s^{m+1}}{\partial \xi^{j+2}} \frac{\partial s^{m+2}}{\partial \xi^{j+1}} \Big)\ , \\ &\quad i, j, m = 1, 2, 3\ . \end{aligned} \tag{10.35}$$

Analogously to the solution of two-dimensional problem (10.27), we find a solution to (10.34) as a limit with $t \to \infty$ of the solution of the corresponding parabolic problem

$$\begin{aligned} &\frac{\partial \boldsymbol{s}}{\partial t} = J^p \Big\{ B_3^{\xi}[\boldsymbol{s}] - \boldsymbol{R}(\boldsymbol{s}) \Big\}\ , \\ &\boldsymbol{s}(\boldsymbol{\xi}, t) = \boldsymbol{\varphi}(\boldsymbol{\xi})\ , \quad \boldsymbol{\xi} \in \partial \Xi^3\ , \quad t \geq 0\ , \\ &\boldsymbol{s}(\boldsymbol{\xi}, 0) = \mathbf{s}_0(\boldsymbol{\xi})\ , \quad \boldsymbol{\xi} \in \Xi^3\ . \end{aligned} \tag{10.36}$$

Initial Transformation

The initial transformation

$$\boldsymbol{s}(\boldsymbol{\xi}, 0) = \boldsymbol{s}_0(\boldsymbol{\xi}) : \Xi^3 \to S^3 \ .$$

can be found by propagating the values of $\boldsymbol{\varphi}(\boldsymbol{\xi})$ into the interior of the unit cube Ξ^3, for example, through the formula of Lagrange transfinite interpolation. In particular, for the simplest expressions of the blending functions

$$\alpha^i_{0j}(s) = 1 - s \ , \quad \alpha^i_{1j}(s) = s \ ,$$

we find from (5.26)

$$\begin{aligned}
&F^i_1(\xi^1, \xi^2, \xi^3) = (1-\xi^1)\varphi^i(0, \xi^2, \xi^3) + \xi^1\varphi^i(1, \xi^2, \xi^3) \ , \\
&F^i_2(\xi^1, \xi^2, \xi^3) = F^i_1(\xi^1, \xi^2, \xi^3) + (1-\xi^2)[\varphi^i(\xi^1, 0, \xi^3) \\
&-F^i_1(\xi^1, 0, \xi^3)] + \xi^2[\varphi^i(\xi^1, 1, \xi^3) - F^i_1(\xi^1, 1, \xi^3)] \ , \\
&x^i(\xi^1, \xi^2, \xi^3) = F^i_2(\xi^1, \xi^2, \xi^3) + (1-\xi^3)[\varphi^i(\xi^1, \xi^2, 0) \\
&-F^i_2(\xi^1, \xi^2, 0)] + \xi^3[\varphi^i(\xi^1, \xi^2, 1) - F^i_2(\xi^1, \xi^2, 1)] \ , \quad i = 1, 2, 3 \ ,
\end{aligned} \tag{10.37}$$

Three–Dimensional Algorithm

A numerical algorithm for solving problem (10.36) is formulated analogously to the two-dimensional algorithm reviewed by formula (10.31), namely, by splitting the process of the numerical solution on the computational domain Ξ^3, exhibited in Figs. 10.5 and 10.6, into three one-dimensional algorithms:

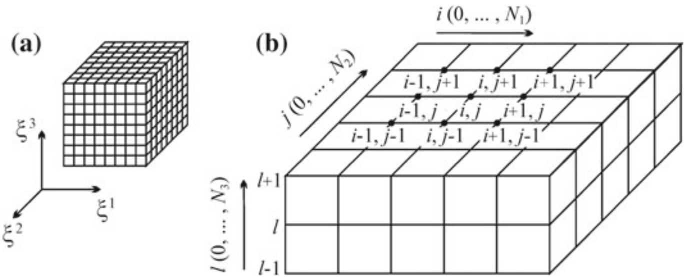


Fig. 10.5 Computational domain Ξ^3 (**a**) and computational stencil (**b**) for generating hexahedral grids

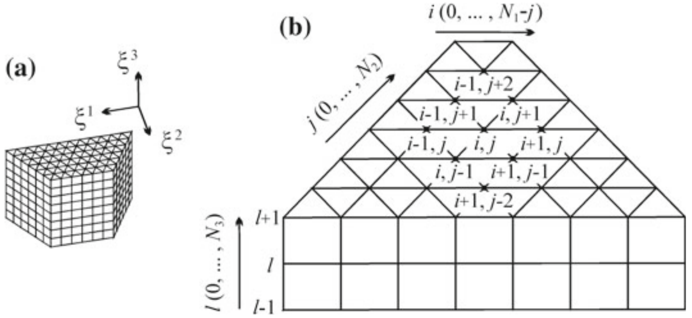


Fig. 10.6 Computational domain $\varXi^3$ (**a**) and computational stencil (**b**) for generating prismatic grids

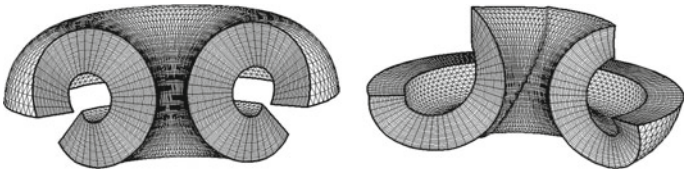


Fig. 10.7 Three-dimensional domain with a prismatic adaptive grid

$$
\begin{aligned}
\frac{\boldsymbol{s}^{k+1/3}-\boldsymbol{s}^k}{\tau/3} &= J^p(\boldsymbol{s}^k)\Big\{a^{11}[\boldsymbol{s}^k]L^h_{11}[\boldsymbol{s}^{k+1/3}] + a^{22}[\boldsymbol{s}^k]L^h_{22}[\boldsymbol{s}^k] \\
&\quad + a^{33}[\boldsymbol{s}^k]L^h_{33}[\boldsymbol{s}^k] + 2a^{12}[\boldsymbol{s}^k]L^h_{12}[\boldsymbol{s}^k] \\
&\quad + 2a^{13}[\boldsymbol{s}^k]L^h_{13}[\boldsymbol{s}^k] + 2a^{23}[\boldsymbol{s}^k]L^h_{23}[\boldsymbol{s}^k]\Big\} \\
&\quad - J^p(\boldsymbol{s}^k)\boldsymbol{R}[\boldsymbol{s}^k]\,, \\
\frac{\boldsymbol{s}^{k+2/3}-\boldsymbol{s}^{k+1/3}}{\tau/3} &= J^p(\boldsymbol{s}^k)\Big\{a^{22}[\boldsymbol{s}^k]L^h_{22}[\boldsymbol{s}^{k+2/3}] - a^{22}[\boldsymbol{s}^k]L^h_{22}[\boldsymbol{s}^k]\Big\}\,, \\
\frac{\boldsymbol{s}^{k+1}-\boldsymbol{s}^{k+2/3}}{\tau/3} &= J^p(\boldsymbol{s}^k)\Big\{a^{33}[\boldsymbol{s}^k]L^h_{33}[\boldsymbol{s}^{k+1}] - a^{33}[\boldsymbol{s}^k]L^h_{33}[\boldsymbol{s}^k]\Big\},
\end{aligned}
\tag{10.38}
$$

where $a^{ij} = g^{\boldsymbol{\xi}} g^{ij}_{\boldsymbol{\xi}} = g^{\boldsymbol{\xi}}_{i+1j+1} g^{\boldsymbol{\xi}}_{i+2j+2} - g^{\boldsymbol{\xi}}_{i+1j+2} g^{\boldsymbol{\xi}}_{i+2j+1}$, $i, j = 1, 2, 3$, i, j, k – fixed, $\boldsymbol{s}^{k+\alpha} = \boldsymbol{s}(\boldsymbol{\xi}, (k+\alpha\tau))$, $k = 0, 1, 2, \ldots,$ $\alpha = 0,\ 1/2,\ 2/3,\ 1$. L^h_{ij} is a finite-difference operator approximating the operator $\partial^2/(\partial\xi^i\partial\xi^j)$, by the central differences. The derivatives in J, a^{ij}, and $\boldsymbol{R}$ are also approximated by the central differences. The initial transformation $\boldsymbol{s}(\boldsymbol{\xi}, 0) = \boldsymbol{s}_0(\boldsymbol{\xi})$ is found through the formulas of transfinite interpolations.

An example of a three-dimensional prismatic spatial grid generated with the use of this scheme is demonstrated in Fig. 10.7.

10.3 Spectral Element Algorithm

The inverted diffusion equations in a divergent form (9.135) may be solved by a parallel code, using spectral elements for spatial discretization, Newton-Krylov methods for solution, and an adaptive time step.

Spatial discretization by high-order spectral elements is a method of exploiting the best features of both grid-based methods and global spectral representation. Grid-based methods, such as the finite difference approach described above, lead to nearest neighbor coupling and its resultant sparse matrix structure, and lends itself to parallelization by domain decomposition and that kind of adaptive gridding. On the other hand, convergence of the spatial truncation error is relatively slow, typically a low power of the grid spacing h. Global spectral methods overcome the latter problem, offering exponential convergence with increasing numbers of basis functions, but lead to large, dense matrices and offer no obvious way to use adaptive gridding and parallelization by domain decomposition. With spectral elements, there is a relatively coarse grid, and within each grid cell, there is a local expansion in basis functions based on orthogonal polynomials. The grid provides nearest-neighbor coupling while the spectral expansion provides exponential convergence.

All equations for spectral elements are to be expressed in flux-source form,

$$\frac{\partial u^k}{\partial t} + \nabla \cdot \mathbf{F}^k = S^k. \tag{10.39}$$

This very form has the following parabolic system:

$$\frac{\partial s^k}{\partial t} - \frac{\partial}{\partial \xi^j}(Jw(\mathbf{s})g_{\boldsymbol{\xi}}^{jk}) = 0, \quad j, k = 1, \ldots, n\ , \tag{10.40}$$

with identification $u = s,\ x^i = s^i$, obtained from grid equations (9.135) in the same manner as the system in (10.17) from the Eq. (10.11). The dependent variables u^k in (10.39) within each grid cell are expanded in a spectral basis $\alpha_j(x)$,

$$u^k(t, \mathbf{x}) \approx \sum_{j=0}^{n} u_j^k(t)\alpha_j(\mathbf{x}). \tag{10.41}$$

Spatially discretized equations are obtained through a Galerkin method, taking the scalar product of (10.39) with each basis function and integrating by parts to obtain

$$\ddot{M}\dot{\mathbf{u}} = \mathbf{r} \equiv \int\limits_{X^n} (S^k\alpha_i + \mathbf{F}^k \cdot \nabla\alpha_i)d\mathbf{x} - \int\limits_{\partial X^n} \mathbf{F}_i^k \cdot \hat{\mathbf{n}}d\mathbf{x}. \tag{10.42}$$

with $\ddot{M}$ the mass matrix, $M_{i,j} \equiv (\alpha_i, \alpha_j)$, and the $\mathbf{u}$ the vector of mode amplitudes $u_j^k(t)$. Integrals are evaluated by Gaussian quadrature to an order appropriate to the degree of the Jacobi polynomials. Fluxes and sources may depend in an arbitrary nonlinear manner on t, $\mathbf{x}$, u^k, and ∇u^k. The code is structured in such a way that the details of discretization and the specification of physics equations are separated into different subroutines, making it as simple as possible to encode complex physics. The discretized flux-source form preserves conservation properties to high order. Elliptic equations are treated by zeroing the mass matrix.

Time discretization of (10.42) is fully implicit in order to treat multiple time scales efficiently and accurately,

$$\ddot{M}(\frac{\mathbf{u}^+ - \mathbf{u}^-}{h}) = \theta \mathbf{r}^+(\mathbf{u}^+) + (1-\theta)\mathbf{r}^-(\mathbf{u}^-) \tag{10.43}$$

with the time-centering parameter θ normally chosen as 1/2 (Crank-Nicholson) for accuracy. Solution of (10.43) requires finding the roots of the nonlinear residual,

$$\mathbf{R}(u^+) \equiv \ddot{M}(\mathbf{u}^+ - \mathbf{u}^-) - h[\theta \mathbf{r}^+ + (1-\theta)\mathbf{r}^-] = \mathbf{0}, \tag{10.44}$$

solved by Newton's iteration,

$$\mathbf{R} + \ddot{J}\delta\mathbf{u}^+ = \mathbf{0}, \quad \delta\mathbf{u}^+ = -\ddot{J}^{-1}\mathbf{R}(\mathbf{u}^+), \quad \mathbf{u}^+ \rightarrow \mathbf{u}^+ + \delta\mathbf{u}^+ \tag{10.45}$$

with the Jacobian defined as $\ddot{J} \equiv \ddot{M} - h\theta\{\partial r_i^+/\partial u_j^+\}$.

Efficient solution of the large sparse linear system in (10.45) is greatly enhanced by the method of static condensation. Because of the C^0 nature of the spectral element representation, discussed above, higher-order elements in one grid cell couple to those in neighboring grid cells only through the shared linear finite elements which straddle cell boundaries. To solve a linear system $\ddot{A}\mathbf{x} = \mathbf{b}$, we partition the dependent variables into (1) element boundary terms and (2) element interior terms, for example, in two dimensions, the system is expressed in the form

$$\ddot{A}_{11}\mathbf{x}_1 + \ddot{A}_{12}\mathbf{x}_2 = \mathbf{b}_1, \tag{10.46}$$

$$\ddot{A}_{21}\mathbf{x}_1 + \ddot{A}_{22}\mathbf{x}_2 = \mathbf{b}_2. \tag{10.47}$$

Solving (10.47) for $\mathbf{x}_2$,

$$\ddot{A}_{22}\mathbf{x}_2 = \mathbf{b}_2 - \ddot{A}_{21}\mathbf{x}_1, \tag{10.48}$$

and substituting it into (10.46), we obtain an equation for the Shur complement,

$$(\ddot{A}_{11} - \ddot{A}_{12}\ddot{A}_{22}^{-1}\ddot{A}_{21})\mathbf{x}_1 = \mathbf{b}_1 - \ddot{A}_{12}\ddot{A}_{22}^{-1}\mathbf{b}_2. \tag{10.49}$$

Equation (10.48), involving the relatively small, dense, local matrix $\ddot{A}_{22}$, is solved locally using LAPACK routines. It parallelizes perfectly over grid cells, requiring no communication once $\mathbf{x}_2$ is determined. Equation (10.49), greatly condensed in size from the original system, is solved globally and iteratively by Krylov subspace routine GMRES, using the PETSc library, preconditioned by additive Schwarz ILU factorization with substantial fill-in and overlap. The most efficient parallel operation is obtained with one grid cell per processor. This is feasible because the use of high-order spectral elements makes it possible to achieve good spatial resolution with relatively few grid cells.

For generating a numerical grid with node clustering in the zones of large values of a function $v(\mathbf{s})$, the measure of departure from the necessary grid can be expressed in the form

$$\sigma(\mathbf{s}) = Z[v](\mathbf{s}) g^{kl}_{sx} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^l}, \quad i, k, l = 1, \dots, n \tag{10.50}$$

where $Z[v]$ is a positive operator such that the function $Z[v](\mathbf{s})$ is large (small) where $v(\mathbf{s})$ is small (large). This measure for generating adaptive grids in domains was introduced in Danaev et al. (1980) and Winslow (1981). Consequently, the contravariant elements of the control metric in S^{xn} are as follows:

$$g^{ij}(\mathbf{s}) = Z[v](\mathbf{s}) g^{ij}_{sx}, \quad i, j = 1, \dots, n\ . \tag{10.51}$$

This contravariant metric tensor can also be used for providing node clustering in the zones of the large variation of a function $\mathbf{f}(\mathbf{s})$ by introducing for this purpose a function $v(grad\ \mathbf{f})$ such that v is large where $|grad\ \mathbf{f}|$ is large, and vice versa.

We choose in the control metric (10.51) the weight function $v(\mathbf{s})$ and assume $Z[v](\mathbf{s}) = 1/v(\mathbf{s})$, to reflect the spatial truncation error in the spectral element representation. In each grid cell $\bar{\Omega}$, we define the spatial truncation error as the ratio of the L^2 norm of the highest-order polynomial $\delta u(\mathbf{s})$ and that of the full solution $u(\mathbf{s})$, but because the spatial discretization error for spectral element methods is exponentially convergent with an increasing number of terms, we use the log of this norm,

$$\delta\bar{\Omega} \equiv \frac{1}{2} \left(\frac{\int\limits_{\bar{\Omega}} \delta u^2(\mathbf{s}) d\mathbf{s}}{\int\limits_{\bar{\Omega}} u^2(\mathbf{s}) d\mathbf{s}} \right). \tag{10.52}$$

Since this function is piecewise constant over each grid cell but we need a smooth function, we use a least-squares bicubic spline fit. Finally, in order to control the range of variation of v, we define

$$v(\mathbf{s}) = 1 + \alpha \left(\frac{\delta - \delta_{min}}{\delta_{max} - \delta_{min}} \right) \tag{10.53}$$

with α an adjustable constant. When $\delta = \delta_{min}$, $v(\mathbf{s}) = 1$, and when $\delta = \delta_{max}$, $v(\mathbf{s}) = 1 + \alpha$. Figure 10.8 (*left-hand*) shows the resulting grid lines obtained by solution of equations (10.40) with $w(\mathbf{s}) = 1/v(\mathbf{s})$. Note that the grid spacing is coarse where v is small and fine where v is large. Thus, the grid is refined where the spatial truncation error is large and rarefied where it is small. Figure 10.8 (*left-hand*) exhibits a grid for both alignment and adaptation and scaled grid density. Figure 10.8 (*center*) shows the

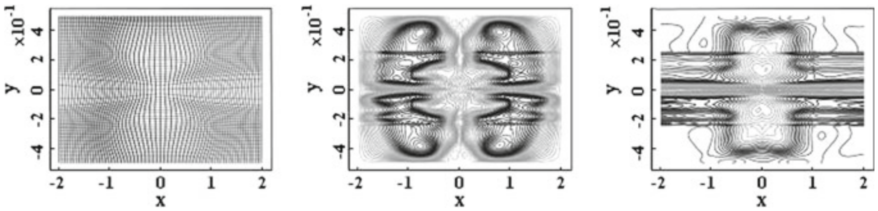


Fig. 10.8 Grid lines for both alignment and adaptation (*left*). Contour plot of alignment error for both alignment and adaptation (*center*). Density of grid lines for both alignment and adaptation (*right*)

resulting weight function for a magnetic reconnection problem. Figure 10.8 (*right-hand*) shows a contour plot of the inverse Jacobian of the transformation $\mathbf{s}(\boldsymbol{\xi})$, which may be interpreted as grid density. The pictures in Fig. 10.8 were formed by A. Glasser who used a spectral element method, developed by Glasser and Tang (2004), for computing plasmas and inverted diffusion grid equations for generating adaptive, field-aligned grids (see Glasser et al. (2005, 2006)).

10.4 Finite Element Method

The finite element method has diverse applications to problems in engineering and science. We demonstrate here its application to numerical grid generation by solution of problem (10.17) for $n = 2$ whose equations are written as

$$\frac{\partial s^l}{\partial t} - (J)^p\Big[g^{\xi}_{22}\frac{\partial^2 s^l}{\partial\xi^1\partial\xi^1} - 2g^{\xi}_{12}\frac{\partial^2 s^l}{\partial\xi^1\partial\xi^2} + g^{\xi}_{11}\frac{\partial^2 s^l}{\partial\xi^2\partial\xi^2} - R^l(\mathbf{s})\Big] = 0\,,\quad l = 1, 2\,. \tag{10.54}$$

These equations are replaced by the following relations:

$$\int\limits_{\Xi^2}\Big(\frac{\partial s^l}{\partial t}v^h - (J)^p\Big[(-1)^{i+j}g^{\xi}_{3-i3-j}\frac{\partial^2 s^l}{\partial\xi^i\partial\xi^j}v^h - R^l v^h\Big]\Big)\mathrm{d}\boldsymbol{\xi} = 0\,,\quad i, j, l = 1, 2\,, \tag{10.55}$$

where v^h are trial functions. Choosing a basis $\varphi_1, \ldots, \varphi_N$ for the trial functions at the interior grid nodes

$$\varphi_p(\boldsymbol{\xi}_k) = \begin{cases} 1, & k = p \\ 0, & k \neq p, \end{cases}$$

and another basis $\phi_{N+1}, \ldots, \phi_{N^r}$, at the boundary grid nodes,

$$\phi_p(\boldsymbol{\xi}_k) = \begin{cases} 1, & k = p \\ 0, & k \neq p, \end{cases}$$

where N^{Γ} is a number of interior and boundary grid nodes, we have an expansion of the functions $s^l(\boldsymbol{\xi}, t)$, $l = 1,\ 2$

$$s^l(\boldsymbol{\xi}, t) = s_p^{l\Gamma}\phi_p + s_k^l\varphi_k\ , \quad k = 1, \dots, N, \quad p = N+1, \dots, N^{\Gamma}\ , \quad l = 1, 2\ . \tag{10.56}$$

Therefore, from (10.55), we obtain the following system of equations:

$$\begin{aligned}\frac{\partial s_k^l}{\partial t}\int\limits_{\Xi^2}\varphi_k\varphi_m \mathrm{d}\boldsymbol{\xi} &= -s_k^l\int\limits_{\Xi^2}\frac{\partial}{\partial\xi^i}\Big((J)^p(-1)^{i+j}g^{\xi}_{3-i3-j}\varphi_k\Big)\frac{\partial\varphi_m}{\partial\xi^j}\mathrm{d}\boldsymbol{\xi}\\ &-\int\limits_{\Xi^2}(J)^pR^l\varphi_m\mathrm{d}\boldsymbol{\xi}, \quad k, m = 1, \dots, N, \quad i, j, l = 1, 2,\end{aligned} \tag{10.57}$$

or in a matrix form $\boldsymbol{M} = \{M_{mk}\}$, $\boldsymbol{K} = \{K_{mk}\}$, $\boldsymbol{F} = \{F_1, \dots, F_m\}$, $k,\ m = 1, \dots, N$

$$\boldsymbol{M}\frac{\partial \boldsymbol{s}^l}{\partial t} = \boldsymbol{K}\boldsymbol{s}^l - \boldsymbol{F}\ , \tag{10.58}$$

where

$$\begin{aligned}M_{mk} &= \int\limits_{\Xi^2}\varphi_k\varphi_m\mathrm{d}\boldsymbol{\xi}, \quad K_{mk} = -\int\limits_{\Xi^2}\frac{\partial}{\partial\xi^i}\Big((J)^p(-1)^{i+j}g^{\xi}_{3-i3-j}\varphi_k\Big)\frac{\partial\varphi_m}{\partial\xi^j}\mathrm{d}\boldsymbol{\xi},\\ F_m &= \int\limits_{\Xi^2}(J)^pR^l\varphi_m\mathrm{d}\boldsymbol{\xi}, \quad k, m = 1, \dots, N, \quad i, j, l = 1, 2.\end{aligned} \tag{10.59}$$

Solving system (10.58) gives the values s_k^l, $l = 1, 2$, $k = 1, \dots, N$, and consequently the values of the grid node coordinates. A more detailed description of the algorithm was originally published in Vaseva and Liseikin (2011).

Figures 10.9 and 10.10 illustrate an application of the finite element method to generation of adaptive triangle grids based on the solution of inverted diffusion equations.

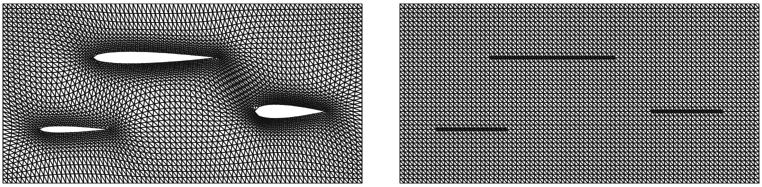


Fig. 10.9 Adaptive grid with node clustering near the boundaries of wings specified analytically (*left*) and the reference grid in Ξ^2 (*right*)

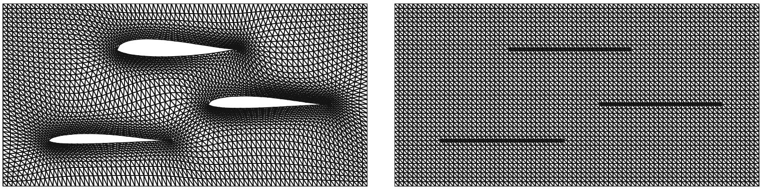


Fig. 10.10 Adaptive grid with node clustering near the boundaries of wings specified discretely (*left*) and the reference grid in Ξ^2 (*right*)

10.5 Inverse Matrix Method

Sections 10.1–10.4 consider algorithms for generating numerical grids by solving a matrix equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \quad \boldsymbol{A} = \{a_{ij}\}, \quad i, j = 1, \dots, M, \tag{10.60}$$

without finding the inverse matrix $\boldsymbol{A}^{-1}$. This section describes an algorithm for finding the inverse matrix $\boldsymbol{A}^{-1}$ provided that there exists a nondegenerate matrix $\boldsymbol{A}(s)$ whose coefficients are dependent on a parameter s, $0 \le s \le 1$, i.e. $\boldsymbol{A}(s) = \{a_{ij}(s)\}$ and $\boldsymbol{A}(1) = \boldsymbol{A}$, while the inverse matrix $\boldsymbol{A}^{-1}(0)$ of the matrix $\boldsymbol{A}(0)$ is known. For example, if $\boldsymbol{A} = \{a_{ij}\}$ is some matrix with a diagonal domination, then $\boldsymbol{A}(s)$ may be defined as

$$\boldsymbol{A}(s) = (1-s)\boldsymbol{D} + s\boldsymbol{A}, \quad \text{i. e.} \quad a_{ij}(s) = (1-s)d_{ij} + sa_{ij}, \quad i, j = 1, \dots, M, \tag{10.61}$$

where $\boldsymbol{D} = \{d_{ij}\}$ is the matrix whose elements equal zero if $i \ne j$, i.e. $d_{ij} = 0$, $i \ne j$, and its diagonal elements coincide with the diagonal elements of the matrix $\boldsymbol{A}$, i.e. $d_{ii} = a_{ii}$ for every fixed index $i = 1, \dots, M$. Thus, $\boldsymbol{A}(0) = \boldsymbol{D}$, $\boldsymbol{D}^{-1} = \{b^{ij}\}$, $b^{ij} = 0$ if $i \ne j$, $b^{ii} = 1/a_{ii}$, i – fixed. Taking into account that $a_{ij}(s)b^{jk}(s) = \delta^i_j$, $i,\ j = 1, \dots, M$, we have

$$\frac{\partial}{\partial a_{lp}}(a_{ij}b^{jk}) = \delta^l_i \delta^p_j b^{jk} + a_{ij}\frac{\partial b^{jk}}{\partial a_{lp}} = \delta^l_i b^{pk} + a_{ij}\frac{\partial b^{jk}}{\partial a_{lp}} = 0,\ i, j, k, l, p = 1, \dots, M\,.$$

Multiplying these equations by b^{ti} and summing over i, we obtain

$$\frac{\partial b^{tk}(s)}{\partial a_{lp}(s)} = -b^{tl}(s)b^{pk}(s)\ , \quad k, l, p, t = 1, \dots, M\ .$$

Therefore, for the elements of the inverse matrix $\boldsymbol{A}^{-1}(s) = \{b^{ij}(s)\}$, we obtain a system of ordinary nonlinear differential equations

$$\frac{\mathrm{d}}{\mathrm{d}s}b^{ij}(s) = \frac{\partial b^{ij}(s)}{\partial a_{kl}(s)}\frac{\mathrm{d}}{\mathrm{d}s}a_{kl}(s) = -b^{ik}(s)b^{lj}(s)\frac{\mathrm{d}}{\mathrm{d}s}a_{kl}(s)\ , \\ i, j, k, l = 1, \dots, M, \quad 0 < s \le 1\ , \tag{10.62}$$

with the initial condition $b^{ij}(0)$, $i,\ \ j = 1, \dots, M$, for $s = 0$. In particular, for the matrix $\{a_{ij}(s)\}$ defined by formula (10.61), we obtain an autonomous system of ordinary differential equations with the initial condition:

$$\frac{\mathrm{d}}{\mathrm{d}s}b^{ij}(s) = -b^{ik}(s)b^{lj}(s)(1-\delta^{kl})a_{kl}, \quad i, j, k, l = 1, \dots, M, \quad 0 < s \le 1,$$
$$b^{ij}(0) = 0, \quad \text{if } i \ne j, \quad b^{ij}(0) = \frac{1}{a_{ij}}, \quad \text{if } i = j, \quad i, j = 1, \dots, M\ .$$

Solving the initial problem for Eq. (10.62), for example, through the Runge Kutta method on the numerical grid $s_i = ih$, $i = 0, \dots, N$, $h = 1/N$, we find for $s = 1$ approximate values of the elements of the matrix $\boldsymbol{A}^{-1}$. Then, an approximate solution of problem (10.60) is obtained as $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$.

Note that the original description of the method was given in Liseikin (2014a, b).

10.6 Method of Minimization of Energy Functional

This section describes another finite-difference grid generation algorithm based on the minimization of inverted energy functional (9.23). Following Charakch'yan and Ivanenko (1988, 1997), the algorithm is first expounded for the two-dimensional version of the functional in the Euler metric, i.e.

$$I_{IS}[\mathbf{s}] = \int\limits_{\Xi^2} \frac{(x_\xi)^2 + (x_\eta)^2 + (y_\xi)^2 + (y_\eta)^2}{J}\mathrm{d}\xi\mathrm{d}\eta, \tag{10.63}$$

where $J = x_\xi y_\eta - x_\eta y_\xi$, and then an explanation is given as to how it can be generalized to monitor metrics and other dimensions. Note that the functional (9.23) in the Euler metric $g^{\mathrm{s}}_{ij} = \delta^i_j$, $i, j = 1, 2$, becomes the functional (10.63) when the following designations are assumed:

$$\xi^1 = \xi, \quad \xi^2 = \eta, \quad \mathbf{s}(\xi, \eta) = [x(\xi, \eta), y(\xi, \eta)].$$

By the algorithm, the functional (10.63) is approximated by a discrete functional $I^h[\ ]$. This is made by approximating the integrand in (10.63) at each grid cell of the logical domain Ξ^2 and then carrying out summation over all cells.

10.6.1 Generation of Fixed Grids

The problem of grid generation is treated as a discrete analog of the problem of finding the components $x(\xi, \eta)$ and $y(\xi, \eta)$ of the intermediate transformation $\mathbf{s}(\xi, \eta)$ producing one-to-one mapping of the logical square

$$0 < \xi < 1, \ 0 < \eta < 1$$

onto a physical domain X^2.

Instead of the logical square on the plane ξ, η, the parametric rectangle

$$1 < \xi < N, \ 1 < \eta < M.$$

is introduced to simplify the computational formulas. This rectangle is associated with the square grid (ξ_i, η_j) on the plane ξ, η such that $\xi_i = i, \ \eta_j = j, \ i = 1, \dots, N; \ j = 1, \dots, M.$

It is readily shown that if a smooth mapping of one domain onto another with a one-to-one transformation between boundaries possesses a positive Jacobian, then such a mapping will be one-to-one. Hence, the grid coordinate system, generated in the domain X^2, will be non-degenerate if the Jacobian of the mapping $\mathbf{s}(\xi, \eta) = [x(\xi, \eta), y(\xi, \eta)]$ is positive:

$$J = x_\xi y_\eta - x_\eta y_\xi > 0. \tag{10.64}$$

Thus, the problem of the construction of the grid coordinates in the domain X^2 can be formulated as the problem of finding a smooth mapping of the parametric rectangle onto the domain X^2, which satisfies the condition of the Jacobian positiveness.

Formulation of Discrete Functional

Let the coordinates $(x, y)_{ij}$ of grid nodes be given. To construct the mapping $x^h(\xi, \eta)$, $y^h(\xi, \eta)$ of the parametric rectangle onto the domain X^2 such that $x^h(i, j) = x_{ij}$ and $y^h(i, j) = y_{ij}$, quadrilateral isoparametric finite elements are used. The square cell numbered as $i + 1/2, j + 1/2$ on the plane ξ, η is mapped onto the quadrilateral cell on the plane x, y, formed by the nodes with coordinates $(x, y)_{ij}$, $(x, y)_{ij+1}$, $(x, y)_{i+1j+1}$, $(x, y)_{i+1j}$. The cell vertices are numbered from 1 to 4 in the clockwise direction. The node (i, j) corresponds to the vertex 1, node $(i, j + 1)$ to vertex 2, and so on. Each vertex is associated with a triangle: vertex 1 with $\triangle_{412}$, vertex 2 with $\triangle_{123}$, and so on. The doubled area $J_k, \ k = 1, 2, 3, 4$, of these triangles is introduced as follows:

$$J_k = (x_{k-1} - x_k)(y_{k+1} - y_k) - (y_{k-1} - y_k)(x_{k+1} - x_k)$$

where one should put $k - 1 = 4$ if $k = 1$, $k + 1 = 1$ if $k = 4$.

The functions x^h, y^h for $i \le \xi \le i+1,\ j \le \eta \le j+1$ are represented in the form

$$\begin{aligned} x^h(\xi,\eta) &= x_1 + (x_4 - x_1)(\xi - i) + (x_2 - x_1)(\eta - j) \\ &\quad + (x_3 - x_4 - x_2 + x_1)(\xi - i)(\eta - j), \\ y^h(\xi,\eta) &= y_1 + (y_4 - y_1)(\xi - i) + (y_2 - y_1)(\eta - j) \\ &\quad + (y_3 - y_4 - y_2 + y_1)(\xi - i)(\eta - j). \end{aligned} \tag{10.65}$$

Each side of the square is linearly transformed onto the appropriate side of the quadrilateral. Consequently, the global transformation x^h, y^h is continuous on the cell boundaries. To check the one-to-one property of the transformation (10.65), we write out the expression for its Jacobian

$$J^h = x^h_\xi y^h_\eta - x^h_\eta y^h_\xi = \det \begin{pmatrix} x_4 - x_1 + A(\eta - j) & x_2 - x_1 + A(\xi - i) \\ y_4 - y_1 + B(\eta - j) & y_2 - y_1 + B(\xi - i) \end{pmatrix},$$

where $A = x_3 - x_4 - x_2 + x_1$, $B = y_3 - y_4 - y_2 + y_1$. The function J^h is linear, not bilinear, since the coefficient before $\xi\eta$ in this determinant is equal to zero. Consequently, if $J^h > 0$ at all corner points of the square, it does not vanish inside this square. At the corner node 1 ($\xi = i,\ \eta = j$) of the cell $i + 1/2, j + 1/2$, the Jacobian equals

$$J^h(i,j) = (x_4 - x_1)(y_2 - y_1) - (y_4 - y_1)(x_2 - x_1),$$

i.e. $J^h(i,j) = J_1$ is the doubled area of the triangle $\triangle_{412}$, introduced above. From this follows that the condition of the Jacobian positiveness $x^h_\xi y^h_\eta - x^h_\eta y^h_\xi > 0$ is equivalent to the system of inequalities

$$[J_k]_{i+1/2 j+1/2} > 0,\ \ k = 1,2,3,4;\ \ i = 1,\dots,N-1;\ \ j = 1,\dots,M-1. \tag{10.66}$$

If conditions (10.66) are satisfied, then all the grid cells are convex quadrilaterals. The set of grids satisfying these inequalities is called a convex grid set and denoted by D. This set belongs to the Euclidean space R^{N_1}, where $N_1 = 2(N-2)(M-2)$ is the total number of degrees of freedom of the grid equal to twice the number of its interior nodes.

Finally, the problem is formulated as follows. The convex grid, satisfying inequalities (10.66), must be generated in the domain X^2 for the given coordinates of the boundary nodes.

The mapping $x(\xi,\eta)$, $y(\xi,\eta)$ is approximated by functions $x^h(\xi,\eta)$, $y^h(\xi,\eta)$ introduced in (10.65). Substituting those expressions in (10.63) and replacing integrals over square cells by the quadrature formulas with nodes coinciding with the grid vertices on the plane ξ,η, the following discrete analog of the functional (10.63) is obtained:

$$I^h = \sum_{i=1}^{N-1} \sum_{j=1}^{M-1} \sum_{k=1}^{4} \frac{1}{4} [F_k]_{i+1/2\ j+1/2}, \tag{10.67}$$

where F_k is the integrand evaluated in the k - th grid node as

$$F_k = [(x_{k+1} - x_k)^2 + (x_k - x_{k-1})^2 + (y_{k+1} - y_k)^2 + (y_k - y_{k-1})^2]J_k^{-1}, \quad (10.68)$$

and J_k is the doubled area of the triangle introduced above.

Notice some properties of the function (10.67). For this purpose, we introduce a parametric rectangle $0 < \xi < 1, 0 < \eta < \alpha$, where $\alpha = (M-1)/(N-1)$ is the constant, instead of the unit logical square as a domain of integration in (10.63). In this case, the continuous limit of the expression $I^h/(N-1)^2$ when $N, M \to \infty$ in such a way, that $(M-1)/(N-1) = \alpha = const$, will be the functional (10.63).

The following identity is readily obtained:

$$\begin{aligned} I &= \int_0^1 \int_0^\alpha \frac{x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2 - 2(x_\xi y_\eta - x_\eta y_\xi) + 2(x_\xi y_\eta - x_\eta y_\xi)}{J} d\xi d\eta \\ &= \int_0^1 \int_0^\alpha \frac{(x_\xi - y_\eta)^2 + (x_\eta - y_\xi)^2}{J} d\xi d\eta + 2\alpha. \end{aligned}$$

From this follows that the functional (10.63) has a lower bound equal to 2α. If this minimum is attained, the mapping $\mathbf{s}(\xi, \eta)$ is conformal:

$$x_\xi = y_\eta, \quad x_\eta = -y_\xi.$$

To find out the corresponding property of discrete analog (10.67) of functional (10.63), let us consider one term in (10.68) for $k = 2$. We can assume that $x_2 = 0$ and $y_2 = 0$, since expression (10.68) contains only finite differences of the grid node coordinates. In this case, we obtain the following identity:

$$\begin{aligned} F_2 &= \frac{x_1^2 + y_1^2 + x_3^2 + y_3^2}{x_1 y_3 - x_3 y_1} \\ &= \frac{x_1^2 + y_1^2 + x_3^2 + y_3^2 - 2(x_1 y_3 - x_3 y_1) + 2(x_1 y_3 - x_3 y_1)}{x_1 y_3 - x_3 y_1} \\ &= \frac{(x_1 - y_3)^2 + (x_3 + y_1)^2}{x_1 y_3 - x_3 y_1} + 2. \end{aligned}$$

From this follows that the function $I^h/(N-1)^2$ has on the set D a lower bound equal to $2(M-1)/(N-1)$. If this minimum is attained, the coordinates of the grid nodes satisfy a discrete analog of the conformal conditions

$$x_1 = y_3, \quad x_3 = -y_1.$$

If these conditions are satisfied for all cells, every grid cell will be a square.

Note that the function (10.67) is not convex and, in principle, multiple solutions may exist.

The function I^h also possesses the following very important property. If $G \to \partial D$ for $G \in D$, where ∂D is the boundary of the set of convex grids D, i.e. if at least one of the quantities J_k tends to zero for some cell while remaining positive, then $I^h(G) \to +\infty$. In fact, suppose that $J_k \to 0$ in (10.68) for some cell, but I^h does not tend to $+\infty$. Then, the numerator in (10.68) must also tend to zero, i.e. the lengths of two sides of the cell tend to zero. Consequently, the areas of all triangles that contain these sides must also tend to zero. Repeating the argument as many times as necessary, we conclude that the lengths of the sides of all grid cells, including those at the boundary of the domain, must tend to zero, which is impossible.

Thus, if the set D is not empty, the system of algebraic equations

$$R_x = \frac{\partial I^h}{\partial x_{ij}} = 0, \;\; R_y = \frac{\partial I^h}{\partial y_{ij}} = 0, \quad i = 2, \ldots, N-1; \;\; j = 2, \ldots, M-1, \tag{10.69}$$

has at least one solution which is a convex grid. To find it, one must first find a certain initial grid $G_0 \in D$, and then use some method of unconstrained minimization. Since the function (10.67) has the infinite barrier on the boundary of the set D, each step of the method can be chosen so that the grid always remains convex. Note that in the common case, the discrete grid-generation Eq. (10.69) may have multiple solutions, but numerical experiments have not met with such opportunity.

Method of Minimization

First, we consider a method for minimizing the function (10.67) assuming that the initial grid $G_0 \in D$ has been found. Suppose the grid at the l-th step of the iterations is determined. For finding the grid nodes at the $(l+1)$-th step, the quasi-Newtonian procedure for each interior node can be used:

$$\begin{aligned} \tau R_x + \frac{\partial R_x}{\partial x_{ij}}(x_{ij}^{l+1} - x_{ij}^l) + \frac{\partial R_x}{\partial y_{ij}}(y_{ij}^{l+1} - y_{ij}^l) = 0, \\ \tau R_y + \frac{\partial R_y}{\partial x_{ij}}(x_{ij}^{l+1} - x_{ij}^l) + \frac{\partial R_y}{\partial y_{ij}}(y_{ij}^{l+1} - y_{ij}^l) = 0 \end{aligned} \tag{10.70}$$

where τ is the iteration parameter. Note that (10.70) is not the Newton-Raphson iteration, because only a part of the second derivatives of (10.67) is taken into account. The rate of convergence for (10.70) is low by comparison. At the same time, the Newton-Raphson method gives us a much more complex system of linear equations at each iteration.

Each of the derivatives in (10.70) is the sum of twelve terms, in accordance with the number of triangles containing the given node as a vertex. Rather than write out such cumbersome expressions, the first and second derivatives of the terms in (10.67) are considered:

$$\frac{\partial F_k}{\partial x_{k-1}} = 2\frac{x_{k-1} - x_k}{J_k} - F_k \frac{y_{k+1} - y_k}{J_k} \;, \tag{10.71}$$

and so on. Arrays storing the derivatives of the function (10.67) were first cleared, and then all grid triangles were scanned and the appropriate derivatives added to the relevant elements of the arrays.

Now, an algorithm is suggested for the choice of the iteration parameter τ in (10.70), which was used only for the problems with moving boundaries. Recall that the minimized function (10.67) has the infinite barrier on the boundary of the set of convex grids D. Since the initial grid G_0 is convex, the iteration (10.70) gives, as a rule, a convex grid for any $\tau < 1$. But in extreme cases when G_0 is very close to the boundary of the set D, the grid $G(\tau)$ can cross the boundary of the set in the first iterations (10.70). Clearly, such a condition is fatal for the method because the same barrier on the boundary of the set D does not allow the iterations to return into the set D in the following iterations. To avoid this, a certain basic parameter τ_0 is chosen so that $G(\tau_0/2) \in D$ and $G(\tau_0) \in D$. In the beginning, $\tau_0 = 1$. If the above-mentioned conditions are violated, we put $\tau_0 = 1/4$ or $\tau_0 = 1/2$, depending on whether the grids $G(\tau_0/2)$ or $G(\tau_0)$ leave the set D, and so on.

In fixed boundary problems, the simple choice $\tau = \text{const} \cdot \tau_0$ is used. For time-dependent problems with moving boundaries, a version of the method of parabolas was developed. As the controlling quantity, the squared residual of the Eq. (10.70)

$$W = \sum_{i,j} (R_x^2 + R_y^2)_{i,j}$$

was used. The parabola $W(\tau)$ is constructed from the grids obtained for $\tau = 0$, $\tau = \tau_0/2$ and $\tau = \tau_0$. The parameter τ is then chosen so that $W(\tau) = \min$ in the interval $\theta\tau \le \tau \le \alpha\tau_0$. The parameter $\theta \sim 0.1$ is given a priori and bounds the value of τ away from zero. The parameter α bounds τ above, i.e. prevents a very large extrapolation along the parabola. If $\tau_0 = 1$, i.e. if the boundary of the set D is not crossed, we put $\alpha = 2$. If $\tau_0 < 1$, then $\alpha = 1$. Finally, if the algorithm gives $\tau < \tau_0/2$, the condition $I^h(\tau_0/2) < I^h(0)$ is checked. In the cases when this condition is found to be valid, $\tau = \tau_0/2$ was used.

For one iteration of the above method, a measurement of the computational cost gives the value of about double (but not three times) the cost of the simple iteration. The reason is that the second derivatives of the function (10.67) are not used in calculating W, while they are used in (10.70) to calculate the direction of minimization.

The algorithm described can be used only if the initial grid is convex. Otherwise, it is necessary either to obtain a convex grid through another algorithm as a preliminary stage of the method or to modify the computational formulas. The first approach is based on the minimization of the following function:

$$I_D = \sum_{i=1}^{N-1} \sum_{j=1}^{M-1} \sum_{k=1}^{4} \left([\epsilon - J_k]_{i+1/2\ j+1/2}\right)_+^2, \ (f)_+ = \max(0, f), \tag{10.72}$$

for some given $\epsilon > 0$. This is accomplished through the gradient method with a suitable choice of the iteration parameter. The iterative process is broken off as soon as all inequalities (10.66) are satisfied. This method was used by Charakch'yan (1993, 1994) for studying gas dynamics problems with moving boundaries when the initial interior grid nodes for minimizing (10.72) were taken from the previous time step. As a result, the initial grid is either convex or such that a convex grid is obtained after a few iterations.

In fixed boundary problems, the starting grid may be non-convex, containing numerous self-intersecting cells. In such a case the preliminary stage of the method based on minimizing (10.72) can be unsuitable. Therefore, another approach had been developed by Ivanenko (1988). The computational formulas (10.70) were modified so that the initial grid need not belong to the set D of convex grids. The quantities J_k appearing in the expressions for R_x, R_y and their derivatives are replaced with new quantities $\tilde{J}_k$

$$\tilde{J}_k = \begin{cases} J_k & if \ J_k > \epsilon, \\ \epsilon & if \ J_k \le \epsilon, \end{cases}$$

where $\epsilon > 0$ is some sufficiently small quantity.

It is quite important to choose an optimal value of ϵ so that the convex grid is constructed as quickly as possible. The method used for specifying the value of ϵ is based on the computation of the absolute value of the average area of triangles with negative areas

$$\epsilon = \max[\alpha S/(N + 0.01), \epsilon_1],$$

where S is twice the absolute value of the total area of triangles with negative areas, and N the number of these triangles. The quantity $\epsilon_1 > 0$ sets a lower bound on ϵ to avoid very large values appearing in the computations. The coefficient α is chosen experimentally and is in the range $0.3 \le \alpha \le 0.7$.

In practical implementation, an arbitrary set of grid nodes can be marked as movable during iterations, while all other nodes are considered as stationary. All the terms in the function (10.67) which become independent on movable nodes are excluded from computations. Since the boundary nodes are always marked as stationary, four terms in (10.67) corresponding to "corner" triangles $\{(1, 2); (1, 1); (2, 1)\}$, $\{(N - 1, 1); (N, 1); (N, 2)\}$, $\{(1, M - 1); (1, M); (2, M)\}$, and $\{(N - 1, M); (N, M); (N, M - 1)\}$ are always excluded from computations. As a result, the method becomes applicable to those domains for which the angle between two intersecting boundaries is greater than or equal to π, despite the fact that the corresponding grid cell becomes non-convex regardless of the positions of interior nodes.

