

Shock Wave and High Pressure Phenomena

Marcello Onofri
Renato Paciorri *Editors*

Shock Fitting

Classical Techniques, Recent
Developments, and Memoirs of Gino
Moretti



Shock Wave and High Pressure Phenomena

Founding Editor

Robert A. Graham, USA

Honorary Editors

Lee Davison, USA

Y. Horie, USA

Editorial Board

Gabi Ben-Dor, Israel

Frank K. Lu, USA

Naresh Thadhani, USA

Shock Wave and High Pressure Phenomena

L.L. Altgilbers, M.D.J. Brown, I. Grishnaev, B.M. Novac, I.R. Smith, I. Tkach, and Y. Tkach:
Magnetocumulative Generators

T. Antoun, D.R. Curran, G.I. Kanel, S.V. Razorenov, and A.V. Utkin: Spall Fracture

J. Asay and M. Shahinpoor (Eds.): High-Pressure Shock Compression of Solids

S.S. Batsanov: Effects of Explosion on Materials: Modification and Synthesis Under High-Pressure Shock Compression

G. Ben-Dor: Shock Wave Reflection Phenomena

L.C. Chhabildas, L. Davison, and Y. Horie (Eds.): High-Pressure Shock Compression of Solids VIII

L. Davison: Fundamentals of Shock Wave Propagation in Solids

L. Davison, Y. Horie, and T. Sekine (Eds.): High-Pressure Shock Compression of Solids

V.L. Davison and M. Shahinpoor (Eds.): High-Pressure Shock Compression of Solids III

R.P. Drake: High-Energy-Density Physics

A.N. Dremin: Toward Detonation Theory

J.W. Forbes: Shock Wave Compression of Condensed Matter

V.E. Fortov, L.V. Altshuler, R.F. Trunin, and A.I. Funtikov: High-Pressure Shock Compression of Solids VII

B.E. Gelfand, M.V. Silnikov, S.P. Medvedev, and S.V. Khomik: Thermo-Gas Dynamics of Hydrogen Combustion and Explosion

D. Grady: Fragmentation of Rings and Shells

Y. Horie, L. Davison, and N.N. Thadhani (Eds.): High-Pressure Shock Compression of Solids VI

J. N. Johnson and R. Cherét (Eds.): Classic Papers in Shock Compression Science

V.K. Kedrinskii: Hydrodynamics of Explosion

C.E. Needham: Blast Waves

V.F. Nesterenko: Dynamics of Heterogeneous Materials

S.M. Peiris and G.J. Piermarini (Eds.): Static Compression of Energetic Materials

M. Sučeska: Test Methods of Explosives

M.V. Zhernokletov and B.L. Glushak (Eds.): Material Properties under Intensive Dynamic Loading

J.A. Zukas and W.P. Walters (Eds.): Explosive Effects and Applications

More information about this series at <http://www.springer.com/series/1774>

Marcello Onofri • Renato Paciorri
Editors

Shock Fitting

Classical Techniques, Recent Developments,
and Memoirs of Gino Moretti



Springer

Editors

Marcello Onofri
Dipartimento di Ingegneria
Meccanica e Aerospaziale
Sapienza University of Rome
Rome, Italy

Renato Paciorri
Dipartimento di Ingegneria
Meccanica e Aerospaziale
Sapienza University of Rome
Rome, Italy

ISSN 2197-9529

ISSN 2197-9537 (electronic)

Shock Wave and High Pressure Phenomena

ISBN 978-3-319-68426-0

ISBN 978-3-319-68427-7 (eBook)

<https://doi.org/10.1007/978-3-319-68427-7>

Library of Congress Control Number: 2017958032

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

*We dedicate this book to the memory of our
master Gino Moretti.*

Preface

This book was born from a series of fortuitous circumstances and after a complex period of gestation. In June 2015, Prof. Gabi Ben-Dor called Marcello Onofri asking him to give an invited lecture for the 30th International Symposium on Shock Waves (ISSW30) which was to be held in July in Israel. Marcello accepted the invitation with enthusiasm and started to evaluate a possible topic for his lecture.

