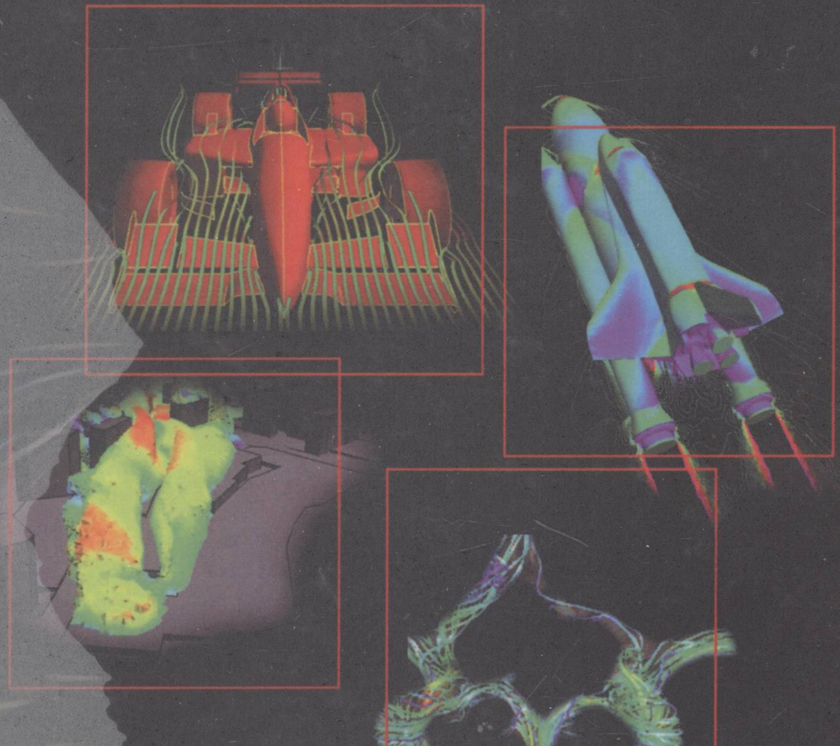
Applied CFD Techniques

An Introduction based on Finite Element Methods



WILEY

Rainald Löhner

Applied Computational Fluid Dynamics Techniques

An Introduction Based on Finite Element Methods

Rainald Löhner
School of Computational Sciences,
George Mason University
Fairfax, Virginia, USA

JOHN WILEY & SONS, LTD
Chichester • Weinheim • New York • Brisbane • Singapore • Toronto

Baffins Lane, Chichester,
West Sussex PO19 1UD, England

National 01243 779777
International (+44) 1243 779777

e-mail (for orders and customer service enquiries): cs-books@wiley.co.uk

Visit our Home Page on http://www.wiley.co.uk
or
http://www.wiley.co.uk

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

Wiley-VCH GmbH, Pappelallee 3,
D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton,
Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01,
Jin Xing Distripark, Singapore 129809

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

Library of Congress Cataloging-in-Publication Data

Löhner, Rainald.
Applied computational fluid dynamics techniques : an introduction based on finite element methods / Rainald Löhner.
p.cm.
Includes bibliographical references and index.
ISBN 0-471-49843-2
1. Fluid dynamics — Mathematics. 2. Numerical analysis. 3. Finite element method. I. Title.

TA357 .L592 2001
620.1′064 — dc21

2001017864

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0 471 49843 2

Typeset in 10/12pt Times by Laser Words Private Limited, Chennai, India
Printed and bound by Antony Rowe Ltd, Eastbourne

Preface

This monograph has its roots in several short courses that were held at the Boeing company, Seattle, in 1988; the IBM Short Course in CFD, held in Monterey in 1990; and the AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows, held at the von Karman Institute in Brussels and the NASA Ames Research Center in 1992. Moreover, large portions of this text were taken from the author's publications in scientific journals, books, and conference proceedings. In much the same way as object-oriented programming, the use of computers has made it a simple matter to compile and edit the material from these publications.