Examples of the grids generated by this method are exhibited in Figs. 10.11 and 10.12. Figure 10.12 demonstrates the application of the algorithm to generation of a grid for computing a high-velocity impact of a thin foil (a) upon a conical target CD Lomonosov et al. (1997).

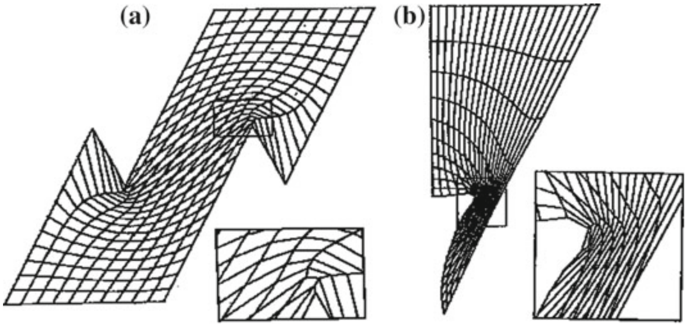


Fig. 10.11 Grids in a model domain (**a**) and for computing a cumulative jet (**b**)

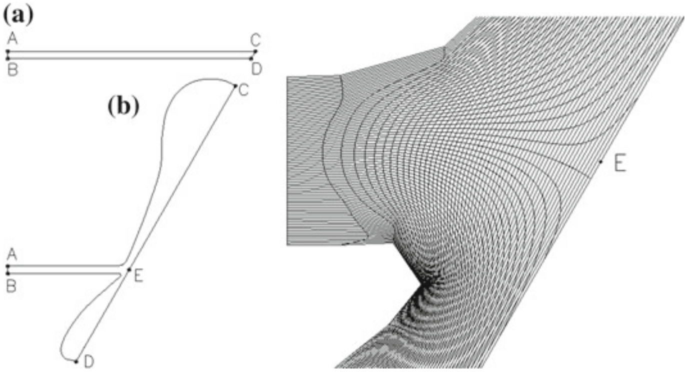


Fig. 10.12 A fragment of the grid (*right*) in the vicinity of the point E (*left*)

10.6.2 Adaptive Grid Generation

Numerical Algorithm

One approach to adaptive grid generation is based on the minimization of the functional (9.23) in the metric of a monitor surface.

Let the monitor surface be defined by a function $z = f(x, y)$ where $f \in C^1$. The expressions for the covariant elements and Jacobian of the monitor metric in the grid coordinates $\xi = \xi^1,\ \eta = \xi^2$ are as follows:

$$\begin{aligned} g^{\xi}_{11} &= g^{\mathrm{s}}_{11}\left(\frac{\partial x}{\partial \xi}\right)^2 + 2g^{\mathrm{s}}_{12}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} + g^{\mathrm{s}}_{22}\left(\frac{\partial y}{\partial \xi}\right)^2, \\ g^{\xi}_{22} &= g^{\mathrm{s}}_{11}\left(\frac{\partial x}{\partial \eta}\right)^2 + 2g^{\mathrm{s}}_{12}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} + g^{\mathrm{s}}_{22}\left(\frac{\partial y}{\partial \eta}\right)^2, \\ g^{\xi} &= (J)^2 g^{\mathrm{s}} = (J)^2[1 + (f_x)^2 + (f_y)^2], \end{aligned}$$

where

$$g^{\mathrm{s}}_{11} = 1 + (f_x)^2, \quad g^{\mathrm{s}}_{12} = f_x f_y, \quad g^{\mathrm{s}}_{22} = 1 + (f_y)^2.$$

Then, functional (9.23) for $n = 2$, with the identification $X^2 = S^2$, $\xi = \xi^1,\ \eta = \xi^2$, has the following form:

$$I_a[\boldsymbol{x}] = \int_0^1\int_0^1 \frac{(x_\xi^2 + x_\eta^2)[1+(f_x)^2] + (y_\xi^2 + y_\eta^2)[1+(f_y)^2] + 2f_x f_y(x_\xi y_\xi + x_\eta y_\eta)}{J[1+(f_x)^2+(f_y)^2]^{1/2}}d\xi d\eta. \tag{10.73}$$

Now we again consider the grid $(x, y)_{ij},\ i = 1, \dots, N;\ j = 1, \dots, M$ and, to simplify the computational formulas, the parametric rectangle $1 < \xi < N,\ 1 < \eta < M$ substitutes for the unit square $0 < \xi < 1,\ 0 < \eta < 1$. The functional I_a is approximated by the function

$$I_a^h = \sum_{i=1}^{N-1}\sum_{j=1}^{M-1}\sum_{k=1}^{4}\frac{1}{4}\,[F_k]_{i+1/2\ j+1/2}\,, \tag{10.74}$$

$$F_k = \frac{D_1[1+(f_x)_k^2] + D_2[1+(f_y)_k^2] + 2D_3(f_x)_k(f_y)_k}{J_k[1+(f_x)_k^2+(f_y)_k^2]^{1/2}}, \tag{10.75}$$

where

$$\begin{aligned}
D_1 &= (x_{k-1}-x_k)^2 + (x_{k+1}-x_k)^2,\\
D_2 &= (y_{k-1}-y_k)^2 + (y_{k+1}-y_k)^2,\\
D_3 &= (x_{k-1}-x_k)(y_{k-1}-y_k) + (x_{k+1}-x_k)(y_{k+1}-y_k),\\
J_k &= (x_{k-1}-x_k)(y_{k+1}-y_k) - (x_{k+1}-x_k)(y_{k-1}-y_k).
\end{aligned}$$

Derivatives $(f_x)_k$ and $(f_y)_k$ in the k-th cell vertex are equal to the corresponding values of derivatives, evaluated at the grid node ij

$$\begin{aligned}
(f_x)_{ij} &= \frac{(f_{i+1j}-f_{i-1j})(y_{ij+1}-y_{ij-1}) - (f_{ij+1}-f_{ij-1})(y_{i+1j}-y_{i-1j})}{(x_{i+1j}-x_{i-1j})(y_{ij+1}-y_{ij-1}) - (x_{ij+1}-x_{ij-1})(y_{i+1j}-y_{i-1j})},\\
(f_y)_{ij} &= \frac{(f_{i+1j}-f_{i-1j})(x_{ij+1}-x_{ij-1}) - (f_{ij+1}-f_{ij-1})(x_{i+1j}-x_{i-1j})}{(x_{i+1j}-x_{i-1j})(y_{ij+1}-y_{ij-1}) - (x_{ij+1}-x_{ij-1})(y_{i+1j}-y_{i-1j})}.
\end{aligned} \tag{10.76}$$

These formulas must be modified for the boundary nodes. Indices "leaving" the computational domain must be replaced by the nearest boundary indices. For example, if $j = 1$, then $(i, j-1)$ must be replaced by (i, j).

Function (10.74) possesses the same property as the function (10.67): $I_a^h(G) \to +\infty$ if $G \to \partial D$ for $G \in D$ where D is the set of convex grids, and ∂D is the boundary of the set.

As before, Eq. (10.70) are used to minimize the function I_a^h. Quantities $(f_x)_{ij}$ and $(f_y)_{ij}$ are assumed to be parameters, and therefore all their derivatives in (10.70) vanish. Note that if $(f_x)_{ij}$ and $(f_y)_{ij}$ vanish, the function I_a^h reduces to the function I^h (10.67).

The adaptive grid generation algorithm is formulated as follows:

1. Generate a grid for the given domain using the unconstrained minimization algorithm described.

2. Compute the values of the control function at each grid node. The result is f_{ij}.

3. Evaluate derivatives $(f_x)_{ij}$ and $(f_y)_{ij}$ using the formulas (10.76).

4. Make one step in the minimization process for the function I_a^h using Eq. (10.70) and compute new values of x_{ij} and y_{ij}.

5. Repeat starting with Step 2 to convergency.

It is important that at each step of the iterative process the grid remains convex.

Redistribution of Boundary Nodes

There are several ways to redistribute the grid nodes along the boundary ∂X^2 of the domain X^2 during adaptation. The simplest one is a fixed position of every point on ∂X^2, referred to as the "fixed position." However, if some physical quantities are not smooth (e.g. shock waves), then some instability in the mesh generation and, consequently, in the physical problem solution near the points where the discontinuity joins ∂X^2 may arise. In some methods, referred to as "unconstrained minimization", the boundary nodes are treated as interior and the vectors of their shift are projected onto ∂X^2. This method can be used only if the discontinuity is nearly orthogonal to ∂X^2. If not, then, when condensing, the boundary nodes overlap, adjacent cells degenerate, and modeling breaks. The next method, referred to as "1-D minimization", relies on using the 1-D functional along ∂X^2. This method is more robust than the two ones discussed above and can usually be used for adaptation. However, the 1-D and 2-D functionals are, as a rule, inconsistent. For this reason, the parameters of adaptation for the interior and boundary nodes should be selected separately. It requires additional work when modeling unsteady flow problems.

In the method suggested by Azarenok (2002), instead of (10.74), the function

$$\tilde{I}_a^h = \sum_{i=1}^{N-1} \sum_{j=1}^{M-1} \sum_{k=1}^{4} \frac{1}{4} [F_k]_{i+1/2 j+1/2} + \sum_{ij \in \mathcal{L}} \lambda_{ij} G_{ij} = I_a^h + \sum_{ij \in \mathcal{L}} \lambda_{ij} G_{ij}, \qquad (10.77)$$

was minimized where the constraints $G_{ij} = G(x_{ij}\ ,\ y_{ij}) = 0$ define ∂X^2, λ_{ij} are the Lagrange multipliers, and $\mathcal{L}$ is the set of the boundary nodes. The function $G(x, y)$ is assumed to be piecewise differentiable, so the function $\tilde{I}_a^h$ holds the infinite barrier on the boundary of the set of convex grids as I_a^h does if $f \in C^1$.

If the set of convex grids is not empty, the system of algebraic equations

$$R_x = \frac{\partial I_a^h}{\partial x_{ij}} + \lambda_{ij} \frac{\partial G_{ij}}{\partial x_{ij}} = 0, \quad R_y = \frac{\partial I_a^h}{\partial y_{ij}} + \lambda_{ij} \frac{\partial G_{ij}}{\partial y_{ij}} = 0, \quad G_{ij} = 0, \quad (10.78)$$

has at least one solution that is a convex mesh. Here, $\lambda_{ij} = 0$ if $ij \notin \mathcal{L}$ and the constraints are defined for the boundary nodes $ij \in \mathcal{L}$.

Consider the method of minimizing the function (10.77) assuming the grid to be convex at the lth step of the iterative procedure. The quasi-Newton procedure for finding, the coordinates x_{ij}^{l+1}, y_{ij}^{l+1} from the system (10.78) was used:

$$\begin{aligned}
&\tau R_x + \frac{\partial R_x}{\partial x_{ij}}\,(x_{ij}^{l+1} - x_{ij}^{l}) + \frac{\partial R_x}{\partial y_{ij}}\,(y_{ij}^{l+1} - y_{ij}^{l}) + \frac{\partial R_x}{\partial \lambda_{ij}}\,(\lambda_{ij}^{l+1} - \lambda_{ij}^{l})=0,\\
&\tau R_y + \frac{\partial R_y}{\partial x_{ij}}\,(x_{ij}^{l+1} - x_{ij}^{l}) + \frac{\partial R_y}{\partial y_{ij}}\,(y_{ij}^{l+1} - y_{ij}^{l}) + \frac{\partial R_y}{\partial \lambda_{ij}}\,(\lambda_{ij}^{l+1} - \lambda_{ij}^{l})=0,\\
&\tau G_{ij} + \frac{\partial G_{ij}}{\partial x_{ij}}\,(x_{ij}^{l+1} - x_{ij}^{l}) + \frac{\partial G_{ij}}{\partial y_{ij}}\,(y_{ij}^{l+1} - y_{ij}^{l}) = 0,
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial R_x}{\partial x_{ij}} &= \frac{\partial^2 I_a^h}{\partial x_{ij}^2} + \lambda_{ij}\frac{\partial^2 G_{ij}}{\partial x_{ij}^2}\,, \quad \frac{\partial R_x}{\partial y_{ij}} = \frac{\partial^2 I_a^h}{\partial x_{ij}\partial y_{ij}} + \lambda_{ij}\frac{\partial^2 G_{ij}}{\partial x_{ij}\partial y_{ij}}\,,\\
\frac{\partial R_y}{\partial x_{ij}} &= \frac{\partial^2 I_a^h}{\partial x_{ij}\partial y_{ij}} + \lambda_{ij}\frac{\partial^2 G_{ij}}{\partial x_{ij}\partial y_{ij}}\,, \quad \frac{\partial R_y}{\partial y_{ij}} = \frac{\partial^2 I_a^h}{\partial y_{ij}^2} + \lambda_{ij}\frac{\partial^2 G_{ij}}{\partial y_{ij}^2}\,,\\
\frac{\partial R_x}{\partial \lambda_{ij}} &= \frac{\partial G_{ij}}{\partial x_{ij}}\,, \quad \frac{\partial R_y}{\partial \lambda_{ij}} = \frac{\partial G_{ij}}{\partial y_{ij}}\,.
\end{aligned}$$

Resolving the last equation of (10.79) with respect to $y_{ij}^{l+1} - y_{ij}^{l}$ and substituting it in the two remaining equations, the system

$$\begin{pmatrix} a_{11} & a_{12}\\ a_{21} & a_{22}\end{pmatrix}\begin{pmatrix} x_{ij}^{l+1} - x_{ij}^{l}\\ \lambda_{ij}^{l+1} - \lambda_{ij}^{l}\end{pmatrix} = \begin{pmatrix} a_{13}\\ a_{23}\end{pmatrix},$$

is obtained, where

$$\begin{aligned}
a_{11} &= \frac{\partial R_x}{\partial x_{ij}} - \frac{\partial R_x}{\partial y_{ij}}\,\frac{\partial G_{ij}}{\partial x_{ij}}\Big/\frac{\partial G_{ij}}{\partial y_{ij}}\,,\\
a_{12} &= \frac{\partial G_{ij}}{\partial x_{ij}}\,,\\
a_{13} &= \tau\left[\frac{\partial R_x}{\partial y_{ij}}\,G_{ij}\Big/\frac{\partial G_{ij}}{\partial y_{ij}} - R_x\right],\\
a_{21} &= \frac{\partial R_y}{\partial x_{ij}} - \frac{\partial R_y}{\partial y_{ij}}\,\frac{\partial G_{ij}}{\partial x_{ij}}\Big/\frac{\partial G_{ij}}{\partial y_{ij}}\,,\\
a_{22} &= \frac{\partial G_{ij}}{\partial y_{ij}}\,,\\
a_{23} &= \tau\left[\frac{\partial R_y}{\partial y_{ij}}\,G_{ij}\Big/\frac{\partial G_{ij}}{\partial y_{ij}} - R_y\right].
\end{aligned}$$

Denoting $\triangle = a_{11}a_{22} - a_{12}a_{21}$, $\triangle_1 = a_{13}a_{22} - a_{23}a_{12}$, $\triangle_2 = a_{11}a_{23} - a_{21}a_{13}$ (since $G_{ij} = 0$, the terms a_{13}, a_{23} are simplified), we obtain

$$x_{ij}^{l+1} = x_{ij}^{l} + \triangle_1/\triangle, \quad \lambda_{ij}^{l+1} = \lambda_{ij}^{l} + \triangle_2/\triangle, \tag{10.79}$$

while y_{ij}^{l+1} is determined from the third equation of (10.79). If the constraints are resolved in y in the form $G(x, y) = y - g(x) = 0$, then

$$\frac{\partial G_{ij}}{\partial x_{ij}} = -\frac{\partial g_{ij}}{\partial x_{ij}}, \quad \frac{\partial G_{ij}}{\partial y_{ij}} = 1,$$

and the upper formulas are simplified. Analogously, the constrains may be resolved in x in the form $G(x, y) = x - \tilde{g}(y) = 0$. Note that the equation $G(x, y) = 0$ can be locally resolved by one of these two forms.

If ∂X^2 is specified by parametric functions $x = x(t)$, $y = y(t)$ or tabular values $(x, y)_{ij}$, the following algorithm can be used. Assume the index j is fixed and i is variable. When calculating the coordinates of the (ij)th node, in the interval (x_{i-1j}, x_{i+1j}), we construct an interpolating parabola $t = t(x)$ using the values in three nodes $(i-1j)$, (ij), and $(i+1j)$. From (10.79), we compute an intermediate value $\tilde{x}_{ij}^{l+1}$; further from the interpolation formula, we determine $t_{ij} = t(\tilde{x}_{ij}^{l+1})$ and final values x_{ij}^{l+1}, y_{ij}^{l+1} from the parametric formulas.

Another way for redistributing the nodes along ∂X^2, given as parametric functions or by tabular values, employs an unconstrained minimization of the function in a parametric form and is based on solving the following system of algebraic equations, referred to as "parametric minimization":

$$R_t = R_x \frac{\partial x_{ij}}{\partial t_{ij}} + R_y \frac{\partial y_{ij}}{\partial t_{ij}} = 0,$$

via the quasi-Newton procedure

$$\tau R_t + \frac{\partial R_t}{\partial t_{ij}} (t_{ij}^{l+1} - t_{ij}^{l}) = 0. \tag{10.80}$$

Here,

$$\frac{\partial R_t}{\partial t_{ij}} = \frac{\partial R_x}{\partial x_{ij}} \left(\frac{\partial x_{ij}}{\partial t_{ij}}\right)^2 + \frac{\partial R_y}{\partial y_{ij}} \left(\frac{\partial y_{ij}}{\partial t_{ij}}\right)^2 + \left(\frac{\partial R_x}{\partial y_{ij}} + \frac{\partial R_y}{\partial x_{ij}}\right) \frac{\partial x_{ij}}{\partial t_{ij}} \frac{\partial y_{ij}}{\partial t_{ij}}$$
$$+ R_x \frac{\partial^2 x_{ij}}{\partial t_{ij}^2} + R_y \frac{\partial^2 y_{ij}}{\partial t_{ij}^2}, \qquad R_x = \frac{\partial I^h}{\partial x_{ij}}, \quad R_y = \frac{\partial I^h}{\partial y_{ij}}.$$

To the analytical control functions, both the constrained and parametric minimization give similar results. Real-world 2-D flow computations have shown that it is better to perform adaptation along the boundary using constrained minimization (10.79), since the procedure (10.80) may not ensure consistent redistribution of the nodes in X^2 and on ∂X^2.

The use of the constrained minimization without adaptation (i.e. when $f{=}const.$) means that we seek the conformal mapping $x(\xi, \eta)$, $y(\xi, \eta)$ of the parametric rectangle onto the domain X^2 with an additional parameter, the so-called conformal modulus.

10.7 Parallel Mesh Generation

Parallel computing is an efficient tool for handling large multidimensional problems by distributing the computational effort and/or the memory requirements over the different computers available.

As to the mesh generation step, one parallelization approach consists of constructing the mesh in parallel by means of using a meshing method under interest which is to be updated in order to incorporate some degree of parallelism. Many classical methods for mesh generation are amenable to being performed in parallel, in particular, the Delaunay and quad-octree methods and the mapping methods based on the numerical solution of elliptic and parabolic equations such as the inverted Beltrami and diffusion equations.

The second approach to the parallelization of the meshing process consists of partitioning the domain by means of sub-domains, whose union forms a covering-up of the entire domain, prior to dispatching these to different processors, each of them generating a mesh on one sub-domain. Different classes of domain partition are encountered. Among these, some are based on graph partitions and some are purely geometric methods directly based on mesh partitions. All these methods apply to finite element type meshes, since a vicinity graph can be constructed based on the connections between the elements in a given mesh.

The partition of the domain as well as constructing the corresponding sub-meshes can be achieved either through a posteriori or a priori partitioning methods. The posteriori processing starts from the data of a large size fine mesh of the entire domain and then splits it into sub-meshes, while a priori processing uses the domain itself or a coarse mesh of it which is split into sub-domains.

For the priori processing, first, a partition of the domain is created. This step may start from the domain geometry or a reasonable coarse mesh of the entire domain. Once this partition is available, some sub-domains are then identified and meshed, each on one processor, thus taking advantage of the parallel capabilities of the computers right from the meshing stage. The global mesh is then achieved by merging all of the local meshes. The interface between two sub-domains is constructed either from the data of the coarse mesh or from the data of the domain boundary discretization. The meshes can also be constructed by using the meshes of the surfaces that constitute the interfaces between the sub-domains extracted from the given boundary mesh. This approach leads to meshing each sub-domain separately after the definition of the various domain interfaces and after a mesh of these interfaces has been constructed.

Provided with a fine mesh of the domain under interest, the posteriori partitioning method consists of splitting this mesh into several sub-meshes in order to distribute the computational effort over several processors, each of them being responsible for the solution of a physical problem for one sub-domain. The global solution is achieved by merging all of the partial solutions. The most frequent case is element-based decomposition in which the fine mesh is partitioned by distributing the cells among the sub-domains i.e. one cell is logically associated with one and only one sub-domain. Another case is node-based decomposition in which the mesh is partitioned by distributing its nodes among the sub-domains, i.e. one node is logically associated with one and only one sub-domain. The main drawback of such a method is related to its memory requirement, as it is necessary to store the initial mesh and, at least, one of the sub-meshes. Nevertheless, the posteriori methods are widely used.

Of course, in practice, these parallelization approaches are often combined by taking into account their advantages and weaknesses.

These and different partition methods are presented in greater detail in the books of Frey and George (2008) and Lo (2015).

References

Azarenok, B. N. (2002). Variational barrier method of adaptive grid generation in hyperbolic problems of gas dynamics. *SIAM Journal on Numerical Analysis*, *40*(40), 651–682.

Charakch'yan, A. A. (1993). Almost conservative difference schems for the equations of gas dynamics. *Computational Mathematics and Mathematical Physics*, *33*, 1473.

Charakch'yan, A. A. (1994). Compound difference schems for time-dependent equations on non-uniform nets. *Communications in Numerical Methods in Engineering*, *10*, 93.

Charakch'yan, A. A., & Ivanenko, S. A. (1988). A variational form of the Winslow grid generator. *Journal of Computational Physics*, *136*, 385–398.

Charakch'yan, A. A., & Ivanenko, S. A. (1997). A variational form of the Winslow grid generator. *Journal of Computational Physics*, *136*, 385–398.

Danaev, N. T., Liseikin, V. D., & Yanenko, N. N. (1980). Numerical solution on a moving curvilinear grid of viscous heat-conducting flow about a body of revolution. *Chisl. Metody Mekhan. Sploshnoi Sredy*, *11*(1), 51–61. (Russian).

Fletcher, C. A. J. (1997). *Computational Techniques for Fluid Dynamics 1: Fundamental and General Techniques*. Berlin: Springer.

Frey, P. J., & George, P.-L. (2008). *Mesh Generation Application to Finite Elements*. Verlag: ISTE Ltd and Wiley Inc.

Glasser, A. H., Liseikin, V. D., & Kitaeva, I. A. (2005). Control of grid properties with the help of monitor metrics. *Computational Mathematics and Mathematical Physics*, *45*(8), 1416–1432.

Glasser, A. H., Liseikin, V. D., Shokin, Ju. I., Vaseva, I. A., & Likhanova, Ju. V. (2006). *Grid Generation with the Use of Beltrami and Diffusion Equations*. Nauka, Novosibirsk. (Russian).

Glasser, A. H., & Tang, X. Z. (2004). The SEL macroscopic modeling code. *Computer Physics Communications*, *164*, 237–243.

Ivanenko, S. A. (1988). Generation of non-degenerate meshes. *USSR Computational Mathematics and Mathematical Physics*, *28*(5), 141.

Kovenya, V. M., Tarnavskii, G. A., & Chernyi, S. G. (1990). *Application of a Splitting Method to Fluid Problems*. Novosibirsk: Nauka. (in Russian).

Langtangen,. (2003). *Computational partial differential equations, numerical methods and diffpack programming*. Berlin: Springer.

Liseikin, V. D. (2014a). *Numerical Grids*. SD RAS, Novosibirsk (in Russia): Theory and Applications.

Liseikin, V. D. (2014b). *Methods for Generating Numerical Grids*. Novosibirsk: NSU. (in Russia).

Lo, S. H. (2015). *Finite Element Mesh Generation*. Boca Raton: CRC Press Taylor and Francis Group.

Lomonosov, I. V., Frolova, A. A., & Charakhch'yan, A. A. (1997). Computation of hight-velocity impact of thin foil upon conical target (survey). *A Mathematical Modeling*, *9*(5), 48–60. (Russian).

Vaseva, I. A., & Liseikin, V. D. (2011). Application of the finite element method for generating adaptive grids. *Computational Technologies*, *16*(5), 3–15. (in Russia).

Winslow, A. M. (1981): Adaptive mesh zoning by the equipotential method. UCID-19062, Lawrence Livermore National Laboratories.

Yanenko, N. N. (1971). *The method of fractional steps. The solution of problems of mathematical physics in several variables*. New York: Springer.

Chapter 11
Control of Grid Properties

This chapter is devoted to practical applications of control metrics g^{s}_{ij} in inverted Beltrami and diffusion equations for finding intermediate transformations

$$\boldsymbol{s}(\boldsymbol{\xi}) : \Xi^n \to S^n, \quad \boldsymbol{s}(\boldsymbol{\xi}) = [s^1(\boldsymbol{\xi}), \dots, s^n(\boldsymbol{\xi})], \quad \boldsymbol{\xi} = (\xi^1, \dots \xi^n), \quad \boldsymbol{\xi} \in \Xi^n \tag{11.1}$$

between computational Ξ^n and parametric S^n domains. The control metrics provide necessary grid properties on arbitrary physical geometries $S^{xn} \subset \mathbb{R}^{n+k}$ specified by parametrizations

$$\boldsymbol{x}(\boldsymbol{s}) : S^n \to S^{xn}, \ \boldsymbol{x}(\boldsymbol{s}) = (x^1(\boldsymbol{s}), \dots, x^{n+k}(\boldsymbol{s})), \ \boldsymbol{s} = (s^1, \dots, s^n) \ , \tag{11.2}$$

in particular, in n-dimensional domains, i.e. when $S^{xn} \equiv X^n$, $\boldsymbol{s} = \boldsymbol{x}$.

11.1 Grid Adaptation to Function Values

11.1.1 Control Operator

Given a function $\boldsymbol{u}(\mathbf{s}) : S^n \to \mathbb{R}^m$ and the grid transformation as a composition of (11.1) and (11.2)

$$\boldsymbol{x}(\boldsymbol{s}(\boldsymbol{\xi})) : \Xi^n \to S^{xn} \ ,$$

consider the quantity

$$m(\mathbf{s}) = Z[\boldsymbol{u}](\mathbf{s})\sqrt{g^{xs}}\, g^{km}_{sx} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^i}{\partial s^m} = Z[\boldsymbol{u}](\mathbf{s})\sqrt{g^{xs}} \sum_{i=1}^{n} g^{ii}_{\xi x}, \quad i, j, k, m = 1, \dots, n \ , \tag{11.3}$$

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_11

where g^{km}_{sx} $(g^{km}_{\xi x})$ is the (km)th element of the contravariant metric tensor of S^{xn} in coordinates $s^1, \dots, s^n$ $(\xi^1, \dots, \xi^n)$,

$$g^{xs} = \det\{g^{xs}_{ij}\} \ , \quad g^{xs}_{ij} = \boldsymbol{x}_{s^i} \cdot \boldsymbol{x}_{s^j} \ , \quad g^{ij}_{\xi x} = g^{km}_{sx} \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^m} \ , \quad i, j, k, m = 1, \dots, n \ ,$$

$Z[\ ]$ is a positive operator whose values are positive scalar functions. We refer to $Z[\ ]$ as a control operator, while referring to $Z[\boldsymbol{u}](\mathbf{s})$ as a control function.

If $S^{xn} \equiv S^n$, then $g^{km}_{sx} = g^{xs}_{km} = \delta^k_m$, so formula (11.3) is presented as

$$m(\mathbf{s}) = Z[\boldsymbol{u}](\mathbf{s}) \sum_{i=1}^{n} g^{ii}_{\xi s}, \quad i = 1, \dots, n, \tag{11.4}$$

where

$$g^{ij}_{\xi s} = \delta^k_m \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^m} = \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^k}, \quad i, j, k, m = 1, \dots, n \ .$$

To elucidate the effect of the operator $Z[\ ]$ on grid adaptation to the values of a function $\boldsymbol{u}(\boldsymbol{s})$, notice that the distance l_k between a point at the coordinate surface (line when $n = 2$) $\xi^k = c$ and the coordinate surface $\xi^k = c + h$ on S^{xn} equals, up to $O(h^2)$, the product of h and the length of the projection of the tangent vector

$$\boldsymbol{x}_{\xi^k} = \frac{\partial}{\partial \xi^k} \boldsymbol{x}(\boldsymbol{s}(\boldsymbol{\xi})), \quad k = 1, \dots, n \ ,$$

onto the unit normal $\boldsymbol{n}_k$ at this point to the coordinate surface $\xi^k = c$ (see Fig. 11.1 for $n = 2$), namely,

$$l_k = h|\boldsymbol{x}_{\xi^k} \cdot \boldsymbol{n}_k| + O(h^2) \ ,$$

where, in accordance with (9.76),

$$\boldsymbol{n}_k = g^{ik}_{\xi x} \boldsymbol{x}_{\xi^i} / \sqrt{g^{kk}_{\xi x}}, \quad i = 1, \dots, n, \quad k \text{ fixed} \ . \tag{11.5}$$

If $S^{xn} \equiv S^n$, then, for $\boldsymbol{n}_k$ in (11.5), we have

$$\boldsymbol{n}_k = \operatorname{grad} \xi^k(\boldsymbol{x}) / \sqrt{g^{kk}_{\xi x}} = \left(\frac{\partial \xi^k}{\partial x^1}, \dots, \frac{\partial \xi^k}{\partial x^n} \right) / \sqrt{g^{kk}_{\xi x}}, \quad k \text{ fixed} \ .$$

Using (11.5) gives for every fixed $k = 1, \dots, n$

$$\boldsymbol{x}_{\xi^k} \cdot \boldsymbol{n}_k = g^{ik}_{\xi x} g^{x\xi}_{ki} / \sqrt{g^{kk}_{\xi x}} = 1 / \sqrt{g^{kk}_{\xi x}}, \quad k \text{ fixed} \ , \tag{11.6}$$

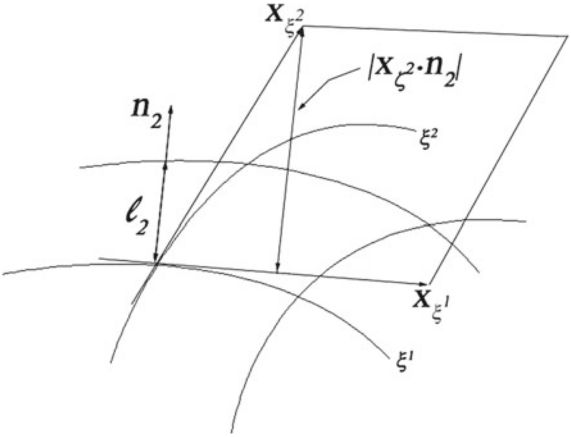


Fig. 11.1 Illustration for defining a grid step

and hence $l_k \approx h/\sqrt{g^{kk}_{\xi x}}$. Therefore, $1/g^{kk}_{\xi x}$ with fixed k is a measure of mesh refinement on S^{xn} in the direction orthogonal to the coordinate surface $\xi^k = const$ on S^{xn} (see Fig. 11.1 for $n = 2$).

Consider the minimization of the following functional:

$$I_D[\boldsymbol{\xi}] = \frac{1}{2}\int\limits_{S^n} m(\boldsymbol{s})\mathrm{d}\boldsymbol{s} = \frac{1}{2}\int\limits_{S^n} \sqrt{g^{xs}}Z[\boldsymbol{u}](\boldsymbol{s})\sum_{i=1}^{n} g^{ii}_{\xi x}\mathrm{d}\boldsymbol{s}; \tag{11.7}$$

with the function $m(\boldsymbol{s})$ from (11.3) for the transformations $\boldsymbol{\xi}(\boldsymbol{s}) : S^n \to \Xi^n$, the expression $\sum\limits_{i=1}^{n} g^{ii}_{\xi x}$ tends to become less at the points where the control function $Z[\boldsymbol{u}](\boldsymbol{s})$ takes larger values, and vice versa. Therefore, for the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$ whose inverse $\boldsymbol{\xi}(\boldsymbol{s})$ is extremal for functional (11.7) to ensure grid clustering in required zones (for example, in the zone of flow turbulence), the values of the control function $Z[\boldsymbol{u}](\boldsymbol{s})$ at the points of these zones must be smaller than at the other points of S^{xn}. Conversely, for mesh rarefaction in required zones of S^{xn}, the values of the control function $Z[\boldsymbol{u}](\boldsymbol{s})$ at the points of these zones must be larger than at the other points of S^{xn}. For example, to ensure grid clustering in regions where the values of $|\boldsymbol{u}(\mathbf{s})|$ are large, we may assume

$$Z[\boldsymbol{u}](\boldsymbol{s}) = \frac{1}{(|\boldsymbol{u}(\boldsymbol{s})| + c_1)^{\alpha}}, \quad c_1 > 0, \quad \alpha > 0\,. \tag{11.8}$$

For grid clustering in regions where the values of $|\boldsymbol{u}(\mathbf{s})|$ are small, for example, within small-depth zones in tsunami problems, we may define $Z[\boldsymbol{u}](\boldsymbol{s})$ as

$$Z[\boldsymbol{u}](\boldsymbol{s}) = (|\boldsymbol{u}(\boldsymbol{s})| + c_1)^{\alpha}, \quad c_1 > 0\,, \quad \alpha > 0\,. \tag{11.9}$$

11.1.2 Grid Equations

Functional (11.7) is, in fact, the diffusion functional (9.121)

$$I_D[\boldsymbol{\xi}] = \frac{1}{2}\int\limits_{S^n} w(\boldsymbol{s}) g_{\mathbf{s}}^{mj} \frac{\partial \xi^i}{\partial s^m}\frac{\partial \xi^i}{\partial s^j} \mathrm{d}\boldsymbol{s} = \frac{1}{2}\int\limits_{S^n} w(\boldsymbol{s}) \sum_{i=1}^{n} g_{\boldsymbol{\xi}}^{ii} \mathrm{d}\boldsymbol{s}, \quad i, j, m = 1, \dots, n \tag{11.10}$$

for $w(\boldsymbol{s}) = 1$ in a control metric whose contravariant elements in coordinates $s^1, \dots, s^n$ are specified as

$$g_{\mathbf{s}}^{ij} = \sqrt{g^{xs}} Z[\boldsymbol{u}](\boldsymbol{s}) g_{sx}^{ij}, \quad i, j = 1, \dots, n\ . \tag{11.11}$$

Consequently, the covariant elements of this metric in coordinates $s^1, \dots, s^n$ are defined as

$$g_{ij}^{\mathbf{s}} = \frac{1}{\sqrt{g^{xs}} Z[\boldsymbol{u}](\boldsymbol{s})} g_{ij}^{xs}, \quad i, j = 1, \dots, n\ . \tag{11.12}$$

Euler–Lagrange equations for functional (11.10) with $w(\boldsymbol{s}) = 1$ in this control metric are diffusion equations

$$\frac{\partial}{\partial s^j}\left(\sqrt{g^{xs}} Z[\boldsymbol{u}](\boldsymbol{s}) g_{sx}^{jk} \frac{\partial \xi^i}{\partial s^k}\right) = 0, \quad i, j, k = 1, \dots, n\ . \tag{11.13}$$

Therefore, the inverted diffusion equations in metric (11.12) for generating grids on S^{xn} will be the following equations:

$$B_n^{x\xi}[s^k] = \frac{J^2\sqrt{g^{xs}}}{Z[\boldsymbol{u}](\boldsymbol{s})} \frac{\partial}{\partial s^j}(\sqrt{g^{xs}} Z[\boldsymbol{u}](\boldsymbol{s}) g_{xs}^{jk}), \quad j, k = 1, \dots, n\ , \tag{11.14}$$

where

$$B_n^{x\xi}[y] = J^2 g^{xs} g_{\xi x}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j}, \quad g_{\xi x}^{ij} = g_{sx}^{kl} \frac{\partial \xi^i}{\partial s^k}\frac{\partial \xi^j}{\partial s^l}, \quad i, j, k, l = 1, \dots, n\ ,$$

while if $S^{xn} \equiv S^n$, they will be

$$B_n^{s\xi}[s^k] = J^2 \frac{1}{Z[\boldsymbol{u}](\boldsymbol{s})} \frac{\partial}{\partial s^k}(Z[\boldsymbol{u}](\boldsymbol{s})), \quad k = 1, \dots, n\ , \tag{11.15}$$

where

$$B_n^{s\xi}[y] = J^2 g_{\xi s}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j}, \quad g_{\xi s}^{ij} = \frac{\partial \xi^i}{\partial s^k}\frac{\partial \xi^j}{\partial s^k}, \quad i, j, k = 1, \dots, n.$$

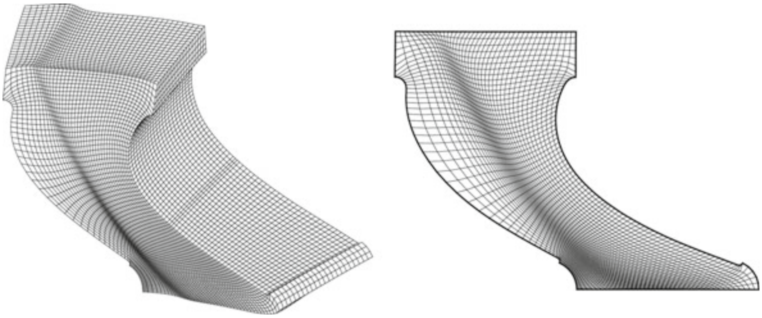


Fig. 11.2 Fragment of a three-dimensional grid between turbine blades with refinement in zones of large values of a function (*left*); a two-dimensional cross section of the grid (*right*)

Figure 11.2 illustrates an adaptive grid in a domain S^3, with node clustering in the zone of large values of a function $\boldsymbol{u}(\boldsymbol{s})$, obtained by the solution of Eq. (11.15) with $Z[\boldsymbol{u}](\boldsymbol{s})$ defined by formula (11.8).

11.2 Grid Generation with Node Clustering Near Isolated Points

Let $\boldsymbol{x}(\boldsymbol{s}) : S^n \to S^{xn}$ be a parametrization of S^{xn}. For generating grids with node clustering in the vicinity of a point $\boldsymbol{x}(\boldsymbol{s}_0) \in S^{xn}$ by solving Eqs. (11.14) and (11.15), we can use control metric (11.12) with a scalar-valued function $u(\boldsymbol{s})$ as a composition of

$$\varphi(\boldsymbol{s}) = \rho(\boldsymbol{s}, \boldsymbol{s}_0) = |\boldsymbol{s} - \boldsymbol{s}_0| \tag{11.16}$$

and a positive univariate scalar function $\sigma(t)$ that takes a small value at $t = 0$ and grows rapidly in the neighborhood of $t = 0$, i.e. $u(\boldsymbol{s}) = \sigma(\varphi(\boldsymbol{s}))$, and then set $Z[u](\boldsymbol{s}) = u(\boldsymbol{s})$. Moreover, the mesh refinement is more localized for a faster growing function $\sigma(t)$. Examples of such univariate functions $\sigma(t)$ are layer-type functions for $|t| < T_0$, as described in Chap. 4:

$$\begin{aligned}
&\sigma_1(t) = |\mathrm{th}(t/\varepsilon)|^{\alpha} + \varepsilon_1, \quad \alpha > 0, \quad \sigma_2(t) = 1 - e^{-|t|^{\alpha}/\varepsilon} + \varepsilon_1, \quad \alpha > 0,\\
&\sigma_3(t) = 1 - \frac{\varepsilon^{\beta}}{(\varepsilon + |t|^{\alpha})^{\beta}} + \varepsilon_1, \quad \alpha, \beta > 0,\\
&\sigma_4(t) = \frac{(\varepsilon + |t|^{\alpha})^{\beta}}{(\varepsilon + (T_0)^{\alpha})^{\beta}}, \quad 0 < \beta < 1, \quad \alpha > 0,\\
&\sigma_5(t) = |t|^{\alpha} + \varepsilon_1, \quad \alpha > 0, \quad \sigma_6(t) = \frac{\ln(1 + |t|^{\alpha}\varepsilon^{-k})}{\ln(1 + (T_0)^{\alpha}\varepsilon^{-k})} + \varepsilon_1, \quad \alpha, k > 0,
\end{aligned} \tag{11.17}$$

where ε and ε_1 are small positive constants, i.e. $0 < \varepsilon, \varepsilon_1 \ll 1$ (see Fig. 11.3).

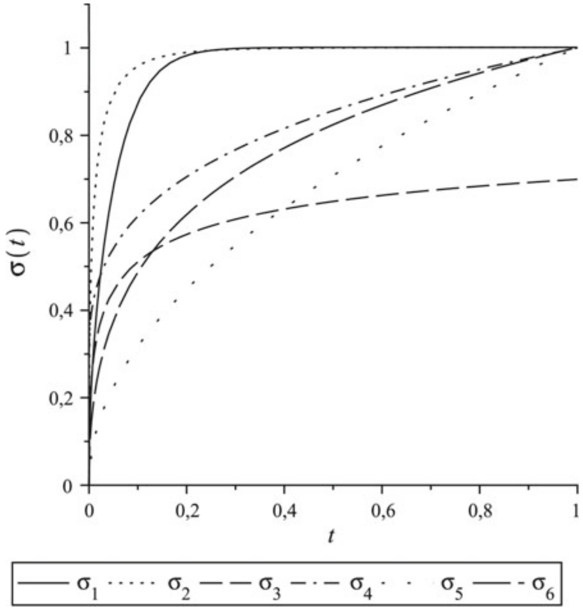


Fig. 11.3 Plots of layer-type functions (11.17) for $1 \geq t \geq 0$, $\alpha = 0,5$, $\varepsilon = 0,1$, $\varepsilon_1 = 0,001$ $\beta = 0,5$

In accordance with the theoretical results, presented in Liseikin (2001a), the functions $\sigma_1(t)$, $\sigma_2(t)$, and $\sigma_3(t)$ are efficient for grid clustering in the zones of exponential boundary and interior layers. The functions $\sigma_3(t)$ and $\sigma_4(t)$ are suitable for providing grid clustering in the zones of power layers, while the function $\sigma_6(t)$ is useful for grid clustering in the zones of logarithmic layers. Moreover, the faster the growth of $\sigma_i(t)$ near $t = 0$, the smaller the zone of grid clustering. The fastest growth has the functions $\sigma_1(t)$ and $\sigma_2(t)$ (see Fig. 11.3).

Since the functions $\sigma_i(t)$, $i = 1, \ldots, 6$ are positive, mesh refinement near the point $\boldsymbol{x}(\boldsymbol{s}_0)$ can be achieved by setting the control function $Z[u](\boldsymbol{s})$ in the inverted diffusion equations (11.14) and (11.15) as a linear combination of functions $\sigma_i(\varphi(\boldsymbol{s}))$, i.e.

$$Z[u](\boldsymbol{s}) = \sum_{i=1}^{6} c_i \sigma_i[\varphi(\boldsymbol{s})], \quad c_i \geq 0\,. \tag{11.18}$$

For mesh rarefaction near the point $\boldsymbol{x}(\boldsymbol{s}_0)$, the control function $Z[u](\boldsymbol{s})$ can be taken as a linear combination of functions $1/\sigma_i(\varphi(\boldsymbol{s}))$, i.e.

$$Z[u](\boldsymbol{s}) = \sum_{i=1}^{6} \frac{c_i}{\sigma_i[\varphi(\boldsymbol{s})]}, \quad c_i \geq 0\,, \tag{11.19}$$

or as a product of some of these functions.

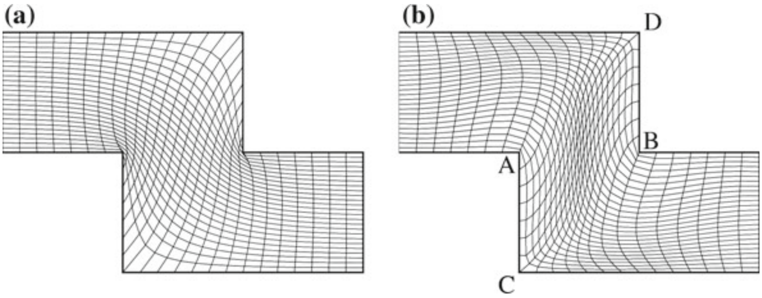


Fig. 11.4 Grid obtained by using the inverted Laplace equations (**a**) and grid obtained by solving the inverted diffusion equations in control metric (11.14) with node clustering near the points C and D and with rarefaction near the points A and B (**b**)

If $\boldsymbol{x}(\boldsymbol{s}_0)$ is a boundary point of S^{xn}, then the use of control function (11.18) in the numerical solution of inverted diffusion equations (11.14) or (11.15) leads to attraction of grid cells to some part of the boundary segment containing $\boldsymbol{x}(\boldsymbol{s}_0)$. Due to the application of control function (11.19), the grid nodes are repelled from a part of the boundary section of S^{xn} containing the point $\boldsymbol{x}(\boldsymbol{s}_0)$. Therefore, the control metric (11.12) can be used to eliminate undesirable refinements or rarefactions obtained by numerically solving the inverted Beltrami equations with $Z[u](\boldsymbol{s}) \equiv 1$. When $S^{xn} \equiv S^n$, these equations are, in fact, the inverted Laplace equations, and the corresponding grid rarefies near convex segments of the boundary coordinate surface (line at $n = 2$) and refines near concave boundary segments (see Fig. 11.4a).