After a brief discussion with colleagues at the Sapienza University of Rome, including Renato Paciorri (second editor of this book), it became clear that the best option was to address the shock-fitting technique. Indeed, a few months prior, sadly Gino Moretti, the scientist who had contributed the most to the development of shock-fitting techniques, had passed away.

Marcello had collaborated for a long time with Gino, helping him in the development of some of his shock-fitting techniques. It was during the 1980s, at the Polytechnic Institute of New York and in his company GMAF (the famous “Gino Moretti And Friends”), along with Frank Marconi and later with Francesco Nasuti, that Marcello helped Gino in developing some complex applications of the floating shock-fitting technique. The collaboration lasted until the early years of the new century. This experience was certainly successful, but at the same time, the work was also a proverbial “pain in the neck.” Indeed, the technique was very precise, maybe too much so: in fact it was overreacting to even the smallest numerical inaccuracies. As an example, minor numerical rearrangements of the local simulation of a shock-shock interaction were translated as real physical perturbations, inducing an evolution of the flow field toward a new steady state physically compatible with them. Moreover, the extension to three-dimensional flow fields was possible only on a theoretical basis.

Fortunately, a few years later, Renato Paciorri and Aldo Bonfiglioli of the Università degli Studi della Basilicata resumed the development of shock-fitting techniques, taking advantage of more modern computer science tools to handle complex moving surfaces in three-dimensional space, similarly to what is shown in sketch illustrating some practical shock configurations. They also used for the first time a gasdynamic solver in the context of unstructured grids.

So the subject and the topics of the ISSW30 invited lecture were chosen: it would have duly celebrated Gino Moretti's seminal work and that of his fellow scientists all the way to the most recent developments of shock-fitting techniques.

The lecture was held in Tel Aviv on July 3, 2015, and raised a good interest. Dr. Christopher T. Coughlin, an editor for Springer Nature, was among the audience and proposed Marcello Onofri to act as editor for a book collecting old and new contributions about shock-fitting.

The idea of a book on shock-fitting was greeted with enthusiasm by Marcello Onofri, who immediately involved Renato Paciorri in the endeavor. In the recent scientific literature, shock-fitting books are almost completely absent. Indeed, to the best of our knowledge, the first and last book explicitly dedicated to shock-fitting is Manuel Salas's *A Shock-Fitting Primer*, published in 2009. As the title suggests, his book is dedicated to researchers and students willing to be introduced to shock-fitting techniques but of course cannot delve into the state of the art of the most recent evolutions and applications. Moreover, Gino Moretti had, in the past, repeatedly expressed the intent to write a book about shock-fitting, and some of his notes circulated among his closest collaborators. Unfortunately the book was never published. The possibility to incorporate in our book some of his unpublished notes can represent the opportunity to present a tribute to his figure and his unrelenting attempts to disseminate his ideas.

Our guideline was to gather the contributions of researchers that had been working, and are still working, in the development of shock-fitting or in its applications to significant practical cases. This collection specifically aims at including all the different variants of shock-fitting techniques and their most significant applications. They indeed range from the early *boundary shock-fitting* for structured meshes, introduced in the late 1960s, to the later versions of *floating shock-fitting*, introduced in the 1970s and largely developed in the 1980s and 1990s, and, finally, to the *shock-fitting for unstructured meshes*, first introduced in 2006.

Although many researchers who had been active during the development of shock-fitting methods for structured grids (boundary and floating shock-fitting) and their application had retired or abandoned this field for a long time, the possibility to collect a sufficient number of technical contributions appeared realistic.

