计算流体力学的新进展

——理论、方法和应用

F. Dubois　　邬华谟　编

高等教育出版社

内 容 提 要

近来计算流体力学已日臻成熟，并在工程设计领域得到广泛应用。1999 年 9 月中法应用数学研究所主办了计算流体力学学术会议. 本书收集了其中 3 个系列讲座的讲稿以及部分大会邀请报告，内容涉及当前计算流体力学的许多活跃的研究方向和最新研究成果。本书可供该领域的学者、学生以及工程技术人员学习参考。

图书在版编目（CIP）数据

计算流体力学的新进展：理论、方法和应用 / 邬华谟，（法）杜布瓦编．—北京：高等教育出版社，2001．8
ISBN 7－04－010056－8

Ⅰ．计…　Ⅱ．①李…　②邬…③杜…　Ⅲ．计算流体力学—国际学术会议—文集　Ⅳ．035－53

中国版本图书馆CIP数据核字（2001）第25450号

计算流体力学的新进展——理论、方法和应用
F.Dubois邬华谟　编

出版发行	高等教育出版社		
社　　址	北京市东城区沙滩后街55号	邮政编码	100009
电　　话	010—64054588	传　　真	010—64014048
网　　址	http://www.hep.edu.cn http://www.hep.com.cn		
经　　销	新华书店北京发行所		
印　　刷	北京联华印刷厂		
开　　本	880×1230　1/32	版　　次	2001 年 8 月第 1 版
印　　张	12. 625	印　　次	2001 年 8 月第 1 次印刷
字　　数	360 000	定　　价	30. 60 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

版权所有　侵权必究

前 言

由中法应用数学研究所(ISFMA)主办的计算流体力学学术会议于1999年9月6日至17日在西安交通大学举行. 会议的目的是联系高速飞行器和透平机械的设计与计算要求, 总结并介绍计算流体力学方面的最新研究成果及计算方法, 促进中法两国学术界和有关应用部门间的交流与协作, 推动计算流体力学在中国的发展. 参加会议的有计算流体力学方面的学者及实际工作人员、高校教师、硕士及博士研究生、博士后等共八十余人, 其中有7位是来自法国的代表.

会议包括4个12小时的系列讲座, 14个1小时的大会邀请报告以及若干个专题交流报告. 本书中收集了3个系列讲座的讲稿以及部分大会邀请报告与专题交流报告的书面材料. 所收集的内容涉及当前计算流体力学的许多活跃的研究方向, 对于在计算流体力学领域工作的学者、学生及实际工作人员相信会有较大的帮助.

我们衷心感谢李大潜教授发起筹备这次会议, 感谢西安交通大学数学系李开泰教授等同志出色地组织了本次会议. 我们还要感谢给予我们各方面帮助的同事们, 特别是在学术活动和会议组织方面作了大量工作的本次会议科学委员会的国内外委员和会议组织委员会的委员们. 我们感谢复旦大学数学系特别是蔡志杰博士在整理出版本书中所做的大量细致的工作.

这次会议得到了法国驻中国大使馆、法国驻沪总领事馆、教育部数学研究与高等人才培养中心、国家自然科学基金委员会、复旦大学、西安交通大学、中国航空计算技术研究所及西北工业大学的支持和资助, 特致谢忱. 最后, 我们对高等教育出版社在出版本书过程中所给予的热情帮助表示诚挚的谢意.

F.Dubois　邬华谟

1999年11月

Preface

The ISFMA Symposium on Computational Fluid Dynamics was held at Xi'an Jiaotong Univresity, Xi'an, China on September 6-17, 1999. According to the demand on the design and computation of high-speed air vehicles and turbine machinery, the purpose of the symposium was to introduce the present state of Computational Fluid Dynamics (CFD); to promote the exchange and collaboration between CFD scientists from China and France and to push forward the development of CFD in China. Over 80 participants attended the symposium. Among them there were CFD researchers and practicians, teachers, graduate students and post-graduate fellows from universities and research institutes, including 7 participants came from France.

The activities of the symposium included four 12-hour plenary lectures; 14 one-hour invited talks and some talks on special topics in CFD. This volume collected the written version of 3 plenary lectures, part of invited talks and talks on special topics. The collected material involves newest results in many important directions for CFD. It will be beneficial to CFD researchers, students and practicians.