The aim of this book is to provide an introduction to the techniques used in applied computational fluid dynamics (CFD). No attempt has been made to provide a comprehensive treatise of all possible methods and algorithms. Given the high rate of innovations and the constant stream of new ideas and publications, such an undertaking would be imprudent at the present time. The emphasis is placed on well-established techniques that have proven their worth in practical applications. In an era that seems more concerned with originality than quality and reliability, this emphasis seems more than justified.

It is my great pleasure to acknowledge the input and stimulus provided by the many colleagues with whom I had the honour to work over the years. From my university team: Drs. Jean Cabello, Dorothée Martin, Helen Rudd, Benoît Petitjean, Eric Mestreau, Jean Favre, Alexander Shostko, Chi Yang, Juan Cebral, Makoto Nagaoka, Eric Darve, Jarek Tuzsinsky, Fernando Camelli, Jacob Waltz, and Orlando Soto. From the Naval Research Laboratory/Berkeley Research Associates/SAIC/ NASA-GSFC teams: Drs. Steven Zalesak, Joseph Baum, Jay Boris, David Book, Richard DeVore, John Ambrosiano, Gopal Patnaik, Ravi Ramamurti, Eric Loth, Hong Luo, and Dmitri Sharov. From the NASA LARC/Vigyan team: Drs. Manuel Salas, Clyde Gumbert, Paresh Parikh and Shaiar Prizadeh. From the Swansea/Imperial College/MIT Teams: Profs. Olgierd Zienkiewicz, Kenneth Morgan, Jaime Peraire, and Dr. Mehdi Vahdati. From the ESI Group: Drs. Ming Zhu, Philippe Ravier, Jean Roger, Ali Tabbal, Daniel Vinteler, Eberhard Haug, and Jan Clinkemaille. From the CIMNE/UPC team: Prof. Eugenio Oñate, Sergio Idelssohn, and Drs. Ramon Ribo, Julio Garcia, and Carlos Sacco.

The work compiled here would not have been possible without the steady support received by the author and his colleagues from such organizations as the Air Force Office of Scientific Research, the Defense Nuclear Agency, the Defense Advanced Research Projects Agency, NASA, and the Office of Naval Research. It is my hope that we have served the taxpayers' money well by developing the techniques described. It takes years to develop a new field. These organizations have shown time and again that they are willing to be patient and optimistic.

I would also like to take the opportunity to thank Cray Research, Inc. for providing many free hours on their machines over the years and IBM for providing me with RISC workstations for private use at home. CFD would never have been the same without this support.

Dr. David Book undertook the difficult task of reading the first draft of this book and provided many comments and suggestions.

Finally, to those unnamed or unreferenced (and there will always be those) — my apologies. You know who you are.

Contents

1

Introduction

1.1 INTRODUCTION AND GENERAL CONSIDERATIONS

Before going into a detailed description of applied computational fluid dynamics (CFD) techniques, it seems proper to define its place among related disciplines. CFD is part of computational mechanics, which in turn is part of simulation techniques. Simulation is used by engineers and physicists to forecast or reconstruct the behavior of an engineering product or physical situation under assumed or measured boundary conditions (geometry, initial states, loads, etc.). A variety of reasons can be cited for the increased importance that simulation techniques have achieved in recent years:

(a) *Need to Forecast Performance*: The inability to forecast accurately the performance of a new product can have a devastating effect on companies. The worst nightmare of an aircraft or car manufacturer is to build a prototype that has some hidden flaw that renders it inoperable or that seriously degrades market appeal. Of the many examples that could be cited here, we just mention flutter or buzz for aircraft and unforeseen noise or vibrations for cars. The development costs for new products are so large (about $\$4 \cdot 10^9$ for a new aircraft, $\$10^9$ for a new car; these and all subsequent quotations are in 2000 US$) that a nonperforming product can quickly lead to bankruptcy. The only way to minimize the risk of unexpected performance is through insight, that is, information. Simulation techniques such as CFD can provide this information.