While we were collecting these contributions, in January 2017, Nick Dizinno organized a special section of the AIAA SciTech Forum 2017 held in Dallas, dedicated to honoring Gino Moretti. There, some scientists who personally met Moretti during his professional life were invited to present their memoirs. Onofri, who participated to this session, had the idea of inviting them to write short memoirs, to be included in the book as a special tribute to the figure of Gino Moretti in occasion of the centenary of his birth.

The book is organized in three parts. The first part focuses on the *floating shock-fitting*, which Moretti devoted great efforts to in the final part of his scientific career. The second part is devoted to the more *recent developments in shock-fitting techniques for unstructured grids*, and, finally, the last is a *tribute to Gino Moretti*, a collection of memoirs of some of the scientists who shared with him the exciting and pioneering early age of computational fluid dynamics.

The first part of the book addresses the shock-fitting techniques for structured grids. This part includes two precious contributions on floating shock-fitting.

The first contribution is an unpublished manuscript by Moretti which, albeit not complete, was part of the book that Moretti intended to publish but never completed. What makes this manuscript interesting and unique is the detail with which Moretti describes his floating shock-fitting technique. Some NASA reports and journal articles exist in which Moretti describes this technique but none of these to the detail of this unpublished manuscript.

The second contribution was written by Francesco Nasuti and Marcello Onofri who continued the development of the floating shock-fitting and applied it to numerous complex flow configurations until the early 2000s. This contribution represents therefore the state of the art reached by the floating shock-fitting in the final phase of its first development. It is focused on shock interaction problems and on the fitting of contact discontinuities. The results reported therein exhibit high quality and detail in the solutions, in spite of them having been obtained over 15 years ago.

The part devoted to shock-fitting for unstructured grids contains a larger number of contributions, as different groups have been studying this technique and are thus currently engaged in its development. The first paper is written by Renato Paciorri and Aldo Bonfiglioli. In 2006 they were the first to propose a *shock-fitting technique for unstructured grids*. Interestingly, this new technique was presented for the first time at the 17th International Shock Interaction Symposium (ISIS17) in the presence of Gino Moretti, who was pleased with the results and encouraged the authors to continue in its development. Indeed this is what Paciorri and Bonfiglioli did, and their contribution is an assessment of the state of the art of their achievements over the past 10 years and an outline of its future developments.

The second contribution is by Andrea Lani and Valentina De Amicis, and it reports their valuable work performed at the von Karman Institute (VKI). For some years, Andrea Lani along with postgraduate students of the universities of Sapienza and Basilicata has been implementing the shock-fitting technique for unstructured grids within the VKI's code CoolFluid. The goal is to apply the technique to atmospheric re-entry flows around space capsules. This work has produced an open-source framework for coupling arbitrary CFD solvers with the unstructured shock-fitting technique.

The third contribution is authored by Campoli et al. who adapted shock-fitting for unstructured meshes to the gasdynamic solver of the *Institut National de Recherche en Informatique et en Automatique* (INRIA) of Bordeaux, developed by Dr. Mario Ricchiuto. The authors show the high quality of unsteady numerical solutions obtained by adding the shock-fitting technique to their code, when compared to solutions computed by shock-capturing solvers.

The group led by Prof. Jun Liu of the Dalian University of Technology has recently extended the shock-fitting technique for unstructured grids to cell-centered solvers. In fact, while the technique proposed by Paciorri and Bonfiglioli, and subsequently implemented in the VKI and INRIA solvers, can be coupled with node-centered CFD solvers (e.g., the solvers based on the residual distribution

approach), the technique proposed by Dalian University researchers can work in conjunction with cell-centered solvers (such as common finite volume solvers). The contribution of Dalian University illustrates recent developments of this algorithm and shows some interesting applications.