Since the values of functions (11.17) are significantly larger then zero outside a small neighborhood of the point $t = 0$, the control function $Z[u](\boldsymbol{s})$ in (11.11) providing grid clustering near several isolated points and grid rarefaction at other isolated points can be defined as the product of the corresponding functions (11.18) and (11.19). Figure 11.4b demonstrates the application of such a control function for reducing grid rarefaction near the points C and D, where the boundary is convex, and for reducing grid clustering near the points A and B, where the boundary is concave, in control metric (11.12) with

$$Z[u](\boldsymbol{s}) = u(\boldsymbol{s}), \quad u(\boldsymbol{s}) = \frac{\sigma_5'(\rho(\boldsymbol{s}, C))\sigma_5'(\rho(\boldsymbol{s}, D))}{\sigma_5''(\rho(\boldsymbol{s}, A))\sigma_5''(\rho(\boldsymbol{s}, B))},$$

where σ_5' and σ_5'' are functions from (11.17), namely,

$$\sigma_5'(\rho(\boldsymbol{s}, \boldsymbol{v})) = 0,0002 + \rho^3(\boldsymbol{s}, \boldsymbol{v}), \quad \sigma_5''(\rho(\boldsymbol{s}, \boldsymbol{v})) = 0,02 + \rho^2(\boldsymbol{s}, \boldsymbol{v}).$$

Figure 11.5 exhibits the application of control metric (11.12) for the elimination of an overlap at a corner point. Specifically, for the corner point $\boldsymbol{s}_1$, the repelling control function

$$u_1(\boldsymbol{s}) = \frac{1}{0,02 + |\boldsymbol{s} - \boldsymbol{s}_1|^2}$$

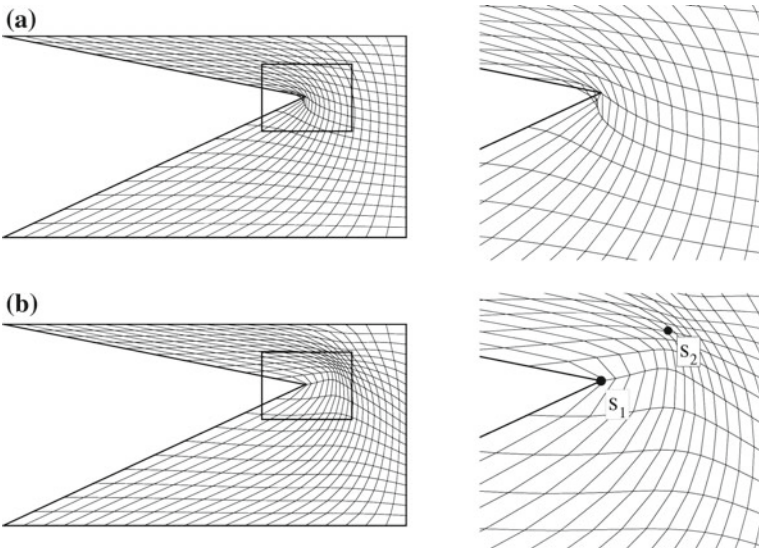


Fig. 11.5 Application of repelling and refining control functions for the elimination of overlaps near acute angles: the grids generated with the help of the inverted Laplace equations (**a**) and inverted diffusion equations (11.15) in metric (11.12) (**b**)

was taken. While for the point $\boldsymbol{s}_2$, we used the attracting control function

$$u_2(\boldsymbol{s}) = 0,002 + |\boldsymbol{s} - \boldsymbol{s}_2|^2 ,$$

and as a result, the control function $Z[u](\boldsymbol{s})$ is as follows:

$$Z[u](\boldsymbol{s}) = u_1(\boldsymbol{s})u_2(\boldsymbol{s}) = \frac{0,002 + |\boldsymbol{s} - \boldsymbol{s}_2|^2}{0,02 + |\boldsymbol{s} - \boldsymbol{s}_1|^2} .$$

To generate a grid that is refined or rarefined near several points $\boldsymbol{s}_i \in S^n$, $i = 1, \dots, N$, the function $\varphi(\boldsymbol{s})$ can be defined as

$$\varphi(\boldsymbol{s}) = \min_{i=1,\dots,N} \rho(\boldsymbol{s}, \boldsymbol{s}_i). \tag{11.20}$$

Specifically, this function can be used to generate grids with refinement on a discrete set specified in S^n. Figure 11.6 demonstrates the application of this function for grid generation with refinement near the boundaries of an aorta and carotid arteries that were discretely specified.

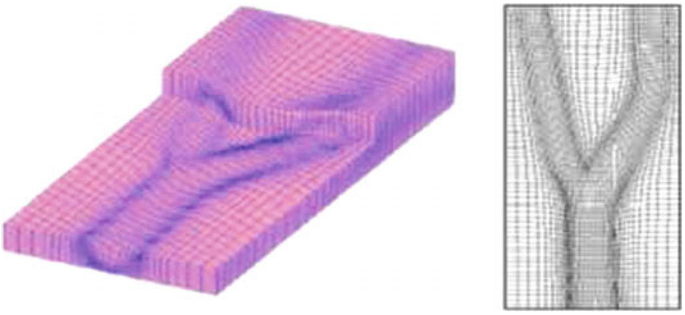


Fig. 11.6 Grid with node clustering near isolated boundary points

11.3 Grids with Node Clustering Near Curves and Surfaces

With the help of control metric (11.12) in inverted diffusion equations (11.14) and (11.15), one can provide grid clustering in the neighborhood of a surface on S^{xn} (curve on S^{x2}) given by an equation $\varphi(\boldsymbol{s}) = 0$. In this case, the control function $Z[u](\boldsymbol{s})$ in Eqs. (11.14) and (11.15) can be defined through a positive univariate function $\sigma(t)$ such that $\sigma(0) < \sigma(t)$, $|t| > 0$ and $\sigma(t)$ varies rapidly in a small vicinity of zero, namely, assuming $Z[u](\boldsymbol{s}) = \sigma(\varphi(\boldsymbol{s}))$. Similarly, the control function $Z[u](\boldsymbol{s}) = 1/\sigma(\varphi(\boldsymbol{s}))$ provides grid rarefaction in the vicinity of the surface $\varphi(\boldsymbol{s}) = 0$. Specifically, any function from (11.17) can be used as $\sigma(t)$. Figure 11.7 demonstrates varied degrees of refinement of grid nodes near the curve defined as

$$\varphi(\boldsymbol{s}) \equiv s^2 - 20(s^1 - 0,5)^3 - 0,5 = 0,$$

depending on the choice of a function from (11.17).

Examples of three-dimensional adaptive grids with node clustering near a surface $\varphi(\boldsymbol{s}) = 0$ for

$$Z[u](\boldsymbol{s}) = \sigma[\varphi(\boldsymbol{s})], \quad \sigma(t) = (t/10)^2 + 0,01, \quad \varphi(\boldsymbol{s}) = s^1 - 3\sin(\pi s^2/10) - 140,$$

and for

$$Z[u](\boldsymbol{s}) = \sigma[\varphi(\boldsymbol{s})], \quad \sigma(t) = 1 - e^{-|t|^{1.1}} + 10^{-5},$$

$$\varphi(\boldsymbol{s}) = (s^1)^3 + (s^2)^3 + (s^3)^3 + 1 - (s^1 + s^2 + s^3 + 1)^3$$

are given in Figs. 11.8 and 11.9, respectively.

If it is necessary to adapt a grid near surfaces (curves for $n = 2$) $\varphi_1(\boldsymbol{s}) = 0$ and $\varphi_2(\boldsymbol{s}) = 0$, then the control function $Z[u](\boldsymbol{s})$ is defined as the product of the corresponding functions $Z_1[u_1](\boldsymbol{s})$ and $Z_2[u_2](\boldsymbol{s})$. Figure 11.10b exhibits a two-dimensional grid with grid clustering near a convex boundary segment and with

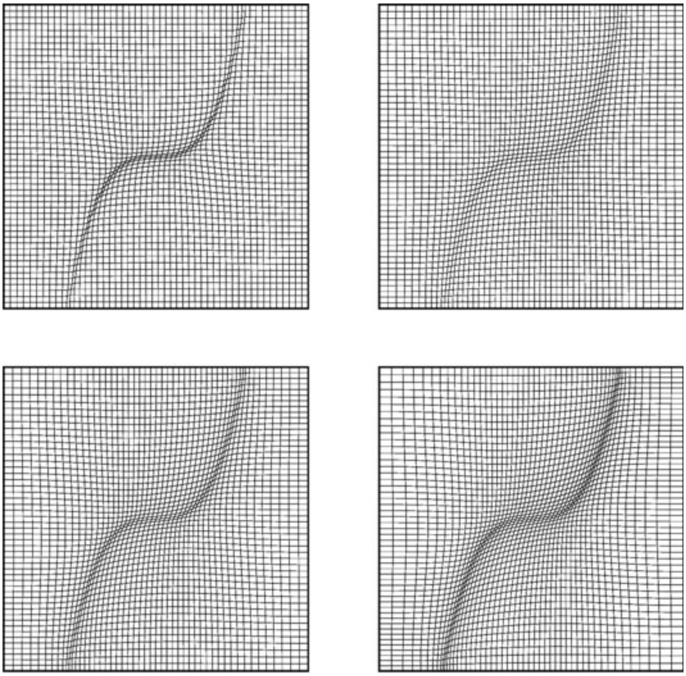


Fig. 11.7 Application of the layer-type functions $\sigma_1,\ \sigma_4,\ \sigma_5,\ \sigma_6$ defined by formula (11.17) for mesh refinement near a given curve for $\alpha = 0, 5$, $\varepsilon = 0, 1$, $\varepsilon_1 = 0, 001$ and $\beta = 0, 5$

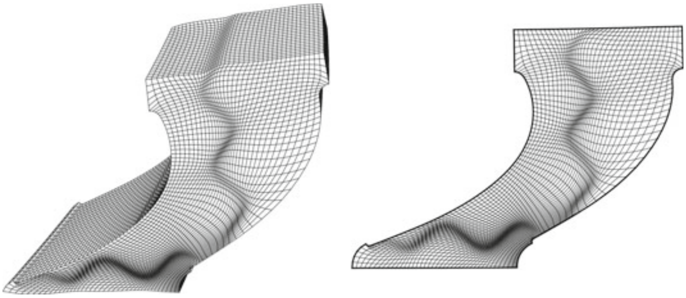


Fig. 11.8 Fragment of a three-dimensional grid between turbine blades with refinement near a given surface (*left*); a two-dimensional cross section of the grid (*right*)

node rarefaction near a concave boundary segment by applying the following control function:

$$Z[u](\boldsymbol{s}) = Z_1[u_1](\boldsymbol{s}) Z_2[u_2](\boldsymbol{s}) \ ,$$

$$Z_1[u_1](\boldsymbol{s}) = u_1(\boldsymbol{s}) = |\mathrm{th}(\rho_{min}^{\mathrm{convex}})| + 0, 1 \ ,$$

$$Z_2[u_2](\boldsymbol{s}) = \frac{1}{u_2(\boldsymbol{s})} = \frac{1}{|\mathrm{th}(\rho_{min}^{\mathrm{concave}})| + 0, 1} \ ,$$

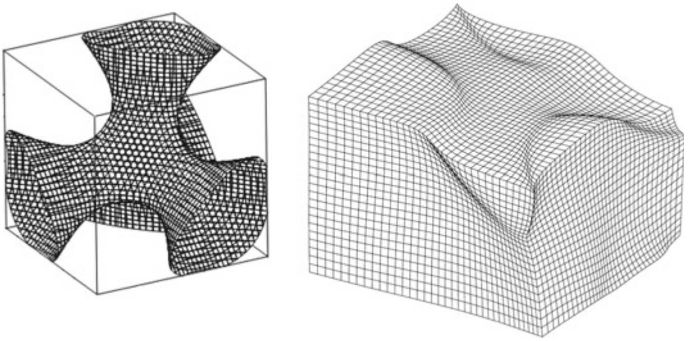


Fig. 11.9 Two-dimensional surface in R^3 (*left*); fragment of a three-dimensional grid with node clustering near this surface (*right*)

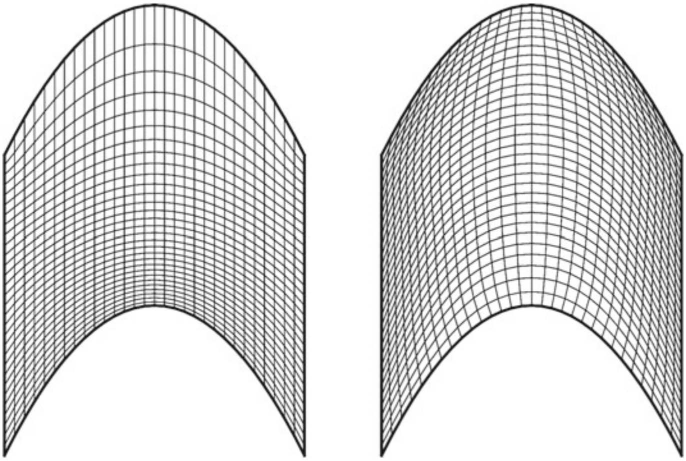


Fig. 11.10 Grid with cell rarefaction near a convex boundary segment and cell clustering near a concave boundary segment obtained by solving the inverted Laplace equations (*left*); grid obtained by solving inverted diffusion equations (11.15) with cell clustering near the convex boundary segment and cell rarefaction near the concave boundary segment (*right*)

$$\rho_{min}^{\text{convex}} = \min_i \rho(\boldsymbol{s}, array_i^{\text{convex}}), \quad \rho_{min}^{\text{concave}} = \min_j \rho(\boldsymbol{s}, array_j^{\text{concave}}),$$

where $array_i^{\text{convex}}$ and $array_j^{\text{concave}}$ are discretizations of convex and concave boundary segments, respectively. Notice that the grid obtained by solving the inverted Beltrami equations with $Z[u](\boldsymbol{s}) = 1$, in fact, by the inverted Laplace equations, rarefies near convex boundary segments and refines near concave boundary segments (Fig. 11.10a).

If a curve in a three–dimensional domain S^3 is defined by two equations $\varphi_1(\boldsymbol{s}) = 0$ and $\varphi_2(\boldsymbol{s}) = 0$, then for generating a grid with node clustering near such a curve, one may assume in inverted diffusion equations (11.14) and (11.15)

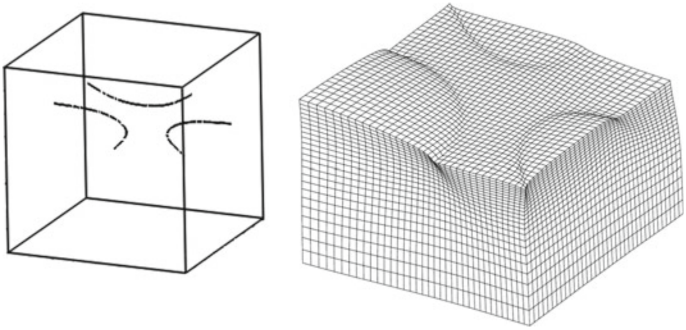


Fig. 11.11 Curves in S^3 (*left*); fragment of a three-dimensional grid with node clustering near the curves (*right*)

$$Z[u](\boldsymbol{s}) = u(\boldsymbol{s}),\ u(\boldsymbol{s}) = \alpha\sigma_i(\varphi_1(\boldsymbol{s})) + \beta\sigma_j(\varphi_2(\boldsymbol{s})),\ \alpha, \beta > 0,\ \alpha + \beta = 1,$$

where $\sigma_i(t)$ are functions from by formula (11.17). An example of such a grid for

$$Z[u](\boldsymbol{s}) = u(\boldsymbol{s}) = 0,7\sigma_2(\varphi_1(\boldsymbol{s})) + 0,3\sigma_6(\varphi_2(\boldsymbol{s})),$$
$$\varphi_1(\boldsymbol{s}) = (s^1)^3 + (s^2)^3 + (s^3)^3 + 1 - (s^1 + s^2 + s^3 + 1)^3, \quad \varphi_2(\boldsymbol{s}) = s^2 + 0,5,$$
$$\sigma_2(t) = 1 - e^{-|t|^{1,1}} + 10^{-5}, \quad \sigma_6(t) = 10^{-6} + |t|^2$$

is exhibited in Fig. 11.11.

If a curve on S^{xn} is specified by a parametrization $\boldsymbol{x}_2(t) = \boldsymbol{x}(\boldsymbol{x}_1(t))$, where $\boldsymbol{x}_1(t) : [0, 1] \to S^n$ is a transformation of the interval [0, 1], while $\boldsymbol{x}(\boldsymbol{s}) : S^n \to S^{xn}$ is a parametrization of S^{xn}, then for generating a grid with node clustering in the vicinity of this curve, one may generate a grid in the interval [0, 1] with nodes $t_0 = 0, t_1, \ldots, t_{N-1}, t_N = 1$, and then use formula (11.20) for the points $\boldsymbol{s}_i = \boldsymbol{x}_1(t_i)$, $i = 0, 1, \ldots, N-1, N$, and assume

$$Z[u](\boldsymbol{s}) = u(\boldsymbol{s}), \quad u(\boldsymbol{s}) = \sigma_i(\varphi(\boldsymbol{s})), \quad i = 1, \ldots, 6,$$

where $\sigma_i(t)$ are functions in formula (11.17).

Similarly, one can generate a grid with node clustering in the vicinity of a surface in S^{xn} specified by a parametrization

$$\boldsymbol{x}(\boldsymbol{x}_1(t_1, t_2)) : [0, 1] \times [0, 1] \to S^2 \to S^{x3}\ .$$

11.4 Generation of Grids with Node Clustering in the Zones of Large Variations of Functions

11.4.1 Control Metric of a Monitor Surface

Efficient control metric for generating grids on S^{xn} specified by parametrization (11.2) with node clustering in the zones of large variations of a function $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^l(\boldsymbol{x}))$ is the metric of a monitor hypersurface S^{rn} over S^{xn} that is defined by parametrization

$$\boldsymbol{r}(\boldsymbol{s}) : S^n \to \mathbb{R}^{n+k+l}, \qquad \boldsymbol{r}(\boldsymbol{s}) = (\boldsymbol{x}(\boldsymbol{s}), \boldsymbol{f}[\boldsymbol{x}(\boldsymbol{s})]), \tag{11.21}$$

for which

$$g^{\mathbf{s}}_{ij} = g^{rs}_{ij} = \boldsymbol{r}_{s^i} \cdot \boldsymbol{r}_{s^j} = g^{xs}_{ij} + \frac{\partial \boldsymbol{f}[\boldsymbol{x}(\boldsymbol{s})]}{\partial s^i} \cdot \frac{\partial \boldsymbol{f}[\boldsymbol{x}(\boldsymbol{s})]}{\partial s^j}, \quad i, j = 1, \ldots, n. \tag{11.22}$$

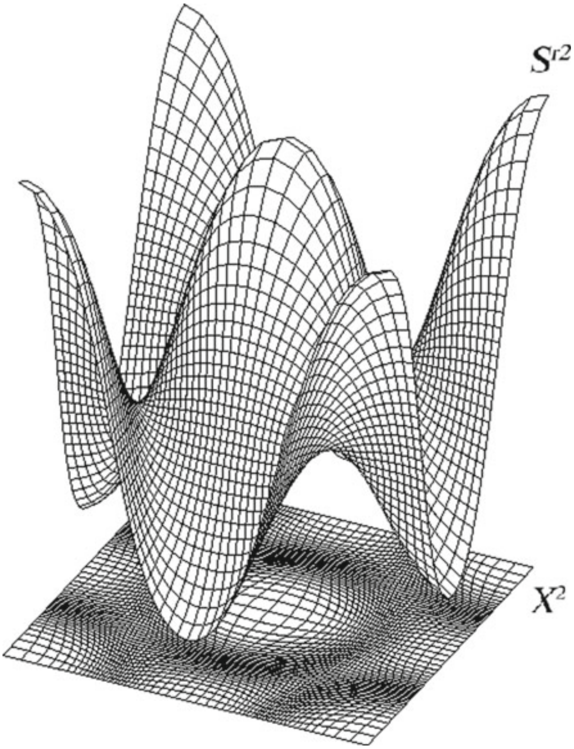


Fig. 11.12 Quasiuniform grid on a monitor surface S^{r2} and adaptive grid in a domain X^2

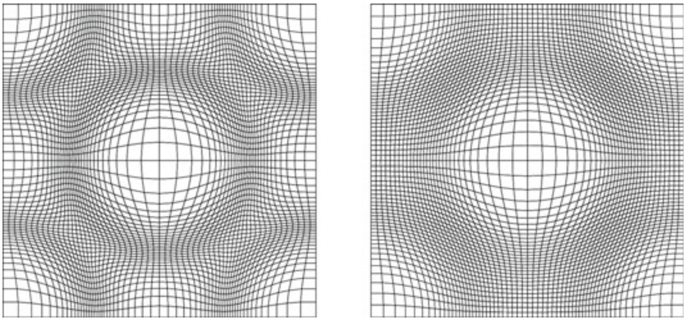


Fig. 11.13 Adaptive grid with node clustering in the zone of large variations of the function defined by (11.25) (*left*); adaptive grid with node clustering in the zone of large gradients of the vector function $\mathbf{f}(\mathbf{x}) = (f^1(\boldsymbol{x}), \dots, f^5(\boldsymbol{x}))$, where $f^i(\boldsymbol{x}),\ i = 1, \dots, 5$ is taken from (11.26) (*right*)

The inverted Beltrami equations in this metric are, in accordance with Chap. 10, presented as

$$B_n^{\boldsymbol{\xi}}[s^i] = -B_n^{\boldsymbol{\xi}}[\boldsymbol{f}] \cdot \frac{\partial \boldsymbol{f}(\boldsymbol{s})}{\partial s^i}, \quad i = 1, \dots, n\ , \tag{11.23}$$

where

$$B_n^{\boldsymbol{\xi}}[y] = g^{\boldsymbol{\xi}} g_{\boldsymbol{\xi}}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j}\ , \quad i, j = 1, \dots, n\ .$$

The quantity

$$m(\boldsymbol{s}) = \sqrt{g^{\mathbf{s}}} g_{\mathrm{s}}^{jm} \frac{\partial \xi^i}{\partial s^j} \frac{\partial \xi^i}{\partial s^m}, \qquad i, j, m = 1, \dots, n \tag{11.24}$$

for metric (11.22) is a local divergency measure from a uniform grid on S^{rn} and, consequently, from an adaptive grid on S^{xn} whose cells are clustering in the zones of large variations of function $\boldsymbol{f}(\boldsymbol{x}(\boldsymbol{s}))$ (see Figs. 11.12 and 11.13, *left-hand* for $n = 2$ and $S^{x2} = X^2$), where

$$f(x_1, x_2) = f_1(x_1, x_2) + f_2(x_1, x_2) + f_3(x_1, x_2) + f_4(x_1, x_2) + f_5(x_1, x_2), \tag{11.25}$$

where

$$\begin{aligned} f_1(x_1, x_2) &= \mathrm{th}(0, 04 - (x_1 - 0, 5)^2 - (x_2 - 0, 5)^2), \\ f_2(x_1, x_2) &= \mathrm{th}(0, 04 - x_1^2 - x_2^2), \\ f_3(x_1, x_2) &= \mathrm{th}(0, 04 - x_1^2 - (x_2 - 1)^2), \\ f_4(x_1, x_2) &= \mathrm{th}(0, 04 - (x_1 - 1)^2 - (x_2 - 1)^2), \\ f_5(x_1, x_2) &= \mathrm{th}(0, 04 - (x_1 - 1)^2 - x_2^2). \end{aligned} \tag{11.26}$$

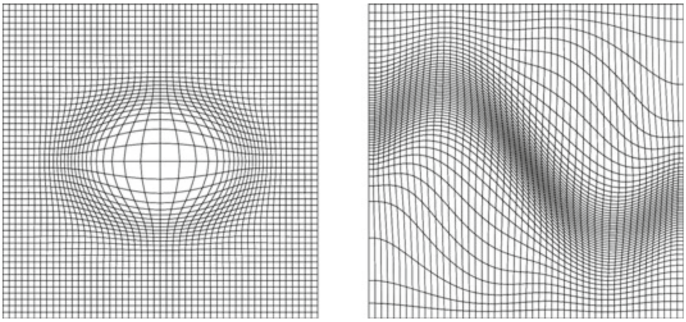


Fig. 11.14 Grids generated with the help of equations (11.23) with node clustering near a closed curve (*left*); near an unclosed curve (*right*)

The metric of a monitor hypersurface can be used for generating grids with node clustering near a surface on S^{xn} specified by the equation $\varphi(\boldsymbol{s}) = 0$ for which grad $\varphi(\boldsymbol{s}) \neq 0$. For this purpose, the monitor function in inverted Beltrami equations (11.23) can be specified as $\boldsymbol{f}(\boldsymbol{s}) = \boldsymbol{v}(\varphi(\boldsymbol{s}))$, where $\boldsymbol{v}(t)$ is a function whose derivative is large at $t = 0$, for example, $\boldsymbol{v}(t) = \mathrm{th}(t/\epsilon)$, thus $\boldsymbol{f}(\boldsymbol{s}) = th(\varphi(\boldsymbol{s})/\varepsilon),\ 1 \gg \varepsilon > 0$.

11.4.2 Spherical Control Metric

Contravariant metric tensor (11.11) can be used in inverted diffusion equations (11.14) and (11.15) for generating grids with node clustering in the zones of large variations of some function $\mathbf{u}(\mathbf{s}) = (u^1(\boldsymbol{s}), \dots, u^l(\boldsymbol{s}))$ (see Fig. 11.13). For this purpose, the control function $Z[\boldsymbol{u}](\mathbf{s})$ should be small at the points where $|\mathrm{grad}\ \mathbf{u}|(\mathbf{s})$ is small, for example, assuming $Z[\boldsymbol{u}](\boldsymbol{s}) = \sigma_i[\mathrm{grad}\ \boldsymbol{u}(\boldsymbol{s})],\ i = 1, \dots, 6$ with the functions $\sigma_i(t)$ from (11.17).

Notice the following fact. When using inverted Beltrami equations (11.23) for generating grids with cell clustering near a closed curve (closed surface when $n = 3$), nodes are moved to the curve from its interior subdomain, and in its exterior part, the grid is nearly unchanged (see Fig. 11.14, *left-hand*). This effect is not observed by applying diffusion equations (11.14) or (11.15) (see Fig. 11.15, *left-hand*).

11.5 Application of Layer-Type Functions to Grid Codes

Formulations of general control metrics (9.116) as well as of the metrics (9.140), (9.162), and (9.163) for generating field-aligned and balanced grids rely on some weight functions whose role is both to specify metrics and to determine the influence

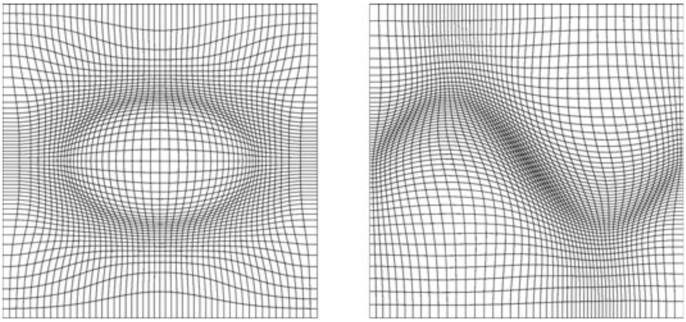


Fig. 11.15 Grids generated with the help of equations (11.15) with node clustering near a closed curve (*left*); near an unclosed curve (*right*)

of individual metrics on the resulting mesh. This section reviews the application of some basic layer-type univariate mappings to definition of the weight functions.

11.5.1 Specification of Basic Functions

A suitable set of weight functions can be formulated by using three basic functions $\varphi_i(x, \varepsilon), i = 1, 2, 3, x \geq 0, 0 < \varepsilon \ll 1$ which model locally the qualitative behavior of solutions to singularly perturbed problems along a coordinate transverse to the layers of their rapid variation.

The first function is the familiar exponential layer-type mapping

$$\varphi_1(x, \varepsilon) = \exp(-bx/\varepsilon^k) \ , \qquad k > 0 \ , \qquad b > 0 \ , \tag{11.27}$$

representing a layer-type function of the first order. The second mapping is a power function, namely,

$$\varphi_2(x, \varepsilon) = \frac{\varepsilon^{kb}}{(\varepsilon^k + x)^b} \ , \qquad k > 0 \ , \qquad b > 0 \ , \tag{11.28}$$

The third mapping is a logarithmic map

$$\varphi_3(x, \varepsilon) = \frac{\ln(1 + x\varepsilon^{-k})}{\ln(1 + \varepsilon^{-k})} \ , \qquad k > 0 \ . \tag{11.29}$$

The interval where any function $\varphi_i(x, \varepsilon), \ i = 1, 2, 3$, provides a rapid stretching of the coordinate x coincides with the interval where the first derivative with respect to x of this function is large. The first derivatives of the basic functions are

$$\begin{aligned}
\frac{\mathrm{d}\varphi_1}{\mathrm{d}x}(x,\varepsilon) &= -b\varepsilon^{-k}\exp(-bx/\varepsilon^k)\,, && k>0\,, \quad b>0\,,\\
\frac{\mathrm{d}\varphi_2}{\mathrm{d}x}(x,\varepsilon) &= -\frac{b\varepsilon^{kb}}{(\varepsilon^k+x)^{b+1}}\,, && k>0\,, \quad b>0\,,\\
\frac{\mathrm{d}\varphi_3}{\mathrm{d}x}(x,\varepsilon) &= \frac{1}{\ln(1+\varepsilon^{-k})(\varepsilon^k+x)}\,, && k>0\,,
\end{aligned}$$

Thus, the lengths of the intervals of the rapid transition of the functions are equal to

$$x_1 = C_1\varepsilon^k ln(\varepsilon^{-1}), \quad x_2 = C_2\varepsilon^{kb/(b+1)}, \quad x_3 = C_3/ln(1+\varepsilon^{-k}),$$

respectively. The intervals $[0, x_1]$, $[0, x_2]$, and $[0, x_3]$ are referred to as layers of singularity of corresponding functions $\varphi_i(x,\varepsilon)$, $i = 1, 2, 3$.

The quantities k and b in the expressions for the functions $\varphi_i(x,\varepsilon)$ are positive constants that control some characteristics of the functions and the layers of their singularity. In particular, the number k exhibits the scale of a layer. The constant b controls the type of stretching nonuniformity and the width of the layer. The parameter ε provides the major contribution to determination of the slopes of the functions in the vicinity of the point $x = 0$.

The basic functions $\varphi_i(x,\varepsilon)$, $i = 1, 2, 3$, have the boundary layers of rapid variation near the point $x = 0$. It is evident that the procedures of scaling, shifting, and matching can yield layer-type functions with arbitrary boundary and interior layers as well. These procedures are described in detail in the monograph by Liseikin (1999, 2001a).

Originally, the layer-type functions $\varphi_i(x,\varepsilon)$ were used in the so-called stretching method for specifying grid node clustering in the zones of boundary and interior layers for the numerical solution of singularly perturbed problems (see Bahvalov 1969 and Liseikin 2001a).

11.5.2 Numerical Grids Aligned to Vector-Fields

Application to Formulation of Field-Aligned Control Metrics

For generating numerical grids in a domain $X^n = S^n$, aligned to a vector-field $\mathbf{B}$, in accordance with Chap. 10, the contravariant metric components (9.118) can be used. For specifying these components, two vector-fields $\mathbf{B}_1$ and $\mathbf{B}_2$ may be chosen by

$$\mathbf{B}_1 = \mathbf{B}, \quad \mathbf{B}_2 = k\mathbf{D}\,,$$

where $\mathbf{D}$ is orthogonal to the field $\mathbf{B}$, and k is a small positive function. Thus, the contravariant components of the monitor metric for generating field-aligned grids in the domain X^n are as follows:

$$g^{ij}_{\mathbf{s}} = \epsilon(\mathbf{s})\delta^i_j + B^iB^j + k^2D^iD^j, \quad i,j = 1,\dots,n. \tag{11.30}$$

Since the vector field $\mathbf{B}$ is orthogonal to $\mathbf{D}$ and $|\mathbf{B}| = |\mathbf{D}|$, we find

$$g_{\mathbf{s}} = \det\{g_{\mathbf{s}}^{ij}\} = [\epsilon(\mathbf{s}) + |\mathbf{B}|^2][\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2]. \tag{11.31}$$

Taking into account that for $n = 2$

$$B^i B^j = |\mathbf{B}|^2\delta^i_j - D^i D^j, \quad i, j = 1, 2,$$

the contravariant metric components (11.30), for $n = 2$ and $l = 2$, are as follows:

$$\begin{aligned} g_{\mathbf{s}}^{ij} &= [\epsilon(\mathbf{s}) + |\mathbf{B}|^2]\delta^i_j + (k^2 - 1)D^i D^j \\ &= [\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2]\delta^i_j + (1 - k^2)B^i B^j, \quad i, j = 1, 2. \end{aligned} \tag{11.32}$$

So, the covariant components of the control metric in the Cartesian coordinates $s^1,\ s^2$ are expressed in both forms

$$\begin{aligned} g^{\mathbf{s}}_{ij} &= \frac{1}{\epsilon(\mathbf{s}) + |\mathbf{B}|^2}\left[\delta^i_j - \frac{k^2 - 1}{\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2}D^i D^j\right], \quad i, j = 1, 2, \\ g^{\mathbf{s}}_{ij} &= \frac{1}{\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2}\left[\delta^i_j - \frac{1 - k^2}{\epsilon(\mathbf{s}) + |\mathbf{B}|^2}B^i B^j\right], \quad i, j = 1, 2, \end{aligned} \tag{11.33}$$

Thus, the covariant metric components in the grid coordinates ξ^1, ξ^2 are presented as

$$\begin{aligned} g^{\xi}_{ij} &= \frac{1}{\epsilon(\mathbf{s}) + |\mathbf{B}|^2}\left[\frac{\partial \mathbf{s}}{\partial \xi^i}\cdot\frac{\partial \mathbf{s}}{\partial \xi^j} - \frac{k^2 - 1}{\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2}D^m\frac{\partial s^m}{\partial \xi^i}D^p\frac{\partial s^p}{\partial \xi^j}\right] \\ &= \frac{1}{\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2}\left[\frac{\partial \mathbf{s}}{\partial \xi^i}\cdot\frac{\partial \mathbf{s}}{\partial \xi^j} - \frac{1 - k^2}{\epsilon(\mathbf{s}) + |\mathbf{B}|^2}B^m\frac{\partial s^m}{\partial \xi^i}B^p\frac{\partial s^p}{\partial \xi^j}\right], \\ & i, j, m, p = 1, 2. \end{aligned} \tag{11.34}$$

For generating a grid with the requirement that grid coordinates are aligned with the vector field $\mathbf{B}$, we assume $k \sim 0.01 - 0.1$ and $\epsilon(\mathbf{s})$ as a function with small positive values when $|\mathbf{B}| \sim 1,$ while $\epsilon(\mathbf{s}) \sim 1$ when $|\mathbf{B}| = 0$. The second condition for $\epsilon(\mathbf{s})$ is stipulated by the effect of the solution of inverted diffusive equations: the grid cells become very small at the points where all elements $g_{\mathbf{s}}^{ij}$ are small. The functions $\epsilon(\mathbf{s})$ satisfying these properties are formulated through the boundary layer-type functions (11.27)–(11.29) assuming $\epsilon(\mathbf{s}) = \varphi(|\mathbf{B}(\mathbf{s})|^2, \delta)$, where

$$\varphi(x, \delta) = \begin{cases} Mexp(-x/\delta), \\ M\delta^\alpha/(\delta + x)^\alpha, \quad \alpha > 0, \\ M\ln(\delta + x)/\ln\delta, \end{cases}$$

for $x \geq 0, \quad 0 < \delta << 1, \ \ M = const$. These boundary layer-type functions help solutions of grid equations switch from one mode to another.

In the case of a two-dimensional domain $X^2 = S^2$ and a square logical domain $\varXi^2$, the boundary conditions at the points of the boundary segments $\xi^2 = 0; 1$ of the logical square $\varXi^2$ are found iteratively to satisfy the requirement of orthogonality

$$\frac{\partial \mathbf{s}}{\partial \xi^1} \cdot \frac{\partial \mathbf{s}}{\partial \xi^2} = 0.$$

The conditions at the boundary points of the other coordinate family can be either specified as fixed or they can be specified at one segment $\xi^1 = const$ and computed iteratively at the points of the other segment to satisfy the requirement of the alignment of grid lines at these points with the vector field

$$\mathbf{B}_3 = [B^1 + sgnB^1\varphi(|B^1|^2, \delta),\ (1 - \varphi(|B^1|^2, \delta))B^2].$$

This vector field is introduced to rule out, at the corresponding boundary points, the direction $(0, B^2)$ parallel to the boundary segment $\xi^1 = const$ for a coordinate curve $\xi^2 = const$ emanating from this segment.

Numerical Experiments

Figures 11.16, 11.17, 11.18 and 11.19 demonstrate isocontours (left-hand) of a vector field $\mathbf{B}$ in a two-dimensional domain S^2 and pictures (right-hand) of the corresponding grids. The vector field $\mathbf{B}$ is formed by a model function $\psi(\mathbf{s})$ via the relation

$$\mathbf{B} = \left(-\frac{\partial \psi}{\partial s^2}, \frac{\partial \psi}{\partial s^1}\right). \tag{11.35}$$

This vector field is subject to the natural equation for magnetic fields $div\mathbf{B} = 0$. In addition, for such a vector field, $\mathbf{D} = grad\ \psi$. Thus, in this case, the Eq. (11.34) become

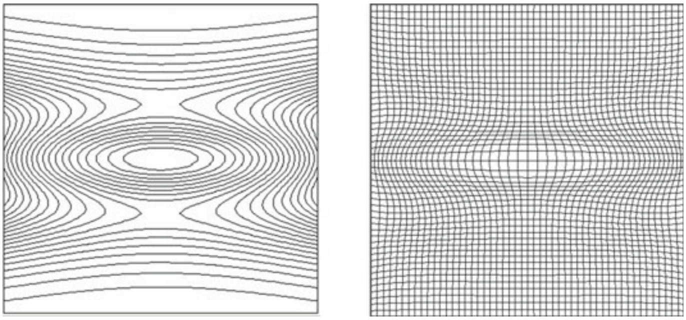


Fig. 11.16 Field-aligned grid (*right*) for a symmetric vector field (*left*)

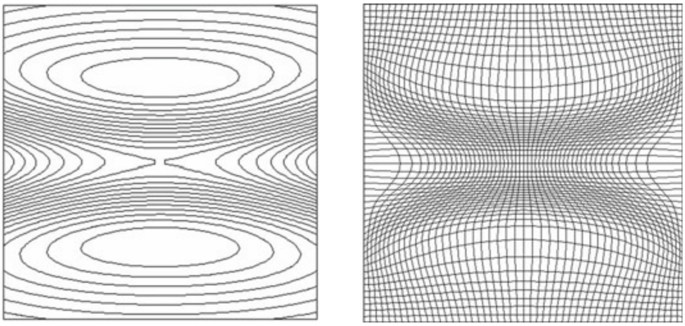


Fig. 11.17 Field-aligned grid (*right*) for a vector field with two islands (*left*)

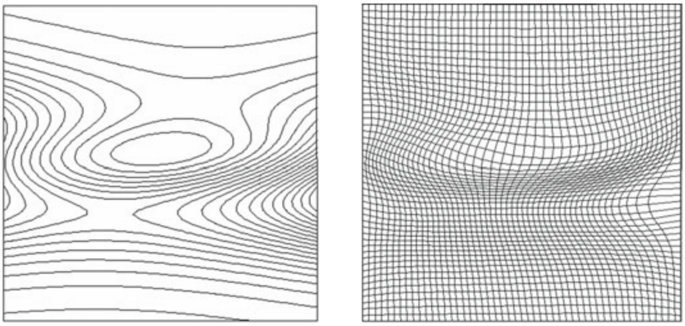


Fig. 11.18 Field-aligned grid (*right*) for a nonsymmetric vector field (*left*)

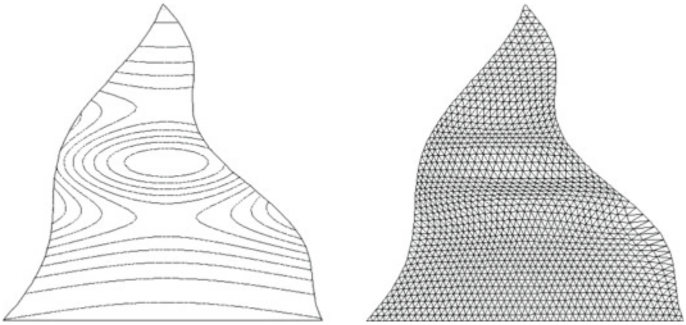


Fig. 11.19 Example of a triangular field-aligned grid

$$g^{\boldsymbol{\xi}}_{ij} = \frac{1}{\epsilon(\mathbf{s}) + |\mathbf{B}|^2}\left[\frac{\partial \mathbf{s}}{\partial \xi^i} \cdot \frac{\partial \mathbf{s}}{\partial \xi^j} - \frac{k^2 - 1}{\epsilon(\mathbf{s}) + k^2|\mathbf{B}|^2}\frac{\partial \psi}{\partial \xi^i}\frac{\partial \psi}{\partial \xi^j}\right], \quad i, j = 1, 2.$$

With these covariant components, the two-dimensional diffusion equations in (9.134) have the following form:

$$\left(g^{\xi}_{22}\frac{\partial^2 s^i}{\partial\xi^1\partial\xi^1} - 2g^{\xi}_{12}\frac{\partial^2 s^i}{\partial\xi^1\partial\xi^2} + g^{\xi}_{11}\frac{\partial^2 s^i}{\partial\xi^2\partial\xi^2}\right) = R^i, \quad i = 1,2, \tag{11.36}$$

where

$$\begin{aligned} R^i &= \frac{(J)^2}{w[\mathbf{s}(\boldsymbol{\xi})]g_{\mathrm{s}}}\frac{\partial}{\partial s^j}[w(\mathbf{s})g^{ij}_{\mathbf{s}}] \\ &= \frac{J}{w[\mathbf{s}(\boldsymbol{\xi})]g_{\mathrm{s}}}\left\{\frac{\partial s^2}{\partial\xi^2}\frac{\partial}{\partial\xi^1}[w(\mathbf{s})g^{i1}_{\mathbf{s}}] - \frac{\partial s^2}{\partial\xi^1}\frac{\partial}{\partial\xi^2}[w(\mathbf{s})g^{i1}_{\mathbf{s}}]\right. \\ &\quad \left. - \frac{\partial s^1}{\partial\xi^2}\frac{\partial}{\partial\xi^1}[w(\mathbf{s})g^{i2}_{\mathbf{s}}] + \frac{\partial s^1}{\partial\xi^1}\frac{\partial}{\partial\xi^2}[w(\mathbf{s})g^{i2}_{\mathbf{s}}]\right\}, \quad i,j = 1,2. \end{aligned}$$

Equations (11.36) become the inverted Beltrami equations after substituting $\sqrt{g^{\mathrm{s}}} = 1/\sqrt{g_{\mathrm{s}}}$ for $w(\mathbf{s})$.

The Eq. (11.36) were solved with the numerical algorithms reviewed in Chap. 10. The following expressions for $\epsilon(\mathbf{s})$ and $\psi(\mathbf{s})$ were used:

$$\epsilon(\mathbf{s}) = \begin{cases} 0.05\ \exp(-|\mathbf{B}|^2/0.1), & \text{Fig. 11.16;} \\ 0.05\left(\dfrac{0.3}{0.3+|\mathbf{B}|^2}\right)^2, & \text{Fig. 11.17;} \\ 0.05\ln(0.005+|\mathbf{B}|^2)/\ln(0.005), & \text{Fig. 11.18;} \\ 0.1 exp\left(-\dfrac{|\mathbf{B}|}{0.07}\right), & \text{Fig. 11.19.} \end{cases}$$

$$\psi(\mathbf{s}) = \begin{cases} \phi(s^2)(1-\phi(s^2))[(s^1-0.5)^2 + 2(\phi(s^2)-0.5)^2], & \text{Fig. 11.16;} \\ \phi(s^2)(1-\phi(s^2))[(s^1-0.5)^2 - 6(\phi(s^2)-0.5)^2], & \text{Fig. 11.17;} \\ \phi(s^2)(1-\phi(s^2))[(s^1-0.5)^2 + 2(\phi(s^2) & \\ -0.5-0.2s^1)^2], & \text{Fig. 11.18;} \\ \phi(s^2)(1-\phi(s^2))[(s^1-0.5)^2 + 1.5(\phi(s^2)-0.5)^2], & \text{Fig. 11.19.} \end{cases}$$

where

$$\phi(s^2) = 0.5\left[1 + tanh\left(\frac{s^2-0.5}{0.2}\right)\right].$$

An example of a field-aligned grid for the tokamak edge region is exhibited in Fig. 11.20 (right-hand). The left-hand picture demonstrates an initial grid presented by A. Glasser. Originally, grids for the tokamak edge region were performed by Petravic (1987) and Rognlien et al. (2002).

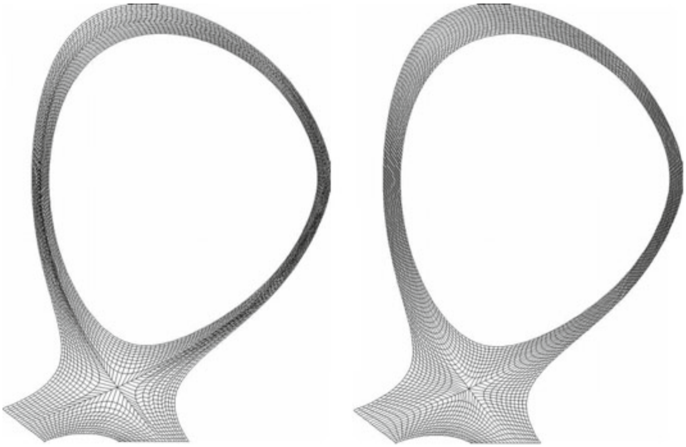


Fig. 11.20 Initial (*left*) and final (*right*) grids for the tokamak edge region

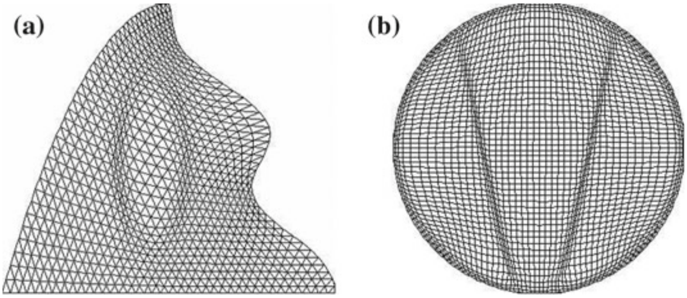


Fig. 11.21 Triangular (*left*) and quadrangular (*right*) adaptive grids

11.5.3 Application to Grid Clustering

The basic layer-type functions $\phi_i(x, \varepsilon),\ i = 1, 2, 3$, can also be used to produce grid clustering in the vicinity of a hypersurface defined by the equation $\phi(\mathbf{s}) = 0$. For this purpose, one can specify a control metric in the form (11.22) proposed for generating grids adapting to the gradient of a function $f(\mathbf{s})$ that has large variation near the hypersurface. One such function is defined by the formula $f(\mathbf{s}) = tanh[\phi(\mathbf{s})/\delta]$, which includes the layer-type map $\phi_1(x, \delta)$. Figure 11.21a demonstrates a grid with node clustering near the curve $\phi(\mathbf{s}) = 0$. Figure 11.21b shows grid clustering near two curves $\phi_1(\mathbf{s}) = 0$ and $\phi_2(\mathbf{s}) = 0$.