(b) *Cost of Experiments*: Experiments, the only other alternative to simulations, are costly. A day in a large transonic wind tunnel costs about $\$10^5$, not counting the personnel costs of planning, preparing the model, analyzing the results, and so forth, as well as the hidden costs of waiting for availability and of the time lost in designing. An underground test for a nuclear device costs about $\$10^8$, and for a conventional weapon, it costs about $\$10^7$. Other large experiments in physics can also command very high prices.

(c) *Impossibility of Experiments*: In some instances, experiments are impossible to conduct. Examples are solar and galactic events, atmospheric nuclear explosions (banned after the Atmospheric Test Ban Treaty of 1960), and biomedical situations that would endanger the patient's life.

(d) *Insight*: Most large-scale simulations offer more insight than experiments. A mesh of $2 \cdot 10^7$ grid points is equivalent to an experiment with $2 \cdot 10^7$ probes or measuring devices. No experiment that the author is aware of has even nearly this many measuring locations. Moreover, many derived diagnostics (e.g., vorticity, shear, residence time, etc.) can easily be obtained in a simulation, but may be unobtainable in experiments.

(e) *Computer Speed and Memory*: Computer speed and memory capacity continue to double every 18 months (Moore's law). At the same time, algorithm development continues to improve accuracy and performance. This implies that ever-more realistic simulations can be performed. Table 1.1 summarizes the size of problem as a function of time from the author's own perspective. Note that in 1983 a problem with more than 1000 finite elements, being run at a university, was considered excessively large!

Although simulations would seem to be more advantageous, the reader should not discount experiments. They provide the only 'reality-check' during the development of new products. But given the steep decline in computing costs, simulations will certainly reduce the number of required experiments. Boeing estimates indicate that the number of wind-tunnel hours required for the development of the B-747 (1963) was reduced by a factor of 10 for the B-767 (1982) (Rubbert (1988)) and by yet another factor of 10 for the B-777 (1998).

Because aerospace is one of the leading fields for simulations, these figures may be indicative of trends to be expected in other manufacturing sectors.

In CFD, the simulation of flows is accomplished by

(a) solving partial differential equations (PDEs) numerically,

(b) following the interaction of a large number of particles, or

(c) a combination of both.

Table 1.1 Increase of problem size

Size	Dim	Code	Year	Problem	Machine
$>10^2$	2-D	FEFLO20	1983	Airfoil	ICL
$>10^3$	3-D	FEFLO30	1985	Forebody	Cyber-205
$>10^4$	2-D	FEFLO27	1986	Train	Cray-XMP
$>10^5$	3-D	FEFLO72	1989	Train	Cray-2
$>10^6$	3-D	FEFLO74	1991	T-62 tank	Cray-2
$>10^7$	3-D	FEFLO96	1994	Garage	Cray-M90
$>10^8$	3-D	FEFLO98	1998	Village	SGI O2000

The first model is used whenever a continuum assumption for the flow can be made. The second model is used for rarefied flows, in which the continuum model is no longer valid. Combinations of fields and particles are used whenever some aspects of a complex problem are best modelled as a continuum and others are modelled by discrete entities or when the motion of passive marker particles is useful for visualizing flows. Examples in which such combinations are commonly employed are plume flows with burning particles and ionized magnetohydrodynamic flows.

Because of its relevance to the aerospace and defense industries, as well as to most manufacturing processes, CFD has been pursued actively ever since the first digital computers were developed. The Manhattan project was a major test bed and a beneficiary of early CFD technology. Concepts such as artificial dissipation date from this time.

CFD, by its very nature, encompasses a variety of disciplines, which have been summarized in Figure 1.1 and may be enumerated in the following order of importance:

(a) *Engineering*: We live in a technology-driven world. Insight for practical engineering purposes is the reason why we pursue CFD. Forget the romantic vision of art for art's sake. This is engineering, physics, medicine, or any such discipline, and if a CFD code cannot guide the analyst to better products or more understanding, it is simply useless.