The last two contributions do not describe specific shock-fitting solvers or application of shock-fitting solvers, but rather focus on two important ingredients of shock-fitting techniques for unstructured grids: the local mesh regeneration and the shock detection. The shock-fitting technique proposed by Paciorri and Bonfiglioli requires runtime local re-meshing of the mesh in the proximity of the shock. This task, which can become complex in three dimensions, can be enucleated as a specific mesh generation problem and relocated to the more general problem of inserting a surface into an existing grid. The group of Prof. Carl Ollivier-Gooch of the University of British Columbia (UBC) of Vancouver has a great experience in the field of mesh generation and has studied this problem extensively. The contribution of UBC researchers explains how the problem has been addressed and shows different application examples.

Finally, the last contribution, written by Prof. Kojiro Suzuki of the University of Tokyo and Dr. Masashi Kanamori of Japan Aerospace Exploration Agency (JAXA), addresses a shock-detection technique. The detection of the time instant and location where a shock wave is formed in an unsteady flow is a key ingredient of an unsteady shock-fitting technique. In Moretti's unpublished manuscript, a shock-detection technique is described based on the characteristics theory applied to one-dimensional flows. Such technique can be extended to multidimensional flows only in case of structured grids but not for unstructured grids. The Japanese researchers propose an interesting shock-detection technique based on the full multidimensional application of the characteristics theory that can be used on unstructured grids.

The last part of the book collects five memoirs. Three are written by his disciples, Manuel Salas, Maurizio Pandolfi, and Pasquale Sforza, whereas the other two come from two scientists, Robert MacCormack and Phil Roe, who shared his professional environment and enjoyed Moretti's unconventional attitude and approach in discussing his own ideas, whether in agreement or in contrast with others. Far from being technical papers, such memoirs set the stage for the historical and scientific context in which Moretti worked and help us explore his personality and outline his human qualities.

Reading such memoirs is a delightful experience that stimulates ideas and personal considerations, and we won't spoil it by anticipating any of such recounts. It is however worth pointing out here an important aspect of Moretti's personality that has left a tangible sign. Beyond a seemingly sharp and clumsy character, these memoirs reveal a very generous man, particularly toward young researchers who were passionately guided by him with continuous detailed suggestions in the early stages of their careers.

This quality is crucial to appreciate the role of mentor that Moretti played during all of his professional life and even well beyond his retirement. This is confirmed by the popularity he had both in the USA and Europe. In the USA, he continued in a sense the tradition opened by Tony Ferri, who escaped from Italy during the fascist government and started pushing many projects and activities in the USA. In Europe Gino met a large consensus and interest mainly in German and in Italian universities.

In Italy, in particular, his role as mentor was particularly effective. Moretti hosted at the Polytechnic Institute of New York many young researchers coming from the Italian Polytechnic Universities of Turin and Bari, as well as from the Sapienza University of Rome, and he enjoyed to maintain long-standing professional and human relations with them until the very end of his life. In fact, some of the best Italian “CFD schools” were founded in Rome, Turin, and Bari, where Moretti’s footprint is still very present and important to this day. Such strong bonds are well documented by the photograph displayed in Fig. 1, taken in 2006, more than 15 years after Moretti’s retirement, at ISIS17, showing Moretti with some members of his “Italian Branch,” who convened from different universities and research centers to meet their “maestro” in what was to be one of the last conferences he attended.



Fig. 1 Gino Moretti in Rome with some of his disciples and friends (from *left to right*: Gino, Antonio di Carlo, Pasquale Sforza, Luca Zannetti, Filippo Sabetta, Maurizio Pandolfi)

We sincerely hope that reading this book and the different contributions included therein will be useful to rediscover the figure of a CFD pioneer like Gino Moretti and to appreciate the great potential of his shock-fitting technique. We address this book especially to young researchers who were born and raised in a world dominated by shock-capturing methods, so that they can appreciate that an alternative route to accurate simulations of flows with shocks is possible: the shock-fitting route.

Rome, Italy

Rome, Italy

July 2017

Marcello Onofri

Renato Paciorri

Acknowledgments

We want to thank all the people who contributed to the realization of this book. In addition to all the contributors, we want to thank Prof. Filippo Sabetta and Dr. Cécile Dobrzynski who helped in the review process, and, finally, we would like to thank Dr. Christopher T. Coughlin for his insistence and patience, without which this book would never have been published.