Figure 11.21 shows grids adapting to the gradient of the following functions:

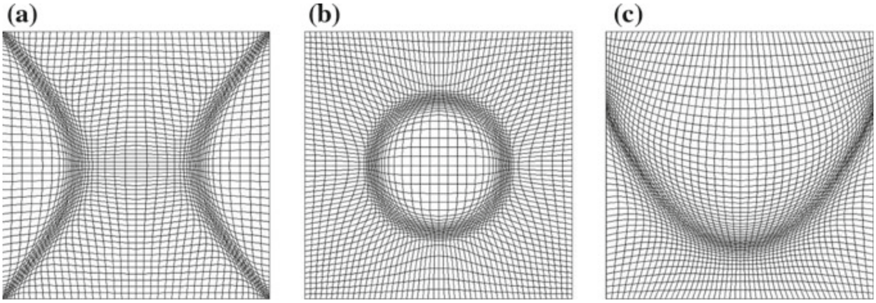


Fig. 11.22 Adaptive grids generated through a spherical control metric

$$f(\mathbf{s}) = \begin{cases} 0.05\, tanh\left(\dfrac{\phi(\mathbf{s})}{0.05}\right), & (a); \\ 0.06\, tanh\left(\dfrac{\phi_1(\mathbf{s})}{0.05}\right) + 0.08\, tanh\left(\dfrac{\phi_2(\mathbf{s})}{0.1}\right), & (b); \end{cases}$$

where

$$\begin{aligned} \phi(\mathbf{s}) &= 100(s^1 - 0.5)^2 + 16(s^2 - 0.5)^2 - 1, \\ \phi_1(\mathbf{s}) &= (s^1 - 0.5)^2 + (s^2 - 0.5)^2 - 0.5, \\ \phi_2(\mathbf{s}) &= s^2 - 0.5 - 0.8\, sin(6(s^1 + 0.3)). \end{aligned}$$

Another (spherical) metric for providing grid clustering is defined by the formula

$$g_{\mathbf{s}}^{ij} = \omega[z(\phi)]\delta^i_j, \quad i, j = 1, \dots, n,$$

where the function $z(\phi)$ is formulated by the basic layer-type mappings $\phi_i(x, \varepsilon)$, $i = 1, 2, 3$. Figures 11.22 and 11.23 exhibit domain grids generated through such a metric by the numerical solution of the inverted diffusion grid equations. We are reminded that, in accordance with Sect. 10.5.3, the three-dimensional diffusion equations are the Beltrami equations in a modified metric.

For Fig. 11.22, it was assumed that

$$\omega[z(\phi)] = \begin{cases} \exp(-z^2/0.5), & z(\mathbf{s}) = \exp\{-[\phi(\mathbf{s})]^2/0.001\}, \quad (a); \\ \left(\dfrac{0.5}{0.5 + z^2}\right)^2, & z(\mathbf{s}) = \exp\{-[\phi(\mathbf{s})]^2/0.0005\}, \quad (b); \\ 2\exp(-z^2/0.3), & z(\mathbf{s}) = \dfrac{\ln\{0.0005 + [\phi(\mathbf{s})]^2\}}{\ln(0.0005)}, \quad (c); \end{cases}$$

$$\phi(\mathbf{s}) = \begin{cases} (s^1 - 0.5)^2 - 0.8(s^2 - 0.5)^2 - 0.05, & (a); \\ (s^1 - 0.5)^2 + (s^2 - 0.5)^2 - 0.0625, & (b); \\ s^2 - 2(s^1 - 0.5)^2 - 0.2, & (c). \end{cases}$$

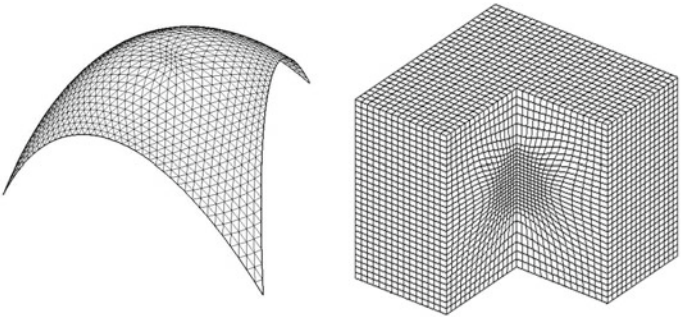


Fig. 11.23 Surface and volume adaptive grids generated through a spherical control metric

Figure 11.23 demonstrates two-dimensional and three-dimensional adaptive grids in the case

$$\begin{aligned}\omega[z(\phi)] &= 0.5\, exp\left[2\, exp\left(-\frac{\phi(\mathbf{s})^2}{0.001}\right)\right],\\ \phi(\mathbf{s}) &= (s^1-0.5)^2+(s^2-0.5)^2+(s^3-0.5)^2,\end{aligned}$$

which provides node clustering near central points of the geometries.

11.6 Generation of Multi-block Smooth Grids

The mapping approaches described above are formulated for generating local single-block grids. This section reviews some approaches for extending the algorithms for generating multi-block smooth grids.

11.6.1 Approaches to Smoothing Grids

In this subsection, we describe two iterative approaches, originally presented in Likhanova et al. (2006), for providing smooth matching of grid lines across adjacent blocks. These approaches are readily applied to generating smooth grids in the vicinity of fictive edges of faces, as in the case of O- or C-types of meshes. The essential difference from the previous methods for generating smooth block-structured grids is that the current approaches are based on the computation of both the position of the joint boundary segments of the blocks and grid distribution at these segments. The position of the segments and grid distribution are found (1) through the numerical solution of grid equations and (2) by the interpolation from the nodes of the grid hypersurfaces neighboring the joint boundary segment.

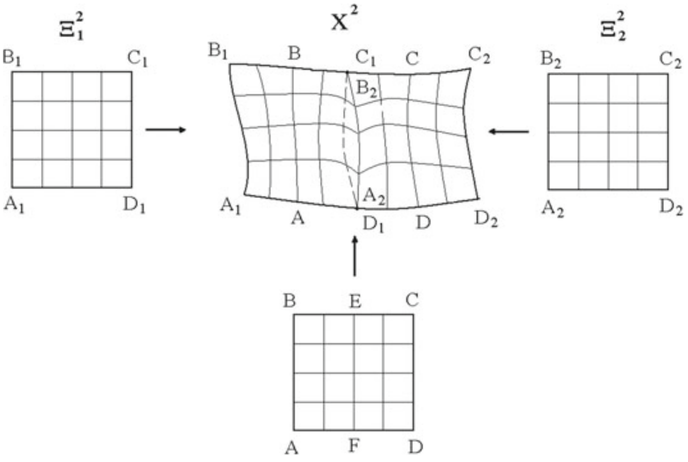


Fig. 11.24 Scheme for generating smooth grids by computing the common points of the block through grid equations

Computation Through Grid Equations

The idea of this version of the approach is demonstrated in Fig. 11.24, representing a two-block structured scheme for generating smooth quadrilateral grids in a domain X^2. The left-hand block of the domain is bounded by the curves $A_1B_1,\ B_1C_1,\ C_1D_1$, and D_1A_1. Similarly, the right-hand block is bounded by the segments $A_2B_2,\ B_2C_2,\ C_2D_2$, and D_2A_2. The curves C_1D_1 and B_2A_2 are identical, presenting the joint boundary of the blocks. By the iterations, the grid in the domain X^2 is computed independently at each block via solving numerically, according to the methods described in Chap. 10 the Dirichlet boundary value problem for grid equations at the nodes of the corresponding logical domains Ξ_1^2 and Ξ_2^2 with the identical boundary node distribution at joint segments in X^2. These segments and their grid points are found in the process of iterations by solving the boundary value problem at the points of a new logical domain $ABCD$.

Thus, during the first iteration, we specify the joint boundary segment $A_2B_2 = D_1C_1$ in X^2, the grid nodes at this segment, and the transformation at these nodes from the grid points of the segments C_1D_1 and A_2B_2 of the corresponding logical domains Ξ_1^2 and Ξ_2^2. Then, we compute independently the grid nodes in the both blocks of X^2 through the grid equations. Having done this, we choose in both blocks the corresponding grid lines AB and CD, for example, neighboring the joint boundary line $C_1D_1 = B_2A_2$. After this, we solve numerically the boundary value problem for the grid equations at the points of a new logical domain $ABCD$. The number of points at the segments AD in the domain X^2 and in this new logical domain coincides. The boundary points at the segments AB and CD in the logical domain are mapped at the computed points of the corresponding segments AB and CD in X^2. The transformation of the boundary points at the segments AD and BC

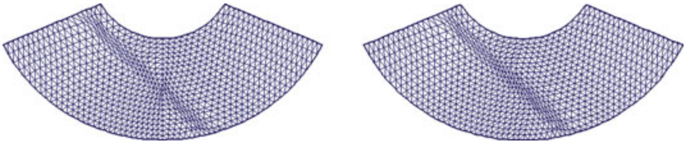


Fig. 11.25 Triangular nonsmooth (*left*) and smooth (*right*) grids

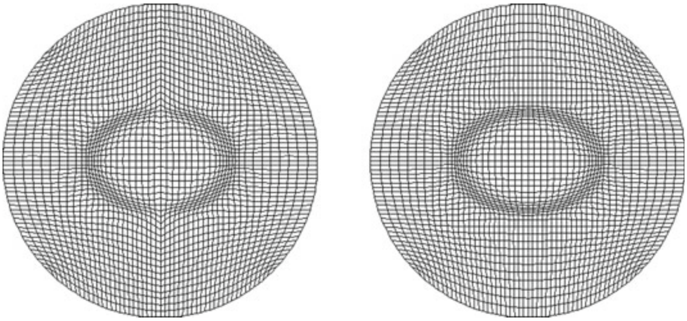


Fig. 11.26 Quadrangular nonsmooth (*left*) and smooth (*right*) grids

coincides with the initial boundary transformation from the corresponding points of the segments A_1D_1, A_2B_2 and B_1C_1, B_2C_2. In particular, the points E and F are mapped at the points C_1 and D_1, respectively. By the computation, the points of the segment EF in the new logical domain are transformed at the points of a new joint boundary segment (dotted line) of two new blocks. Then, the iterations continue up to the point of satisfying a tolerance condition. Similarly, smooth block-structured triangular grids are generated.

Figures 11.25, 11.26, and 11.27 demonstrate nonsmooth grids (left-hand) computed in two-dimensional domains without the use of the smoothing algorithm and smooth grids (right-hand) found through application of the algorithm.

An analogous procedure is formulated for generating smooth block-structured surface and three-dimensional domain grids. Figure 11.28 exhibits nonsmooth (left-hand) and smooth (right-hand) grids on a surface.

An essential feature of this approach is that the equations for generating grids in the blocks and for computing new joint boundary segments of the blocks are the same. So, such a smoothing process does not breach the properties of the grids realized by particular control metrics.

11.6.2 Computation by Interpolation

In this approach, the grid point at the i-th level of a new joint boundary segment (dotted line in Fig. 11.29) is computed through the following procedure: after com-

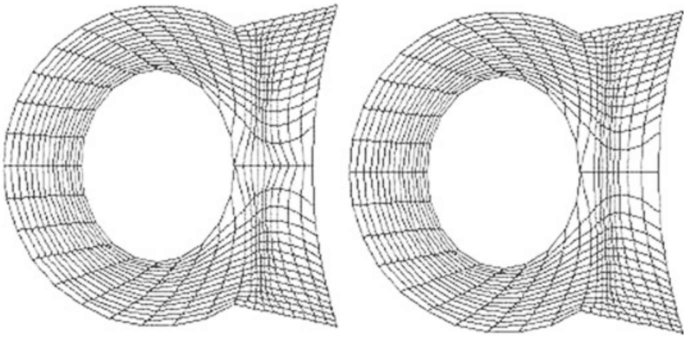


Fig. 11.27 C-type nonsmooth (*left*) and smooth (*right*) grids

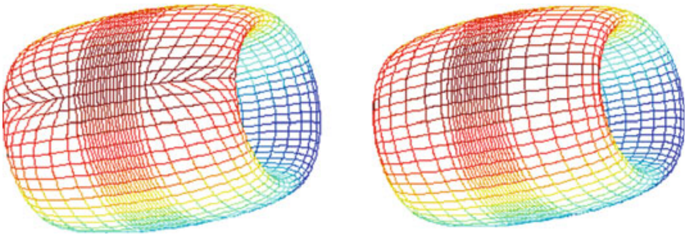


Fig. 11.28 Surface nonsmooth (*left*) and smooth (*right*) grids

puting the grid points at both blocks having a joint boundary line with a grid node D, we choose at the i-th level of grid lines two points neighboring the previous joint segment point D of one block (the points B and C in Fig. 11.29) and one neighboring a point of another block (point F in Fig. 11.29). Through these points, we draw a smooth line, for example, a circle segment. The i-th level of a new joint boundary point E will lie on this segment. Its position at the segment is defined from the relation $\overline{CE}/\overline{EF} = \overline{AB}/\overline{BC}$, where A is the third grid point of the first block computed at the i-th level (overline means the distance between the corresponding points). When all grid points of the new joint boundary segment are found in such a way then the grid points in the blocks are found independently by solving grid equations. The process continues up until the point of satisfying a required tolerance.

Figure 11.30 illustrates an O-type nonsmooth (left-hand) and smooth (right-hand) surface triangular grid generated by this approach.

Similarly, smooth block-structured three-dimensional domain grids are generated.

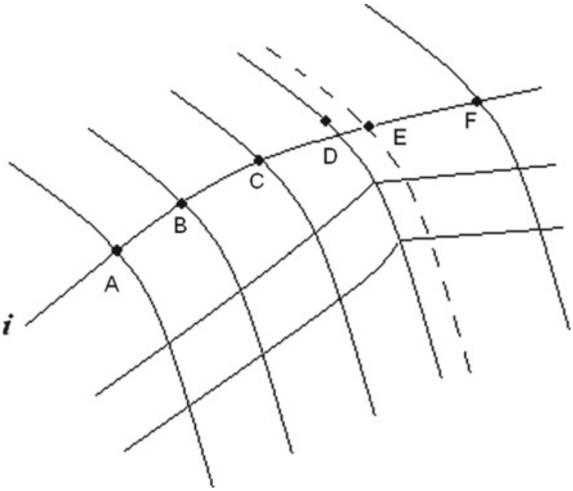


Fig. 11.29 Schema for smoothing grids

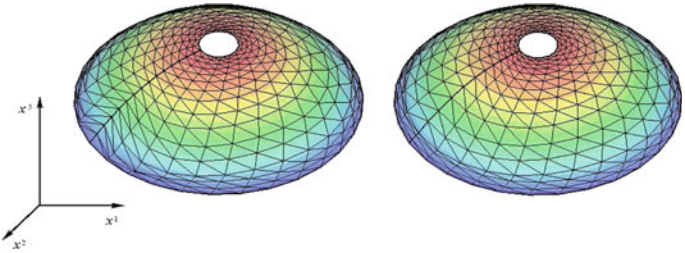


Fig. 11.30 Surface triangular nonsmooth (*left*) and smooth (*right*) grids

References

Bahvalov, N. S. (1969). On optimization of the methods of the numerical solution of boundary-value problems with boundary layers. *Computational Mathematics and Mathematical Physics*, *9*(4), 842–859 (Russian) [English transl.: *USSR Computational Mathematics and Mathematical Physics*, *9* (1969)].

Likhanova, Yu. V, Liseikin, V. D., Patrakhin, D. V., & Vaseva, I. A. (2006). Generation of block structured smooth grids. *Compindia Technologies*, *11*(4), 3–12.

Liseikin, V. D. (1999). *Grid generation methods*. Berlin: Springer.

Liseikin, V. D. (2001a). *Layer resolving grids and transformations for singular perturbation problems*. Utrecht: VSP.

Petravic, M. (1987). Orthogonal grid construction for modeling of transport in Tokamaks. *Journal of Computational Physics*, *73*, 125–130.

Rognlien, T. D., Xu, X. Q., & Hinmarsh A. C. (2002). Application of parallel implicit methods to edge-plasma numerical simulations. *Journal of Computational Physics*, *175*, 249–268.

Chapter 12
Unstructured Methods

12.1 Introduction

Unstructured mesh techniques occupy an important niche in grid generation. The major feature of unstructured grids consists, in contrast to structured grids, of a nearly absolute absence of any restrictions on grid cells, grid organization, or grid structure. Figuratively speaking, unstructured grids manifest the domination of anarchy while structured grids demonstrate adherence to order. The concept of unstructured grids allows one to place the grid nodes locally irrespective of any coordinate directions, so that curved boundaries can be handled with ease and local regions in which the solution is turbulent or its variations are large can be resolved with a selective insertion of new points without unduly affecting the resolution in other parts of the physical domain.

Unstructured grid methods were originally developed in solid mechanics. Nowadays, these methods influence many other fields of application beyond solid modeling, in particular, computational fluid dynamics, where they are becoming widespread.

Unstructured grids can, in principle, be composed of cells of arbitrary shapes built by connecting a given point to an arbitrary number of other points, but are generally formed from tetrahedra and hexahedra (triangles and quadrilaterals in two dimensions). The advantages of these grids lie in their ability to deal with complex geometries, while allowing one to provide natural grid adaptation through the insertion of new nodes.

At the present time, the methods of unstructured grid generation have reached the stage in which three-dimensional domains with complex geometry can be successfully meshed. The most spectacular theoretical and practical achievements have been connected with the techniques for generating tetrahedral (or triangular) grids. There are at least two basic approaches that have been used to generate these meshes: Delaunay and advancing-front. This chapter presents a review of some popular techniques realizing these approaches.

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_12

Note that the chapter addresses only some general aspects of unstructured grid methods. The interested reader who wishes to learn more about the wider aspects of unstructured grids should study, for example, the monographs by Carey (1997) George and Borouchaki (1998a, b), Frey and George (2008), and Lo (2015).

12.2 Methods Based on the Delaunay Criterion

Much attention has been paid in the development of methods for unstructured discretizations to triangulations which are based upon the very simple geometrical constraint that the hypersphere of each n-dimensional simplex defined by $n + 1$ points is void of any other points of the triangulation. For example, in three dimensions, the four vertices of a tetrahedron define a circumsphere which contains no other nodes of the tetrahedral mesh. This restriction is referred to as the Delaunay or incircle criterion, or the empty-circumcircle property. Triangulations obeying the Delaunay criterion are called Delaunay triangulations. They are very popular in practical applications, owing to the following properties being valid in two dimensions:

(1) Delaunay triangles are nearly equilateral;
(2) the maximum angle is minimized;
(3) the minimum angle is maximized;
(4) the triangulation is unique if the points are in a general position, i.e. no four points are cyclic;
(5) if every triangle in a triangulation is non-obtuse, it is a Delaunay triangulation;
(6) any two-dimensional triangulation can be transformed into a Delaunay triangulation by locally flipping of the diagonals of adjacent triangles.

These properties give some grounds to expect that the grid cells of a Delaunay triangulation are not too deformed.

Based on a sound geometrical concept and the optimality properties, Delaunay triangulation has important applications in many fields, including data visualization, terrain modelling, mesh generation, surface reconstruction and structural networking for arbitrary point sets. The popularity of Delaunay triangulation is attributed to its nice geometric properties as a dual of Voronoi tessellation and the speed with which it can be constructed in two or higher dimensions.

The Delaunay criterion itself is not an algorithm for mesh generation. It merely provides a rule for connecting a set of existing points in space to form a triangulation. As a result, although the boundary of the domain is well specified, it is necessary to devise a scheme to determine the number and the locations of node points to be inserted within the domain of interest.

The Delaunay criterion does not give any indication as to how the grid points should be defined and connected. One more drawback of the Delaunay criterion is that it may not be possible to realize it over the whole region with a prespecified boundary triangulation. This disadvantage gives rise to two grid generation approaches of constrained triangulation which preserve the boundary connectivity and take into account the Delaunay criterion. In the first approach of constrained Delaunay triangulation, the Delaunay property is overriden at points close to the boundaries, and consequently the previously generated boundary grid remains intact. Alternatively, or in combination with this technique, points can be added in the form of a skeleton to ensure that breakthroughs of the boundary do not occur. Another approach, which observes the Delaunay criterion over the whole domain, is to postprocess the mesh by recovering the boundary simplexes which are missed during the generation of the Delaunay triangulation and by removing the simplexes lying outside the triangulated domain.

There are a number of algorithms for generating unstructured grids based on the Delaunay criterion in constrained or unconstrained forms.

Some methods for Delaunay triangulations are formulated for a preassigned distribution of points which are specified by means of some appropriate technique, in particular, by a structured grid method. These points are connected to obtain a triangulation satisfying certain specific geometrical properties which, to some extent, are equivalent to the Delaunay criterion.

Many Delaunay triangulations use an incremental Bowyer–Watson algorithm which can be readily applied to any number of dimensions. It starts with an initial triangulation of just a few points. The algorithm proceeds at each step by adding points one at a time into the current triangulation and locally reconstructing the triangulation. The process allows one to provide both solution-adaptive refinement and mesh quality improvement in the framework of the Delaunay criterion. The distinctive characteristic of this method is that point positions and connections are computed simultaneously.

One more type of algorithm is based on a sequential correction of a given triangulation, converting it into a Delaunay triangulation.

12.2.1 Dirichlet Tessellation

A very attractive means for generating a Delaunay triangulation of an assigned set of points is provided by a geometrical construction first introduced by Dirichlet (1850).

Consider an arbitrary set of points P_i, $i = 1, \dots, N$, in the n-dimensional domain. For any point P_i, we define a region $V(P_i)$ in R^n characterized by the property that it is constituted by the points from R^n which are nearer to P_i than to any other P_j, i.e.

$$V_i = \{x \in R^n | \mathrm{d}(x, P_i) \le \mathrm{d}(x, P_j) \ , \quad i \ne j \ , \quad j = 1, \dots, N\} \ ,$$

where $\mathrm{d}(a, b)$ denotes the distance between the points a and b. These areas V_i are called the Voronoi polyhedrons (see Fig. 12.1 for $n = 2$). Thus, the polyhedra are intersections of half-spaces, and therefore they are convex, though not necessarily bounded. The set of Voronoi polyhedra corresponding to the collection of points P_i is called the Voronoi diagram or Dirichlet tessellation. The common boundary of

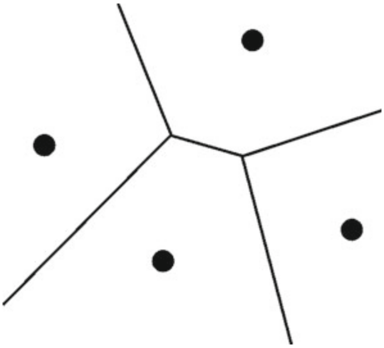


Fig. 12.1 Voronoi polyhedron for 4 points

two facing Voronoi regions $V(P_i)$ and $V(P_j)$ is an $(n-1)$-dimensional polygon. A pair of points P_i and P_j whose Voronoi polyhedra have a face in common is called a configuration pair. By connecting only the contiguous points, a network is obtained. In this network, a set of $n+1$ points which are contiguous with one another forms an n-dimensional simplex. The circumcenter, i.e. the center of the hypersphere, of any simplex is a vertex of the Voronoi diagram. The hypersphere of the simplex is empty, that is, there is no point inside the hypersphere. Otherwise, this point would be nearer to the circumcenter than the points on the hypersphere. Thus, the set of simplexes constructed in such a manner from the Dirichlet tessellation constitutes a new tessellation which satisfies the Delaunay criterion and is, therefore, a Delaunay triangulation. The boundary of the Delaunay triangulation built from the Voronoi diagram is the convex hull of the set of points P_i (see Fig. 12.2 for $n=2$).

It should be noted that Delaunay triangulations and Dirichlet tessellations can be considered the geometrical duals of each other, in the sense that for every simplex S_i, there exists a vextex P_i of the tessellation and, conversely, for every Voronoi region $V(P_j)$, there exists a vertex P_j of the triangulation. In addition, for every edge of the triangulation, there exists a corresponding $(n-1)$-dimensional segment of the Dirichlet tessellation.

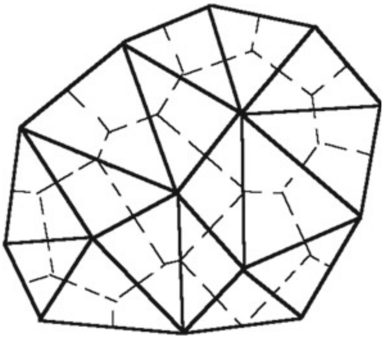


Fig. 12.2 Voronoi diagram and Delaunay triangulation

12.2.2 Incremental Techniques

The empty-hypercircle criterion of the Delaunay triangulation can be utilized to create incremental triangulation algorithms for arbitrary dimensions. Recall that by the Delaunay triangulation of a set V_N of N points in n-dimensional space, we mean the triangulation of V_N by simplexes with the vertices taken from V_N such that no point lies inside the hypersphere of any n-dimensional simplex.

Here, two incremental methods are presented. In the first method, a new n-dimensional simplex is constructed during each stage of the triangulation, using the given set of points for this purpose. In the second technique, each step produces several simplexes which are generated after inserting a new point.

A-Priori-Given Set of Points

Let a set of points V_N in a bounded n-dimensional domain X^n be given. We assume that these points do not lie in any $(n-1)$-dimensional hyperplane. The incremental technique starts by taking an $(n-1)$-dimensional face e (edge in two dimensions and triangle in three dimensions), commonly the one with the smallest size, and constructing hyperspheres through the vertices of e and any one of the remaining points of V_N. One of these hyperspheres formed by a point, say, P_1, does not contain any point of V_N inside it. The $(n-1)$-dimensional simplex e and P_1 define a new n-dimensional simplex. In the next step, the $(n-1)$-dimensional simplex e is taken out of consideration. The algorithm stops, and the triangulation is complete, when every boundary face corresponds to the side of one simplex and every internal $(n-1)$-dimensional simplex forms the common face of precisely two n-dimensional simplexes. It is clear that this algorithm is well suited to generate a Delaunay triangulation with respect to a prescribed boundary triangulation.

The set of points used to generate the triangulation can be built with a structured method or an octree approach, or by embedding the domain into a Cartesian grid. However, the most popular approach is to utilize the strategy of a sequential insertion of new points.

Modernized Bowyer–Watson Technique

Another incremental method, proposed by Baker (1989) and which is a generalization of the Bowyer–Watson technique, starts with some triangulation, not necessarily that of Delaunay, of the set of N points $V_N = \{P_i | i = 1, \dots, N\}$ by an assembly of simplexes $T_N = \{S_j\}$. For any simplex $S \in T_N$, let R_S be the circumradius and Q_S the circumcenter of S. In the sequential-insertion technique, a new point P is introduced inside the convex hull of V_N. Let $B(P)$ be the set of the simplexes whose circumspheres contain the point P, i.e.

$$B(P) = \{S | S \in T_N,\ \mathrm{d}(P, Q_S) < R_S\},$$

where $\mathrm{d}(P, Q)$ is the distance between P and Q. All these simplexes from $B(P)$ form a region $\Gamma(P)$ surrounding the point P. This region is called the generalized cavity. The maximal simply connected area of $\Gamma(P)$ that contains the point P is called the principal component of $\Gamma(P)$ and denoted by Γ_P. The point P is checked to determine if it is visible from all boundary segments of the principal component or if it is obscured by some simplex. In the former case, the algorithm generates new simplexes associated with P by joining all of the vertices of the principal component with the point P. In the latter case, either this point is rejected and a new one is introduced or the principal component Γ_P is reduced by excluding the redundant simplexes from $B(P)$ to obtain an area whose boundary is not obscured from P by any simplex. Then, the new simplexes are formed as in the former case. The union of these simplexes and those which do not form the reduced region of the retriangulation defines a new triangulation of the set of $N+1$ points $V_{N+1} = V_N \bigcup\{P\}$. In this manner, the process proceeds by inserting new points, checking visibility, adjusting the principal component, and generating new simplexes. The new triangulation differs from the previous one only locally around the newly inserted point P.

In two dimensions, we have that if the initial triangulation is the Delaunay triangulation, then the region $\Gamma(P)$ is of star shape, and consequently the boundary is visible from the point P and each step of the Bowyer–Watson algorithm produces a Delaunay triangulation. Thus, in this case, the Bowyer–Watson algorithm is essentially a "reconnection" method, since it computes how an existing Delaunay triangulation is to be modified because of the insertion of new points. In fact, the algorithm removes from the existing grid all the simplexes which violate the empty-hypersphere property because of the insertion of the new point. The modification is constructed in a purely sequential manner, and the process can be started from a very simple initial Delaunay triangulation enclosing all points to be triangulated (for example, that formed by one very large simplex or one obtained from a given set of boundary points) and adding one point after another until the necessary requirements for grid quality have been satisfied.

12.2.3 Approaches for Insertion of New Points

The sequential nature of the Bowyer–Watson algorithm gives rise to a problem of choosing the position where the new point in the existing mesh will be inserted, because a poor point distribution can eventually lead to an unsatisfactory triangulation. The new point should be chosen according to some suitable geometrical and physical criteria which depend on the existing triangulation and the behavior of the physical solution. The geometrical criteria commonly consist in the requirement for the grid to be smooth and for the cells to be of a standard uniform shape and a necessary size. The physical criterion commonly requires the grid cells to be concentrated in some specific zones as the zones of turbulence, large solution variations, or large solution errors. With respect to the geometrical criterion of generating uniform cells, the vertices and segments of the Dirichlet tessellation are promising locations for placing a new point, since they represent a geometrical locus which falls, by construction, midway between the triangulation points. Thus, in order to control the size and shape of the grid cells, two different ways in which the new point is inserted are commonly considered. In the first, the new point is chosen at the vertex of the Voronoi polyhedron corresponding to the “worst” simplex. In the second way, the new point is inserted into a segment of the Dirichlet tessellation, in a position that guarantees the required size of the newly generated simplexes.

12.2.4 Two-Dimensional Approaches

This subsection discusses the major techniques delineated in Sects. 12.2.1–12.2.3 for generating planar triangulations based on the Delaunay criterion.

Voronoi Diagram

The Delaunay triangulation has a dual set of polygons referred to as the Voronoi diagram or the Dirichlet tessellation.

The Voronoi diagram can be constructed for an arbitrary set of points in the domain. Each polygon of the diagram corresponds to the point that it encloses. The polygon for a given point is the region of the plane which is closer to that point than to any other points. These regions have polygonal shapes and the tessellation of a closed domain results in a set of nonoverlapping convex polygons covering the convex hull of the points. It is clear that the edge of a Voronoi polygon is equidistant from the two points which it separates, and is thus a segment of the perpendicular bisector of the line joining these two points. The Delaunay triangulation of the given set of points is obtained by joining with straight lines all point pairs whose Voronoi regions have an edge in common. For each triangle formed in this way, there is an associated vertex of the Voronoi diagram which is at the circumcentre of the three points which form the triangle. Thus, each Delaunay triangle contains a unique vertex of the Voronoi diagram, and no other vertex within the Voronoi structure lies within the circle centered at this vertex. Figure 12.2 depicts the Voronoi polygons and the associated Delaunay triangulation.

It is apparent from the definition of a Voronoi polygon that degeneracy problems can arise in the triangulation procedure when

(1) three points of a potential triangle lie on a straight line;
(2) four or more points are cyclic.

These cases are readily eliminated by rejecting or slightly moving the point which causes the degeneracy from its original position.

Incremental Bowyer–Watson Algorithm

The two-dimensional incremental technique, introduced independently by Bowyer (1981) and Watson (1981), triangulates a set of points in accordance with the requirement that the circumcircle through the three vertices of a triangle does not contain any other point. The accomplishment of this technique starts from some Delaunay triangulation which is considered to be an initial triangulation. The initial triangulation commonly consists of a square divided into two triangles which contain the

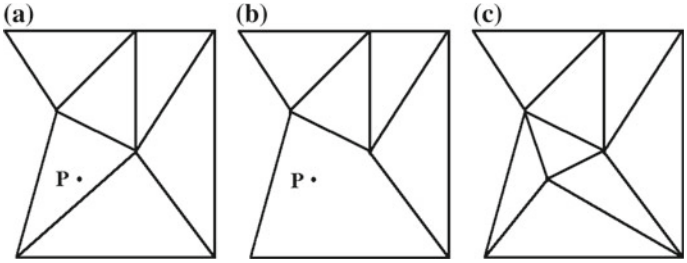


Fig. 12.3 Stages of the planar incremental algorithm

given points. With this starting Delaunay triangulation, a new grid node is chosen from a given set of points or is found in accordance with some user-specified rule to supply new vertices. In order to define the grid cells which contain this point as a vertex, all the cells whose circumcircles enclose the inserted point are identified and removed. The union of the removed cells forms the region which is referred to as the Delaunay or inserting cavity. A new triangulation is then formed by joining the new point to all boundary vertices of the inserting cavity created by the removal of the identified triangles. Figure 12.3 represents the stages of the planar incremental algorithm.

The distinctive feature of the two-dimensional Delaunay triangulations is that all edges of the Delaunay cavity are visible from this inserted point, i.e. each point of the edges can be joined to it by a straight line which lies in the cavity.

Properties of the Planar Delaunay Cavity

In order to prove the fact that all boundary edges of the Voronoi cavity are visible from the introduced point, we consider an edge AB lying on the boundary of the cavity. Let ABC be the triangle with the vertices A, B, and C, which lies in the Delaunay cavity formed by the insertion of the point, denoted by P (Fig. 12.4). It is obvious that all edges of triangle ABC are visible if P lies inside the triangle. Let P

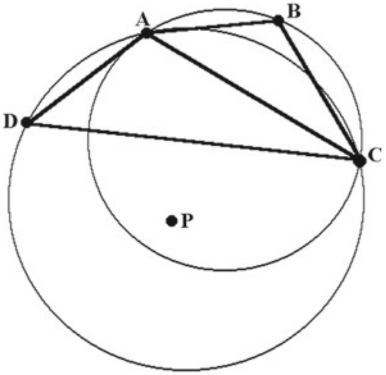


Fig. 12.4 Illustration of the inserted point P and the triangles of the Delaunay cavity

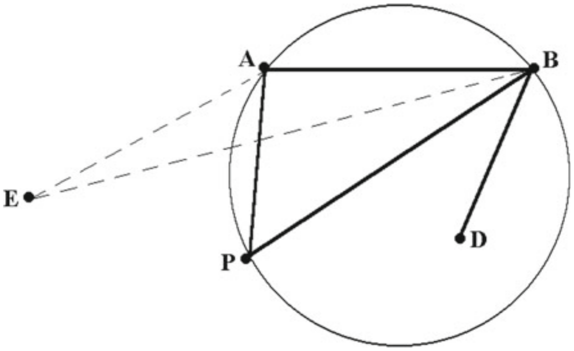
lie outside the triangle. As this triangle lies in the Delaunay cavity, it follows that P lies inside circle ABC. In this case, the quadrilateral whose vertices are the points $ABCP$ is convex. Thus, P has to be visible from edge AB unless we have a situation like the one depicted in Fig. 12.4, in which some triangle ACD separates the edge from P. As triangle ACD belongs to the initial Delaunay triangulation, the vertex D lies outside circle ABC. However, since a chord of a circle subtends equal angles at its circumference, we readily find that P belongs to circle ACD, i.e. the triangle lies inside the Delaunay cavity formed by P. Thus, triangle ACD does not prevent those edges of ABC which are the boundary edges of the cavity from being visible from P. Repeating the argument with the other triangles, the number of which is finite, we come to the conclusion that there are no triangles between the boundary of the Delaunay cavity and P which do not lie in the cavity. Also, we find that the Delaunay cavity is simply connected. We emphasize that these facts are valid if the original triangulation satisfies the Delaunay criterion.

Thus, in accordance with the incremental algorithm, the Delaunay cavity is triangulated by simply connecting the inserted point with each of the nodes of the initial grid that lie on the boundary of the cavity. The union of these triangles with those which lie outside of the cavity (Fig. 12.3c) completes one loop of the incremental grid construction. The subsequent steps are accomplished in the same fashion.

It is apparent that in two dimensions, the creation of these new cells results in a Delaunay triangulation, i.e. the Delaunay criterion is valid for all new triangles. Here, we present a schematic proof of this fact.

Let AB be an edge of the Delaunay cavity formed by the insertion of point P. Suppose that the new triangle ABP does not satisfy the Delaunay criterion. Then, there exists some point D on the same side of AB as P and which lies inside circle ABP (Fig. 12.5). Consider the original triangle that had AB as an edge. There are two possibilities: either ABD is this original triangle or there is another point, say, E, on the cavity boundary lying outside circle ABP. In the former case, P lies outside circle ABD, i.e. triangle ABD does not lie in the Delaunay cavity, and consequently edge AB is not the edge of the cavity, contrary to our assumption. In the latter case, arc ABP lies inside circle ABE. However, this contradicts the assumption that

Fig. 12.5 Illustration of the proof that the Delaunay criterion is satisfied by all new triangles created by the incremental algorithm



the original triangulation was of Delaunay type. Therefore,,,, circle ABP does not contain other points, i.e. the triangle ABP satisfies the Delaunay criterion.

Thus, we find that the planar Bowyer–Watson algorithm is a valid procedure for generating Delaunay triangulations. One more issue that has received attention is that the point placement selected to generate Delaunay triangulations can be used to generate meshes with a good aspect ratio.

Initial Triangulation

Because the mesh points are introduced in a sequential manner, in the initial stages of this construction, an extremely coarse grid containing a small subset of the total number of mesh points and consisting of a small number of very large triangles can be chosen. For example, for generating grids in general two-dimensional domains, an initial triangulation can be formed by dividing a square lying in the domain or containing it into two triangles. Then, interior and boundary points are successively added to build successive triangulations until the necessary requirements of domain approximation are observed.

It is desirable to make the initial triangulation boundary-conforming, i.e. all boundary edges are included in the triangulation. One natural way is to triangulate initially only the prescribed boundary nodes, by means of the Bowyer–Watson algorithm. Since the Delaunay triangulation of a given set of points is a unique construction, there is no guarantee that the triangulation built through the boundary points will be boundary-conforming. However, through repeated insertion of new mesh points at the midpoints of the missing boundary edges, a boundary-conforming triangulation may be obtained. Another way to maintain boundary integrity is obtained by rejecting any point that would result in breaking boundary connectivity.

Diagonal-Swapping Algorithm

Diagonal swapping is a topological operation in which the diagonal of a quadrilateral formed by two adjacent triangles is swapped to the other position to improve the overall quality of the two triangles. The diagonal-swapping algorithm makes use of the equiangular property of a Delaunay-type triangulation, which states that the minimum angle of each triangle in the mesh in maximized.

Assuming we have some triangulation of a given set of points, the swapping algorithm transforms it into a Delaunay triangulation by repeatedly swapping the positions of the edges in the mesh in accordance with the equiangular property. For this purpose, each pair of triangles which constitutes a convex quadrilateral is considered. This quadrilateral produces two of the required triangles when one takes the diagonal which maximizes the minimum of the six interior angles of the quadrilaterals, as shown in Fig. 12.6. Each time an edge swap is performed, the triangulation becomes more equiangular. The end of the process results in the most equiangular (the Delaunay) triangulation.

This technique based on the Delaunay criterion retriangulates a given triangulation in a unique way, such that the minimum angle of each triangle in the mesh is maximized. This has the advantage that the resulting meshes are optimal for the

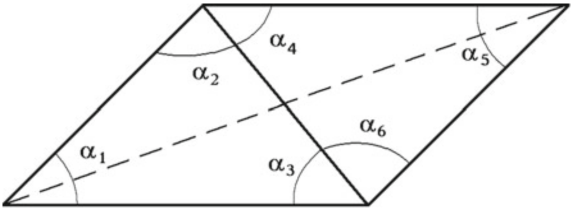


Fig. 12.6 The triangulation which maximizes the minimum angle. The dashed line indicates a possible original triangulation

given point distribution, in that they do not usually contain many extremely skewed cells.

12.2.5 Constrained Form of Delaunay Triangulation

One way to ensure that the boundary triangulation remains intact in the process of retriangulation by inserting new points is to use a constrained version of the Delaunay triangulation algorithm that does not violate the point connections made near the boundary.

Principal Component

For the purpose of generating a constrained two-dimensional triangulation, we consider the modernized Bowyer–Watson algorithm for an arbitrary triangulation T that may not satisfy the Delaunay criterion. Let P be a new, introduced point. The Delaunay cavity is the area constituted by all triangles whose circumcircles contain P. Let this be denoted by $\Gamma(P)$.

An important fact is that the Delaunay cavity created by the introduction of the point P contains no points other than P in its interior. In order to show this, we consider a point A in the triangulation T that is a vertex of at least one triangle in $\Gamma(P)$. If there is a triangle $S \notin \Gamma(P)$ that has A as a vertex, then the point A is not an interior point of $\Gamma(P)$. Thus, we need to show that there exists such a triangle. Let $\{S_i\}$ be the set of all triangles that have A as a vertex, and let C_i be the circumcircle associated with triangle S_i. Now $S_i \in \Gamma(P)$ if and only if the new point P lies inside C_i. Thus, for vertex A to be an interior point of $\Gamma(P)$, point P must lie inside $\bigcap C_i$. However, if the point A is an interior point of $\Gamma(P)$, then the interior of $\bigcap C_i$ is empty, since the vertex A is the only point that lies on all the circles of $\{S_i\}$. Thus, at least one triangle of $\{S_i\}$ does not lie in $\Gamma(P)$, and hence the vertex A is not an interior point of $\Gamma(P)$.

In the case of a general triangulation, the cavity $\Gamma(P)$ need no longer be simply connected. For the purpose of retriangulation, we consider the maximal simply connected region of the cavity that contains the new point P. This region is called the principal component of the Delaunay cavity and is designated by Γ_P.

It is apparent that the principal component possesses the property that all its boundary edges are visible from P. To prove this, we first note that Γ_P is not empty,

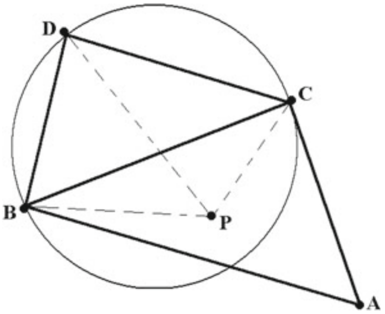


Fig. 12.7 Illustration of the principal component

since it includes the triangle containing P. Let this be the triangle ABC (Fig. 12.7). Now consider all neighboring triangles sharing a common edge with the triangle ABC. In particular, let triangle BCD lie in $\varGamma_P$. Point P must, therefore, lie inside circle BCD. As points P, B, C, and D define a convex quadrilateral, all edges of this quadrilateral are visible from P. Continuing this process by means of a tree search through all triangles in $\varGamma_P$, we clearly see that all edges of $\varGamma_P$ are visible from P.

Formulation of the Constrained Triangulation

Now we can formulate the generation of a constrained planar Delaunay triangulation developed by Baker (1989).

We assume that certain triangles of a triangulation T are fixed, in particular, those adjacent to the boundary. Let this subset of T be denoted by $\overline{T}$. The triangles from $\overline{T}$ do not participate in the building of any Delaunay cavity, i.e. if the cavity created by the introduction of a new point contains one or more of the fixed triangles, we restrict the reconnections to the part of the cavity that does not contain any fixed triangle. Let $\varUpsilon(P)$ be this part of the cavity, i.e. $\varUpsilon(P) = \varGamma(P) - \overline{T}$. By $\varUpsilon_P$, we denote the maximal simply connected region of $\varUpsilon(P)$ that contains P. In analogy with $\varGamma_P$, we call the region $\varUpsilon_P$ the principal component of $\varUpsilon(P)$. It is clear that the principal component $\varUpsilon_P$ exists only if P does not lie inside any of the triangles belonging to the collection $\overline{T}$ of the fixed triangles.

It is apparent that the boundary edges of the principal component $\varUpsilon_P$ are visible from P. As the analogous fact has been proved for $\varGamma_P$, we can restrict our consideration to the case $\varGamma_P \bigcap \overline{T} \neq \emptyset$. Let the edges of the principal component $\varGamma_P$ be given by $\overline{A_1, A_2}$, $\overline{A_2, A_3}, \dots, \overline{A_{n-1}, A_n}$, $\overline{A_n, A_1}$, where $\{A_i\} i = 1, n$ are the vertices on the boundary of $\varGamma_P$. These edges, and consequently the vertices A_i, are visible from P. The subcavity obtained by removing one of the triangles from $\varGamma_P$ contains at most three new edges. These internal edges lie wholly inside the cavity $\varGamma_P$ and divide $\varGamma_P$ into disjoint polygonal regions. The principal component $\varUpsilon_P$ is the polygonal region that contains the point P, and this polygon is made up of one of these internal cavity edges and other edges which come from the cavity boundary. The vertices of the polygon containing P must, therefore, remain visible from P. Hence, all edges of the polygon are visible from P. By repeating this argument with other triangles removed from Γ_P, we conclude that the boundary edges of Υ_P are visible from P.

Now the vertices of Υ_P can be connected with P, thus building the constrained retriangulation. This retriangulation keeps the fixed triangles of $\overline{T}$ intact.

Boundary-Conforming Triangulation

A key requirement of a mesh generation procedure is to ensure that the mesh is boundary-conforming, i.e. the edges of the assembly of triangles conform to the boundary curve. The procedure of constrained triangulation allows one to keep a subset of the boundary triangles, built from the edges forming the boundary, intact. These boundary triangles can be generated by any one of the suitable procedures. Thus, the resulting triangulation will be boundary-conforming and its interior triangles obey the Delaunay criterion.

Another approach developed by Weatherill and Hassan (1994) to applying the Delaunay criterion to generate boundary-conforming grids consists in recovering the boundary edges which are missing during the process of Delaunay triangulation and then deleting all triangles that lie outside the domain.

12.2.6 Point Insertion Strategies

The Bowyer–Watson algorithm proceeds by sequentially inserting a point inside the domain at selected sites and reconstructing the triangulation so as to include new points. This subsection presents two approaches to sequential point insertion which provide a refinement of planar Delaunay triangulations. In both cases, bounds on some measures of grid quality, such as the minimum angle, the ratio of maximum to minimum edge length, and the ratio of circumradius to inradius, are estimated.

Point Placement at the Circumcenter of The Maximum Triangle

One simple but effective approach consists in placing a new point at the circumcenter of the cell with the largest circumradius and iterating this process until the maximum circumradius is less than some prescribed threshold. In this way, by eliminating bad triangles, the quality of the grid is improved at every new point insertion, terminating with a grid formed only by suitable triangles. In this subsection, it will be shown that the Bowyer–Watson incremental algorithm together with point insertion at the circumcenters of maximal triangles will lead to a triangulation with a guaranteed level of triangle quality.

Unconstrained Triangulation

Let $\{T_n\}$, $n = 0, 1, \ldots,$ be a sequence of Delaunay triangulations built by the repeated application of the Bowyer–Watson algorithm with point insertion at the circumcenter of the maximal triangle. By the maximal triangle of a triangulation, we mean the triangle with the maximum value of its circumradius. We assume that the initial Delaunay triangulation T_0 conforms to a prescribed set of boundary edges.

Now let $l_n,\ L_n,\ n = 0, 1, \ldots,$ be the minimum and maximum edge lengths, respectively, of T_n, and let R_n be the radius of the maximal triangle of T_n. Furthermore, for any triangle S, we denote its circumradius by R_S and its inradius by r_S. Thus, $R_n = \max\{R_S,\ S \in T_n\}$. We have the following relations:

(1) $R_{i+1} \leq R_i$;
(2) when $R_{n-1} \geq l_0$, then $l_n = l_0$, and when $R_{n-1} < l_0$, then $l_n = R_{n-1}$;
(3) when $R_n \leq l_0$, then $L_n/l_n \leq 2$, $\theta \geq 30°$ for all angles of the triangulation T_n, and $\min R_S/r_S \leq 2 + 4\sqrt{3}$ for all triangles S of T_n.