(b) *Physics*: Physics explains the phenomena to be simulated for engineering purposes and provides possible approximations and simplifications to the equations describing the flow fields. For example, the potential approximation, where applicable, represents central processing unit (CPU) savings of several orders of magnitude as compared with full Reynolds-averaged Navier-Stokes (RANS) simulations. It is the task of this discipline to outline the domains of validity of the different assumptions and approximations that are possible.

(c) *Mathematics*: Mathematics has three different types of input for CFD applications. These are as follows:

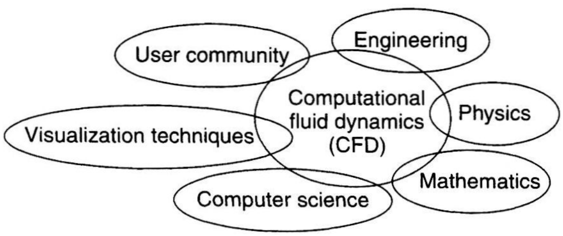


Figure 1.1 The multidisciplinary nature of CFD

- *classical analysis*, which discusses the nature, boundary conditions, Green kernels, underlying variational principles, adjoint operators, and so forth of the PDEs;
- *numerical analysis*, which describes the stability, convergence rates, uniqueness of solutions, well-posedness of numerical schemes, and so forth; and
- *discrete mathematics*, which enables the rapid execution of arithmetic operations.

(d) *Computer Science*: Computer science has mushroomed into many subdisciplines. The most important ones for CFD are as follows:

- *algorithms*, which describe how to perform certain operations in an optimal way (e.g., search of items in a list or in space);
- *coding*, so that the final code is portable, easy to modify and/or expand, easy to understand, user-friendly, and so forth;
- *software*, which encompasses not only compilers, debuggers, and operating systems but also advanced graphics libraries (e.g., OpenGL); and
- *hardware*, which not only drives the realm of ever-expanding applications that would have been unthinkable a decade ago but also influences to a large extent the algorithms that are employed and the way codes are written.

(e) *Visualization Techniques*: The vast amount of data produced by modern simulations need to be displayed in a sensible way. This refers not only to optimal algorithms to filter and traverse the data at hand but also to ways of seeing this data (plane-cuts, isosurfaces, X-rays, stereovision, etc.).

(f) *User Community*: The final product of any CFD effort is a code that is to be used for engineering applications. Successful codes tend to have a user community. This introduces human factors that have to be accounted for: confidence and benchmarking, documentation and education, the individual motivation of the end users, ego factors, not-invented-here syndrome, and so forth.

1.2 THE CFD CODE

The end product of any CFD effort is a code that is to be used for engineering applications or for the understanding of physical phenomena that were inaccessible previously. The quality of this tool will depend on the quality of ingredients listed earlier. Just as a chain is only as strong as its weakest link, a code is only as good as the worst of its ingredients. Given the breadth and variety of disciplines required for a good code, it is not surprising that only a few codes make it to a production environment, although many are written worldwide. Once a CFD code leaves the confines of research, it becomes a *tool*, that is, a part of the *service industry*. CFD codes, like other tools, can be characterized and compared according to the properties considered important by the user community. Some of these are as follows:

- EU: ease of use (problem setup, user interface, etc.)
- DO: documentation (manuals, help, etc.)
- GF: geometric flexibility
- TT: turnaround time (setup to end result)
- BM: benchmarking
- AC: accuracy
- SP: speed
- EX: expandability to new areas/problems

Like any other product, CFD codes have a customer base. This customer base can be categorized by the number of times a certain application has to be performed. Three main types of end users may be identified:

(a) those that require a few occasional runs on new configurations to guide them in their designs (e.g., flow simulations in the manufacturing industries and process control);

(b) those that require a large number of runs to optimize highly sophisticated products (e.g., airfoil or wing optimization); and

(c) those that require a few very detailed runs on extremely simple geometries to understand or discover new physics. These end users are typically associated with government laboratories. Runs of this kind typically push the limits of tolerance for other users, and their lengths are often the subject of 'war stories' (e.g., more than two weeks of continuous CPU-time on the fastest machine available).