Contents

Part I Floating Shock-Fitting

Shock-Fitting Analysis	3
Gino Moretti	
Steady and Unsteady Shock Interactions by Shock Fitting Approach	33
Francesco Nasuti and Marcello Onofri	

Part II Recent Developments in Shock-Fitting Techniques for Unstructured Grids

Basic Elements of Unstructured Shock-Fitting: Results Achieved and Future Developments	59
Renato Paciorri and Aldo Bonfiglioli	
SF: An Open Source Object-Oriented Platform for Unstructured Shock-Fitting Methods	85
Andrea Lani and Valentina De Amicis	
Shock-Fitting and Predictor-Corrector Explicit ALE Residual Distribution	113
L. Campoli, P. Quemar, A. Bonfiglioli, and M. Ricchiuto	
A Shock-Fitting Technique for ALE Finite Volume Methods on Unstructured Dynamic Meshes	131
Jun Liu and Dongyang Zou	
Inserting a Shock Surface into an Existing Unstructured Mesh	151
Daniel Zaide and Carl Ollivier-Gooch	
Shock Wave Detection Based on the Theory of Characteristics	171
Masashi Kanamori and Kojiro Suzuki	

Part III Tribute to Gino Moretti

Gino Moretti: A Man for All Seasons	193
Manuel D. Salas	
A Renaissance Man: Personality and Generosity of a Dear Friend	203
Maurizio Pandolfi	
Gino Moretti at GASL, Brooklyn Poly, and the Later Years	215
Pasquale M. Sforza	
Gino Moretti, the Maestro of Gas Dynamics	223
Robert MacCormack	
There Was a Conference in Europe	227
Philip Roe	

Contributors

Aldo Bonfiglioli Scuola di Ingegneria, Università degli Studi della Basilicata, Potenza, Italy

Lorenzo Campoli Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Rome, Italy

Valentina De Amicis Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienze University of Rome, Rome, Italy

Masashi Kanamori Institute of Aeronautical Technology, Japan Aerospace Exploration Agency, Chofu, Tokyo, Japan

Andrea Lani Von Karman Institute for Fluid Dynamics, Belgium

Jun Liu Dalian University of Technology, Dalian, P.R. China

Robert MacCormack Stanford University, Stanford, CA, USA

Gino Moretti New York University, New York, NY, USA

Francesco Nasuti Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Rome, Italy

Carl Ollivier-Gooch Advanced Numerical Simulation Laboratory, The University of British Columbia, Vancouver, BC, Canada

Marcello Onofri Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Rome, Italy

Renato Paciorri Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Rome, Italy

Maurizio Pandolfi Politecnico di Torino, Torino, Italy

Pierrick Quemar LAGA, Université Paris 13 et LNHE, EDF R&D, Paris, France

Mario Ricchiuto Team CARDAMOM, Inria Bordeaux Sud-Ouest, Talence, France

Philip Roe University of Michigan, Ann Arbor, MI, USA

Manuel D. Salas (retired) NASA Langley Research Center, Hampton, VA, USA

Pasquale M. Sforza University of Florida, Gainesville, FL, USA

Koじro Suzuki Graduate School of Frontier Sciences, The University of Tokyo, Kashiwa, Chiba, Japan

Daniel Zaide Advanced Numerical Simulation Laboratory, The University of British Columbia, Vancouver, BC, Canada

Synopsys, Inc., Mountain View, CA, USA

Dongyang Zou Dalian University of Technology, Dalian, P.R. China

Part I
Floating Shock-Fitting

Shock-Fitting Analysis

Gino Moretti

Introduction to Gino Moretti's Manuscript

This chapter reports an unpublished manuscript by Gino Moretti taken from the book on the shock-fitting that Gino Moretti tried to write after his retirement but that he never completed and published. The manuscript is not complete, there are two figures missing, some of final sections and the references. We repropose in this book the manuscript as it is, without any attempt to restore the missing parts. Indeed, we think that it has a high value, not only historical but also scientific, as it represents the most complete description of the floating shock-fitting technique proposed and developed by Gino Moretti in the last years of his scientific career.