To prove the first relation, we consider an edge e_n of the Delaunay cavity of the triangulation T_n formed by an inserted point P. There exist triangles S_1 and S_2 in T_n which share the common edge e_n, such that S_1 lies inside while S_2 lies outside the Delaunay cavity. Let S_1 be defined by the points $A,\ B$, and C and S_2 be defined by the points $A,\ B$, and D. Then, edge e_n is the line segment $\overline{AB}$. Since P lies outside circle ABD, P lies on the same side of e_n as C. If the center of circle ABP lies on the same side of e_n as D, then angle APB is obtuse and, consequently, the circumradius of triangle ABP is smaller than the circumradius of triangle ABD. We denote these circumradii by R_{ABP} and R_{ABD}, respectively.

If the center of circle ABP lies on the same side of e_n as C, then the angle θ_1 subtended by chord AB at C is less than the angle θ_2 subtended at P. Since the centers of circles ABP and ABC lie on the same side of $\overline{AB}$ as points C and P, it follows that $\theta_1 < \pi/2$ and $\theta_2 < \pi/2$. If the length of chord $\overline{AB}$ is l, then

$$R_{ABP} = \frac{l}{2\sin\theta_2} < \frac{l}{2\sin\theta_1} = R_{ABC}\ ,$$

where R_{ABC} is the circumradius of ABC. Thus, we obtain

$$R_{ABP} < R_{ABD} \quad \text{and} \quad R_{ABP} < R_{ABC}\ .$$

Since this is true for all edges of the Delaunay cavity, we obtain the proof of the first relation, that the maximum circumradius R_n decreases, i.e. $R_{n+1} \leq R_n$, with strict inequality if there is only one triangle with the maximum radius R_n. As there can be only a limited number of maximal triangles in T_n, after several applications of the procedure, we obtain $R_{n+k} < R_n$.

It follows that the maximum radius can be reduced to any required size after a sufficiently large number of iterations. When R_n falls below the value of l_0, so that $l_{n+1} = R_n$, we obtain the following obvious inequality:

$$L_{n+1} \leq 2R_{n+1} \leq 2R_n = 2l_{n+1}\ . \tag{12.1}$$

It is evident that repeated point insertion at the circumcenter reduces the value $\lambda = L_n/l_n$ to a value no greater than 2. The upper bound of 2 for λ is achieved when $R_n \leq l_0$. Let $\theta_{\min}$ be the minimum angle. We have

$$\sin\theta_{\min} \geq \frac{l_{n+1}}{2R_{n+1}} , \tag{12.2}$$

with equality if the minimum edge length of any maximal triangle is equal to l_{n+1}, the minimum edge length for the triangulation T_{n+1}. From the inequalities (12.1) and (12.2), we obtain

$$\sin\theta_{\min} \geq \frac{l_{n+1}}{2R_{n+1}} = \frac{R_n}{2R_{n+1}} \geq \frac{1}{2} , \tag{12.3}$$

so that

$$\theta_{\min} \geq \pi/6 .$$

For each triangle, the quantity $\mu = R/r$, where R is the circumradius and r is the inradius, is a characteristic of cell deformity. The maximum value of μ occurs for an isosceles triangle with an angle between sides of $\theta_{\min}$ and assumes the value

$$\mu_{\max} = \frac{1}{2\cos\theta_{\min}(1-\cos\theta_{\min})} .$$

From (12.3), we obtain

$$\mu \leq 2 + 4/\sqrt{3}$$

after a sufficient number of retriangulations with the insertion of new points at the circumcenters of maximal triangles.

These considerations prove the properties (2) and (3) stated above.

Generalized Choice of the Insertion Triangles

In the approach considered, a new point is inserted at the circumcentre of the largest triangle. The choice of the insertion triangle, namely the triangle where the point is inserted, can be formulated in accordance with more general principles.

One simple formulation is based on the specification of a function $f(\boldsymbol{x})$ which prescribes a measure of grid size or quality, say the radius of the circumscribed circle, at the point $\boldsymbol{x}$. The actual expression for $f(\boldsymbol{x})$ can be obtained by interpolating prescribed nodal values over a convenient background mesh. The function $f(\boldsymbol{x})$ defines a quantity $\alpha(S)$ for each triangle S:

$$\alpha(S) = \frac{R_S}{f(Q_S)} ,$$

where Q_S is the position of the centre of the circle circumscribed around the triangle S. The largest value of $\alpha(S)$ determines the choice of the insertion triangle S. By repeatedly inserting the new point at the circumcenters of such triangles, it is possible eventually to reach a mesh in which $\max_S \alpha(S) < 1$.

Voronoi-Segment Point Insertion

The second approach proposed by Rebay (1993) to placing a new point consists in inserting the point along a segment of the Dirichlet tessellation. In contrast to the first approach, in which the position of the inserted point is predetermined, and the required cell size is reached after a number of iterations, this technique provides an opportunity to generate one or possibly several new triangles having, from the very beginning, the size prescribed for the final grid. This is achieved by choosing a suitable position for point placement in the Dirichlet tessellation, between a triangle whose circumradius falls below the required value and a neighboring triangle whose circumradius is still too large. This point insertion results in almost equilateral triangles over most of the interior of the domain.

Formulation of the Algorithm

At each stage of the process of generating the triangulations $T_n,\ n = 1, 2, \ldots,$ the triangles of T_n are divided into two groups, which are referred to as the groups of accepted (small enough) and nonaccepted (too large) triangles, respectively. In most cases, the accepted triangles are the boundary triangles and those whose circumradii are below $3/2$ times the prescribed threshold. The remaining triangles constitute the group of nonaccepted triangles.

The algorithm proceeds by always considering a maximal nonaccepted triangle which borders one of the accepted triangles (Fig. 12.8). Let ABC be the accepted triangle and ADB the nonaccepted triangle. The Voronoi segment connecting the circumcenters of these triangles is the interval EF which is perpendicular to the common edge AB and divides it into two equal parts. In the algorithm, a new point X is inserted on the Voronoi segment edge EF in a position chosen so that the triangle

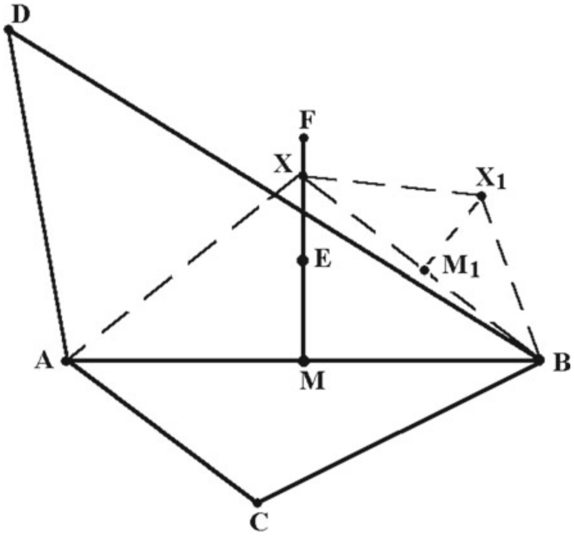


Fig. 12.8 Voronoi-segment point insertion

formed by connecting X with A and B has the prescribed size. This point is inserted into the interval between the midpoint M of the common edge and the circumcenter F of the nonaccepted triangle ADB.

Let p be one half the length of edge AB, and q the length of FM. As point F is the circumcenter of the triangle ADB, we find that $q \geq p$. Let f_M be the prescribed threshold value for the circumradius at the point M. It may seem that we can locate the new point X on segment EF at the intersection of EF with the circle that passes through points A and B and has a radius equal to f_M. However, it might happen that this exact value f_M for the circumradius is not appropriate, since any circle through A and B has a radius $\rho \geq p/2$. Furthermore, a real intersection point X exists only for circles having a radius ρ smaller than that of the circle passing through AB and F, i.e. $\rho \leq (p^2 + q^2)/2q$. For these reasons, the circumradius for the triangle AXB is defined by the equation

$$R_{AXB} = \min\left[\max(f_M, p)\ ,\quad \frac{p^2 + q^2}{2q}\right] . \tag{12.4}$$

Since

$$\frac{p^2 + q^2}{2q} = \frac{(p - q)^2 + 2pq}{2q} \geq p\ ,$$

we find that $R_{AXB} \geq p$. In accordance with the algorithm, the new point X will lie on the interval EF between M and F at a distance

$$d = R_{AXB} + \sqrt{(R_{AXB})^2 - p^2} \tag{12.5}$$

from the point M.

Properties of the Triangulation

The condition

$$R_{AXP} \leq \frac{p^2 + q^2}{2q}$$

and (12.5) ensure that $d \leq q$. We also have, from (12.5), that $d \geq p$. Angle AXB is a right angle when $d = p$, and it decreases as d increases.

If the accepted triangle ABC is equilateral, then angle AFB must be no greater than $2\pi/3$, since otherwise the Delaunay triangulation would have given rise to an edge connecting C to F.

At the first stage, we expect $p \ll q$. Recall that the threshold of f_M is such that $f_M < p < 3f_M/2$. It follows that $f_M < p \leq (p^2 + q^2)/2q$, and hence $d = p$ and $R_{AXB} = p$. Thus, triangle AXB has a right angle at vertex X. Since $2 < 3f_M/2$, triangle AXB will be tagged as accepted and each segment AX and XB will be a candidate for the next accepted triangle, built in the same way as AXB. Now we denote the quantity p, equal to one half the length of the accepted edge of the ith iteration, by p_i, and thus $p_1 = p_0/\sqrt{2}$. Analogously, we use d_i, R_i, and M_i at the ith iteration of the procedure. It turns out that on repeating the procedure, p_i and d_i show the following behavior:

$$p_i \to \sqrt{3} f_{M_i}/2 \ , \qquad d_i \to 3 f_{M_i}/2 \ ,$$

i.e. the generated triangles tend to become equilateral, with circumradius f_{M_i}. To show this, let

$$f_{M_n} = \Big(\frac{2}{\sqrt{3}} + \epsilon_n\Big) p_n = \Big(\frac{2}{\sqrt{3}} + \epsilon_{n+1}\Big) p_{n+1} \ . \tag{12.6}$$

If $|\epsilon_n|$ is sufficiently small, we obtain $p_n < f_{M_n}$ so that $R_n = f_{M_n}$ and, from (12.5),

$$d_n = f_{M_n} + \sqrt{f_{M_n}^2 - p_n^2} \ .$$

Furthermore, we have

$$4p_{n+1}^2 = p_n^2 + d_n^2 = 2\Big(f_{M_n}^2 + f_{M_n}\sqrt{f_{M_n}^2 - p_n^2}\Big) \ .$$

Thus,

$$\frac{p_{n+1}^2}{p_n^2} = \frac{1}{2}\frac{f_{M_n}^2}{p_n^2} + \frac{1}{2}\frac{f_{M_n}}{p_n}\sqrt{\frac{f_{M_n}^2}{p_n^2} - 1} \ .$$

Using (12.6), we obtain

$$\frac{p_{n+1}^2}{p_n^2} = \frac{1}{2}\Big(\frac{2}{\sqrt{3}} + \epsilon_n\Big)^2 + \frac{1}{2}\Big(\frac{2}{\sqrt{3}} + \epsilon_n\Big)\sqrt{\Big(\frac{2}{\sqrt{3}} + \epsilon_n\Big)^2 - 1} \ ,$$

which results in

$$\frac{p_{n+1}}{p_n} = 1 + \frac{3}{4}\sqrt{3} + \epsilon_n + O(\epsilon_n^2) \ .$$

From (12.6), we also have

$$\frac{p_{n+1}}{p_n} = \frac{2/\sqrt{3} + \epsilon_n}{2/\sqrt{3} + \epsilon_{n+1}} \ .$$

Comparing the last two equations and neglecting terms $O(\epsilon_n^2)$, we find that

$$\epsilon_{n+1} \simeq -\epsilon_{n/2} \ .$$

Thus, for $|\epsilon_n|$ sufficiently small, the algorithm ensures that $\epsilon_n \to 0$ and

$$p_n \to \sqrt{3} f_{M_n}/2 .$$

Therefore, it can be expected that a large number of the interior triangles will be nearly equilateral. Close to the boundary, there may be isosceles right-angled triangles, and in regions where the boundary has large curvature, there may be some obtuse triangles. A maximum angle of 120° and minimum angle of 30° may be realized by an obtuse triangle formed when the vertex D of a nonaccepted triangle is sufficiently close to an active edge.

In analogy with the first approach to inserting new points, the choice of the triangle into which the new point is inserted can be modified by introducing a quality measure function $f(\boldsymbol{x})$ and a corresponding control quantity $\alpha(S)$.

12.2.7 Surface Delaunay Triangulation

A surface Delaunay triangulation is defined by analogy with the planar Delaunay triangulation.

Let P_i be the vertices of the surface triangulation T. A triangle S from T satisfies the Delaunay criterion if the interior of the circumsphere through the vertices of S and centered on the plane formed by S does not contain any points. If all triangles satisfy the Delaunay criterion, then the triangulation T is called a surface Delaunay triangulation.

In practice, all methods for planar Delaunay triangulations are readily modified and extended for a surface Delaunay triangulation taking into account various surface geometric characteristics (see Frey and George (2008) and Lo (2015)).

12.2.8 Three-Dimensional Delaunay Triangulation

In three dimensions, the network of the Delaunay triangulation is obtained by joining the vertices of the Voronoi polyhedrons that have a common face. Each vertex of a Voronoi polyhedron is the circumcenter of a sphere that passes through four points which form a tetrahedron and no other point in the construction can lie within the sphere.

Unconstrained Technique

The most popular three-dimensional algorithm providing a Delaunay structure is the one based on the Bowyer–Watson sequential process: each point of the grid is introduced into an existing Delaunay triangulation, which is broken and then reconnected to form a new Delaunay triangulation.

In general, the algorithm follows the same steps as in the two-dimensional construction described above. It starts with an initial Delaunay triangulation formed by a supertetrahedron or supercube, partitioned into five tetrahedrons which contain all other points. The remaining points which comprise the mesh to be triangulated are introduced one at a time, and the Bowyer–Watson algorithm is applied to create the Delaunay cavity and the corresponding retriangulation after each point insertion.

An important feature of a mesh generation procedure is its ability to produce a boundary-conforming mesh, i.e. the triangular faces of the assembly of tetrahedrons conform to the boundary surface. Unfortunately, the unconstrained technique does not guarantee that the boundary faces will be contained within such a triangulation. Thus, a modified procedure must be introduced to ensure that the resulting triangulation is boundary-conforming.

Constrained Triangulation

The purpose of the constrained Delaunay triangulation is to generate a triangulation which preserves the connections imposed on the boundary points. The three-dimensional constrained triangulation is carried out in the same way as for two-dimensional triangulations.

In the first approach, the tetrahedrons whose faces constitute the boundary surface are fixed during the process of retriangulation. These boundary tetrahedrons are generated in the first step of triangulation. The next steps include the insertion of a point, the definition of a star-shaped cavity containing the point, and retriangulation of the cavity. The resulting grid is boundary-conforming and its interior subtriangulation is a Delaunay triangulation.

The second approach to the constrained triangulation of a domain developed by Weatherill and Hassan (1994) starts with inputting the boundary points and boundary point connectivities of the faces of the boundary triangulation. After performing a Delaunay triangulation of the boundary points, a new Delaunay triangulation is built by inserting interior points and applying the Bowyer–Watson algorithm. After this, the tetrahedrons intersecting the boundary surface are transformed to recover the boundary triangulation. If a boundary face is not present in the new Delaunay triangulation, this is due to the fact that edges and faces of the tetrahedrons of the Delaunay triangulation intersect this face. Since the face is formed from three edges, it is necessary first to recover the face edges and then the face. This is achieved by first finding the tetrahedrons which are intersected by the face edges. There is a fixed combination of possible standard intersections of each tetrahedron by any mixed boundary edge, which allows one to perform direct transformations to recover the edge. Having established the intersection types, these tetrahedrons are then locally transformed into new tetrahedrons so that the required edges are present. A similar procedure then follows to recover the boundary faces.

12.3 Advancing-Front Methods

Advancing-front techniques extend the grid into the region in the form of marching layers, starting from the boundary and proceeding until the whole region has been covered with grid cells. The region separating the part of the domain already meshed from those that are still unmeshed is referred to as a front. Advancing-front techniques need some initial triangulation of the boundaries of the geometry, and this triangulation forms the initial front. The name of this class of methods refers to a strategy that consists of creating the mesh sequentially, element by element, creating new points and connecting them with previously created elements, thus marching into as-yet-unmeshed space and sweeping a front across the domain. The marching process includes the construction of a new simplex, which is built by connecting either some appropriate points on the front or some inserted new point with the vertices of a suitable face on the front. The process stops when the front is empty, i.e. when the domain is entirely meshed.

The advancing-front approaches offer the advantages of high-quality point placement and integrity of the boundary. The efficiency of the grid-marching process largely depends on the arrangement of grid points in the front, especially at sharp corners. A new grid point is placed at a position which is determined so as to result in a simplex with prescribed optimal quality features. In some approaches, the grid points are positioned along a set of predetermined vectors. To ensure a good grid quality and to facilitate the advancing process, these vectors are commonly determined once at each layer mesh point by simply averaging the normal vectors of the faces sharing the point and then smoothing the vectors. Other approaches to selecting new points for moving the front use the insertion techniques applied in the Delaunay triangulations described above.

The fronts continue to advance until either

(1) opposite fronts approach to within a local cell size; or
(2) certain grid quality criteria are locally satisfied.

Grid quality measures which are to be observed in the process of grid generation by means of the advancing-front method include the cell spacing and sizes of angles. The desired mesh spacings and other gridding preferences in the region are commonly specified by calculations on a background grid.

12.3.1 Procedure of Advancing-Front Method

In order to generate cells with acceptable angles and lengths of edges by a marching process, the advancing-front concept inherently requires a preliminary specification of local grid spacing and directionality at every point of the computational mesh. The spacing is prescribed by defining three (two in two dimensions) orthogonal directions together with some length scale for each direction. The directions and length scales are commonly determined from background information, in particular, by carrying out computations on a coarse grid and interpolating the data.

The advancing-front procedure proceeds by first listing all faces which constitute the front and then selecting an appropriate face (edge in two dimensions) on the front. The operation of the selection is very important, since the quality of the final grid may by affected by the choice. According to a common rule, the face is selected where the grid spacing is required to be the smallest. A collection of vertices on the front which are appropriate for connection to the vertices of the selected face to form a tetrahedron (triangle in two dimensions) is searched. The collection may be formed by the vertices which lie inside a sphere centered at the barycenter of the face, with an appropriate radius based upon the height of a unit equilateral tetrahedron. A new point is also created which is consistent with the ideal position determined from the background information about grid spacing and directionality. The selected vertices and the new point are ordered according to their distance from the barycenter of the selected face. Each sequential tetrahedron formed by the face and the ordered points is then checked to find out whether it intersects any face in the front. The first point which satisfies the test and gives a tetrahedron of good quality is chosen as the fourth vertex for the new tetrahedron. The current triangle is then removed from the list of front faces, since it is now obscured by the new tetrahedron. This process continues until there are no more faces in the list of front faces.

In many cases, the use of the background mesh to define the local grid spacing can be replaced by sources in the form of points, lines, and surfaces.

One of the advantages of such a procedure is that all operations are performed locally, on neighboring faces only. Additionally, boundary integrity is observed, since the boundary triangulation constitutes the initial front.

The disadvantages of the advancing-front approach relate mainly to the phase in which a local direction and length scale are determined and to the checking phase for ensuring the acceptability of a new tetrahedron.

12.3.2 Strategies for Selecting Out-of-Front Vertices

One of the critical items of advancing-front methods is the placement of new points. Upon generating a new simplex, a point is placed at a position which is determined so as to result in the required shape and size of the new simplex. The parameters which define the desirable cell at each domain position are specified by a function which is determined a priori or found in the process of computation.

In one approach, the new point is placed along a line which is orthogonal to a chosen face on the front and passes through its circumcenter. This placement is aimed at the creation of a new simplex whose boundary contains the chosen face.

If the simplex generated with the new point results in a crossover with the front, it is discarded. Alternately, if the new point is located very close to a vertex on the front, it is replaced by this vertex in order to avoid the appearance of a cell with a very small edge at some later stage.

Another approach, generally applied in two dimensions, takes into account a vertex on the front and the angle at which the edges cross at this point. The point is created with the aim of making the angles in the new triangles as near to 60° as possible. In particular, a very large angle between the edges is bisected or even trisected. On the other hand, if the vertex has a small interior angle, the two adjacent vertices on the front are connected. This approach can be extended to three dimensions by analyzing a dihedral angle at the front.

12.3.3 Grid Adaptation

The frontal approach is well suited to generating adaptive grids near the boundary segments, where the grid cells are commonly required to be highly stretched.

Highly stretched grid cells begin forming individually from the boundary and march into the domain. However, unlike the conventional procedure in which cells are added in no systematic sequence, the construction of a stretched grid needs to be performed by advancing one layer of cells at a time, with the minimum congestion of the front and a uniform distribution of stretched cells. The new points are positioned along a set of predetermined vectors in accordance with the value of a stretching function. The criterion by which the points are evaluated has a significant impact on the grid quality and the marching process. Because of the requirement of a high aspect ratio of cells in the boundary layer, the conventional criteria based on the cell angles are not appropriate for building highly stretched cells.

In a criterion based on a spring analogy, the points forming a new layer are assumed to be connected to the end points of the face by tension springs. Among these points, the one with the smallest spring force is considered the most suitable for forming the new cell, and consequently for changing the front boundary. The spring concept allows one to indicate when an opposing front is very close to the new location, namely, when an existing point on the front has the smallest spring force. The adaptive advancing process terminates on a front face when the local grid characteristics on the front, influenced by the stretching function, no longer match those determined by the background grid in that location. When the proximity and/or grid quality criteria are satisfied on all faces of the front, the process switches from an advancing-layers method to the conventional advancing-front method to form regular isotropic cells in the rest of the domain.

12.3.4 Advancing-Front Delaunay Triangulation

A combination of the advancing-front approach and the Delaunay concept gives rise to the advancing-front Delaunay methods.

If the boundary of a domain is triangulated and a set of points to be triangulated is given in the interior of the domain, then the advancing-front Delaunay triangulation is carried out by forming the cells adjoining the front in accordance with the empty-circumcircle property.

The procedure for the triangulation can be outlined as follows. A face on the front is chosen, and a new simplex is tentatively built by joining the vertices of the face to an arbitrary point on the front, in the interior of the domain with regard to the front. If this simplex contains any points within its circumcircle, it is not added to the triangulation. By checking all points, the appropriate vertex which produces a simplex containing no points interior to its circumcircle is eventually found. The simplex formed through this vertex is accepted and the front is advanced.

Another algorithm is based on the strategy of placing new points ahead of the front and triangulating them according to the Delaunay criterion.

12.4 Meshing by Quadtree-Octree Decomposition

Quadtree-octree meshing is based on the idea of partitioning a domain in a progressive manner so as to produce cells of size compatible with the node spacing requirement. The use of quadtree decomposition for two-dimensional mesh generation was developed in the 1980s by Shephard et al. (1988).

In this approach, applied to mesh generation, the n-dimensional domain to be gridded is first enclosed in a bounding root box (an n-dimensional parallelepiped) which is approximated with a union of disjoint and variably sized cells whose union constitutes the final mesh of the domain. The cells are obtained from a recursive refinement of the root parallelepiped. The current cell is subdivided into four equally sized cells in a two-dimensional case and into eight equally sized cells in a three-dimensional case. The stopping criterion used to subdivide a cell can be based on the local geometric properties of the boundary of the domain (e.g. the local curvature of the boundary) or user defined level of refinement.

The set of cells composes the tree structure associated with the spatial decomposition. At each stage of the tree construction, each cell of the tree is analyzed and refined into 2^n (with n the space dimension) equally sized cells, based on a specific criterion. The level of a cell corresponds to its depth in the related tree (i.e. the number of subdivisions required to reach a cell of this size). The bounding box is at level 0. The depth of the tree corresponds to the maximum level of subdivision.

12.5 Three-Dimensional Prismatic Grid Generation

The use of prismatic cells is justified by the fact that the requirement of high aspect ratio can be achieved without reducing the values of the angles between the cell edges.

The procedure for generating a prismatic grid begins by triangulating the boundary surface of a domain. The next stage in the procedure computes a quasinormal direction at each node of the surface triangulation. Then, the initial surface is shifted along these quasinormal directions by a specified distance d. This gives the first layer of prismatic cells. This shifting process is repeated a number of times using suitable values of d at each stage and either the same or newly computed normal directions. The value of the quantity d can be chosen in the form of any of the stretching functions described in Chap. 4.

The efficiency of the algorithm is essentially dependent on the choice of quasinormal directions. The generation of the quasinormals is carried out in three stages, depending on a position of the vertices.

(1) Normals are first computed at the vertices which lie on the corners of the boundary. These are calculated as the angle-weighted average of the adjacent surface normals. The angle used is the one between the two edges adjacent to the boundary surface and meeting at the corner.
(2) Normals at grid points on the geometrical edges of the boundary surface are computed. These normals are the average of the two adjacent surface normals.
(3) Finally, the normals at grid nodes on the boundary surfaces are calculated.

12.6 Comments

Unstructured grid methods were originally developed in solid mechanics. The paper by Field (1995) reviews some early techniques for unstructured mesh generation that rely on solid modelling. An informal survey that illustrates the wide range of unstructured mesh generation was conducted by Owen (1998) and described in the handbook of grid generation edited by Thompson et al. (1999).

Though unstructured technology deals chiefly with tetrahedral (triangular in two dimensions) elements, some approaches rely on hexahedrons (or quadrilaterals) for the decomposition of arbitrary domains. Recent results have been presented by Tam and Armstrong (1991) and Blacker and Stephenson (1991).

Properties of n-dimensional triangulations were reviewed by Lawson (1986). The relations between the numbers of faces were proved in the monograph by Henle (1979) and in the papers by Steinitz (1922), Klee (1964), Lee (1976).

The Delaunay triangulation and Voronoi diagram were originally formulated in the papers of Delaunay (1934, 1947), Voronoi (1908), respectively. Algorithms for computing Voronoi diagrams have been developed by Green and Sibson (1978), Brostow et al. (1978), Finney (1979), Bowyer (1981), Watson (1981), Tanemura et al. (1983), Sloan and Houlsby (1984), Fortune (1985) and Zhou et al. (1990). Results of studies of geometrical aspects of Delaunay triangulations and their dual Voronoi diagrams were presented in the monographs by Edelsbrunner (1987), Du and Hwang (1992), Okabe et al. (1992), Preparata and Shamos (1985). Proofs of the properties of planar Delaunay triangulations were given by Guibas and Stolfi (1985) and by Baker (1987, 1989).

A technique for creating the Delaunay triangulation of an a priori given set of points was proposed by Tanemura et al. (1983). The incremental two-dimensional Delaunay triangulation which starts with an initial triangulation was developed by Bowyer (1981), Watson (1981). Watson has also shown the visibility of the edges of the cavity associated with the inserted point. Having demonstrated that the Delaunay criterion is equivalent to the equiangular property, Sibson (1978) devised and later Lee and Schachter (1980) investigated a diagonal-swapping algorithm for generating a Delaunay triangulation by using the equiangular property.

A novel approach, based on the aspect ratio and cell area of the current triangles, to the generation of points as the Delaunay triangulation proceeds was developed by Holmes and Snyder (1988). In their approach, a new point is introduced into the existing triangulation at the Voronoi vertex corresponding to the worst triangle. Ruppert (1992), Chew (1993) have shown that in the planar case, the procedure leads to a Delaunay triangulation with a minimum-angle bound of 30 degrees. An alternative procedure of inserting the new point on a Voronoi segment was proposed by Rebay (1993). A modification of the Rebay technique was made by Baker (1994). Haman et al. (1994) inserted points into a starting Delaunay grid in accordance with the boundary curvature and distance from the boundary, while Anderson (1994) added nodes while taking into account cell aspect ratio and proximity to boundary surfaces.

Approaches to the generation of boundary-conforming triangulations based upon the Delaunay criterion have been proposed by Lee (1978), Lee and Lin (1986), Baker (1989), Chew (1989), Cline and Renka (1990), George et al. (1990), Weatherill (1990), George and Hermeline (1992), Field and Nehl (1992), Hazlewood (1993), Weatherill and Hassan (1994). All techniques and methods considered in the present chapter for proving the results associated with the constrained Delaunay triangulation were described on the basis of papers by Weatherill (1988), Baker (1989, 1994), Mavriplis (1990), Rebay (1993), Weatherill and Hassan (1994).

Further development of unstructured grid techniques based on the Delaunay criterion and aimed at the solution of three-dimensional problems has been performed by Cavendish et al. (1985), Shenton and Cendes (1985), Perronet (1988), Baker (1987, 1989), Jameson et al. (1986), Weatherill (1988), Frey and George (2008), Lo (2015). The application of the Delaunay triangulation for the purpose of surface interpolation was discussed by DeFloriani (1987).

The octree approach originated from the pioneering work of Yerry and Shephard (1985). The octree data structure has been adapted by Lohner (1988b) to produce efficient search procedures for the generation of unstructured grids by the moving-front technique. Octree-generated cells were used by Shephard et al. (1988); Yerry and Shephard (1990) to cover the domain and the surrounding space, and then to derive a tetrahedral grid by cutting the cubes. The generation of hexahedral unstructured grids was developed by Schneiders and Bunten (1995).

The moving-front technique has been successfully developed in three dimensions by Peraire et al. (1987), Lohner (1988a) and Formaggia (1991). Some methods using Delaunay connectivity in the frontal approach have been created by Merriam (1991), Mavriplis (1991, 1993), Rebay (1993), Muller et al. (1993), Marcum and Weatherill (1995).

Advancing-front grids with layers of prismatic and tetrahedral cells were formulated by Lohner (1993). A more sophisticated procedure, basically using bands of prismatic cells and a spring analogy to stop the advancement of approaching layers, was described by Pirzadeh (1992). The application of adaptive prismatic meshes to the numerical solution of viscous flows was demonstrated by Parthasarathy and Kallinderis (1995).

Some procedures for surface triangulations have been developed by Peraire et al. (1988), Lohner and Parikh (1988), Weatherill et al. (1993).

A survey of adaptive mesh refinement techniques was published by Powell et al. (1992). The combination of the Delaunay triangulation with adaptation was performed by Holmes and Lamson (1986), Mavriplis (1990), Muller (1994). The implementation of solution adaptation into the advancing-front method with directional refinement and regeneration of the original mesh was studied by Peraire et al. (1987). Approaches based on the use of sources to specify the local point spacing have been developed by Pirzadeh (1993), Pirzadeh (1994), Weatherill et al. (1993).

The prospects and trends for unstructured grid generation in its application to computational fluid dynamics were discussed by Baker (1995), Venkatakrishan (1996). The first application of the Delaunay triangulation in computational fluid dynamics was carried out by Bowyer (1981), Baker (1987). The advancing-front technique was introduced, in computational fluid dynamics, primarily by Peraire et al. (1987), Lohner (1988a), Lohner and Parikh (1988). The techniques of George (1971); Wordenweber (1981), Wordenweber (1983), Lo (1985), Peraire (1986) foreshadowed the more recent advancing-front methods. Muller (1994), Marchant and Weatherill (1994) applied a combination of frontal and Delaunay approaches to treat problems with boundary layers. Muller (1994) generated triangular grids in the boundary layer through a frontal technique, with high-aspect-ratio triangles, and filled the remainder of the domain with triangles built through the Delaunay approach. Another way to treat a boundary layer with the advancing-front approach was applied by Hassan et al. (1994). In the first step, the boundary layer is covered by a single layer of tetrahedral cells. Then, the newly generated nodes are moved along the cell edges towards the boundary by a specified distance. These steps, in the original layer, are repeated until a required resolution has been reached. After this, the advancing front proceeds to fill up the remainder of the domain.

An algorithm for the generation of a high-quality well-graded quadrilateral element mesh from a triangular element mesh was presented by Lee and Lo (1994), Lo (2015). Very important applications to parallel unstructured mesh generation were discussed by Chrisochoides (2006) and Ivanov (2008).

References

Anderson, W. K. (1994). A grid generation and flow solution method for the Euler equations in unstructured grids. *Journal of Computational Physics*, *110*, 23–38.

Baker, T. J. (1987). Three-dimensional mesh generation by triangulation of arbitrary points sets. AIAA Paper 87-1124-CP.

Baker, T. J. (1995). Prospects and expectations for unstructured methods. In *Proceedings of the Surface Modeling, Grid Generation and Related Issues in Computational Fluid Dynamics Workshop.* NASA Conference Publication 3291. (pp. 273–287). Cleveland, OH: NASA Lewis Research Center.

Baker, T. J. (1989). Automatic mesh generation for complex three-dimensional region using a constrained Delaunay triangulation. *Engineering with Computers*, *5*, 161–175.

Baker, T. J. (1994). Triangulations, mesh generation and point placement strategies. In D. Caughey (Ed.), *Computing the Future* (pp. 1–15). New York: Wiley.

Blacker, T. D., & Stephenson, M. B. (1991). Paving a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, *32*, 811–847.

Bowyer, A. (1981). Computing Dirichlet tessellations. *The Computer Journal*, *24*(2), 162–166.

Brostow, W., Dussault, J. P., & Fox, B. L. (1978). Construction of Voronoi polyhedra. *Journal of Computational Physics*, *29*, 81–92.

Carey, G. F. (1997). *Computational grids: Generation, adaptation, and solution strategies*. London: Taylor and Francis.

Cavendish, J. C., Field, D. A., & Frey, W. H. (1985). An approach to automatic three-dimensional finite element mesh generation. *International Journal for Numerical Methods in Engineering*, *21*, 329–347.

Chew, P. (1993). Mesh generation, curved surfaces and guaranteed quality triangles. Technical report, IMA, Workshop on Modeling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations, University of Minnesota, Minneapolis.

Chew, L. P. (1989). Constrained Delaunay triangulations. *Algorithmica*, *4*, 97–108.

Chrisochoides, N. (2006). *Chapter 7. Parallel Mesh Generation.* In *Numeriacal Solution of PDEs on Parallel Computers. Lecture Notes in Computational Science and Engineering* (vol. 51, pp. 237–264).

Cline, A. K., & Renka, R. L. (1990). A constrained two-dimensional triangulation and the solution of closest node problems in the presence of barriers. *SIAM Journal on Numerical Analysis*, *27*, 1305–1321.

DeFloriani, L. (1987). Surface representations on triangular grids. *The Visual Computer*, *3*, 27–50.

Delaunay, B. (1947). *Petersburg School of Number Theory*. USSR, Moscow (Russian): Ak. Sci.

Delaunay, B. (1934). Sur la sphere vide. *Bull. Acad Sci. USSR VII: Class Sci. Mat. Nat.*, *6*, 793–800.

Dirichlet, G. L. (1850). Uber die Reduction der positiven quadratischen Formen mit drei underbestimmten ganzen Zahlen. *Z. Reine Angew. Math.*, *40*(3), 209–227.

Edelsbrunner, H. (1987). *Algorithms in combinatorial geometry*. Berlin: Springer.

Du, D.-Z., & Hwang, F. (Eds.). (1992). *Computing in euclidean geometry*. Singapore: World Scientific.

Field, D. A., Nehl, T. W. (1992). Stitching together tetrahedral meshes. In Field, D., Komkov, V. (Eds.), *Geometric Aspects of Industrial Design.* Philadelphia: SIAM, Chap. 3, 25–38

Field, D. A. (1995). The legacy of automatic mesh generation from solid modeling. *Computer Aided Geometric Design*, *12*, 651–673.

Finney, J. L. (1979). A procedure for the construction of Voronoi polyhedra. *Journal of Computational Physics*, *32*, 137–143.

Formaggia, L. (1991). An unstructured mesh generation algorithm for three-dimensional aeronautical configurations. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 249–260). New York: North-Holland.

Fortune, S. (1985). *A sweepline algorithm for Voronoi diagrams*. Murray Hill, NJ: AT&T Bell Laboratory Report.

Frey, P. J., & George, P.-L. (2008). *Mesh generation application to finite elements*. ISTE Ltd and Wiley Inc.

George, A. J. (1971). Computer Implementation of the Finite Element Method. Stanford University Department of Computer Science, STAN-CS-71-208.

George, P. L., Hecht, F., & Saltel, E. (1990). Automatic 3d mesh generation with prescribed meshed boundaries. *IEEE Transactions on Magnetics*, *26*(2), 771–774.

George, P. L., & Borouchaki, H. (1998a). *Delaunay triangulation and meshing: application to finite elements*. Paris: Editions Hermes.

George, P. L., & Borouchaki, H. (1998b). *Delaunay triangulation and meshing*. Paris: Editions Hermes.

George, P. L., & Hermeline, F. (1992). Delaunay's mesh of convex polyhedron in dimension d: application for arbitrary polyhedra. *International Journal for Numerical Methods in Engineering*, *33*, 975–995.

Green, P. J., & Sibson, R. (1978). Computing Dirichlet tessellations in the plane. *The Computer Journal*, *21*(2), 168–173.

Guibas, L., & Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, *4*, 74–123.

Haman, B., Chen, J.-L., & Hong, G. (1994). Automatic generation of unstructured volume grids inside or outside closed surfaces. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulation and Related Fields* (p. 187). Swansea: Pineridge.

Hassan, O., Probert, E. J., Morgan, K., & Peraire, J. (1994). Unstructured mesh generation for viscous high speed flows. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulation and Related Fields* (p. 779). Swansea p: Pineridge.

Hazlewood, C. (1993). Approximating constrained tetrahedrizations. *Computer Aided Geometric Design*, *10*, 67–87.

Henle, M. (1979). *A combinatorial introduction to topology*. San Francisco: W.H Freeman.

Holmes, D. G., & Lamson, S. H. (1986). Adaptive triangular meshes for compressible flow solutions. In J. Hauser & C. Taylor (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics* (p. 413). Swansea: Pineridge.

Holmes, D. G., & Snyder, D. D. (1988). The generation of unstructured triangular meshes using Delaunay triangulation. In S. Sengupta, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics* (pp. 643–652). Swansea: Pineridge.

Ivanov, E. (2008). *Parallel Tetrahedral meshing based on a-priori domain decomposition: from scratch to results by utilizing off-the-shelf sequential software*. Saarbrucken: VDM Verlag Dr. Muller.

Jameson, A., Baker, T. J., Weatherill, N. P. (1986). Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103.

Klee, V. (1964). The number of vertices of a convex polytope. *Can. Math.*, *16*, 37.

Lawson, C. L. (1986). Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, *3*, 231–246.

Lee, K. D. (1976). On finding k-nearest neighbours in the plane. Technical Report 76–2216. University of Illinois, Urbana, IL.

Lee, D. T. (1978). Proximity and reachibility in the plane. Techical Report R-831, University of Illinois, Urbana, IL.

Lee, D. T., & Lin, A. K. (1986). Generalized Delaunay triangulation for planar graphs. *Discrete & Computational Geometry*, *1*, 201–217.

Lee, C. K., & Lo, S. H. (1994). A New Scheme for the Generation of a Graded Quadrilateral Mesh. *Computers and Structures.*, *52*, 847–857.

Lee, D. T., & Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Science*, *9*(3), 219–241.

Lo, S. H. (1985). A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, *21*, 1403–1426.

Lo, S. H. (2015). *Finite element mesh generation*. Boca Raton: CRC Press Taylor and Francis Group.

Lohner, R. (1988a). Generation of three-dimensional unstructured grids by the advancing-front method. AIAA Paper 88-0515.

Lohner, R. (1993). Matching semi-structured and unstructured grids for Navier–Stokes calculations. AIAA Paper 933348-CP.

Lohner, R. (1988b). Some useful data structures for the generation of unstructured grids. *Communications in Applied Numerical Methods*, *4*, 123–135.

Lohner, R., & Parikh, P. (1988). 3-dimensional grid generation by the advancing front method. *Intertnational Journal for Numerical Methods in Fluids*, *8*, 1135–1149.

Marchant, M. J., & Weatherill, N. P. (1994). Unstructured grid generation for viscous flow simulations. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulations and Related Fields* (p. 151). Swansea, UK: Pineridge.

Marcum, D. L., & Weatherill, N. P. (1995). Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal*, *33*(9), 1619–1625.

Mavriplis, D. J. (1993). An advancing front Delaunay triangulation algorithm designed for robustness. AIAA Paper 93-0671.

Mavriplis, D. J. (1990). Adaptive mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics*, *90*, 271–291.

Mavriplis, D. J. (1991). Unstructured and adaptive mesh generation for high Reynolds number viscous flows. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 79–92). Amsterdam: North-Holland.

Merriam, M. (1991). An efficient advancing front algorithm for Delaunay triangulation. Technical report, AIAA Paper 91-0792.

Muller, J. D., Roe, P. L., & Deconinck, H. (1993). A frontal approach for internal node generation in Delaunay triangulations. *International Journal for Numerical Methods in Fluids*, *17*(3), 241–256.

Muller, J. D. (1994). Quality estimates and stretched meshes based on Delaunay triangulation. *AIAA Journal*, *32*, 2372–2379.

Okabe, A., Boots, B., & Sugihara, K. (1992). *Spatial tessellations concepts and applications of voronoi diagrams*. New York: Wiley.

Owen, S. (1998). A survey of unstructured mesh generation technology. In 7th International Roundtable

Parthasarathy, V., & Kallinderis, Y. (1995). Directional viscous multigrid using adaptive prismatic meshes. *AIAA Journal*, *33*(1), 69–78.

Peraire, J. (1986). *A Finite Element Method for Convection Dominated Flows.* Ph.D. dissertation, University of Wales.

Peraire, J., Peiro, J., Formaggia, L., Morgan, K., Zienkiewicz, O. C. (1988). Finite element Euler computations in three dimensions. AIAA Paper 88-0032.

Peraire, J., Vahdati, M., Morgan, H., & Zienkiewicz, O. C. (1987). Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, *72*, 449–466.

Perronet, A. (1988). A generator of tetrahedral finite elements for multimaterial objects or fluids. In S. Sengupta, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Mechanics* (pp. 719–728). Swansea: Pineridge.

Pirzadeh, S. (1992). Recent progress in unstructured grid generation. AIAA Paper 92-0445.

Pirzadeh, S. (1994). Viscous unstructured three-dimensional grids by the advancing-layers method. AIAA Paper 94-0417.

Pirzadeh, S. (1993). Structured background grids for generation of unstructured grids by advancing front method. *AIAA Journal*, *31*(2), 257–265.

Powell, K. G., Roe, P. L., & Quirk, J. J. (1992). Adaptive-mesh algorithms for computational fluid dynamics. In M. Y. Hussaini, A. Kumar, & M. D. Salas (Eds.), *Algorithmic Trends in Computational Fluid Dynamics* (pp. 301–337). New York: Springer.

Preparata, F. P., & Shamos, M. I. (1985). *Computational geometry: an introduction*. New York: Springer.

Rebay, S. (1993). Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer–Watson algorithm. *Journal of Computational Physics*, *106*, 125–138.

Ruppert, J. (1992). *Results on Triangulation and High Quality Mesh Generation.* Ph.D. thesis, University of California, Berkeley.

Schneiders, R., & Bunten, R. (1995). Automatic generation of hexahedral finite element meshes. *Computers Aided Geometry Design*, *12*(7), 693.

Shenton, D. N., Cendes, Z. J. (1985). Three-dimensional finite element mesh generation using Delaunay tessellation. IEEE Transactions on Magnetics **MAG-21**, 2535–2538.

Shephard, M. S., Guerinoni, F., Flaherty, J. E., Ludwig, R. A., & Bachmann, P. L. (1988). Finite octree mesh generation for automated adaptive three-dimensional flow analysis. In S. Sengupta, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical grid generation in computational fluid mechanics* (pp. 709–718). Swansea: Pineridge.

Sibson, R. (1978). Locally equiangular triangulations. *The Computer Journal*, *21*(3), 243–245.

Sloan, S. W., & Houlsby, G. T. (1984). An implementation of Watson's algorithm for computing 2D Delaunay triangulations. *Advances in Engineering Software*, *6*(4), 192–197.

Steinitz, E. (1922). Polyeder and Raumeintailungen. *Enzykl. Mathematischen Wiss.*, *3*, 163.

Tam, T. K. H., & Armstrong, C. G. (1991). 2D finite element mesh generation by medial axis subdivision. *Advance in Engineering Software and Workstations*, *13*, 313–344.

Tanemura, M., Ogawa, T., & Ogita, N. (1983). A new algorithm for three-dimensional Voronoi tessellation. *Journal of Computational Physics*, *51*, 191–207.

Thompson, J. F., Soni, B. K., & Weatherill, N. P. (Eds.). (1999). *Handbook of grid generation*. Boca Raton: CRC Press.

Venkatakrishan, V. (1996). Perspective on unstructured grid flow solvers. *AIAA Journal*, *34*(3), 533–547.

Voronoi, G. F. (1908). Nouvelles applications des parameters countinus a la theorie des formes quadratiques. Deuxieme Memoire: Recherches sur la parallelloedres primitifs. *J. Reine Angew. Math.*, *134*, 198–287.

Watson, D. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, *24*(2), 167–172.

Weatherill, N. P., Marchant, M. F., Hassan, O., Marcum, D. L (1993). Adaptive inviscid flow solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes. AIAA Paper 93-3390

Weatherill, N. P. (1988). A method for generating irregular computational grids in multiply connected planar domains. *International Journal for Numerical Methods Fluids*, *8*, 181–197.

Weatherill, N. P. (1990). The integrity of geometrical boundaries in the two-dimensional Delaunay triangulation. *Communications in Applied Numerical Methods*, *6*, 101–109.

Weatherill, N. P., & Hassan, O. (1994). Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods Engineering*, *37*, 2005–2039.

Wordenweber, B. (1981). Automatic mesh generation of 2- and 3-dimensional curvilinear manifolds. Ph.D. Dissertation (available as Computer Laboratory Report N18), University of Cambridge).

Wordenweber, B. (1983). Finite-element analysis from geometric models. *The International Journal for Computational and Mathematics in Electrical and Electronics Engineering*, *2*, 23–33.

Yerry, M. A., & Shephard, M. S. (1985). Automatic three-dimensional mesh generation for three-dimensional solids. *Computers & Structures*, *20*, 31–39.

Yerry, M. A., & Shephard, M. S. (1990). Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, *20*, 1965–1990.

Zhou, J. M., Ke-Ran, S., Ke-Ding, Z., & Quing-Hua, Z. (1990). Computing constrained triangulation and Delaunay triangulation: a new algorithm. *IEEE Transactions on Magnetics*, *26*(2), 692–694.

Chapter 13
Applications of Adaptive Grids to Solution of Problems

This chapter discusses applications of mapping approaches to numerical solutions of some problems of mechanics, physics, fluids, plasmas, and nano-technologies.

13.1 Application to Unsteady Gas Dynamics Problems

The calculation of hydrodynamical problems on the adaptive moving meshes requires special conservative numerical schemes which update directly the flow parameters on the moving mesh at the new time level. Another way, in which interpolation of parameters from the fixed mesh to the moving one is used at every time step, smears the singularities in the solution, causing a decrease in accuracy of modeling. We describe here a modification of the Godunov scheme of the second-order accuracy in time and space on moving meshes, suggested by Azarenok (2000), to compute a two-dimensional gas flow in the Euler approach.