According to the frequency of runs, the priorities change, as can be seen from the following Table 1.2:

The message is clear: before designing or comparing codes, one should ask how often the code is to be used on a particular application, how qualified the personnel are, what the maximum allowed turnaround time is, what the expected accuracy is, and what are the resources that are available. Only then can a proper design or choice of codes be made.

Table 1.2 Priorities for different user environments

Type of run	No. of runs	Run time	Desired properties
General purpose analysis	O(1)	Hours	EU, DO, GF, EX, TT, BM, AC, SP
Design/ optimization	O(1000)	Seconds	SP, TT, GF, AC, BM, EU, EX, DO
New physics	O(10)	Months	AC, BM, SP, TT, EU, GF, DO, EX

1.3 PORTING RESEARCH CODES TO AN INDUSTRIAL CONTEXT

Going from a research code to an industrial code requires a major change of focus. Industrial codes are characterized by

- extensive manuals and other documentation;
- 24-hour hot line answering service;
- customer-support team for special requests/applications; and
- incorporation of changes through releases and training.

In short, they require an *organization* to support them. The CFD software and consulting market already exceeds $300M/year and is expected to grow rapidly in the coming decade.

1.4 SCOPE OF THE BOOK

This book treats the different topics and disciplines required to carry out a CFD run in the order they appear or are required during a run:

(a) Data structures (to represent, manage, generate, and refine a mesh),

(b) Grid generation (to create a mesh),

(c) Approximation theory and flow solvers (to solve the PDEs and push particles on the mesh),

(d) Interpolation (for particle-mesh solvers and applications requiring remeshing),

(e) Adaptive mesh refinement (to minimize CPU and memory requirements), and

(f) Efficient use of hardware (to minimize CPU requirements).

This order is different from the historical order in which these topics first appeared in CFD and the order in which most CFD books are written.

Heavy emphasis is placed on CFD using unstructured (i.e., unordered) grids of triangles and tetrahedra. A number of reasons can be given for this emphasis:

- The only successfully industrialized CFD codes that provide user support, updates, and an evolving technology to a large user base are based on unstructured grids. This development parallels the development of finite element codes for computational structural dynamics (CSD) in the 1960s.
- Once the problem has been defined for this more general class of grids, reverting to structured grids is a simple matter.
- A large number of very good books on CFD based on structured grids exists (e.g., Book (1981), Roache (1982), Anderson, Tannehill and Pletcher (1984),

Oran and Boris (1987), and Hirsch (1991)), and there is no point writing yet another one.

As with any technological product, the final result is obtained after seemingly traversing a maze of detours. After all, why use a car (which has to be painted after assembly after mining/producing the iron and all other raw materials ...) to go to the grocery shop when one can walk the half mile? The answer is that we want to do more with a car than drive half a mile. The same is true for CFD. If the requirement consists of a few simulations of flows past simple geometries, then all this development is not needed. To go the distance to realistic 3-D simulations of flows in or past complex geometries, no other way will do. The reader is therefore asked to be patient. The relevance of some parts will only become apparent in subsequent chapters.

REFERENCES

Anderson, D. A., Tannehill, J. C. and Pletcher, R. H. *Computational Fluid Mechanics and Heat Transfer*, McGraw-Hill, New York, 1984.

Book, D. L. ed., *Finite Difference Techniques for Vectorized Fluid Dynamics Calculations*, Springer-Verlag, New York, 1981.

Hirsch, C., *Numerical Computation of Internal and External Flow*, Wiley & Sons, New York, 1991.

Oran, E. and Boris, J. P. *Numerical Simulation of Reactive Flow*, Elsevier, New York, 1987.

Roache, P. J, *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, NM, 1982.

Rubbert, P. *IBM Short Course in CFD*, Monterey, IBM, CA, 1988.