We would like to thank Prof. Li Tatsien for his initiation of organizing this symposium. We give our thanks to Prof. Li Kaitai and his colleagues from Xi'an Jiaotong University for their successful organization. We would like to take this opportunity to express our appreciation to all colleaques, especially to the members of the Scientific Committee and the Organizing Committee. Our sincere thanks are due to the Department of Mathematics of Fudan University, especially to Dr. Cai Zhijie for his heavy work in editing this volume.

The symposium was supported by the French Embassy in Beijing, the Consulate General of France in Shanghai, the Mathematical Center of the Education Ministry of China, the National Natural Science Foundation of China, Fudan University, Xi'an Jiaotong University, Aeronautics Computing Technique Research Institute, Northwestern Polytechnical University. We are indebted to all of these supporters. Finally, we would like to give our sincere thanks to Higher Education Press for their help in publishing this book.

François Dubois   Wu Huamo

November, 1999

此为试读，需要完整PDF请访问：www.ertongbook.com

目　　录

Contents

Mesh Generation and Related Topics Application to Finite Elements

P. J. Frey P. L. George

ABSTRACT

The numerical simulation of the physical problems expressed in terms of *partial differential equations* (so-called PDEs) using a *finite element, finite volume, boundary element*, or any other numerical method requires the discretization of the domain of interest into a set of elements, i.e. a mesh. The differential equations are approximated by a set of algebraic equations on this mesh, this set being then solved to provide the approximate solution of the partial differential system over the field. The discretization requires some properties for the solution to be exploitable and must at least conform to all domain boundaries in order to accurately represent boundary conditions. Consequently, the mesh generation stage, as an essential pre-requisite, is of utmost importance in the computational schemes, as it is related to the convergence of the computational scheme as well as to the accuracy of the numerical solutions.

There is indeed a large variety of algorithms suitable to produce such meshes. Some of these methods are designed to handle specific geometric situations while others can be used in a more general context. User-driven, semi-automatic as well as fully automatic methods exist leading to structured, unstructured or mixed meshes. The mesh generation problems are mainly related to the boundary meshing (line, curve

and surface meshing) and domain meshing issues (planar domain or volumetric domain).

Numerous computational issues must be carefully addressed for designing reliable and robust meshing algorithms. These issues concern computer-related data structures and algorithms (low-level routines) as well as advanced data structures and computational schemes (high-level routines). In this regard, basic computational tools, geometric and discrete geometric notions, computational and mesh data structures, element and mesh definitions are of significant importance.

H

H H

The aim of this paper is to provide a comprehensive survey of the different algorithms and data structures usefull for triangulation and meshing construction and to serve as an introductory support for the course about mesh generation at the ISFMA Symposium on Computational Fluid Dynamics held at Xi'an, China, 6-17 sept. 1999. In addition, several aspects will be described too, for instance, mesh modification tools, mesh evaluation criteria, mesh optimization, even including adaptive mesh construction as well as parallel meshing techniques.

The text which follows is extracted from Chapter 3 of a book by Paul Louis George and Pascal Frey, published in French as "Maillages. Applications aux éléments finis" whose translation in English is in progress and is scheduled end of 1999.

Introduction

Mesh generation has evolved rapidly over the last two decades and meshing techniques seem to have recently reached a level of maturity that allows them to calculate complete solutions to complex three-dimensional problems. Thus, unstructured meshes for complex three-dimensional domains of arbitrary shape can be completed on current workstations in reasonable time. Further improvements may still be expected, for instance regarding the robustness, reliability and optimality of the meshing techniques.

Early mesh generation methods employed meshes consisting of quadrilaterals in two dimensions or hexahedra in three dimensions. Each vertex of such meshes can be readily defined as an array of indices and these types of meshes are commonly referred to as *structured* meshes. By extension, any mesh having a high degree of ordering (for example, a Cartesian grid) is said to be structured. More recent developments have tried to cope with the complex geometries (for instance, in CAD models involving multiple bounding surfaces) that were difficult to handle (*i.e.*, to mesh) with fully structured meshes. Nowadays unstructured meshes can be associated with finite element methods to provide an efficient alternative to structured meshes.