System of Equations

Two-dimensional equations of gas dynamics, namely, the laws of conservation of mass, momentum, and total energy, are written in the integral form which can be derived by transformation of the volume integrals in the space (x, y, t) to the surface integrals by virtue of Gauss's theorem:

$$\int_{\Omega} \left(\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{\partial \boldsymbol{a}}{\partial x} + \frac{\partial \boldsymbol{b}}{\partial y} \right) d\Omega = \oint\!\!\!\iint_{\partial\Omega} \boldsymbol{\sigma} dxdy + \boldsymbol{a} dydt + \boldsymbol{b} dtdx = \boldsymbol{0}, \tag{13.1}$$

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_13

where Ω is an arbitrary control volume in space (x, y, t), and $\partial\Omega$ is the boundary of Ω,

$$\boldsymbol{\sigma} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}.$$

Here, u and v are the velocity components, p and ρ are the pressure and density, and $E{=}\rho[e{+}0.5(u^2{+}v^2)]$ is the total energy, while e is the specific internal energy. The equation of state is $p = (\gamma{-}1)\rho e$ where γ is the ratio of specific heats. Denote the vector-valued unknown function as $\boldsymbol{f} = (u, v, p, \rho)^\top$. The conservation laws (13.1) hold for any functions $\boldsymbol{f}$ both smooth and discontinuous describing an ideal gas flow.

Numerical Scheme

Let a curvilinear moving grid in the x-y plane be introduced with the coordinate lines ξ, η, the $(i{+}1/2,\, j{+}1/2)$th cell of which at the time range (t^n, t^{n+1}) is shown in by a domain Ω in R space (x, y, t), being a hexahedron with planar top and bottom faces. The bottom (top) face of the hexahedron Ω is the control volume at the time t^n (t^{n+1}).

Integrating (13.1) over the boundary $\partial\Omega$ of the hexahedron gives a cell-centered finite-volume approximation of the governing gas dynamics equations

$$\begin{aligned} &\boldsymbol{\sigma}^{n+1}_{i+1/2j+1/2} A_{1'2'3'4'} - \boldsymbol{\sigma}^{n}_{i+1/2j+1/2} A_{1234} \\ &+\boldsymbol{Q}_{411'4'} + \boldsymbol{Q}_{233'2'} + \boldsymbol{Q}_{122'1'} + \boldsymbol{Q}_{344'3'} = 0, \end{aligned} \tag{13.2}$$

where $\boldsymbol{\sigma}^{n+1}_{i+1/2j+1/2}$ and $\boldsymbol{\sigma}^{n}_{i+1/2j+1/2}$ are the average values of $\boldsymbol{\sigma}$ at the time t^{n+1} and t^n in the center of the top and bottom faces, respectively; $A_{1'2'3'4'}$ and A_{1234} are the areas of the corresponding faces. Each of the four vector values $\boldsymbol{Q}_{411'4'}$, $\boldsymbol{Q}_{233'2'}$, $\boldsymbol{Q}_{122'1'}$ and $\boldsymbol{Q}_{344'3'}$ is the amount of the mass, momentum, and energy which flows into and out of the quadrilateral cell 1234 within time $\triangle t = t^{n+1} - t^n$ through the corresponding moving edges of the cell.

For example, the vector-valued quantity $\boldsymbol{Q}_{122'1'}$ that is the change of the parameters due to the flux through the edge 12 within time $\triangle t$ is given by

$$\boldsymbol{Q}_{122'1'} = \boldsymbol{\sigma}^{n+1/2}_{i+1/2j} A^{xy}_{122'1'} + \boldsymbol{a}^{n+1/2}_{i+1/2j} A^{yt}_{122'1'} + \boldsymbol{b}^{n+1/2}_{i+1/2j} A^{tx}_{122'1'}, \tag{13.3}$$

where $\boldsymbol{\sigma}^{n+1/2}_{i+1/2j}$, $\boldsymbol{a}^{n+1/2}_{i+1/2j}$, and $\boldsymbol{b}^{n+1/2}_{i+1/2j}$ are calculated using the parameters $\boldsymbol{f} = (u, v, p, \rho)^\top$ in the center of the face $122'1'$, i.e., at the mid-point of edge 12 at the time $t^{n+1/2}$ (or at the mid-point of edge $1''2''$); $A^{xy}_{122'1'}$, $A^{yt}_{122'1'}$, $A^{tx}_{122'1'}$ are the areas of the projections of the face $122'1'$ onto the coordinate planes x-y, y-t, and t-x, respectively, given by

$$
\begin{aligned}
A^{xy}_{122'1'} &= \int\limits_{122'1'} dxdy = 0.5[(x_{2'} - x_1)(y_{1'} - y_2) - (x_{1'} - x_2)(y_{2'} - y_1)], \\
A^{yt}_{122'1'} &= \int\limits_{122'1'} dydt = 0.5\triangle t(y_{2'} + y_2 - y_1 - y_{1'}), \\
A^{tx}_{122'1'} &= \int\limits_{122'1'} dtdx = -0.5\triangle t(x_{2'} + x_2 - x_1 - x_{1'}).
\end{aligned}
$$

These expressions are obtained from the formula for area of the quadrangle 1234

$$
\begin{aligned}
A_{1234} &= A(x_1, y_1; x_2, y_2; x_3, y_3; x_4, y_4) \\
&= 0.5[(x_3 - x_1)(y_4 - y_2) - (x_4 - x_2)(y_3 - y_1)] \,,
\end{aligned}
$$

when passing its contour in the anticlockwise direction.

The values $\boldsymbol{f}^{n+1}_{i+1/2j+1/2}$ are updated by two stages using a predictor-corrector procedure. At the first stage (predictor), we compute the intermediate values $\bar{\boldsymbol{f}}^{n+1}_{i+1/2j+1/2}$ at the $(n+1)$th level by using (13.2).

Let us consider the curvilinear coordinate ξ. Assume the function $\boldsymbol{f}$ to be linear within the cell $(i+1/2,\ j+1/2)$ in the ξ-direction. The values $\boldsymbol{f}^n_{ij+1/2}$ and $\boldsymbol{f}^n_{i+1j+1/2}$, specified at the left and right ends of the segment $((i,\ j+1/2),\ (i+1,\ j+1/2))$ at the time t^n, are defined as

$$
\begin{aligned}
\boldsymbol{f}^n_{ij+1/2} &= \boldsymbol{f}^n_{i+1/2\ j+1/2} - 0.5\delta \boldsymbol{f}^n_{i+1/2} h^n_{i+1/2} \,, \\
\boldsymbol{f}^n_{i+1j+1/2} &= \boldsymbol{f}^n_{i+1/2\ j+1/2} + 0.5\delta \boldsymbol{f}^n_{i+1/2} h^n_{i+1/2} \,.
\end{aligned}
$$

Here, $\delta \boldsymbol{f}^n_{i+1/2}$ is the "effective" derivative in the ξ-direction, while the spacing $h^n_{i+1/2}$ is the length of the underlying segment. Note that $\delta \boldsymbol{f}^n_{i+1/2}$ and $h^n_{i+1/2}$ are the notations for $(\delta \boldsymbol{f}^n_{\xi})_{i+1/2j+1/2}$ and $(h^n_{\xi})_{i+1/2j+1/2}$, respectively. When determining $\delta \boldsymbol{f}^n_{i+1/2}$, to suppress spurious oscillations in the vicinity of discontinuities, the monotonicity algorithm should be applied. The spacing $h^n_{i+1/2}$ is given by

$$
\begin{aligned}
&h^n_{i+1/2} = \\
&0.5\sqrt{(x^n_{i+1j} + x^n_{i+1j+1} - x^n_{ij} - x^n_{ij+1})^2 + (y^n_{i+1j} + y^n_{i+1j+1} - y^n_{ij} - y^n_{ij+1})^2}.
\end{aligned}
$$

By analogy, the values $\boldsymbol{f}^n_{i+1/2j}$ and $\boldsymbol{f}^n_{i+1/2j+1}$ are calculated at the left and right ends of the segment in the η-direction in the cell. Note that since we interpolate $\boldsymbol{f}$ along the curvilinear coordinate lines ξ and η the order of interpolation, in general, is less than 2 and equals 2 only if the mesh is rectangular and quasiuniform.

In order to find the values $\boldsymbol{Q}_{122'1'}$ we substitute in (13.3) the determined values of $\boldsymbol{f}$ at the mid-point of the lateral edge 12 of the quadrilateral 1234, i.e., at the time t^n instead of the ones at the time $t^{n+1/2}$. The values $\boldsymbol{Q}_{411'4'}$, $\boldsymbol{Q}_{233'2'}$, and $\boldsymbol{Q}_{344'3'}$ can be found in a similar way. Finally, from (13.2), we obtain the intermediate values $\bar{\boldsymbol{f}}^{n+1}_{i+1/2j+1/2}$ at the $(n+1)$th level.

We now discuss the second stage corrector. For this purpose, we set the effective derivatives at t^{n+1} equal to the ones at t^n, i.e., $\delta\bar{\boldsymbol{f}}^{n+1}_{i+1/2}=\delta\boldsymbol{f}^{n}_{i+1/2}$. Then, the values in the center of the faces $122'1'$ and $344'3'$, namely, at the mid-point of the edges 12 and 34 at the time $t^{n+1/2}$, are

$$\begin{aligned}
\boldsymbol{f}^{n+1/2}_{ij+1/2} &= 0.5[\boldsymbol{f}^{n}_{i+1/2j+1/2}+\bar{\boldsymbol{f}}^{n+1}_{i+1/2j+1/2}-0.5\delta\boldsymbol{f}^{n}_{i+1/2}(h^{n}_{i+1/2}+h^{n+1}_{i+1/2})]\ ,\\
\boldsymbol{f}^{n+1/2}_{i+1j+1/2} &= 0.5[\boldsymbol{f}^{n}_{i+1/2j+1/2}+\bar{\boldsymbol{f}}^{n+1}_{i+1/2j+1/2}+0.5\delta\boldsymbol{f}^{n}_{i+1/2}(h^{n}_{i+1/2}+h^{n+1}_{i+1/2})]\ .
\end{aligned}$$

We can obtain $\boldsymbol{f}^{n+1/2}_{i+1/2j}$ and $\boldsymbol{f}^{n+1/2}_{i+1/2j+1}$ in a similar way. These four vector values are used as the pre-wave states in the center of the corresponding lateral faces of the hexahedron for the Riemann problem.

Let us consider the face $122'1'$. To get the postwave states $\boldsymbol{f}^{n+1/2}$ in the center of this face (for brevity we omit subscripts i, j), i.e., at the mid-point of the segment $(1'', 2'')$, we solve the Riemann problem with the pre-wave states $(r,\, p,\, \rho)^{n+1/2}$ at this point on both sides of the face. One state $(r,\, p,\, \rho)^{n+1/2}_{+}$ relates to the underlying hexahedron and the other $(r,\, p,\, \rho)^{n+1/2}_{-}$ to the hexahedron adjacent to the face $122'1'$ (corresponding to the $(i+1/2,\, j-1/2)$th cell). Here, $r^{n+1/2}$ is the normal component of the velocity to the segment $(1'', 2'')$. We also use the tangential components of the velocity $q^{n+1/2}$ on those sides. The normal and tangential components of the velocity are given by

$$r^{n+1/2}=n_x u^{n+1/2}+n_y v^{n+1/2}\ , \qquad q^{n+1/2}=n_y u^{n+1/2}-n_x v^{n+1/2}\ ,$$

where n_x, n_y are the components of the outward unit normal vector to the segment $(1'', 2'')$.

After solving the Riemann problem, the post-wave values $(r,\, p,\, \rho)^{n+1/2}_{R}$ in the face center are defined. The post-wave tangential component of the velocity $q^{n+1/2}_{R}$ is given by

$$q^{n+1/2}_{R}=\begin{cases} q^{n+1/2}_{+} & \text{if } w_{12}\le d_{cont}\ ,\\ q^{n+1/2}_{-} & \text{otherwise}\ ,\end{cases} \tag{13.4}$$

where d_{cont} is the contact discontinuity speed in the Riemann problem, w_{12} is the velocity of the edge 12 in the normal direction to this edge, and $q^{n+1/2}_{+}$, $q^{n+1/2}_{-}$ are the pre-wave tangential components of the velocity in the underlying hexahedron and the one adjacent to the face $122'1'$, respectively. This condition expresses the fact that the tangential component of the velocity is discontinuous across the tangential discontinuity. The velocity w_{12} can be derived from the equality

$$\triangle t l_{1''2''} w_{12}=A^{xy}_{122'1'}\ , \tag{13.5}$$

where $l_{1''2''}$ is the length of the segment $(1'', 2'')$. Next, we restore the Cartesian components of the post-wave velocity in the center of the face $122'1'$

$$u_R^{n+1/2} = n_x r_R^{n+1/2} + n_y q_R^{n+1/2}\,, \qquad v_R^{n+1/2} = n_y r_R^{n+1/2} - n_x q_R^{n+1/2}\,.$$

Given the post-wave values $(u, v, p, \rho)_R^{n+1/2}$ in the center of the face $122'1'$, we calculate $\boldsymbol{Q}_{122'1'}$ via (13.3). Similarly, we treat the Riemann problem in the center of the other three faces to obtain $\boldsymbol{Q}_{411'4'}$, $\boldsymbol{Q}_{233'2'}$, and $\boldsymbol{Q}_{344'3'}$.

The final values of $\boldsymbol{f}_{i+1/2j+1/2}^{n+1}$ at the time t^{n+1} are obtained by using (13.2). This scheme is of second-order accuracy in the domains of smooth flow provided that the mesh is quasiuniform and close to rectangular.

Riemann Problem on the Moving Mesh

To demonstrate how to take into account the movement of grid nodes, let us consider the midpoint of the segment $(1''2'')$ within the time interval $(t^{n+1/2}, t^{n+1})$. There are 5 cases of location of the segment $((x, y)_{ij+1/2}^{n+1/2}, (x, y)_{ij+1/2}^{n+1})$ in the wave pattern depending on the velocity w_{12} of the edge 12. As the post-wave values are $(r, p, \rho)_R^{n+1/2}$, we take:

1. $(r, p, \rho)_R^{n+1/2} = (r, p, \rho)_-^{n+1/2}$ if $w_{12} < d_{\text{sh}}$, where d_{sh} is the speed of the left shock in the l-axis direction.
2. $(r, p, \rho)_R^{n+1/2} = (r, p, \rho)_2^{n+1/2}$ if $d_{\text{sh}} < w_{12} < d_{\text{cont}}$, where the vector $(r, p, \rho)_2^{n+1/2}$ defines the flow parameters behind the shock, and d_{cont} is the speed of the contact discontinuity which equals the velocity u in that domain.
3. $(r, p, \rho)_R^{n+1/2} = (r, p, \rho)_3^{n+1/2}$ if $d_{\text{cont}} < w_{12} < d_{\text{rar}}^{\text{lft}}$, where the vector $(r, p, \rho)_3^{n+1/2}$ defines the parameters in the domain between the contact discontinuity and left characteristic of the rarefaction wave expanding with the speed $d_{\text{rar}}^{\text{lft}}$.
4. $(r, p, \rho)_R^{n+1/2} = \phi(\alpha)$ if $d_{\text{rar}}^{\text{lft}} < w_{12} < d_{\text{rar}}^{\text{rght}}$, i.e., we calculate the flow parameters in the rarefaction wave using the similarity variable $\alpha = l/(t - t^{n+1/2})$. Here, $d_{\text{rar}}^{\text{rght}}$ is the speed of the right characteristic in the rarefaction fan.
5. $(r, p, \rho)_R^{n+1/2} = (r, p, \rho)_+^{n+1/2}$ if $w_{12} > d_{\text{rar}}^{\text{rght}}$.

Note that in the first-order Godunov scheme the above algorithm is applied at the time t^n.

Stability Condition

To demonstrate how the stability condition on moving mesh is obtained, let us consider the one-dimensional case with the $(i+1/2)$th cell. At the nth time level in this cell, the local time step is determined by

$$\triangle t_{i+1/2} = \frac{h_{i+1/2}^n}{\max(d_i^{rght} - w_{i+1}\ ,\ -d_{i+1}^{lft} - w_i)}\,, \tag{13.6}$$

where d_i^{rght} and d_{i+1}^{lft} are the extreme right and left wave speeds at the points x_i^n and x_{i+1}^n, respectively, obtained by solving the Riemann problem at $t^{n+1/2}$, and w_i is the velocity of the node x_i, i.e., the slope of the intercell boundary (x_i^n, x_i^{n+1}). The condition (13.6) implies that we estimate the time within which the left-going characteristic (in the linearized analysis, this is a straight line), emanating from the

$(i+1)$th node, arrives at the ith node moving with the velocity w_i, as well as the time within which the right characteristic, emanating from the ith node, arrives at the $(i+1)$th node moving with the velocity w_{i+1}. From these two time steps, we take the minimal one. The resulting time step over the mesh is given by

$$\triangle t = c_{cfl}\ \min_i\ \triangle t_{i+1/2}\ . \tag{13.7}$$

The coefficient c_{cfl} is a correction to the non-linearity of the Eq. (13.1). To calculate the node velocity w_i, on one hand, it is necessary to know the time step $\triangle t$, and on the other hand, to take into account that w_i participates in determining $\triangle t$. By these reasons, at the time level $n+1$, we use $\triangle t$ obtained at the preceding level n. The coefficient $c_{\text{cfl}}<1$, usually about 0.5, may be corrected during the computation.

In the 2D case, the choice of the admissible step $\triangle t$ may be estimated in the energetic norm to the underlying Eq. (13.1) written in a differential form as a t-hyperbolic by Friedrichs's system. The step $\triangle t$ in the $(i+1/2,\, j+1/2)$th cell is given by

$$\triangle t_{i+1/2j+1/2} = \frac{\triangle t^{'}\ \triangle t^{''}}{\triangle t^{'} + \triangle t^{''}}\ , \tag{13.8}$$

where

$$\begin{aligned}
\triangle t^{'} &= \frac{h^{'}}{\max(d_{14}^{\text{rght}} - w_{23};\, -d_{23}^{\text{lft}} + w_{14})}\ ,\\
\triangle t^{''} &= \frac{h^{''}}{\max(d_{12}^{\text{rght}} - w_{34};\, -d_{34}^{\text{lft}} + w_{12})},\\
h^{'} &= \frac{A_{1234}}{0.5\sqrt{(x_4+x_3-x_1-x_2)^2+(y_4+y_3-y_1-y_2)^2}},\\
h^{''} &= \frac{A_{1234}}{0.5\sqrt{(x_3+x_2-x_4-x_1)^2+(y_3+y_2-y_4-y_1)^2}}.
\end{aligned} \tag{13.9}$$

Here, $\triangle t^{'}$ and $\triangle t^{''}$ are the admissible time steps to the one-dimensional scheme in the ξ and η-direction, respectively; $h^{'}$, $h^{''}$ are the "average heights" of the bottom face 1234, and w is the velocity of the corresponding cell edge. For example, w_{12} is the velocity of the edge 12 in the normal direction determined via (13.5). Next, d_{12}^{rght} and d_{14}^{rght} are the "extreme right wave" speeds defined from solving the Riemann problem to the faces $122'1'$ and $11'4'4$, respectively; d_{23}^{lft} and d_{34}^{lft} are the "extreme left wave" speeds to the faces $233'2'$ and $433'4'$, respectively.

The resulting time step over the mesh is given by

$$\triangle t = c_{\text{cfl}}\ \min_{ij}\ \triangle t_{i+1/2j+1/2}\ .$$

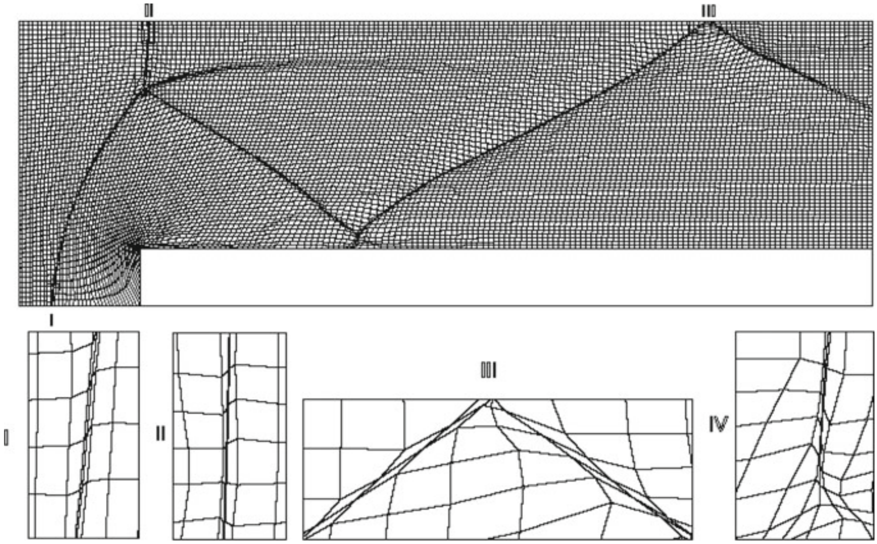


Fig. 13.1 Supersonic flow in the mach wind tunnel containing a step. The boundary nodes are distributed using constrained minimization. Adaptive mesh with fragments I, II, III near the boundary and IV comprising the triple point

13.1.1 Numerical Examples

Robustness, of the adaptive mesh method is demonstrated in the two numerical examples.

The first is a test presented by Fig. 13.1 of the planar unsteady supersonic flow in the wind tunnel containing a step (for details, see Colella and Woodward 1984). This test was performed by Azarenok and Ivanenko (2001) on the adaptive grids by applying the above flow solver when, as a monitor function f in the inverted energy functional (9.32), the modulus of velocity $|V|$ was used. The boundary nodes were adapted by applying 1-D minimization. One of the main difficulties was to capture the triple point, caused by the irregular reflection of the bow shock from the top wall, with clustered grid lines that required special efforts.

The use of constrained minimization for the boundary nodes allows us both to eliminate the above difficulty connected with capturing the triple point (see fragment IV of the mesh in Fig. 13.1) and to perform robust node clustering in the domains where the shocks are attached to the boundary or reflected from it (see fragments I–III). The shock waves are smeared over 2 to 3 cells. Compression of grid lines to the contact discontinuity emanating from the triple point is also demonstrated.

The second example is related to the modeling motion of a detonation wave. The adaptive mesh, obtained when modeling the unstable detonation wave motion (for details, see Azarenok and Tang (2005)), is presented in Fig. 13.2. The pressure is used as a monitor function. To perform a stable mesh adaptation, the constrained

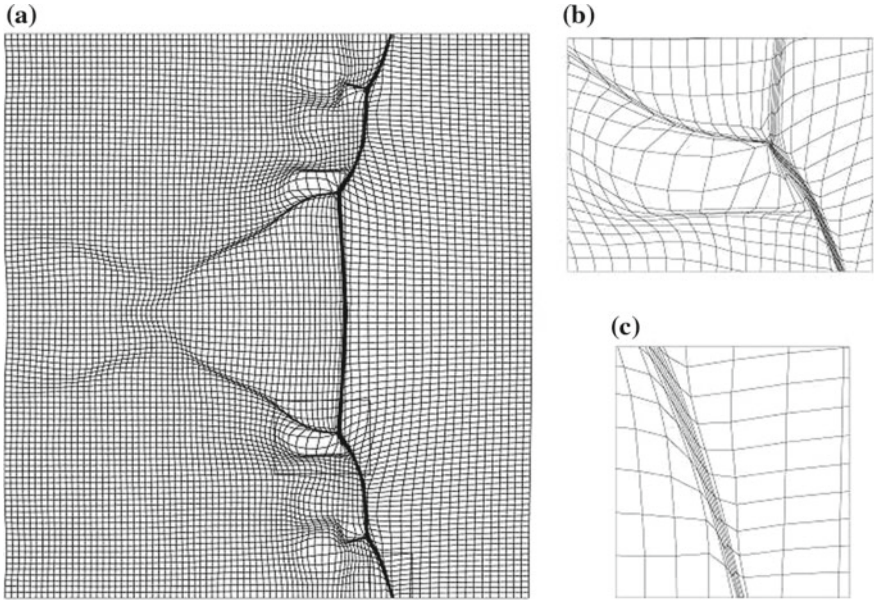


Fig. 13.2 Adaptive mesh (**a**) and its fragments (**b**), (**c**)

minimization for the redistribution of boundary nodes was also employed. The figure exhibits clustering of the grid lines to the main incident shock and transverse waves.

The calculations related to Figs. 13.1 and 13.2 were carried out by B. Azarenok.

13.2 Applications to Numerical Simulations of Tsunami Run-Up

This section is devoted to an implementation of adaptive grid technologies to numerical simulations of the successive reproduction of the edge of inundation caused by a tsunami wave.

13.2.1 Mathematical Model

It has been shown that many coastal effects of tsunamis can be described by a set of depth-averaged hydrostatic equations of motion, known as shallow-water wave equations (Leffe et al. 2010):

$$\begin{aligned}
&\frac{\partial H}{\partial t} + \frac{\partial Hu}{\partial x} + \frac{\partial Hv}{\partial y} = 0,\\
&\frac{\partial Hu}{\partial t} + \frac{\partial Hu^2}{\partial x} + \frac{\partial Huv}{\partial y} + \frac{g}{2}\frac{\partial H^2}{\partial x} = gH\frac{\partial h}{\partial x} - uHC_R,\\
&\frac{\partial Hv}{\partial t} + \frac{\partial Huv}{\partial x} + \frac{\partial Hv^2}{\partial y} + \frac{g}{2}\frac{\partial H^2}{\partial y} = gH\frac{\partial h}{\partial y} - vHC_R,
\end{aligned} \tag{13.10}$$

where $H(x, y, t)$, $h(x, y)$ are functions of the water depth and bottom elevation, respectively, $u(x, y, t)$, $v(x, y, t)$ the cartesian components of the velocity vector $\boldsymbol{u}(x, y, t)$, g the gravitational acceleration, C_R, the bottom friction coefficient determined by the formula:

$$C_R = \frac{gn^2}{H^{4/3}}|\boldsymbol{u}|,$$

in which n denotes an empirical roughness coefficient (Chesy coefficient). The solution domain of the system (13.10) is a rectangle $D\{0 \le x \le L_x,\ 0 \le y \le L_y\}$. All parameters in (13.10) for the subdomain in D corresponding to the dry land area are considered as being equal to zero.

13.2.2 Dynamically Adaptive Numerical Grid

The solution of system (13.10) was carried out by a method of large particles represented as water columns (Leffe et al. 2010; Belotserkovsky et al. 1982) on nonstationary grid cells clustering only in the zone of D corresponding to the tsunami run-up edge. The numerical grid is obtained through the use of a time-dependent coordinate transformation

$$\boldsymbol{x}(t, \boldsymbol{\xi}) : \Xi^2 \to X^2, \quad \boldsymbol{x}(t, \boldsymbol{\xi}) = (x^1(t, \boldsymbol{\xi}), x^2(t, \boldsymbol{\xi})), \quad \boldsymbol{\xi} = \xi^1, \xi^2, \tag{13.11}$$

with the identification $x^1 = x, x^2 = y$. The coordinate transformation is found at the points of the uniform grid in the reference domain $\Xi^2\{0 \le \xi^1,\ \xi^2 \le 1\}$ by numerical solution of two-dimensional inverted diffusion equations:

$$g_{22}\frac{\partial^2 x^k}{\partial \xi^1 \partial \xi^1} - 2g_{12}\frac{\partial^2 x^k}{\partial \xi^1 \partial \xi^2} + g_{11}\frac{\partial^2 x^k}{\partial \xi^2 \partial \xi^2} = J^2 \frac{1}{Z(\boldsymbol{x}, t)} \frac{\partial}{\partial x^k} Z(\boldsymbol{x}, t), \quad k = 1, 2, \tag{13.12}$$

where

$$g_{ij} = \frac{\partial x^1}{\partial \xi^i}\frac{\partial x^1}{\partial \xi^j} + \frac{\partial x^2}{\partial \xi^i}\frac{\partial x^2}{\partial \xi^j}, \quad J = \det\left\{\frac{\partial x^i}{\partial \xi^j}\right\}, \quad i, j = 1, 2,$$

and $Z(\boldsymbol{x}, t)$ is a control function. The function $Z(\boldsymbol{x}, t)$ is determined by the formula

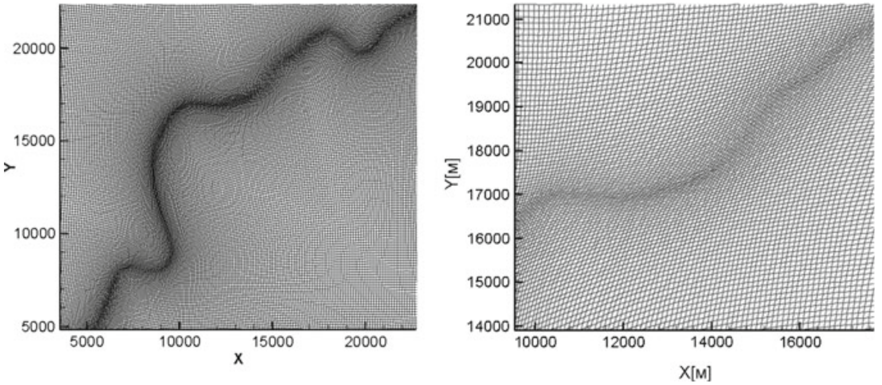


Fig. 13.3 Adaptive grid with node clustering near the wave edge (*left*); fragment of the grid (*right*)

$$Z(\boldsymbol{x}, t) = (\varphi(\boldsymbol{x}, t))^{\alpha} + \varepsilon, \quad \varphi(\boldsymbol{x}, t) = \min_{i=1,\dots,N} \rho(\boldsymbol{x}, \boldsymbol{x}_i(t)), \quad \alpha > 0, \quad 0 < \varepsilon \ll 1, \tag{13.13}$$

where $\boldsymbol{x}_i(t)$, $i = 1, \dots, N$, are the points in D at time t corresponding to the wave run-up edge (dry-wet interface), $\alpha = 1.7, \varepsilon = 0.01$. The coordinates of these points in the domain D are calculated by the solution of equations (13.10). Thus, in accordance with Sect. 11.2, the nodes of the grid obtained by the solution of equations (13.12) are clustering in the vicinity of the points $\boldsymbol{x}_i(t)$, $i = 1, \dots, N$, which approximate the projection of the wave run-up edge on the domain D, and vice-versa, rarefying in other parts of the domain, including parts corresponding to dry land area. The solution to equations (13.12) at each time step t was found through the scheme described in Sect. 11.2. Figure 13.3 demonstrates such a grid in D with cells clustering on the projection of the wave run-up edge.

13.2.3 Equations in Dynamic Curvilinear Coordinates

The coordinate transformation (13.11) also allows one to find solutions to the tsunami problem on a uniform grid in the reference domain Ξ^2. For this purpose, Eq. (13.10) are rewritten in the curvilinear coordinates t, ξ^1, ξ^2 in a divergent form with the help of the tensor identity (2.97) connecting the expressions in the cartesian $t, x^1, \dots, x^n$ and curvilinear $t, \xi^1, \dots, \xi^n$ coordinates:

$$\frac{\partial A^0}{\partial t} + \sum_{i=1}^{n} \frac{\partial A^i}{\partial x^i} = \frac{1}{J}\Big(\frac{\partial}{\partial t}(J A^0) + \sum_{j=1}^{n} \frac{\partial}{\partial \xi^j}[J(\overline{A}^j - A^0 \overline{w}^j)]\Big),$$

where

$$J = \det\left\{\frac{\partial x^i}{\partial \xi^j}\right\}, \quad \overline{A}^j = \sum_{i=1}^{n} A^i \frac{\partial \xi^j}{\partial x^i}, \quad \overline{w}^j = \sum_{i=1}^{n} \frac{\partial x^i}{\partial t}\frac{\partial \xi^j}{\partial x^i}.$$

With the help of this identity, system (13.10) is rewritten in the variables t, ξ^1, ξ^2 in the following divergent form:

$$\begin{aligned}
\frac{\partial(JH)}{\partial t} &+ \frac{\partial}{\partial \xi^1}\left\{H\left[\left(u - \frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial \xi^2} - \left(v - \frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial \xi^2}\right]\right\} + \\
&+ \frac{\partial}{\partial \xi^2}\left\{H\left[\left(v - \frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial \xi^1} - \left(u - \frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial \xi^1}\right]\right\} = 0, \\
\frac{\partial(JHu)}{\partial t} &+ \frac{\partial}{\partial \xi^1}\left\{Hu\left[\left(u - \frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial \xi^2} - \left(v - \frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial \xi^2}\right]\right\} + \\
&+ \frac{\partial}{\partial \xi^2}\left\{Hu\left[\left(v - \frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial \xi^1} - \left(u - \frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial \xi^1}\right]\right\} + \\
&+ \frac{g}{2}\left[\frac{\partial H^2}{\partial \xi^1}\frac{\partial y}{\partial \xi^2} - \frac{\partial H^2}{\partial \xi^2}\frac{\partial y}{\partial \xi^1}\right] = \\
&= gH\left[\frac{\partial h}{\partial \xi^1}\frac{\partial y}{\partial \xi^2} - \frac{\partial h}{\partial \xi^2}\frac{\partial y}{\partial \xi^1}\right] - JuHC_R, \\
\frac{\partial(JHv)}{\partial t} &+ \frac{\partial}{\partial \xi^1}\left\{Hv\left[\left(u - \frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial \xi^2} - \left(v - \frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial \xi^2}\right]\right\} + \\
&+ \frac{\partial}{\partial \xi^2}\left\{Hv\left[\left(v - \frac{\partial y}{\partial t}\right)\frac{\partial x}{\partial \xi^1} - \left(u - \frac{\partial x}{\partial t}\right)\frac{\partial y}{\partial \xi^1}\right]\right\} + \\
&+ \frac{g}{2}\left[\frac{\partial H^2}{\partial \xi^2}\frac{\partial x}{\partial \xi^1} - \frac{\partial H^2}{\partial \xi^1}\frac{\partial x}{\partial \xi^2}\right] = \\
&= gH\left[\frac{\partial h}{\partial \xi^2}\frac{\partial x}{\partial \xi^1} - \frac{\partial h}{\partial \xi^1}\frac{\partial x}{\partial \xi^2}\right] - JvHC_R,
\end{aligned} \tag{13.14}$$

where $J = \dfrac{\partial x}{\partial \xi^1}\dfrac{\partial y}{\partial \xi^2} - \dfrac{\partial x}{\partial \xi^2}\dfrac{\partial y}{\partial \xi^1}$.

The components $x(t, \boldsymbol{\xi})$, $y(t, \boldsymbol{\xi})$ of the coordinate transformation (13.11), as well as the parameters of the water flow in system (13.14), are found at the points of the uniform rectangular grid in the reference domain Ξ^2. There is therefore no need for interpolation of the flow parameters, in contrast to the solution of equations (13.10) on a nonuniform moving grid in the physical domain D.

13.2.4 Numerical Algorithm

A realization of the method of large particles for solution of equations (13.14) was carried out in two stages. In the first stage, all convective terms in (13.14) were deleted, resulting in the system:

$$
\begin{aligned}
&\frac{\partial JH}{\partial t} = 0,\\
&\frac{\partial u}{\partial t} + g[\frac{\partial(H-h)}{\partial\xi_1}\frac{\partial y}{\partial\xi_2} - \frac{\partial(H-h)}{\partial\xi_2}\frac{\partial y}{\partial\xi_1}] = -uC_R,\\
&\frac{\partial v}{\partial t} + g[\frac{\partial(H-h)}{\partial\xi_2}\frac{\partial x}{\partial\xi_1} - \frac{\partial(H-h)}{\partial\xi_1}\frac{\partial x}{\partial\xi_2}] = -vC_R.
\end{aligned}
\tag{13.15}
$$

During the second stage, the transition equations

$$
\begin{aligned}
\frac{\partial(JH)}{\partial t} &+ \frac{\partial}{\partial\xi_1}\left\{H\left[(u-\frac{\partial x}{\partial t})\frac{\partial y}{\partial\xi_2} - (v-\frac{\partial y}{\partial t})\frac{\partial x}{\partial\xi_2}\right]\right\}\\
&+ \frac{\partial}{\partial\xi_2}\left\{H\left[(v-\frac{\partial y}{\partial t})\frac{\partial x}{\partial\xi_1} - (u-\frac{\partial x}{\partial t})\frac{\partial y}{\partial\xi_1}\right]\right\} = 0,\\
\frac{\partial(JHu)}{\partial t} &+ \frac{\partial}{\partial\xi_1}\left\{Hu\left[(u-\frac{\partial x}{\partial t})\frac{\partial y}{\partial\xi_2} - (v-\frac{\partial y}{\partial t})\frac{\partial x}{\partial\xi_2}\right]\right\}\\
&+ \frac{\partial}{\partial\xi_2}\left\{Hu\left[(v-\frac{\partial y}{\partial t})\frac{\partial x}{\partial\xi_1} - (u-\frac{\partial x}{\partial t})\frac{\partial y}{\partial\xi_1}\right]\right\} = 0,\\
\frac{\partial(JHv)}{\partial t} &+ \frac{\partial}{\partial\xi_1}\left\{Hv\left[(u-\frac{\partial x}{\partial t})\frac{\partial y}{\partial\xi_2} - (v-\frac{\partial y}{\partial t})\frac{\partial x}{\partial\xi_2}\right]\right\}\\
&+ \frac{\partial}{\partial\xi_2}\left\{Hv\left[(v-\frac{\partial y}{\partial t})\frac{\partial x}{\partial\xi_1} - (u-\frac{\partial x}{\partial t})\frac{\partial y}{\partial\xi_1}\right]\right\} = 0,
\end{aligned}
\tag{13.16}
$$

were solved. An explicit scheme of the first order on the uniform rectangular grid in Ξ^2 was applied to solution of equations (13.15) and (13.16). The grid functions $H_{i,j}$ and $h_{i,j}$ were specified at the grid nodes (circles in Fig. 13.4), the velocity component u was specified between the nodes at the points marked by squares (in Fig. 13.4), and the velocity component v by triangles.

The numerical scheme for Eq. (13.15) is written in the following form:

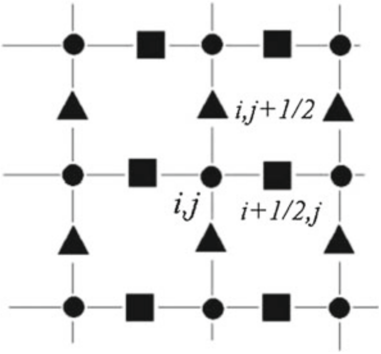


Fig. 13.4 Structure of the reference grid

$$
\begin{aligned}
J\tilde{H}_{i,j} &= (JH)^n_{i,j},\\
\tilde{u}_{i+1/2,j} &= \frac{1}{1+\Delta t\,(C_R)^n_{i+1/2,j}}\Big\{u^n_{i+1/2,j} - \Delta t g\Big[\frac{(H-h)^n_{i+1,j}-(H-h)^n_{i,j}}{\Delta\xi_1}\\
&\quad\times\frac{y^n_{i+1/2,j+1/2}-y^n_{i+1/2,j-1/2}}{\Delta\xi_2}\\
&\quad-\frac{(H-h)^n_{i+1/2,j+1/2}-(H-h)^n_{i+1/2,j-1/2}}{\Delta\xi_2}\frac{y^n_{i+1,j}-y^n_{i,j}}{\Delta\xi_1}\Big]\Big\},\\
\tilde{v}_{i,j+1/2} &= \frac{1}{1+\Delta t\,(C_R)^n_{i,j+1/2}}\Big\{v^n_{i,j+1/2} - \Delta t g\Big[\frac{(H-h)^n_{i,j+1}-(H-h)^n_{i,j}}{\Delta\xi_2}\\
&\quad\times\frac{x^n_{i+1/2,j+1/2}-x^n_{i-1/2,j+1/2}}{\Delta\xi_1}\\
&\quad-\frac{(H-h)^n_{i+1/2,j+1/2}-(H-h)^n_{i-1/2,j+1/2}}{\Delta\xi_1}\frac{x^n_{i,j+1}-x^n_{i,j}}{\Delta\xi_2}\Big]\Big\}.
\end{aligned}
$$

Equations (13.16) are approximated by a similar scheme.

In order to fulfill the condition of hydrostatic fluid equilibrium, the value of the Jacobian J in system (13.16) at each time step was calculated from the differential identity

$$
\frac{\partial J}{\partial t}+\frac{\partial}{\partial\xi_1}\Big\{\Big[-\frac{\partial x}{\partial t}\frac{\partial y}{\partial\xi_2}+\frac{\partial y}{\partial t}\frac{\partial x}{\partial\xi_2}\Big]\Big\}+\frac{\partial}{\partial\xi_2}\Big\{\Big[-\frac{\partial y}{\partial t}\frac{\partial x}{\partial\xi_1}+\frac{\partial x}{\partial t}\frac{\partial y}{\partial\xi_1}\Big]\Big\}\equiv 0.
$$

In order to provide the stability criteria for schemes (13.15) and (13.16), the step size was taken from the inequality, following from (2.91) for $n = 2$

$$
\Delta t \le \alpha\cdot\min\Big\{\frac{\Delta x}{|u|+\sqrt{gH}},\ \frac{\Delta y}{|v|+\sqrt{gH}}\Big\},
$$

computed at all grid nodes, where $0 < \alpha < 1$ is an empirical coefficient. In the present calculations, $\alpha = 0.5$.

13.2.5 Some Results of Calculations

The method of adaptive grids described above was applied to calculations of a solitary wave run-up on the coastal region of Tohoku (Japan tsunami, 2011) (Shimozono et al. 2012). The height of the wave traveling across the ocean to the coast was taken as 0.6 m. Figure 13.5 demonstrates the projection of the wave-front run-up calculated by the solution of equations (13.10) on an adaptive grid with 300 × 225 nodes, represented in Fig. 13.3. The identical results are obtained by the solution of equations (13.15) and (13.16) with the adaptive coordinate transformation but on the uniform rectangular grid (300 × 225 nodes) in the logical domain Ξ^2, to the solution of equations (13.10) on the adaptive grid generated by the same coordinate

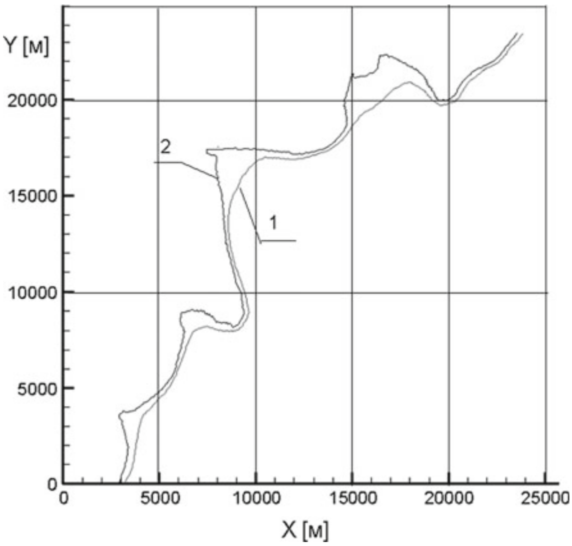


Fig. 13.5 Computation with an adaptive coordinate transformation (300 × 225 nodes): *1* – initial line of wave run-up edge, *2* – final line of wave run-up edge

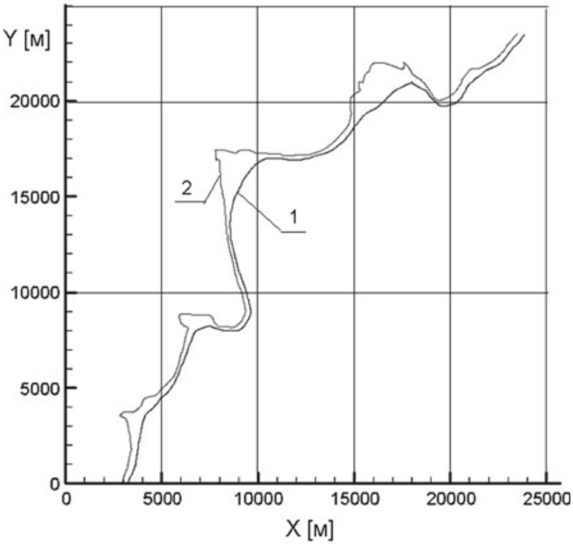


Fig. 13.6 Computation on a uniform grid (1500 × 1200 nodes) in *D*: *1* – initial line of wave run-up edge, *2* – final line of wave run-up edge

transformation. Figure 13.6 exhibits the similar results obtained by the solution of equations (13.10) on the uniform grid in the physical domain D but with 1500 × 1200 nodes. Thus, the application of adaptive coordinate transformations allows one to reduce the number of cells up to 20 times.

13.3 Application to Singularly-Perturbed Equations

As an example of the application of the comprehensive grid technology described, we consider a numerical solution of the following boundary value problem for a singularly perturbed equation:

$$\begin{aligned}&-\varepsilon \varDelta u + \mathbf{a}(\mathbf{s}) \cdot \nabla u + f(\mathbf{s}, u) = 0, \quad \mathbf{s} \in S^n,\\ &u(\mathbf{s}) = u_0(\mathbf{s}), \quad \mathbf{s} \in \partial S^n,\end{aligned} \tag{13.17}$$

where

$$\varDelta u = \sum_{i=1}^{n} \frac{\partial^2 u}{\partial s^i \partial s^i}, \quad i = 1, \dots, n, \quad \nabla u = \Big(\frac{\partial u}{\partial s^1}, \dots, \frac{\partial u}{\partial s^n}\Big),$$

$S^n \subset R^n$ is a bounded domain, $\mathbf{a}(\mathbf{s}) = (a^1(\boldsymbol{s}), \dots, a^n(\boldsymbol{s}))$ is the convection vector, $f(\mathbf{s}, u)$ is a specified function, while $0 < \varepsilon \ll 1$ is the coefficient of diffusion.

Depending on the function $\boldsymbol{a}(\boldsymbol{s})$, the solution to (13.17) has boundary and/or interior layers when the coefficient ε is small (Liseikin 2001).