The manuscript focuses exclusively on the algorithmic description of the technique and no results and applications are shown. A number of applications of this technique can be found in these Refs. [1–3] of the late 80s. Later on, the technique was further developed and applied during the 90s by Marcello Onofri and Francesco Nasuti. The next chapter will show you these further developments and some interesting applications of the floating shock-fitting.

The analysis of the manuscript allows us to observe some interesting aspects of Moretti's "modus operandi." What is most evident is the extreme precision and detailed information, shown in the algorithm description. The manuscript contains also some pieces of code and the description of the data structure. At the same time it is also evident the strong link between numerical algorithm and mathematical theory or the natural evolution of the physical phenomenon (e.g., see the Sect. 7.1 concerning shocks generated by coalescence of waves). Even when Moretti deals

G. Moretti

Emeritus Professor at New York University, New York, NY, USA

The author Gino Moretti was deceased. For questions and comments about this chapter please contact R. Paciorri at renato.paciorri@uniroma1.it.

with purely numerical problems (e.g., see Sect. 9.1 on wrinkles on a shock), he tries to provide a mathematical interpretation of the numerical phenomenon, although in the light of the differences between continuous and discrete models. For this reason the explanations and solutions that Moretti proposes are always very convincing and elegant.

We hope that the publication of this manuscript could be appreciated by the scientists interested in the shock-fitting technique and that it will be a source of inspiration and example.

Rome, Italy

Rome, Italy

July 2017

Marcello Onofri

Renato Paciorri

References

1. Moretti, G.: A technique for integrating two-dimensional Euler equations. *Comput. Fluids* **15**(1), 59–75 (1987)
2. Moretti, G.: A general-purpose technique for two dimensional transonic flows (1987). NASA-CR-194186
3. Moretti, G.: Efficient Euler solver with many applications. *AIAA J.* **26**(6), 655–660 (1988). <http://arc.aiaa.org/doi/abs/10.2514/3.9950>

1 Introduction

Simple, accurate, and fast-working codes can be written for Euler's equations if they are formulated in characteristic terms prior to discretization. A typical code is the λ -scheme [1, 2] with its variations [3–6]. It is well-known that such codes cannot properly capture strong shocks and cannot describe evolutions where a shock has to move upstream in a supersonic environment. Therefore, the formulation of Euler's equations in divergence form and shock-capturing codes are generally preferred. Simplicity, accuracy, and speed of shock-capturing codes, however, are not a challenge for the λ -scheme; in addition, the lack of accuracy in the regions surrounding the captured shock jeopardizes certain extensions, for example to non-equilibrium flows. The only advantage of shock-capturing codes is their robustness, that gives peace of mind to the analyst (to put it, bluntly, shock-capturing codes generally produce results, but not necessarily good results). One may be induced to compute using shock-capturing codes and to fit the captured shocks where necessary. I do not agree, for the following reasons:

1. The inaccuracies of shock-capturing codes, that pervade the entire flow field, are not corrected by occasional fittings.

2. From a practical viewpoint, the shock-capturing code (that is always more cumbersome than the λ -scheme) needs additional working to be made compatible with shock-fitting.
3. The λ -scheme itself is able to capture oblique isentropic shocks and to give the shock detection procedure time to initialize the fitting, if necessary. Therefore, there is no reason to fear a loss of accuracy, for having missed the fitting of a shock.
4. Once an oblique shock has been captured by a shock-capturing code, the transition is smeared over many cells and the fitting technique (at least, the one I am going to describe in this paper) may not find a sufficiently strong jump to start working; this is particularly evident near, shock interactions and reflections from rigid wall where even regular reflections tend to be interpreted by a shock-capturing technique as Mach reflections.