H

H H

The purpose of this chapter is to provide a comprehensive overview of the current techniques for both structured and unstructured mesh generation and to discuss their intrinsic advantages or weaknesses. These techniques will be further discussed in more detail in the relevant chapters of our book. First, a preliminary classification of existing meshing

此为试读，需要完整PDF请访问: www.ertongbook.com

techniques is proposed. One section is dedicated to surface meshing as surfaces play an important role in unstructured mesh generation techniques. Finally, a brief outline a mesh adaptation approaches is given.

1.1. Classes of methods

Despite many conceptual differences (since mesh generation methods have been developed in different contexts and were aimed at different field of applications), the classification of these techniques into seven classes has been proposed, for instance, in [George-1991]. Although this classification reflects the main approaches published, it appears that several techniques can be gathered together (due to their intrinsic properties), thus leading to a modified classification into only five categories:

Class 1. *manual* or *semi-automatic* methods.

They are applicable to geometrically simple domains. Enumerative methods (mesh entities are explicitly user-supplied) and explicit methods (which take advantage of the geometric features of the domain) are representative of this class.

Class 2. *parameterization* (mapping) methods.

The final mesh is the result of the inverse transformation *mapping* of a regular lattice of points in a parametric space to the physical space. Two main approaches belong to this class, depending on whether the mapping function is implicitly or explicitly defined:

- *algebraic interpolation* methods. The mesh is obtained using a transfinite interpolation from boundary curves (surfaces) or other related techniques explicitly defined,
- *solution-based* methods. The mesh is generated based on the numerical solution of a partial differential system of equations (elliptic, hyperbolic or parabolic), thus relying on an analytically defined function.

Class 3. *domain decomposition* methods.

The mesh is the result of a top-down analysis that consists in splitting the domain to be meshed into smaller domains, geometrically close to a domain of reference (in terms of shape). Two main approaches have been proposed, the difference being the structured or unstructured nature of the mesh used to cover the small domains:

- *block decomposition* methods: the domain is decomposed into several simpler subdomains (blocks), each of which is then covered with a structured mesh (obtained for instance using a mapping technique, as seen above).
- *spatial decomposition* methods: the domain is approximated with a union of disjoint cells that are subdivided to cover a spatial region object, each cell then being further decomposed into mesh elements. Quadtree and octree-based techniques are representative of this class.

Class 4. *point-insertion / element creation* methods.

The related methods generally start from a discretization of the boundary of the domain (although this feature is not strictly required) and mainly consist in creating and inserting internal nodes (elements) in the domain. Advancing-front (element creation) and Delaunay-based (point insertion) approaches are two methods belonging to this class.

Class 5. *constructive* methods.

The final mesh of the domain is the result of merging several meshes using topological or geometric transformations, each of these meshes being created by any of the previous methods.

Remark 1.1. Needless to say, this classification is necessarily arbitrary. However, while not unique, it does account for the different approaches published. Other methods not included in this classification exist, which are designed to handle specific situations.

A difficult task consists of clearly identifying the method capable of providing an adequate mesh, depending on the field of application. Basically, the geometry of the domain and the physical problem direct the user towards which method to apply.

On the other hand, the emphasis can be put on the type of meshes created by any of the proposed methods. From this point of view, two classes of meshing techniques can be identified, depending on whether they lead to structured or unstructured meshes. The following sections provide additional details on this aspect.

1.2. Structured mesh generators

In this section, we briefly describe the main approaches generally used to create structured meshes. While not claiming exhaustivity, the techniques mentioned here are representative of the current and latest developments in this field.

The basic idea common to all structured mesh generation methods consists of meshing a canonical domain (*i.e.*, a simple geometry) and mapping this mesh to a physical domain defined by its boundary discretization. Numerous types of such transformations exist and have been successfully applied to computational domains, for instance parametric space for surfaces (Bézier patches, B-splines), Lagrange or transfinite interpolation formula, quasi-conformal transformations, etc.

The first problem to solve is *where* to place the mesh points in such a way as to achieve a *natural* ordering appropriate to the problem considered. A trivial observation shows that simple domains such as squares and discs, in two dimensions, have an intrinsic curvilinear coordinate system. In this sense, the mapping techniques described below provide a basis for mesh generation.