13.3.1 Numerical Algorithm

The boundary value problem (13.17) is written out in the grid coordinates ξ^i as

$$\begin{aligned}&-\varepsilon B_n^{s\xi}[u] + F[u] = 0,\\ &u(\boldsymbol{\xi}) = \varphi(\boldsymbol{\xi}) \ , \quad \boldsymbol{\xi} \in \partial \varXi^n \ ,\end{aligned} \tag{13.18}$$

where $\varphi(\boldsymbol{\xi}) = u_0[\boldsymbol{s}(\boldsymbol{\xi})]$,

$$\begin{aligned}&F[u] = \{(J)^2 a^i[\boldsymbol{s}(\boldsymbol{\xi})] + \varepsilon B_n^{s\xi}[s^i]\} \frac{\partial u}{\partial \xi^j} \frac{\partial \xi^j}{\partial s^i} + (J)^2 f(\boldsymbol{s}(\boldsymbol{\xi}), u) \ , \quad i, j = 1, \dots, n \ ,\\ &B_n^{s\xi}[u] = g^{s\xi} g_{\xi s}^{ij} \frac{\partial^2 u}{\partial \xi^i \partial \xi^j} \ , \quad i, j = 1, \dots, n \ ,\\ &g_{\xi s}^{ij} = \frac{\partial \xi^i}{\partial s^k} \frac{\partial \xi^j}{\partial s^k} \ , \quad g^{s\xi} = (J)^2 \ , \quad J = \det\Big\{\frac{\partial s^i}{\partial \xi^j}\Big\} \ , \quad i, j, k = 1, \dots, n \ .\end{aligned}$$

Similarly to (10.17), the problem (13.18) is replaced by the following nonstationary problem with respect to the function $u(\boldsymbol{\xi}, t)$:

$$\begin{aligned}&\frac{\partial u}{\partial t} = (J)^p\{\varepsilon B_n^{s\xi}[u] - F[u]\},\\ &u(\boldsymbol{\xi}, t) = \varphi_0(\boldsymbol{\xi}) \ , \quad \boldsymbol{\xi} \in \partial \varXi^n,\\ &u(\boldsymbol{\xi}, 0) = u_0(\boldsymbol{\xi}) \ , \quad \boldsymbol{\xi} \in \varXi^n \ .\end{aligned} \tag{13.19}$$

Since the operator $B_n^{s\xi}$ is, in fact, the operator $B_n^{\boldsymbol{\xi}}$ in (10.17) with the identification $g_{ij}^{\boldsymbol{\xi}} = g_{ij}^{s\xi}$, the problem (13.19) is solved by the same method described in Chap. 11 for problem (10.17) with $n = 3$ and the condition (10.15), namely, for

$$\begin{aligned}
&\frac{\partial s^i}{\partial t} = (J)^p \left\{ B_n^{s\xi}[s^i] - \frac{J^2}{Z[u](\boldsymbol{s})} \frac{\partial}{\partial s^i} Z[u](\boldsymbol{s}) \right\}, \quad i = 1, \dots, n\,, \\
&s^i(\xi, t) = \psi^i(\xi), \quad \xi \in \partial \Xi^n, \quad t \geq 0\,, \\
&s^i(\xi, 0) = s_0^i(\xi)\,, \quad \xi \in \Xi^n\,.
\end{aligned} \tag{13.20}$$

This method was applied to the generation of the intermediate transformation $\boldsymbol{s}(\boldsymbol{\xi})$.

The grids are obtained by the numerical solution of the inverted Beltrami equations in (13.20) in the spherical metric tensor (9.149) with $n = 3$ and

$$\begin{aligned}
Z[u](\boldsymbol{s}) &= \sqrt{c + \left(\frac{\partial u}{\partial s^1}\right)^2 + \left(\frac{\partial u}{\partial s^2}\right)^2 + \left(\frac{\partial u}{\partial s^3}\right)^2}\,, \\
Z[u](\boldsymbol{s}(\boldsymbol{\xi})) &= \sqrt{c + g_{\xi s}^{ij} \frac{\partial u(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^i} \frac{\partial u(\boldsymbol{s}(\boldsymbol{\xi}))}{\partial \xi^j}}\,, \quad i, j = 1, 2, 3
\end{aligned}$$

(Figure 13.7) and in control metric (9.10) of the monitor hypersurface S^{r3} over S^3 with $\boldsymbol{f}(\boldsymbol{s}) = u(\boldsymbol{s})$ (Fig. 13.8). Both metrics provide node clustering in the zones of the interior and boundary layers of $u(\boldsymbol{s})$. The expressions $J \dfrac{\partial \xi^j}{\partial s^m}$ in (13.19) and (13.20) were replaced in accordance with the following formula:

$$J \frac{\partial \xi^j}{\partial s^m} = \frac{\partial s^{m+1}}{\partial \xi^{j+1}} \frac{\partial s^{m+2}}{\partial \xi^{j+2}} - \frac{\partial s^{m+1}}{\partial \xi^{j+2}} \frac{\partial s^{m+2}}{\partial \xi^{j+1}}, \quad m, j = 1, \dots, n, \tag{13.21}$$

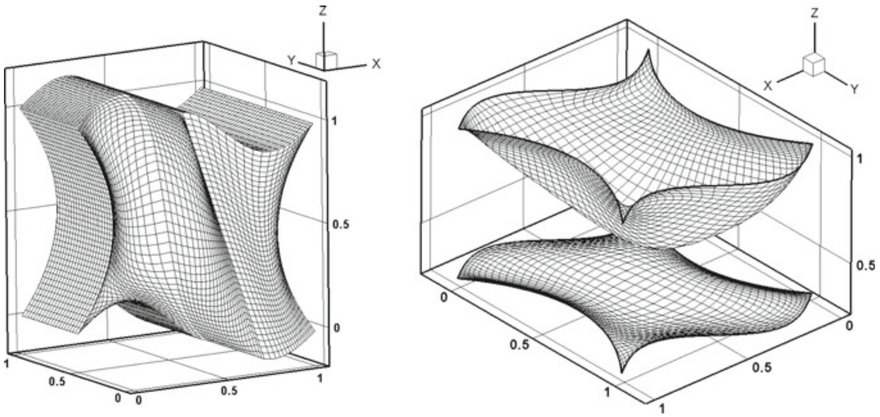


Fig. 13.7 Three-dimensional adaptive grid with node clustering near the boundary layers of the solution $u(\mathbf{s})$ to problem (13.17) with $\boldsymbol{a}(\boldsymbol{s}) = \mathbf{0}$ (*left*). The solution at the points of a two-dimensional coordinate surface (*right*)

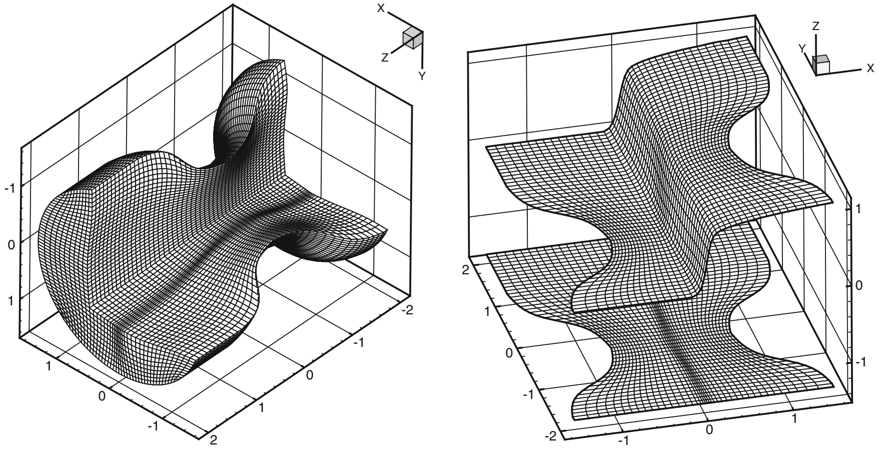


Fig. 13.8 Three-dimensional adaptive grid with node clustering near an interior layer of the solution of problem (13.17) (*left*). The solution at the points of a two-dimensional coordinate surface (*right*)

where, for each superscript or subscript index, say, l, $l + 3$ is equivalent to l.

Figures 13.7 and 13.8 show numerical solutions to the coupled problems (13.19) and (13.20) for $n = 3$, both with $p = 1$, and the resulting three-dimensional adaptive grids.

13.4 Problem of Heat Transfer in Plasmas

Adaptive grid generation for computational studies of magnetized plasmas is a novel challenge because of their extreme degree of anisotropy. The nature of magnetic confinement is to restrict the motion of particles across, but not along, the magnetic field. For relatively collisional plasmas described by fluid equations in the grid coordinates $\xi^1, \dots, \xi^n$

$$\frac{\partial T}{\partial t} = \nabla \cdot (\chi \cdot \nabla T) = \frac{1}{J}\frac{\partial}{\partial \xi^i}\Big(J\chi : \nabla \xi^i \nabla \xi^j \frac{\partial T}{\partial \xi^j}\Big), \quad i, j = 1, 2, 3 \;, \qquad (13.22)$$

this results in a ratio of parallel to transverse thermal conductivities 10^{10}. Discretization errors which cause a small amount of the large parallel heat flux to “leak” into the transverse direction can fatally compromise the validity of the results. Anisotropy also strongly affects the propagation of magnetohydrodynamic waves; waves propagating normal to the magnetic field have much smaller phase velocity than those propagating along the field. Since these slow waves are the most readily destabilized by magnetic field gradients and curvature, accurate representation of small parallel gradients is essential for the accurate modeling of linear and nonlinear instabilities.

Such accurate representation of the small parallel gradients requires the generation of magnetic-field-aligned numerical grids. One more requirement for the grids consists in their nonsingularity. In addition to alignment and nonsingularity, grid adaptation to large transverse gradients characteristic of resistive instabilities and magnetic reconnection is also needed.

The most successful method of dealing with anisotropy has been the use of a flux coordinate system in which at least one coordinate is chosen to be orthogonal to the magnetic field. This has previously been implemented only for static, axisymmetric magnetic fields. As the magnetic field evolves due to currents in the plasma and external coils, the static grid is left behind and loses its alignment. While the initial configuration may be axisymmetric, for which nested flux surfaces are known to exist, the field may evolve into a nonaxisymmetric configuration in which no exact flux surfaces exist, but rather there are regions of multiple small islands and stochasticity. For such situations, the approach that uses the flux lines as a family of coordinate lines produces singular numerical grids.

The method applied to generating numerical grids with the requirements of magnetic field alignment, adaptation, and nonsingularity is based on the solution of inverted diffusion and Beltrami equations with respect to control metrics that serve to provide these grid properties. The principal task in applying this method to a specific problem is the choice of a suitable control metric. It has been illustrated that the efficiency of the inverted diffusion and Beltrami's nonlinear elliptic equations for harmonic grid generation produces nonsingular grids aligned with a given magnetic field. Also, it has been demonstrated that the nature of the grid depends strongly on the topology of the initial conditions, which is preserved in the process of iterating to a solution. Thus, the quality of the alignment obtained for a logically rectangular grid in the magnetic reconnection problem is not as good as that obtained for a more complicated topology in the tokamak edge problem. The results accomplished have also illustrated the use of a nonlinear diffusion-type equation to obtain a grid adapted to a specified function related to spatial truncation error, producing a fine grid where that function is large and a coarse grid where it is small.

Figure 13.9 (*left-hand*) is a contour plot of the flux function for an intermediate stage in a magnetic reconnection problem, the Eq. (13.22) computed numerically with the same spectral element code described in Chap. 10 used to solve the grid generation system (10.40) for $n = 2$. The top and bottom boundaries are surfaces of

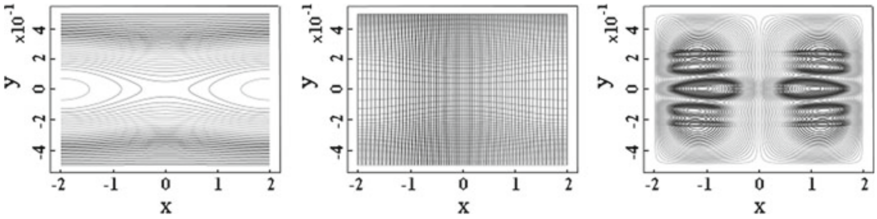


Fig. 13.9 Magnetic flux function $\psi(\mathbf{s})$ (*left*). Grid lines aligned with the magnetic field (*center*). Contour plot of alignment error $|\mathbf{B} \cdot \nabla \xi^2|$ (*right*)

constant ψ, while the left and right boundaries are periodic. There is an x-point in the middle of the computational domain, where $\mathbf{B} = \mathbf{0}$, and a separatrix dividing the domain into non-simply-connected regions, where ψ is, respectively, positive and negative.

Figure 13.9 (*center*) shows the resulting grid lines obtained by the solution of the system (10.40) with g_{s}^{ij} specified by (10.51). Using the spectral element code, a smooth, continuous solution is found for the mapping $\mathbf{s}(\xi)$ and used to map a uniform Cartesian grid in Ξ^2 into the parametric domain S^2.

The initial condition used for relaxation to this solution is a simple Cartesian grid, $s^1 = \xi^1, \ s^2 = \xi^2$. The final grid is constrained to have the same topology as the initial condition, which forces it to cut across the magnetic field lines. This occurs in the region where $\mathbf{B}$ is small but nonzero, minimizing the overall misalignment. Figure 13.9 (*right-hand*) is a contour plot of the alignment error $|\mathbf{B} \cdot \nabla \xi^2|$, with maximum value 0.025 in the region near the midplane and RMS average value 0.0085. Outside the separatrix, the alignment is very good. This type of aligned grid is adequate for small magnetic islands for which alignment inside of the island is not essential and simplicity of grid topology is more important.

The pictures in Fig. 13.9 were created by A. Glasser, who used a spectral element method for computing plasmas and inverted diffusion grid equations for generating adaptive, field-aligned grids (see Glasser et al. 2005a, b).

13.4.1 The Tokamak Edge Region

The region at the edge of the tokamak is of critical importance to its performance as a potential fusion reactor. Figure 13.10 shows the tokamak (*left-hand*) and magnetic field lines (*right-hand*) in this region. There is an x-point connected to a separatrix dividing the domain into closed field lines in the hot core where the main fusion reactions occur, and open field lines further out, which make contact with material walls. MHD edge-localized modes (ELMs) bridging the pedestal region at the edge of the core and the scrape-off layer outside create a region of turbulence which determine the presence or absence of a transport barrier, and thus the quality of confinement in the core. Hot plasma from the core escapes from the pedestal region to the scrape-off region and encounters metallic divertor plates (see Fig. 13.10 (*left-hand*)), where ionization and recombination occur and the plasma is subjected to impurities. Accurate modeling of this region is essential to an understanding of the tokamak. Adaptive grid generation can make important contributions to the accuracy and efficiency of such a model.

The final grid preserves the topology of the initial conditions, which is a simply connected and logically rectangular grid for the case of Fig. 13.9, resulting in imperfect alignment inside the separatrix. For the tokamak edge region, that is not an acceptable solution; it is essential to have good alignment in the whole neighborhood of the x-point and the separatrix, and that cannot be done with a single simply-connected, logically rectangular grid. Figure 13.10 (*right-hand*) shows the

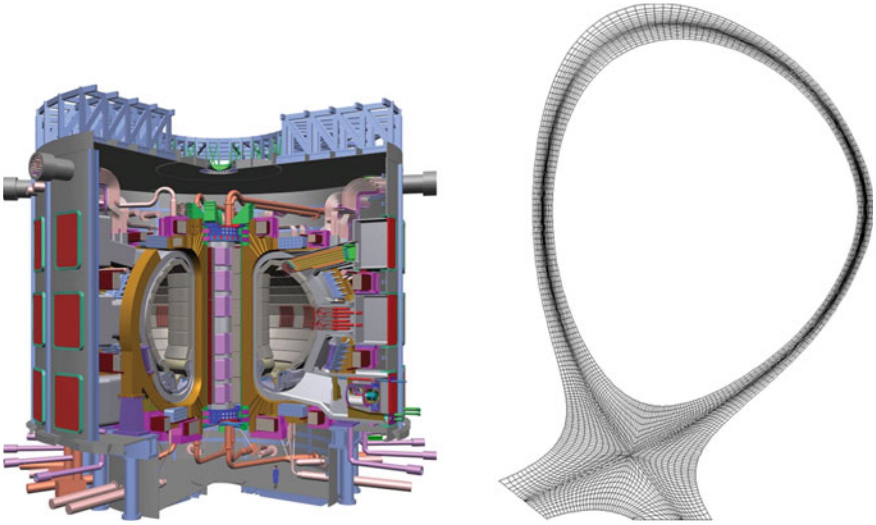


Fig. 13.10 Tokamak and numerical grid aligned with magnetic field

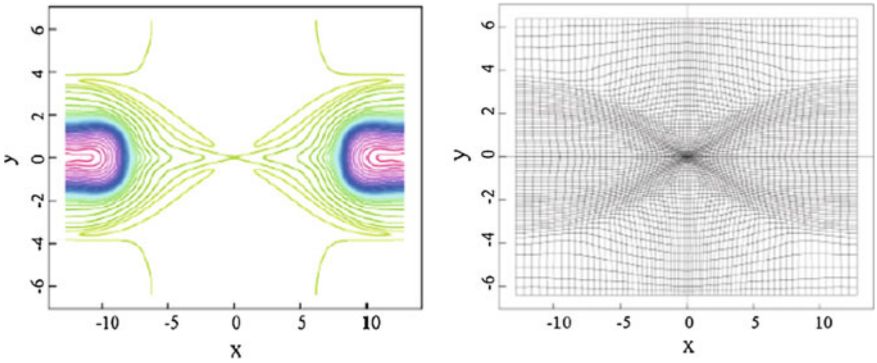


Fig. 13.11 Magnetic vector field flux (*left*) and balanced grid (*right*)

result of the grid generated through the control metric (11.32) with the coefficient $\epsilon(\mathbf{s})$ defined by the layer type function (11.29):

$$\epsilon(\mathbf{s}) = 0.001\frac{ln(1000|\mathbf{B}|^2+1)}{ln(1000)}.$$

13.4.2 Computations on Balanced Grids

Figures 13.11 (*right-hand*) and 13.12 (*below*) exhibit adaptive grids aligned with a magnetic field and clustering in the zones of large solution error. Computations of Eq. (13.22) and adaptive numerical grids were carried out by A. Glasser.

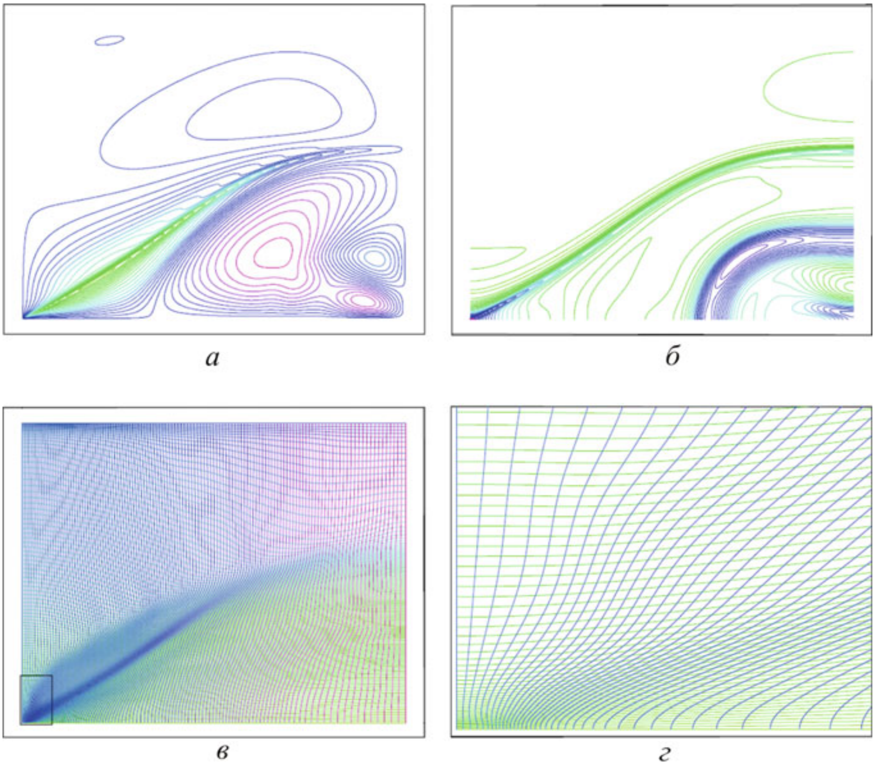


Fig. 13.12 Magnetic vector field flux (*top*) and balanced grid (*bottom*)

13.5 Evaluations of Temperature-Profile Discrepancies

This section demonstrates some results of a numerical modeling of temperature gauged in burning solid fuel by an inserted thermocouple. Their detailed description is presented in Liseikin et al. (2011a, b).

Subsurface thermocouple sensors are very important technical devices employed to gauge heat fluxes in complicated heat-stressed frameworks (Borovkova et al. 2008), in various heat-diffusion mechanisms (Franco et al. 2007), and in burning solid fuels (Zenin 1962, 1963; Asay et al. 2005). Incidental problems related to the accuracy of the sensors' temperature data may appear. The major source of inaccuracy is the difference in thermal properties of the materials of the thermocouple and the surrounding medium. High gradients of temperature in heated materials lead, as a rule, to increased heat transfer from the surface, since the thermal conductivity of metallic thermocouples is higher than that of the surrounding substance. Additional problems may appear due to variation in distances to the surface of heat transfer caused by the pyrolysis of the substance. A numerical investigation of this problem in the axisymmetric formulation was discussed in Rychkov et al. (2010).

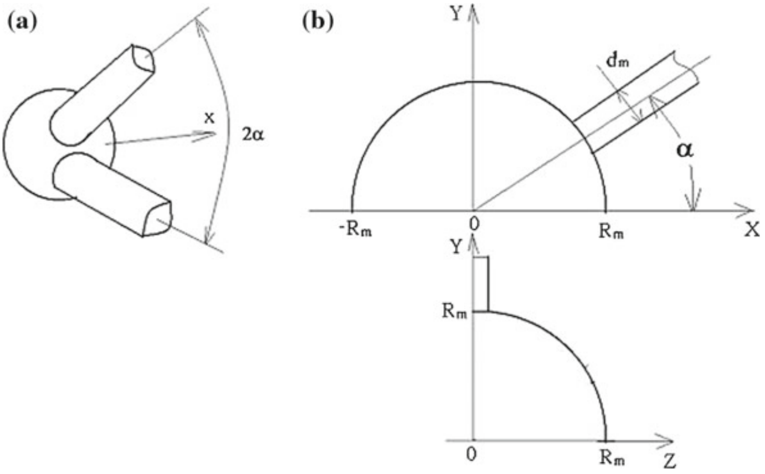


Fig. 13.13 Principal scheme of a thermocouple (*left*) and a domain of solution (*right*)

13.5.1 Mathematical Model for the Interaction of Heat Wave with Thermocouple

Let us consider a process of temperature-profile gauging of a condensed substance pyrolyzed by an external heat source. This process is formulated as an unsteady three-dimensional problem of heat transfer between the solid material and an embedded thermocouple (Fig. 13.13), assuming that both the pyrolysis velocity and the surface heat flux are constant.

The thermocouple head is a bead in radius R_m, which is connected under angle 2α to two cylindrical conductors (wires), each in radius r_m (Fig. 13.13 *left-hand*). Since there are two planes of symmetry, the domain for physical modelling can be taken as a quarter of the whole domain (Fig. 13.13 *right-hand*). The magnitudes of the angle α range from $\alpha = 0^0$ to $\alpha = 60^0$. In the case $\alpha = 0^0$, both conductors are assumed to be a single conductor with an area double its cross-section. In the Cartesian coordinates x, y, z, the modelling domain is a parallelepiped $D\{x_s(y, z, t) \leq x \leq x_{max}, 0 \leq y \leq y_{max}, 0 \leq z \leq z_{max}, t \geq 0\}$, whose external boundary segments are separated from the thermocouple bead by a significant distance to make their influence on the temperature data negligible. The left-hand boundary of the domain is a plane surface of pyrolysis which moves with a constant velocity r_b, and thus $x_s(y, z, t) = x_s(y, z, 0) + r_b t$. The center of the coordinates is settled at the center of the thermocouple bead. The process is simulated by the following heat equation:

$$\frac{\partial(\rho C T)}{\partial t} - \Big(\frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(\lambda\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(\lambda\frac{\partial T}{\partial z})\Big) = 0\,, \tag{13.23}$$

where $C, \ \rho, \ \lambda$ - are the specific heat, density, and thermal conductivity, respectively. These coefficients are not continuous, but break at the boundary surface of the thermocouple. The divergent form of Eq. (13.23) provides a correct computation of the thermal fluxes in the case of discontinuous coefficients. The applicable boundary conditions for Eq. (13.23) are specified as follows:

$$\begin{aligned} &T(x_s, y, z, t) = T_s \ ; \quad \frac{\partial T(x, y, z, t)}{\partial x}\Big|_{x=x_{max}} = 0 \ ; \\ &\frac{\partial T(x, y, z, t)}{\partial y}\Big|_{y=y_{max}} = \frac{\partial T(x, y, z, t)}{\partial y}\Big|_{y=y_0} = 0 \ ; \\ &\frac{\partial T(x, y, z, t)}{\partial z}\Big|_{z=z_{max}} = \frac{\partial T(x, y, z, t)}{\partial z}\Big|_{z=z_0} = 0 \ . \end{aligned} \tag{13.24}$$

The initial conditions are specified in accordance with Mihelson's distribution, which describes the temperature of a plane heat wave moving along the axis OX with a constant velocity r_b:

$$T(x, y, z, 0) = T_0 + (T_s - T_0) exp(-r_b(x - x_s(0))C_p\rho_p/\lambda_p) \ ,$$

where the index p is related to the parameters of the substance; T_s, T_0 are the temperature of pyrolysis and the initial temperature, respectively, and both are constant. The initial placement of the left-hand boundary segment $x = x_s(0)$ is sufficiently far from the thermocouple bead to provide a temperature distribution in the thermocouple close to T_0.

Numerical Algorithm

A numerical solution of equation (13.23) with boundary conditions (13.24) is found using the method of finite volumes, which allows one to apply arbitrary meshes. The equation was rewritten in the integral form for every cell V of the mesh:

$$\int_V \frac{\partial}{\partial t} Q dV + \oint_S \boldsymbol{F} d\boldsymbol{S} = 0 \ , \tag{13.25}$$

where $Q = C\rho T$; $\boldsymbol{F} = -\lambda\nabla T$ is the heat flux through a normally oriented segment $d\boldsymbol{S}$ of the boundary S of V. Let us designate by $V_{i,j,k}$ the volume of the cell V, and by $Q^n_{i,j,k}$ the middle quantity of Q at the center of V at the time $n\tau$. Then, Eq. (13.25) is approximated by the following scheme of the second order with respect to space and time:

$$\begin{aligned} &\frac{4Q^{n+1}_{i,j,k} - 3Q^n_{i,j,k} + Q^{n-1}_{i,j,k}}{2\tau} V_{i,j,k} + \big[(\boldsymbol{F}\cdot\boldsymbol{S})^{n+1}_{i+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1}_{i-1/2} \\ &+(\boldsymbol{F}\cdot\boldsymbol{S})^{n+1}_{j+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1}_{j-1/2} + (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1}_{k+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1}_{k-1/2}\big] = 0 \ , \end{aligned} \tag{13.26}$$

where τ is the time step. The scalar products in the square brackets are heat fluxes through the corresponding faces of V, factored by the unit normals to these faces. A description of their calculations was given in Vinokur (1989). Scheme (13.26) is implicit, and to its solution was applied the following iterative algorithm, based on the introduction of quasitime at each τ_n:

$$
\begin{aligned}
&\frac{Q^{n+1,s+1}_{i,j,k} - Q^{n+1,s}_{i,j,k}}{\tau_1} + \frac{4Q^{n+1,s}_{i,j,k} - 3Q^{n,s}_{i,j,k} + Q^{n-1,s}_{i,j,k}}{2\tau} V_{i,j,k} \\
&+\big[(\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{i+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{i-1/2} + (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{j+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{j-1/2} \\
&+(\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{k+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{k-1/2}\big] = 0\ ,
\end{aligned}
\tag{13.27}
$$

where τ_l is the quasitime step, and s is the number of the iteration. A solution to (13.27) was found by the prediction-correction method (Yanenko 1971):

$$
\begin{aligned}
&\frac{\delta^{s+1/3} - \delta^{s}}{\tau_1} V_{i,j,k} - \Lambda_1 \delta^{s+1/3} = \frac{4Q^{n+1,s}_{i,j,k} - 3Q^{n,s}_{i,j,k} + Q^{n-1,s}_{i,j,k}}{2\tau} V_{i,j,k} \\
&+\big[(\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{i+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{i-1/2} + (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{j+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{j-1/2} \\
&+(\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{k+1/2} - (\boldsymbol{F}\cdot\boldsymbol{S})^{n+1,s}_{k-1/2}\big]\ , \\
&\frac{\delta^{s+2/3} - \delta^{s+1/3}}{\tau_1} V_{i,j,k} + \Lambda_2 \delta^{s+2/3} = 0, \\
&\frac{\delta^{s+1} - \delta^{s+2/3}}{\tau_1} V_{i,j,k} + \Lambda_3 \delta^{s+1} = 0, \quad Q^{n+1,s+1} = Q^{n+1,s} + \delta^{s+1}\ ,
\end{aligned}
\tag{13.28}
$$

where δ^s is a correction of Q, and Λ_1, Λ_2 and Λ_3 are the difference operators, each taking into account the second derivatives in one corresponding direction only. The solution to the problem (13.28) relaxes to the solution to (13.23) as $\delta^s \to 0$. At the points of the left-hand boundary segment, we assume $\delta^s = 0$, while "mild" boundary conditions were specified on the other boundary segments. Since quantities $Q^{-1}_{i,j,k}$ are not known, a scheme of the first order with respect to τ was applied for calculating $Q^{0}_{i,j,k}$ in (13.28).

13.5.2 Generation of Adaptive Grid

The nodes of a grid were calculated by a numerical solution of inverted diffusion equations (10.11) for $n = 3$, namely,

$$
\begin{aligned}
&[g^{s\xi}_{22} g^{s\xi}_{33} - (g^{s\xi}_{23})^2] \frac{\partial^2 s^k}{\partial\xi^1 \partial\xi^1} + [g^{s\xi}_{11} g^{s\xi}_{33} - (g^{s\xi}_{13})^2] \frac{\partial^2 s^k}{\partial\xi^2 \partial\xi^2} \\
&+[g^{s\xi}_{11} g^{s\xi}_{22} - (g^{s\xi}_{12})^2] \frac{\partial^2 s^k}{\partial\xi^3 \partial\xi^3} + 2[g^{s\xi}_{23} g^{s\xi}_{13} - g^{s\xi}_{12} g^{s\xi}_{33}] \frac{\partial^2 s^k}{\partial\xi^1 \partial\xi^2} \\
&+2[g^{s\xi}_{12} g^{s\xi}_{23} - g^{s\xi}_{22} g^{s\xi}_{13}] \frac{\partial^2 s^k}{\partial\xi^1 \partial\xi^3} + 2[g^{s\xi}_{13} g^{s\xi}_{12} - g^{s\xi}_{23} g^{s\xi}_{11}] \frac{\partial^2 s^k}{\partial\xi^2 \partial\xi^3} \\
&= \frac{J^2}{Z(\boldsymbol{s})} \frac{\partial Z(\boldsymbol{s})}{\partial s^k}, \\
&i, j = 1, 2, 3.
\end{aligned}
\tag{13.29}
$$

Equations (13.29) were solved by means of the iterative algorithm applied to (10.17). The control function $Z(\boldsymbol{s})$ was used to provide grid clustering near the boundary of the thermocouple. This function $Z(\boldsymbol{s})$ was specified as

$$\begin{aligned}
Z(\boldsymbol{s}) &= u_1(\boldsymbol{s})u_2(\boldsymbol{s}),\\
u_1(\boldsymbol{s}) &= \frac{1}{0.01 + \left|x^2(\boldsymbol{s}) + z^2(\boldsymbol{s}) - r_m^2\right|},\\
u_2(\boldsymbol{s}) &= \frac{1}{0.01 + \left|x^2(\boldsymbol{s}) + y^2(\boldsymbol{s}) + z^2(\boldsymbol{s}) - R_m^2\right|},
\end{aligned}$$

where

$$\begin{aligned}
x(\boldsymbol{s}) &= s^1 \sin\alpha - s^2 \cos\alpha \ ,\\
y(\boldsymbol{s}) &= s^1 \cos\alpha + s^2 \sin\alpha \ ,\\
z(\boldsymbol{s}) &= s^3 \ .
\end{aligned}$$

The function $u_1(\boldsymbol{s})$ makes the grid nodes cluster near the conductor, while $u_2(\boldsymbol{s})$ does this near the bead. The final grid is demonstrated in Fig. 13.14.

Since the grid nodes may not lie on the boundary of the thermocouple, the coefficients of the thermal conductivity on the boundaries of cells were taken to be middle values of the coefficients at the centers of the neighboring cells. The quantity $C \cdot \rho$ was calculated at the center of the cell. The accuracy of numerical solutions was estimated through calculations on sequences of grids. It was established that the grid $75 \times 100 \times 50$ provides a relative accuracy of 0.1%.

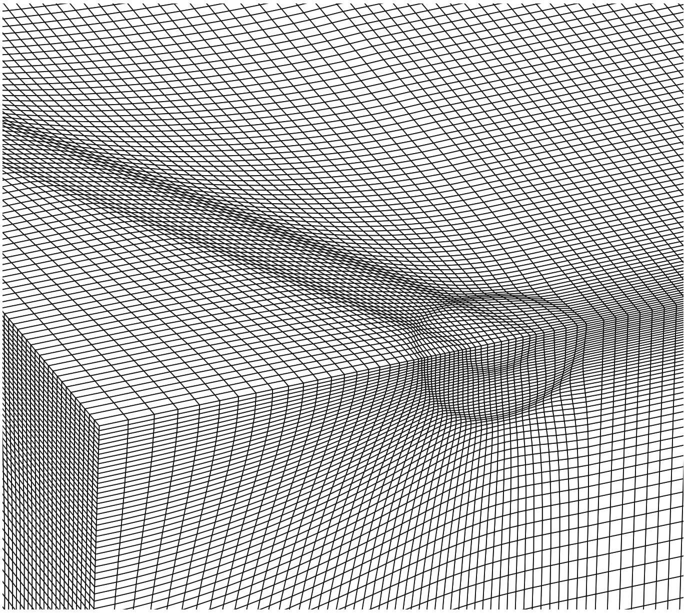


Fig. 13.14 Fragment of a three-dimensional adaptive grid ($\alpha = 60^0$)

13.5.3 Results of Numerical Experiments

Calculations were run for various radiuses R_m of the thermocouple bead. The radiuses of the wire conductors were specified by the relation $r_m/R_m = 0.2$ and $r_m/R_m = 0.75$. The thermal parameters were taken as follows:

$$\rho_p = 1.6\,[\mathrm{g/sm^3}],\ \ \rho_m = 8\,[\mathrm{g/sm^3}],\ \ C_p = 0.3\,[\mathrm{kal/g\cdot K}],\ \ C_m = 0.2\,[\mathrm{kal/g\cdot K}],$$
$$\lambda_p = 0.00072\,[\mathrm{kal/(sm\cdot s\cdot K)}],\ \ \lambda_m = 0.16\,[\mathrm{kal/(sm\cdot s\cdot K)}],$$

where the indexes m and p are related to the material of the thermocouple and of the surrounding substance, respectively. The temperature of the pyrolysis surface was $T_s = 650\,\mathrm{K}$, and the initial temperature was $T_0 = 300\,\mathrm{K}$.

The calculations were carried out up to the point when the pyrolysis surface touched the thermocouple bead. The temperature at the center of the bead was defined by the formula

$$T_{av} = \iiint\limits_V T(x,y,z,t)dx\,dy\,dz \ \ / \iiint\limits_V dx\,dy\,dz,$$

while the relative discrepancy of the temperature measurement was determined as

$$\delta = \frac{T_\infty - T_{av}}{T_\infty}\cdot 100\%,$$

where $T_\infty = T(0, y_{max}, z_{max}, t)$.

A total of two series of calculations were run, namely, for the pyrolysis velocities $r_b = 0.1\,[\mathrm{sm/s}]$ and $r_b = 1\,[\mathrm{sm/s}]$. The results are illustrated in Figs. 13.15, 13.16, 13.17, 13.18, 13.19 and 13.20 for the function $\delta(\xi)$, where $\xi = \dfrac{R_m - x_s(t)}{\Delta}$ is the dimensionless distance between the pyrolysis surface and the thermocouple bead, and $\Delta = \dfrac{\lambda_p}{C_p\rho_p r_b}$ is the width of the thermal front of the pyrolysis wave. The dotted lines correspond to the quantity q

$$q = \left[\left(-\lambda_p\frac{\partial T}{\partial x}\right)\Big|_{\substack{x=x_s\\ y=0\\ z=0}}\right] \Big/ \left[\left(-\lambda_p\frac{\partial T}{\partial x}\right)\Big|_{\substack{x=x_s\\ y=y_{max}\\ z=z_{max}}}\right].$$

In the first series of calculations, the width of the heat front was $\Delta = 150\,\mu\mathrm{m}$, and the pyrolysis velocity was $r_b = 0.1\,[\mathrm{sm/s}]$, while in the second series, the width of the heat front was $\Delta = 15\,\mu\mathrm{m}$, and the pyrolysis velocity $r_b = 1\,[\mathrm{sm/s}]$.

The figures show that the surface temperature discrepancy depends on the angle α and the radius of the bead. Figures 13.15 and 13.16, illustrating the results related to the pyrolysis velocity $r_b = 1\,[\mathrm{sm/s}]$, demonstrate that the temperature discrepancy decreases when the radius of the bead is increased. In particular, the temperature

Fig. 13.15 Plot of $\delta(\xi)$ for $R_m = 45\,\mu\text{m}$

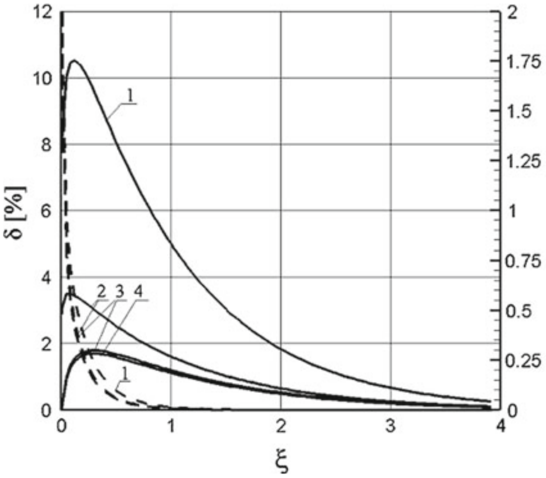
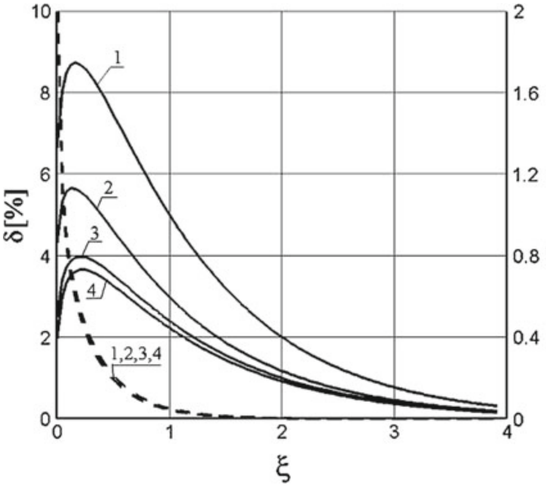


Fig. 13.16 Plot of $\delta(\xi)$ for $R_m = 120\,\mu\text{m}$



discrepancy is negative when $R_m > 40\,\mu\text{m}$ because heating is delayed when the width of the heat wave is larger than the size of the bead. When the pyrolysis front nears the thermocouple, heat loss from the bead increases, whereas when ξ is small, the heating of the bead increases, leading to a decrease of $\delta(\xi)$. This accounts for the pike shape of the quantity $\delta(\xi)$ in Figs. 13.17, 13.18, 13.19 and 13.20, with its maximum value in the interior of the interval ξ.

Analogous behaviour of $\delta(\xi)$, depicted in Fig. 13.21, occurs when the pyrolysis velocity is low while the bead is large. When the quantities r_m and R_m are subject to the relation $r_m/R_m = 0.75$ and r_m is large, then the measurement error is significantly larger (Fig. 13.22), because, in this case, the heat transfer is faster from the bead.

Fig. 13.17 Plot of $\delta(\xi)$ for $R_m = 12\,\mu\text{m}$ $(r_m/R_m = 0.2)$. *1* $-\ \alpha = 0^0$; *2* $-\ \alpha = 15^0$; *3* $-\ \alpha = 45^0$; *4* $-\ \alpha = 60^0$

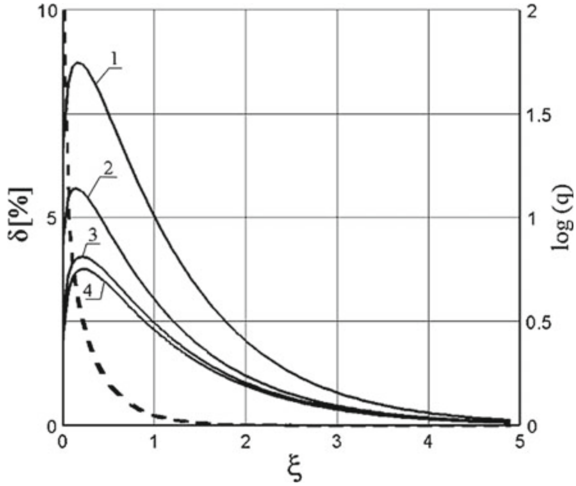
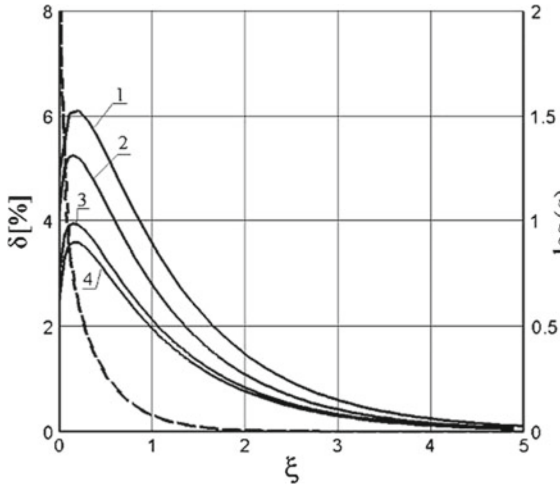


Fig. 13.18 Plot of $\delta(\xi)$ for $R_m = 20\,\mu\text{m}$ $(r_m/R_m = 0.2)$. *1* $-\ \alpha = 0^0$; *2* $-\ \alpha = 15^0$; *3* $-\ \alpha = 45^0$; *4* $-\ \alpha = 60^0$



The results obtained show that the temperature data in the solid material subsurface layer may be incorrect when the thickness of the layer is of the same order as the width of the heat wave front of the pyrolysis. Furthermore, heat transfer to the thermocouple leads to a significant loss of heat from the pyrolysis surface when it is close to the thermocouple, which can result in a change of the pyrolysis velocity near the bead.

The behaviour of this quantity is also dependent on the bead radius and pyrolysis velocity, which may be an additional source of the temperature discrepancy.

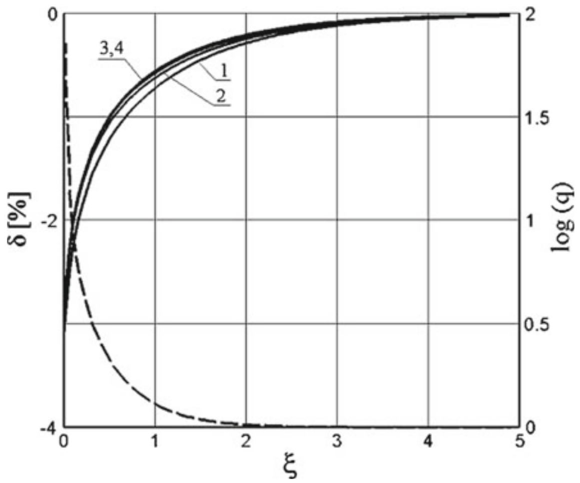


Fig. 13.19 Plot of $\delta(\xi)$ for $R_m = 45\,\mu\text{m}$ $(r_m/R_m = 0.2)$. *1* $- \alpha = 0^0$; *2* $- \alpha = 15^0$; *3* $- \alpha = 45^0$; *4* $- \alpha = 60^0$

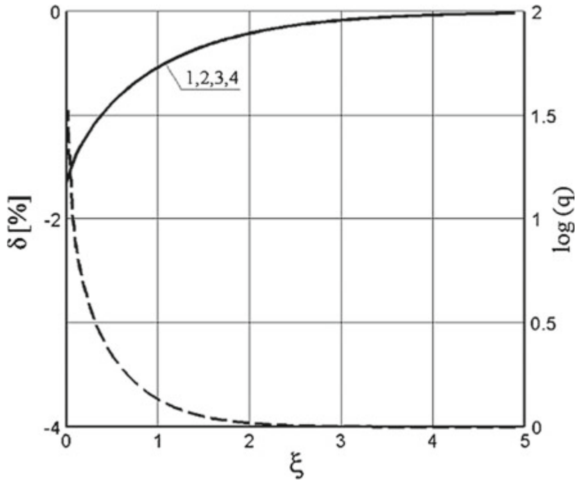


Fig. 13.20 Plot of $\delta(\xi)$ for $R_m = 120\,\mu\text{m}$ $(r_m/R_m = 0.2)$. *1* $- \alpha = 0^0$; *2* $- \alpha = 15^0$; *3* $- \alpha = 45^0$; *4* $- \alpha = 60^0$

Conclusions

- Numerical experiments demonstrate that temperature measurements of solid substances by thermocouples may be highly inaccurate, and that the discrepancy is dependent on both the pyrolysis velocity and the geometric size of the thermocouple.
- The results obtained can be used for correcting temperature measurements of pyrolized solids by thermocouples.

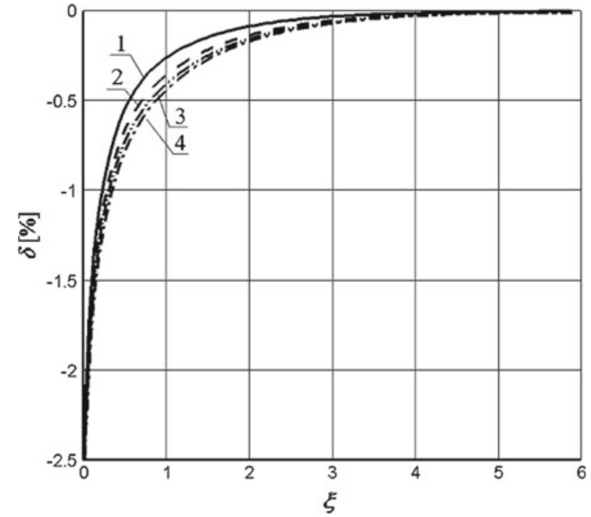


Fig. 13.21 Plot of $\delta(\xi)$ for $R_m = 400\,\mu\text{m}$ ($r_b = 0.1\,[\text{sm/s}]$; $r_m/R_m = 0.2$). *1* $- \alpha = 0^0$; *2* $- \alpha = 15^0$; *3* $- \alpha = 45^0$; *4* $- \alpha = 60^0$

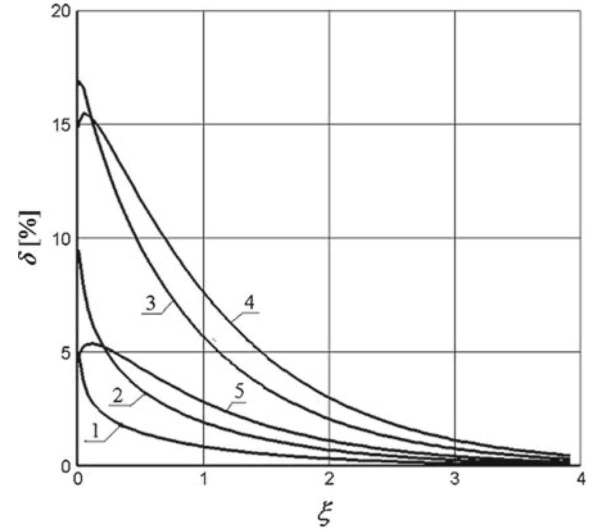


Fig. 13.22 Plot of $\delta(\xi)$ for $R_m = 400\,\mu\text{m}$ ($r_b = 0.1\,[\text{sm/s}]$; $r_m/R_m = 0.2$). *1* $- \alpha = 0^0$; *2* $- \alpha = 15^0$; *3* $- \alpha = 45^0$; *4* $- \alpha = 60^0$

13.6 Numerical Modeling of Nanopore Formation in Aluminium Oxide Films

13.6.1 Introduction

Fabrication of ordered nanostructures has received a great deal of attention due to its potential application in various fields, such as in electronic, photonic, magnetic, biochemical devices, etc. Anodic alumina is known as a typical nanoporous material that shows a tendency to form self-organized pore arrays of triangular symmetry, one that has been extensively studied, especially with bulk aluminum foils. Alumina pores grown with aluminum films that are deposited on foreign substrates would potentially offer much broader application than those on bulk aluminum foils.