Curvilinear coordinates

The physical domain discretization requires some level of organization to efficiently compute the solution of the PDEs. This organization is usually provided through a Cartesian or cylindrical coordinate system. More precisely, the grid points are defined using coordinate line intersections, which allow all numerical computations to be performed in a fixed (square or rectangular) grid. Hence, the Cartesian coordinates used to represent the PDEs have been replaced by the curvilinear coordinates.[1] A constant value of one curvilinear coordinate (and a monotonic variation of the other) in the physical space corresponds to vertical or horizontal lines in the *logical* space.[2]

Theoretically, two procedures can be used to generate a system of curvilinear coordinates, algebraic interpolation techniques [Gordon,Hall-1973] and solution-based techniques [Thompson-1982a]. From the computational point of view, the classical algebraic method is usually faster than the differential equation methods.

1.2.1. Algebraic interpolation methods

A simple, though efficient, way to achieve a structured mesh is to use a sequence of mappings to reduce the possibly complex domain to simple generic shapes (e.g., a triangle, a quadrilateral, a hexahedron, etc.). After a structured mesh has been defined in the logical space, the mapping function is used to generate a mesh conforming to all do-

[1] Note: the mapping of the physical space onto the *logical* space must be one-to-one.

[2] Also called transformed or parametric space.

main boundaries. This technique has proved useful for two-dimensional domains as well as in three dimensions.

The mapping function(s) and the mesh point distribution in the logical space can be chosen arbitrarily. However, it may be of some interest (and sometimes more efficient) to force the boundary discretization in the logical space to match the given domain boundary discretization.[3] The control of the mesh point distribution in the parametric space makes it possible to control the density of mesh vertices in the real domain (for instance, to obtain a finer mesh in regions of high curvature).

Remark 1.2. In general, the domain discretization must be a convex polygon (polyhedron, in three dimensions)[4] to guarantee the validity and the conformity of the resulting mesh.

The problem of finding a proper mapping function is equivalent to finding a specific function of the curvilinear coordinates. This function contains coefficients that enable the function to match specific values of the Cartesian coordinates on the boundary (and possibly elsewhere). Algebraic grid generation is thoroughly discussed in [Shmit-1982] and [Eriksson-1982].

To emphasize the algebraic method feature, we simply mention one particular mapping function, the transfinite interpolation scheme. This approach was first investigated by [Gordon,Hall-1973] and, then, by [Eriksson-1983], among others. Its most significant feature is its ability to control the mesh point distribution and particularly the slope of the mesh lines meeting the boundary surfaces [Baker-1989a]. In this chapter, we do not pursue the notion of transfinite elements. However, we describe its application to the mapping of a unit square, in two dimensions.

[3]This feature makes it possible to conform exactly to the given domain boundaries.
[4]or at least close to a convex shape.

Unit square mapping by transfinite interpolation

Here, we are concerned with a continuous transformation which maps the unit square $(\xi, \eta) \in [0,1] \times [0,1]$ one-to-one onto a simply connected, bounded two-dimensional domain. The mapping can be seen as a topological distortion of the square into the domain. The problem is to construct the mapping function that matches the boundary of the domain, and more precisely, the boundary discretization of this domain.

Let $\phi_i(\xi, \eta)$, $i = 1, \cdots, 4$ be the parameterization of the si

de i of the real domain, for which four such sides have been identified, and let a_i be the corresponding edge endpoints (corners). For the sake of simplicity, we have assumed that the discretizations of any two opposite edges of the domain have the same number of points[5]. A discretization of the unit square is constructed, analogous to that of the real domain (*i.e.*, each side of the square conforms to the discretization of the corresponding real side, in terms of relative distances between successive points). A quadrilateral mesh is then formed in the logical space by joining the matching points on opposite edges, the internal nodes thus being the line intersections.

Then, the mapping function takes the lattice of points in the parametric space (unit square) and maps it to the physical space (real domain) using transfinite interpolation based on the Lagrange interpolation formula as follows:

$$F(\xi,\eta) = (1-\eta)\phi_1(\xi) + \xi\phi_2(\eta) + \eta\phi_3(\xi) + (1-\xi)\phi_4(\eta)$$
$$-((1-\xi)(1-\eta)a_1 + \xi(1-\eta)a_2 + \xi\eta a_3 + (1-\xi)\eta a_4.) \quad (0.1)$$

Figure 1.1 shows a mesh of a domain mapped by applying a transfinite interpolation formula.

[5]One can always obtain such a situation by adding more points along a boundary edge, if needed.

此为试读,需要完整PDF请访问: www.ertongbook.com