Although a general description of anodization electrochemistry can be found in standard references, the precise chemical, physical, and electrical processes occurring in porous oxide growth during aluminum anodization are still not well understood. Mechanistic and kinetic investigations of the aluminum anodization system have established the presence of two dynamic, coupled interfaces (metal oxide and oxide electrolyte), but the nature of their coupling and their roles in the pore formation mechanism remain largely unknown.

One such mechanism is the dependence of the activation energies of the interfacial reactions on the Laplace pressure, arising at the curved interfaces due to surface energy. Although the surface energy is known to have an important stabilizing influence on the development of patterns observed in other electrochemical systems, such as porous silicon and electropolished aluminum, its effect on the formation of porous aluminum oxide has not yet been examined.

Another possibly significant factor in porous aluminum oxide formation is the elastic stress caused by the volume expansion associated with the oxidation reaction at the metal-oxide interface.

13.6.2 Mathematical Model

One mathematical model of nanopore formation on the surface of aluminum oxide was described in detail in Sample and Golovin (2006), Singh et al. (2006). Here, we give a concise excerption related to the model from these papers.

Since the electrical conductivities of the aluminum and the electrolyte are much larger than that of the oxide, it can be assumed that the main part of the voltage drop occurs in the oxide layer. Consequently, the potential from the metal side of the metal-oxide interface is fixed at the applied voltage V, and the potential from the electrolyte side of the oxide-electrolyte interface is fixed at zero. The charge flux, or current density, **J** in the oxide layer is given by

$$\mathbf{J} = -\sigma \mathrm{grad}\, \varphi,$$

where $\varphi(x, y, z, t)$ is the electric potential and σ is the oxide conductivity that, for simplicity, is assumed to be constant. The conservation of charge in the oxide layer,

$$\mathrm{div} \cdot \boldsymbol{J} = 0,$$

then reduces to the Laplace equation

$$\nabla^2 \varphi = 0. \tag{13.30}$$

The perturbation of the electric field due to ion migration is neglected, and at the metal-oxide interface, the effect of the electric double layer is ignored, as well as the dependence of the oxidation reaction rate on the electric potential (due to the low activation energy of the oxidation reaction). The potential at this interface is simply fixed at the applied voltage,

$$\varphi = V \text{ on } z = \xi^1(x, y, t), \tag{13.31}$$

where $\xi^1(x, y, t)$ is an evolving metal-oxide interface. At an oxide-electrolyte interface $\xi^2(x, y, t)$, however, the double layer induces a jump in the potential across the interface. The electric current across this interface is sustained by the transport of oxygen or hydroxyl ions and depends exponentially on the potential jump, as prescribed by the Butler–Volmer relation,

$$-\sigma \frac{\partial \varphi}{\partial \boldsymbol{n}} = k_+ e^{\alpha\varphi} - k_- e^{-\alpha\varphi} \quad \text{on } z = \xi^2(x, y, t), \tag{13.32}$$

where $k_\pm$ are kinetic coefficients characterizing the interfacial current of oxygen or hydroxyl ions produced by the forward (+) and reverse (−) reactions, respectively. The constant $\alpha = q_e/(2k_B T)$, where q_e is the electron charge, k_B is the Boltzmann constant, T is the absolute temperature, and $1/2$ is the symmetry factor, assumed equal for the forward and reverse reactions. The differential operator $\partial/(\partial \boldsymbol{n})$ is the normal derivative, where $\boldsymbol{n}$ is the outward normal pointing into the electrolyte for the surface $z = \xi^2(x, y, t)$.

The normal velocity $\boldsymbol{v}_n = (v_n^1,\ v_n^2)$ of an interface is given by

$$\begin{aligned} v_n^1 &= a\sigma \frac{\partial \varphi}{\partial \boldsymbol{n}} \quad \text{on } z = \xi^1(x, y, t), \\ v_n^2 &= -b_+ k_+ e^{\alpha\varphi} + b_- k_- e^{-\alpha\varphi} \quad \text{on } z = \xi^2(x, y, t), \end{aligned} \tag{13.33}$$

where a and b_+, b_- are Faradaic coefficients relating the rate of the interface motion to the interfacial currents.

We now study the stability of the stationary basic state of the oxide layer. The steady velocity v_s is used to introduce a moving coordinate frame attached to the planar metal-oxide interface. In this frame,

$$\frac{\partial l_s}{\partial t} = 0, \quad l_s = \xi^2 - \xi^1.$$

The stationary basic-state solution in the moving frame is perturbed as

$$\begin{aligned} &\varphi = (U - E_s) + \tilde{\varphi}(z)\phi(x, y, t), \quad \xi^{(1)}(x, y, t) = 0 + \tilde{\xi}^{(1)}\phi(x, y, t), \\ &\xi^{(2)}(x, y, t) = l_s + \tilde{\xi}^{(2)}\phi(x, y, t), \end{aligned} \tag{13.34}$$

where the hats denote the perturbations, $\phi(x, y, t) = e^{i\boldsymbol{q}\cdot\boldsymbol{x}+\omega t}$ is the normal mode $\boldsymbol{x} = (x, y)$, $q = |\boldsymbol{q}|$, $\boldsymbol{q} = (q_x,\ q_y)$ is the wave vector, and q_x and q_y are the perturbation wave numbers in the x and y directions, respectively.

We insert the expansions (13.34) into the governing equations (13.30)–(13.33), linearize in the perturbed variables, and solve the resulting linear system to obtain the dispersion relation in the form,

$$\omega^2 + A(q)\omega + B(q) = 0, \tag{13.35}$$

where the coefficients $A(q)$ and $B(q)$ are also functions of the physical parameters.

The two roots of the quadratic equation (13.35) describe two modes $\omega_1(q)$ and $\omega_2(q)$. The wavelength selection exhibited in $\omega_1(q)$ is determined by the balance between the destabilizing effect of the electric-field-dependent chemical reactions and the stabilizing effect of the surface energy, expressed through the dependence of the activation energies on the Laplace pressure. The presence of the zero mode, $\omega_1(0) = 0$, indicates the translation symmetry of the problem, and a long-wave expansion of $\omega_1(q)$ gives

$$\omega_1(q) = a_1 q^2 - a_2 q^4 \quad \text{for } q \ll 1\ . \tag{13.36}$$

Finally, the solvability condition gives the Kuramoto–Sivashinsky equation

$$\partial_t u + a_1 \nabla^2 u + a_2 \nabla^4 u - \frac{v_s}{2}(\nabla u)^2 = 0. \tag{13.37}$$

Note that the coefficients a_1 and a_2 in (13.37) are determined by the long-wave limit of the dispersion relation (13.36). The nonlinear term describes the correction of the interface velocity projection in the z direction for a slightly slant interface. The Kuramoto–Sivashinsky equation (13.37) is a generic equation that describes the weakly nonlinear evolution of many systems characterized by translation invariance and a monotonic instability with the long-wave spectrum of the type (13.36). Solutions to the Kuramoto–Sivashinsky equation (13.37) are known to have the form of spatiotemporally chaotic cells, splitting and merging, and having a well-defined average size.

In the case of two variables x and y, Eq. (13.37) can be written as

$$u_t = -a[u_{xx} + u_{yy}] - b[u_{xxxx} + 2u_{xxyy} + u_{yyyy}] + c[u_x u_x + u_y u_y]. \tag{13.38}$$

Using periodic boundary conditions allows one to model the process on the whole surface composed of equal $d \times d$ cells:

$$u(x + d, y, t) = u(x, y, t); \quad u(x, y + d, t) = u(x, y, t).$$

The initial condition for (13.38) is defined as $u(x, y, 0) = l_s$.

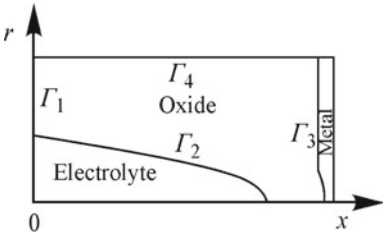


Fig. 13.23 Scheme of nanopore formation

We consider an axially symmetric problem whose domain $D = \{\Gamma_1(r, t) \leq x \leq \Gamma_3(r, t), \Gamma_2(x, t) \leq r \leq \Gamma_4\}$ has left, right and bottom movable boundary segments, shown schematically in Fig. 13.23. As the process is axially symmetric, the function φ is dependent on x, r, that is, $\varphi = \varphi(x, r)$, and consequently the Laplace equation (13.30) in the coordinates x, r is presented as

$$\frac{\partial^2}{\partial x^2}(r\varphi) + \frac{\partial}{\partial r}(r\frac{\partial \varphi}{\partial r}) = 0 \tag{13.39}$$

with boundary conditions

$$\varphi\Big|_{\Gamma_1} = \varphi\Big|_{\Gamma_2} = 0; \quad \varphi\Big|_{\Gamma_3} = V; \quad \frac{\partial \varphi}{\partial r}\Bigg|_{\Gamma_4} = 0, \tag{13.40}$$

where V – is specified voltage.

Electric current density $\boldsymbol{J}_c$ into the oxide layer is defined as

$$\boldsymbol{J}_c = -[A\ \exp(k_d E_c) - B\ \exp(k_c E_c)]\boldsymbol{n}_c,$$

where $E_c = |grad\,\varphi|$, $\boldsymbol{n}_c$ is the unit normal vector to the interface, $A,\ B,\ k_d,\ k_c$ are empirical coefficients.

Electric current density $\boldsymbol{J}_m$ from the oxide–metal interface Γ_3 is defined as

$$\boldsymbol{J}_m = \boldsymbol{J}_c \frac{E_m}{E_c},$$

where $E_m = |grad\,\varphi|$ on the interface Γ_3.

Displacement of the electrolyte–oxide interface caused by $\boldsymbol{J}_c$, $\boldsymbol{J}_m$ is defined by equations

$$\frac{d\boldsymbol{R}_c}{dt} = [\alpha A\ \exp(k_d E_c) - \beta B\ \exp(k_c E_c)]\boldsymbol{n}_c, \tag{13.41}$$

$$\frac{d\boldsymbol{R}_m}{dt} = -\gamma\,\boldsymbol{J}_m, \tag{13.42}$$

where $\boldsymbol{R}_c,\ \boldsymbol{R}_m$ are the radius–vectors of the points at the electrolyte–oxide interface and oxide–metal, respectively; $\alpha,\ \beta$ are coefficients satisfying Faraday relations; γ is an empirical coefficient.

13.6.3 Numerical Approximation

Equation (13.39) with boundary conditions (13.40) was solved through an iterative process. For this purpose, Eq. (13.39) was replaced by the following parabolic equation:

$$\frac{\partial \varphi}{\partial t} = \frac{\partial^2}{\partial x^2}(r\varphi) + \frac{\partial}{\partial r}\left(r\frac{\partial \varphi}{\partial r}\right), \tag{13.43}$$

which was written in the integral form

$$\int\limits_S \frac{\partial}{\partial t}\varphi\, dS - \oint\limits_\Gamma \boldsymbol{J}\, d\boldsymbol{l} = 0, \tag{13.44}$$

where $\boldsymbol{J} = \operatorname{grad} r\varphi$ is the current flux, and $\boldsymbol{l}$ is a normal to the boundary segment Γ of an area S. In the domain D, a two-dimensional quadrilateral grid was generated. Equation (13.44) was approximated by the following scheme:

$$\begin{aligned}&\frac{\varphi_{i,j}^{n+1} - \varphi_{i,j}^{n}}{\tau} S_{i,j} - \big[(\boldsymbol{J}\cdot\boldsymbol{l})_{i+1/2}^{n+1} - (\boldsymbol{J}\cdot\boldsymbol{l})_{i-1/2}^{n+1} + \\ &+ (\boldsymbol{J}\cdot\boldsymbol{l})_{j+1/2}^{n+1} - (\boldsymbol{J}\cdot\boldsymbol{l})_{j-1/2}^{n+1}\big] = 0,\end{aligned} \tag{13.45}$$

where τ is a time step.

For solving system (13.45), the following iterative scheme based on the introduction of a pseudo time s was used:

$$\begin{aligned}&[\frac{\varphi_{i,j}^{n+1,s+1} - \varphi_{i,j}^{n+1,s}}{\tau_1} + \frac{\varphi_{i,j}^{n+1,s} - \varphi_{i,j}^{n}}{\tau}]S_{i,j} - \big[(\boldsymbol{J}\cdot\boldsymbol{l})_{i+1/2}^{n+1,s} - \\ &-(\boldsymbol{J}\cdot\boldsymbol{l})_{i-1/2}^{n+1,s} + (\boldsymbol{E}\cdot\boldsymbol{l})_{j+1/2}^{n+1,s} - (\boldsymbol{J}\cdot\boldsymbol{l})_{j-1/2}^{n+1,s}\big] = 0,\end{aligned} \tag{13.46}$$

where $S_{i,j}$ is the area of a cell, $\varphi_{i,j}^n$ is the middle value of φ on the nth layer, and τ is the pseudo time step.

For a realization of (13.46), a splitting scheme was used:

$$\begin{aligned}&\frac{\delta^{s+1/2} - \delta^{s}}{\tau_1} S_{i,j} - \Lambda_1 \delta^{s+1/2} = \frac{\varphi_{i,j}^{n+1,s} - \varphi_{i,j}^{n}}{\tau} S_{i,j} - \big[(\boldsymbol{J}\cdot\boldsymbol{l})_{i+1/2}^{n+1,s} \\ &-(\boldsymbol{J}\cdot\boldsymbol{l})_{i-1/2}^{n+1,s} + (\boldsymbol{J}\cdot\boldsymbol{l})_{j+1/2}^{n+1,s} - (\boldsymbol{J}\cdot\boldsymbol{l})_{j-1/2}^{n+1,s}\big], \\ &\frac{\delta^{s+1} - \delta^{s+1/2}}{\tau_1} S_{i,j} - \Lambda_2 \delta^{s+1} = 0, \quad \varphi^{n+1,s+1} = \varphi^{n+1,s} + \delta^{s+1}.\end{aligned} \tag{13.47}$$

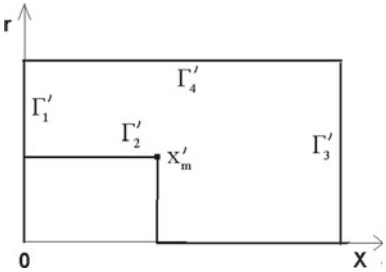


Fig. 13.24 Computational domain

Here, δ^s is a correction to φ; Λ_1, Λ_2 are difference operators of second derivatives related to corresponding directions. Boundary conditions for δ^s were specified as follows: $\delta^s = 0$ on Γ_1, Γ_2, Γ_3, on Γ_4 –"soft" boundary condition.

13.6.4 Grid Generation

For solving Eq. (13.39) numerically, an adaptive quadrilateral grid with node clustering near the interface Γ_2 was generated in D. A computational domain Ξ^2 was specified as a rectangle with a cut (see Fig. 13.24). The reference grid in Ξ^2 is uniform with square cells. The boundary segments Γ_1', Γ_2', Γ_3', and Γ_4' in the computational domain Ξ^2 were mapped onto the corresponding boundary segments Γ_1, Γ_2, Γ_3, and Γ_4 in the domain D. The corner point at the segment Γ_2' in Ξ^2 was mapped on the point x_m on the curve Γ_2 in D. The point x_m was defined in such way that the cells were not too skewed (Fig. 13.25). For convenience, the interface Γ_2 was partitioned into two segments Γ_2 and Γ_2^{ox}, $\Gamma_2 = \Gamma_2 \cup \Gamma_2^{ox}$. The nodes of the reference grid in Ξ^2 were mapped in the domain D by means of the numerical solution of two-dimensional inverted diffusion equations (11.15) for $n = 2$ with respect to a spherical control metric:

$$g_{22}^{s\xi}\frac{\partial^2 s^k}{\partial\xi^1\partial\xi^1} - 2g_{12}^{s\xi}\frac{\partial^2 s^k}{\partial\xi^1\partial\xi^2} + g_{11}^{s\xi}\frac{\partial^2 s^k}{\partial\xi^2\partial\xi^2} = \frac{J^2}{Z[u](\boldsymbol{s})}\frac{\partial Z[u](\boldsymbol{s})}{\partial s^k}, \qquad (13.48)$$
$$k = 1, 2,$$

where s^1, s^2 is identified with x, r, respectively.

The control function $Z(u(s))$ for providing grid clustering near the interface $\Gamma_2 = \Gamma_2 \cup \Gamma_2^{ox}$ was defined in the following form:

$$\begin{aligned}
&Z[u(\boldsymbol{s})] = u_1(\boldsymbol{s})u_2(\boldsymbol{s}),\\
&u_1(\boldsymbol{s}) = \alpha + (\rho_{\min})^2, \quad 10^{-4} \le \alpha \le 10^{-2},\\
&u_2(\boldsymbol{s}) = \beta + (\rho_{\min}^{ox})^{1.2}, \quad 10^{-2} \le \beta \le 10^{-1},\\
&\rho_{\min} = \min_i \rho(\boldsymbol{s}, \text{array}_i),\\
&\rho_{\min}^{ox} = \min_j \rho(\boldsymbol{s}, \text{array}_j^{ox}),
\end{aligned}$$

where array_i and array_j^{ox} are discretizations of Γ_2 and Γ_2^{ox}, respectively, α and β are parameters.

13.6.5 Numerical Experiments

Numerical computation of a nonstationary nanopore formation was carrying out in the following way. It was assumed initially that the nanopore shape (interface Γ_2) was specified. Then, an adaptive numerical grid was generated in the domain D, and Eq. (13.40) was solved on this grid, after which with the help of Eqs. (13.41) and (13.42), new locations of interfaces Γ_1, Γ_2, and Γ_3 were defined at time $t + \Delta t$. Then, a new grid was generated in D and the process continued on. The number of nodes was taken from 3600 up to 7500, which provided a calculation accuracy on nearly 0.5%.

Figures 13.26, 13.27 and 13.28 demonstrate some numerical results obtained by Rychkov (Liseikin et al. 2011a, b) at time $t = 0$, $t = 10\,\text{min}$, and $t = 20\,\text{min}$.

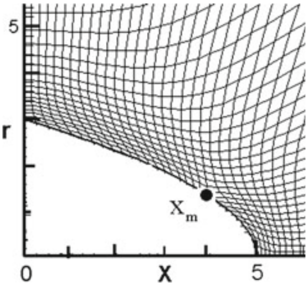


Fig. 13.25 Fragment of a numerical grid

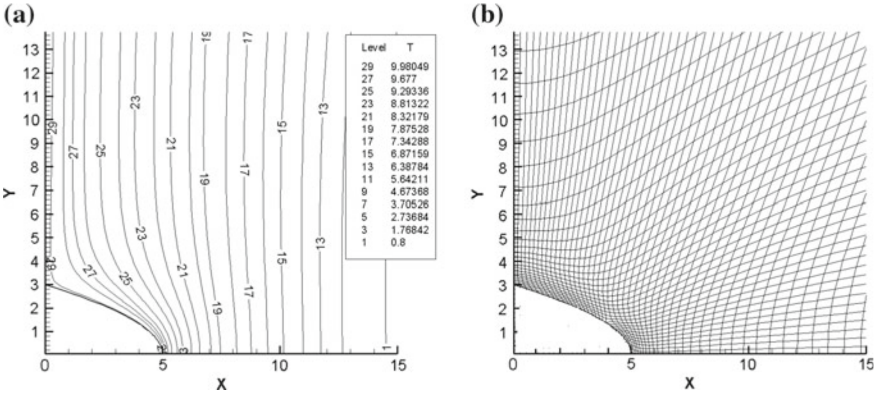


Fig. 13.26 Illustrations: **a** – isolines of field potential, **b** – numerical grid at time t = 0 min

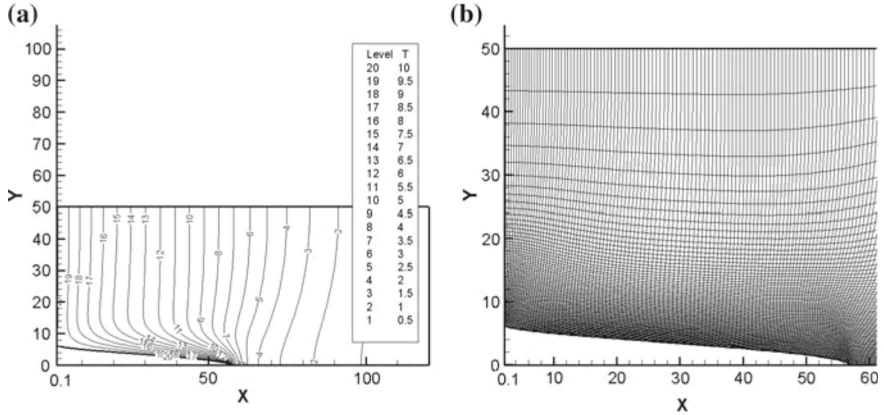


Fig. 13.27 Illustrations: **a** – isolines of field potential, **b** – numerical grid at time t = 10 min

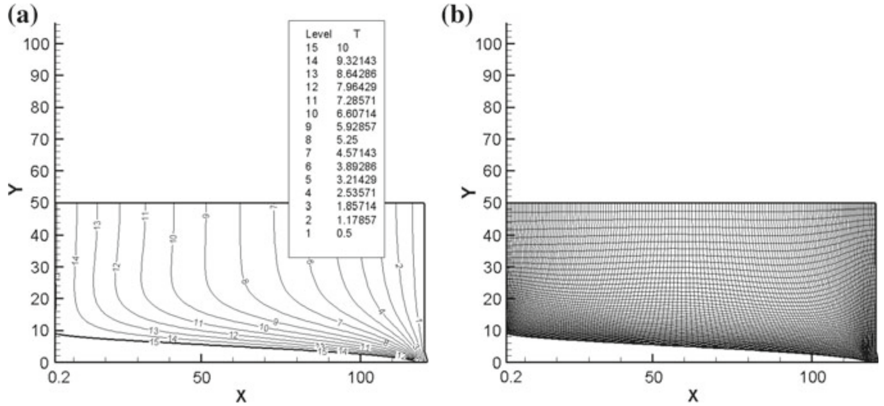


Fig. 13.28 Illustrations: **a** – isolines of field potential, **b** – numerical grid at time t = 20 min

13.7 Grids for Boundary Immersing Methods

13.7.1 Introduction

Despite the considerable success achieved in grid-generation technologies, development of more efficient and sophisticated algorithms and computer codes for generating grids remains an important problem. Serious difficulties arise in grid generation in domains with complicated boundary geometries, especially in geometries with discretely defined segments and in cases when grids have to be adapted to singularities such as boundary and interior layers, shock-waves, detonation waves, combustion fronts, high-speed jets, and phase-transition zones. For domains with discretely specified boundaries, popular methods are based on applying the Delaunay criterion to the whole domain, or, if this is not possible, to some part of the domain. Such methods involve triangulation of the domain boundary, followed by triangulation of the interior of the domain.

For a domain with discretely specified boundary sections, an approach to grid generation is described here that does not involve the initial triangulation of its boundary. In this approach, a global numerical grid is first constructed in a larger domain, with mesh refinement near the specified boundary points. Next, boundary and interior cells of the domain under consideration are selected, which will comprise its numerical grid with immersed boundary points. As a result, a grid with thin cells near the boundary is obtained. The thinner these cells, the better the approximation of the domain. The global grid may be generated by using the inverted diffusion equations for a spherical metric tensor. Following this approach, one can generate grids with the help of other equations, for example, Beltrami equations. The approach is also suitable for domains with boundaries specified by an implicit analytic function $\varphi(\boldsymbol{x}) = 0$. Originally, the approach was formulated in Kofanov and Liseikin (2013).

This approach to grid generation is also suitable for generating grids for solving problems through boundary immersing methods which were originally founded by Peskin (1972), who developed the technique in 1972 to study blood flow around heart valves.

The term Immersed Boundary Methods designates the class of boundary methods for which the calculations are performed on a grid that does not conform to the shape of a domain. The boundary conditions on the domain are not imposed directly; instead, an extra term, called the forcing function, is added to the governing equations or the discrete numerical scheme is altered near the boundary.

13.7.2 Formulation of the Method

In accordance with Chap. 11 the inverted diffusion equations

$$B_n^{\xi}[x^k] = J^2 \frac{1}{Z(\boldsymbol{x})} \frac{\partial}{\partial x^k} Z(\boldsymbol{x}) \ , \quad k = 1, \dots, n \ , \tag{13.49}$$

where

$$B_n^{\xi}[y] = J^2 \sum_{i,j=1}^{n} g_{\xi x}^{ij} \frac{\partial^2 y}{\partial \xi^i \partial \xi^j}, \quad J = \det\left\{\frac{\partial x^i}{\partial \xi^j}\right\},$$

$$g_{\xi x}^{ij} = \sum_{k=1}^{n} \frac{\partial \xi^i}{\partial x^k} \frac{\partial \xi^j}{\partial x^k}, \quad i, j = 1, \dots, n \ ,$$

can be used for generating grids in an n-dimensional domain X^n with refinement in the zone where the control function $Z(\boldsymbol{x})$ is small and with rarefaction in the zone where $Z(\boldsymbol{x})$ is large. The control function in the examples considered in Chap. 11 was specified as

$$Z(\boldsymbol{x}) = \sum_{i=1}^{5} c_i \sigma_i[\varphi(\boldsymbol{x})], \quad c_i \geq 0, \tag{13.50}$$

and

$$Z(\boldsymbol{x}) = \sum_{i=1}^{5} \frac{c_i}{\sigma_i[\varphi(\boldsymbol{x})]}, \quad c_i \geq 0, \tag{13.51}$$

where $\sigma_i(t)$ are the following layer-type functions:

$$\begin{aligned} &\sigma_1(t) = |\tanh(t/\varepsilon)|^{\alpha} + \varepsilon_1, \quad \alpha > 0, \quad \sigma_2(t) = 1 - e^{-|t|/\varepsilon} + \varepsilon_1, \\ &\sigma_3(t) = 1 - \frac{\varepsilon^{\beta}}{(\varepsilon + |t|^{\alpha})^{\beta}} + \varepsilon_1, \quad \sigma_4(t) = |t|^{\alpha} + \varepsilon_1, \quad \alpha, \beta > 0, \\ &\sigma_5(t) = \frac{\ln(1 + |t|^{\alpha}\varepsilon^{-k})}{\ln(1 + (T_0)^{\alpha}\varepsilon^{-k})} + \varepsilon_1, \quad \alpha, k > 0, \end{aligned} \tag{13.52}$$

ε and ε_1 are small positive constants, i.e., $0 < \varepsilon, \varepsilon_1 \ll 1$. With such specification, the grid obtained by solving Eq. (13.49) is refined if the control function $Z(\boldsymbol{x})$ is defined by (13.50) and rarefied if $Z(\boldsymbol{x})$ is defined by (13.51) in the zones where the values of $\varphi(\boldsymbol{x})$ are small. In particular, for generating a grid with refinement near several points $\boldsymbol{x}_i$, $i = 1, \ldots, N$, the function $\varphi(\boldsymbol{x})$ in formula (13.50) can be defined as

$$\varphi(\boldsymbol{x}) = \min_{i=1,\ldots,N} \rho(\boldsymbol{x}, \boldsymbol{x}_i). \tag{13.53}$$

To generate a grid with rarefaction near the same points, the function $\varphi(\boldsymbol{x})$ in formula (13.51) can be defined by the same formula (13.53). In particular, functions (13.50) and (13.53) can be used to generate grids with refinement near a discretely specified set in the domain X^n.

To solve Eq. (13.49), the values of function (13.53) have to be found at all grid nodes. Accordingly, for each grid node, the nearest point from the given set $\boldsymbol{x}_i$, $i = 1, \ldots, N$ has to be found at each iteration step. The more points in this set, the higher the computational costs, which are especially critical in the three-dimensional case, when the grid cells can number in the billions. However, since the set of points $\boldsymbol{x}_i$, $i = 1, \ldots, N$, is given and remains unchanged, the search for a minimum in (13.53) can easily be parallelized over this set at a single iteration step. Moreover, since the shift in the grid after each iteration step is small, there is no need to verify all the points in the set in order to find the nearest one to each node at the next iteration. For example, for each $\boldsymbol{x}_i$, $i = 1, \ldots, N$, we can store the set of points lying in its δ-neighborhood for some $\delta > 0$ and verify only the points from this set, since the other points are farther away. In this approach, more storage is needed for such sets, but the CPU time required for grid generation can be reduced significantly.

Examples of such zones, which are typically found through numerical computations, are interfaces between different media, high-speed jets, combustion fronts, shock waves, detonation waves, etc. Let Γ denote the boundary of such a zone in

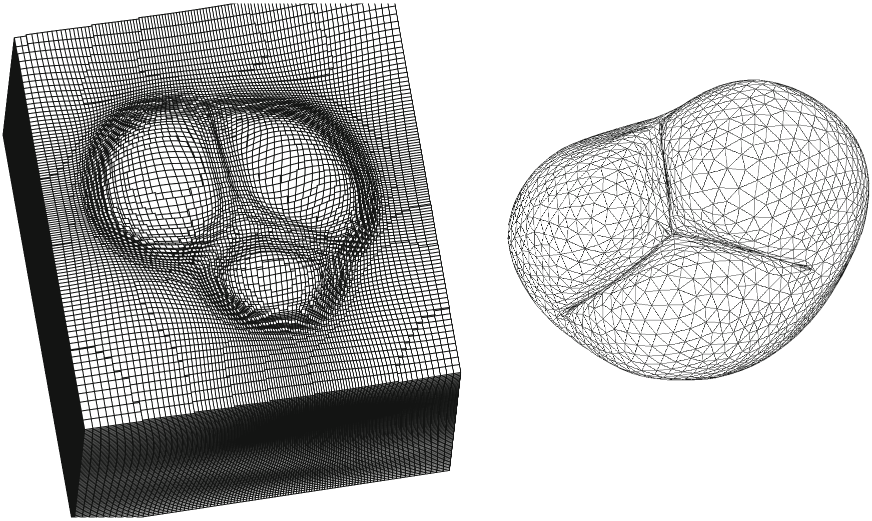


Fig. 13.29 A section of an adaptive grid with node clustering in the vicinity of a valve and blood vessel (*left*). A fragment of the heart valve surface (*right*)

some domain X^n and let B be a discrete set of points approximating Γ. To generate a grid in X^n with refinement near Γ, there is no need to first construct cells approximating Γ and then add cells of higher dimension covering the domain X^n. Instead, we can generate a global grid in X^n with refinement near B, as was proposed above. In this case, the cells containing the points of B approximate Γ better when the grid near B is finer and when Γ is better approximated by B.

This approach can be used to generate a grid in a physical domain X^n whose boundary is specified by a discrete set of points B. For engineering applications, the boundary surfaces defined by discrete data are widely adopted, for instance, in the presentations of complex molecules, human organs, and visualizations. For this purpose, the set B is enclosed in a domain Y^n of a simple shape, for example, in an n-dimensional parallelepiped (see Fig. 13.29 for a heart vessel with the valve closed). Then, a global grid with refinement in the neighborhood of B is constructed in this domain. Next, two sets of cells are determined, namely, the cells approximating a thin layer covering the boundary of X^n and the cells lying inside X^n. These cells comprise a grid in X^n. This approach to grid generation resembles the well-known octree method (Yerry and Shephard 1985). Note that the discrete 3D-models required for the generation of the adaptive grids demonstrated below are freely available for research purposes and can be found at http://www-roc.inria.fr/gam.ma/gamma/download/.

With an arbitrary chosen set B approximating the boundary Γ in a zone of interest within a domain X^n, we may not generate an adequately refined mesh. This occurs, for example, if there is a point that is far away from the nearest point in B, in which case, the mesh refinement lines go around this point and poorly approximate Γ

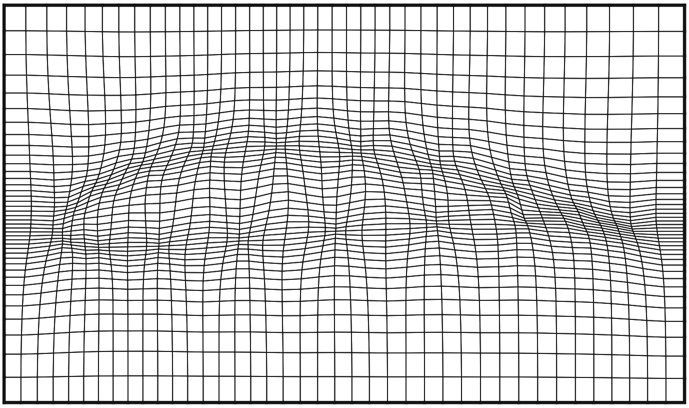


Fig. 13.30 Numerical grid for rarefied points

(Fig. 13.30). In such situations, more points have to be added to B, which can be accomplished by triangulating Γ by the points from B and then adding points to the resulting cells by applying standard methods. These operations can be executed using software packages, most of which can be found at http://www.robertschneiders.de/meshgeneration/software.html.

13.7.3 Determination of Boundary Cells

The union of all cells, each including at least one point from the set B approximating the boundary Γ of X^n or a certain zone of X^n, may not include all the points of Γ. In this case, for a more complete choice of cells approximating the layer covering Γ, we can set a small constant $c > 0$ and choose all the nodes of cells separated from B by a distance less than c. The grid cells containing these nodes comprise the set of cells approximating the layer covering Γ. The resulting configuration is a narrow strip containing Γ in the case of a two-dimensional domain, and a thin-walled layer in the case of a three-dimensional domain. The strip or the wall is thinner for smaller values of c. However, if c is too small, we can obtain a configuration with holes. Note also that the width of the strip, or the thickness of the wall, depends substantially on the constant ε in equations (13.12) and on the steepness of the function $\sigma_i(t)$ near $t = 0$. The functions $\sigma_1(t)$ and $\sigma_2(t)$ have the greatest steepness; hence, they ensure the smallest values of the width or thickness. Figure 13.31 demonstrates a boundary grid simulating the walls of an aorta and carotid arteries. The coordinates of the boundaries were taken from the site http://www-roc.inria.fr/gamma/gamma/download/ANATOMY/index0.php. Figure 13.32 illustrates a boundary grid for an aircraft.

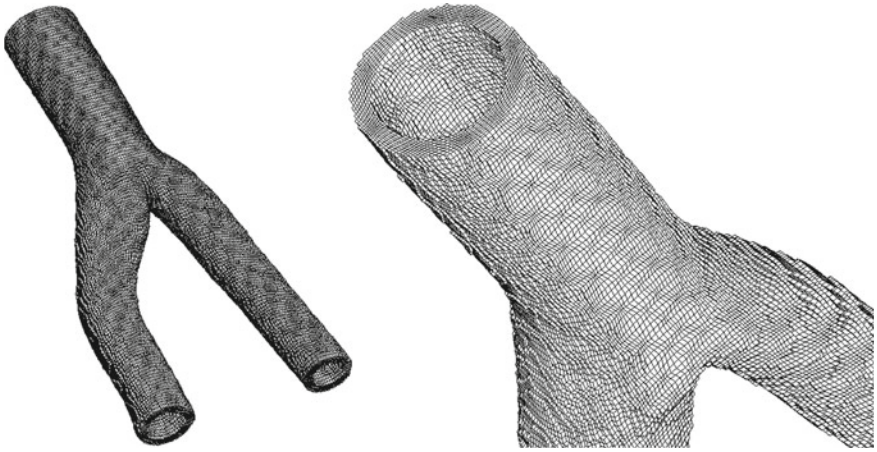


Fig. 13.31 Hexahedral mesh modelling the walls of an aorta and carotid arteries

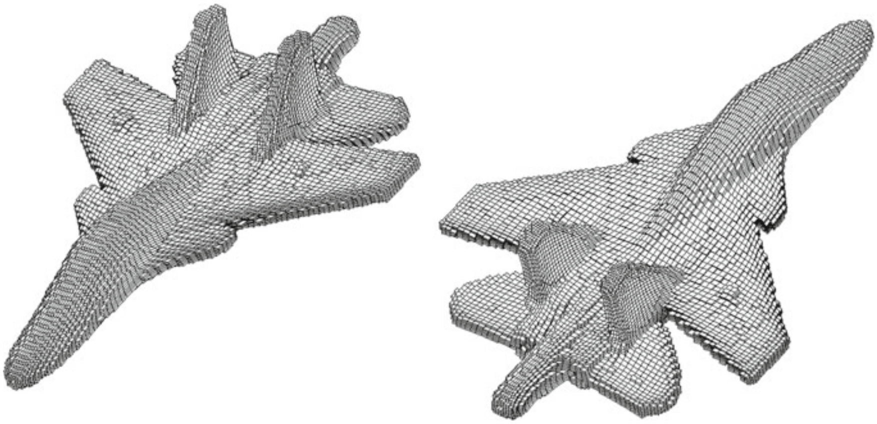


Fig. 13.32 A surface of an SU-34 aircraft formed by hexahedral cells

13.7.4 Algorithm for Determining Interior Cells

In the considered algorithm for mesh generation in X^n, we know all the cells adjacent to a given cell, i.e., the cells sharing a face in the case of a three-dimensional domain, and an edge in the case of a two-dimensional domain. If this cell is interior, it is adjacent to six cells in the three-dimensional case and to four cells in the two-dimensional case. Therefore, after the boundary cells of X^n are chosen, to select all interior cells of this domain, we begin by finding at least one interior cell. The other cells are found by means of the adjacency relationship, in a manner similar to the advancing-front method, adjacent cells are sequentially added to the set of interior cells if they do not belong to the set of chosen boundary cells. If X^n is a single-connected domain, all its interior cells are found in this way. If X^n is multi-connected, then this operation is executed in each single-connected subdomain.

Alternatively, we can obtain the interior cells of X^n by using the same approach to find cells lying outside of X^n and those covering the boundary of X^n, and eliminating them from the set of all cells.

13.7.5 Mesh Adaptation

In the above algorithm for mesh generation, cells are refined near a discretely defined boundary, specifically, near the boundary of the domain. Mesh refinement in other zones can be achieved by applying the same inverted diffusion equations (13.49) with

$$Z(\boldsymbol{x}) = \prod_{l=1}^{k} Z_l(\boldsymbol{x}),$$

where $Z_l(\boldsymbol{x})$ is a control function for mesh refinement in the lth zone. Figures 13.33 and 13.34 give an example of such a grid generated in the domain containing an aircraft with mesh refinement near the boundary of the aircraft and around a surface that qualitatively simulates a shock wave.

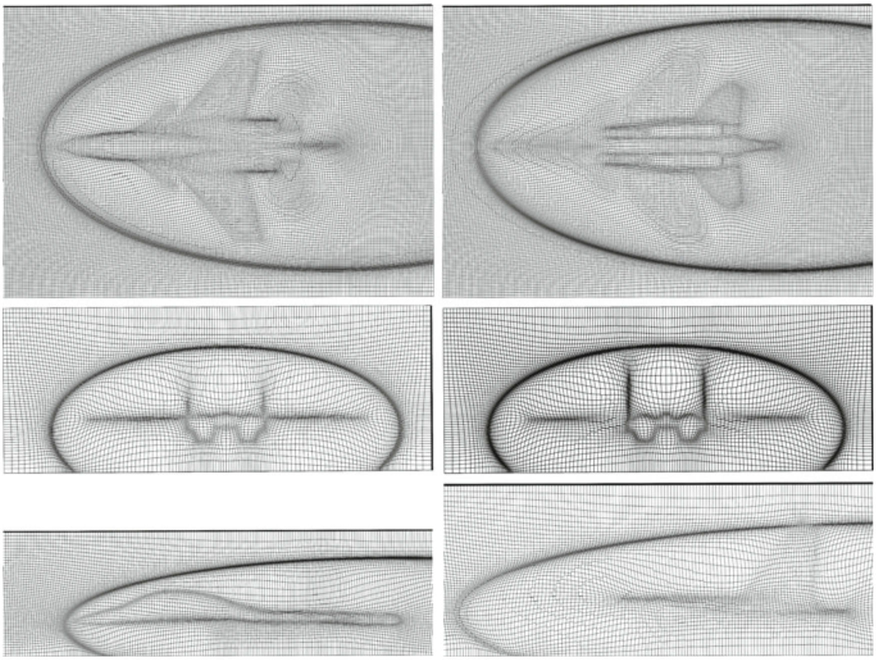


Fig. 13.33 Longitudinal and transversal sections of an adaptive grid in the vicinity of an aircraft

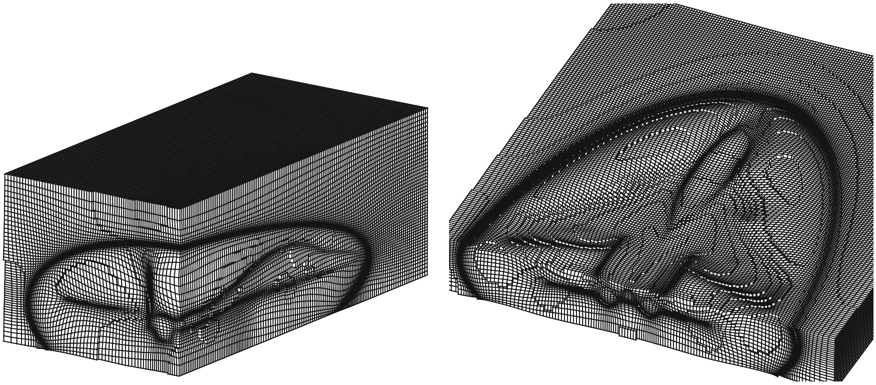


Fig. 13.34 Fragments of an adaptive grid in the vicinity of an airplane

This subsection only gives examples of hexahedral three-dimensional grids. Grids with cells of other shapes can be generated in a similar manner. For this purpose, a reference grid with the required cells is to be specified in the computational domain Ξ^n.

References

Asay, B. W., Son, S. F., Dickson, P. M., Smilowitz, L. B., & Henson, B. F. (2005). An investigation of the dynamic response of thermocouples in inert and reacted condensed phase energetic materials. *Propellants, Explosives, Pyrotechnics*, *3*(30), 199–208.

Azarenok, B. N. (2000). Realization of second-order Godunov's scheme. *Computer Methods in Applied Mechanics and Engineering*, *189*, 1031–1052.

Azarenok, B. N., & Ivanenko, S. A. (2001). Application of moving adaptive grids for simulation of supersonic gas flow. *Computational Fluid Dynamics Journal Japanese*, *10*(3), 400–404.

Azarenok, B. N., & Tang, T. (2005). Second-order Godunov-type scheme for reactive flow calculations on moving meshes. *Journal of Computational Physics*, *206*, 48–80.

Belotserkovsky, O. M., & Davidov, Yu M. (1982). *Method of Large Particles in Gas Dynamics*. Moscow: Nauka. (Russian).

Borovkova, T. V., Yeliseyev, Y. N., & Lopukhov, I. I. (2008). Mathematical modeling of contact thermocouple. *Physics of Particles and Nuclei Letters*, *1*(3), 274–277.

Colella, P., & Woodward, P. R. (1984). The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, *54*, 115–173.

Franco, G. A., Caron, E., & Wells, M. A. (2007). Quantification of the surface temperature discrepancy caused by surface thermocouples and methods for compensation. *Metallurgical and Materials Transactions B*, *338*, 949–956.

Glasser, A. H., Kitaeva, I. A., Likhanova, Yu V, Liseikin, V. D., & Lukin, V. S. (2005a). Specification of monitor metrics for generating balanced numerical grids. *Joint Bulletin of NCC and IIS, Numerical Analysis*, *13*, 1–13.

Glasser, A. H., Liseikin, V. D., & Kitaeva, I. A. (2005b). Control of grid properties with the help of monitor metrics. *Computational Mathematics and Mathematical Physics*, *45*(8), 1416–1432.

Kofanov, A. V., & Liseikin, V. D. (2013). Grid construction for discretely defined configurations. *Computational Mathematics and Mathematical Physics*, *53*(6), 938–945.

Leffe, M., Touze, D., & Alessandrini, B. (2010): SPH modeling of shallow-water coastal flows. Journal of Hydraulic Research **48** (Extra Issue), 118–125.

Liseikin, V. D. (2001). *Layer resolving grids and transformations for singular perturbation problems*. Utrecht: VSP.

Liseikin, V. D., Rychkov, A. D., & Kofanov, A. V. (2011a). *Technology of adaptive grids for numerical solution of applied problems*. Novosibirsk: Novosibirsk State University. (Russian).

Liseikin, V. D., Rychkov, A. D., & Kofanov, A. V. (2011b). Applications of a comprehensive grid method to solution of three-dimensional boundary value problems. *Journal of Computational Physics*, *230*, 7755–7774.

Peskin, C. S. (1972). Flow patterns around heart valves: A digital computer method for solving the equations of motion. Ph.D. Thesis, Albert Einstein College of Medicine, July, 211.

Rychkov, A. D., Liseikin, V. D., & Kofanov, A. V. (2010). On inaccuracy of temperature measurements via thermocouples in the case of large gradients. *Computational Technologies*, *4*, 25–35.

Sample, C., Golovin, A. A. (2006). Formation of porous metal oxides in the anodization process. *Physical Review. Part. E*, *74*, 041606.

Shimozono, T., Sato, S., Okayasy, A., Tajima, Y., Friz, H.M., Liu, H., & Takagawa, T. (2012). Propagation and inundation characteristics of the 2011 Tohoku tsunami on the central Sanriku Coast. *Coastal Engineering Journal*, *54*(1) 1250004 (17 pages). World Scientific Publishing Company and Japan Society of Civil Engineers. doi:10.1142/S0578563412500040.

Singh, G. K., Golovin, A. A., & Aranson, I. S. (2006). Formation of self-organized nanoscale porous structures in anodic aluminum oxide. *Physical Review. Part. B*, *73*(205422), 1–12.

Vinokur, M. (1989). An analysis of finite-difference and finite-volume formulations of conservation laws. *Journal of Computational Physics*, *81*, 1–52.

Yanenko, N. N. (1971). *The method of fractional steps: The solution of problems of mathematical physics in several variables*. New York: Springer.

Yerry, M. A., & Shephard, M. S. (1985). Automatic three-dimensional mesh generation for three-dimensional solids. *Computers and Structures*, *20*, 31–39.

Zenin, A. A. (1962). On errors of temperature measurements of flames via thermocouples. *Journal of Engineering Physics and Thermophysics.*, *5*(5), 67–74.

Zenin, A. A. (1963). On temporal response of thermocouples in a solid fuel combustion wave. *Journal of Applied Mechanics and Technical Physics.*, *5*, 125–131.

Index

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0