There are many theoretical and practical advantages in shock-fitting. Fitting of shocks agrees with their physical nature (in an inviscid model) as a part of strong solutions. Their evaluation is very accurate and fast if the Rankine-Hugoniot conditions are used explicitly. The automatic separation of certain domains of dependence in the vicinity of a shock prevents the formation of wiggles and makes the use of arbitrary damping factors (artificial viscosity) unnecessary. Therefore, some years ago, I started working on the problem of combining a simply, highly accurate, fast, and robust λ -scheme technique with a shock-fitting technique capable of detecting and handling as large a variety of complex shock patterns as possible. Progress reports have been published [7–9] presenting a number of applications of increasing complexity. No details have been given extensively, both because of space limitations and because I wanted first to validate the technique in a greater number of problems. Unfortunately, the time scale for each problem is measured in months, not so much for the fitting of shock as for the creation of the geometry and a suitable grid, the proper enforcement of boundary conditions, a large number of parametric studies and an analysis of the possible dependence of the solution on initial conditions, grid fineness, etc. Meanwhile, I have been reprimanded for being too cryptic and, consequently, for preventing other people from contributing to the development of the technique.

The present paper tries to correct the omission. The matter contained herein is an extended version of the Final Report submitted to NASA Lewis BC, in December 1987; it supersedes all my other papers containing partial details.

2 Basic Ideas

The following problems arise in considering shock-fitting:

1. Detection of formation of a shock, either by coalescence of pressure waves or by abrupt changes in rigid wall geometry,
2. Conversely, disappearance of increasingly weaker shocks,

3. Local treatment of shocks according to the Rankine-Hugoniot conditions,
4. Shock topology: location of shocks with respect to the computational grid and to one another,
5. Definition of the normal to a shock,
6. Shock interactions,
7. Stability of shocks,
8. Treatment of points in the vicinity of a shock.

I will consider them in detail in the following sections, limiting the analysis to problems depending on two space dimensions and time. Many of the above problems are tightly tied together; no section can be written without some knowledge of other sections. I attempted to arrange the material to avoid browsing around back and forth as much as possible. Therefore, the sequence of sections is not the same as the topics listed above, that appear instead in the following order: 8 (Sect. 5), 3 (Sect. 6), 1 (Sect. 7), 4 (Sect. 8), 2 (Sect. 9), 5 and 7 (Sect. 10), and 6 (Sect. 11). In addition, the actual sequence of subroutines and operations in any of my computational codes will be described and rediscussed in Sect. 11.

3 Grids, Shock Points, Notations

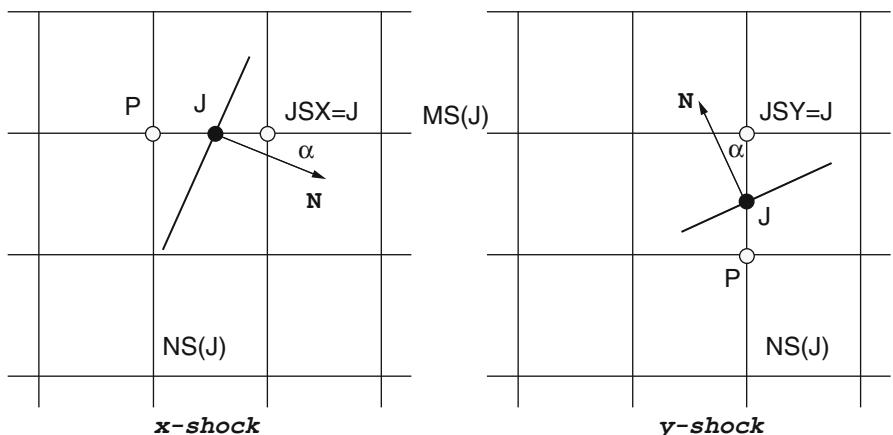
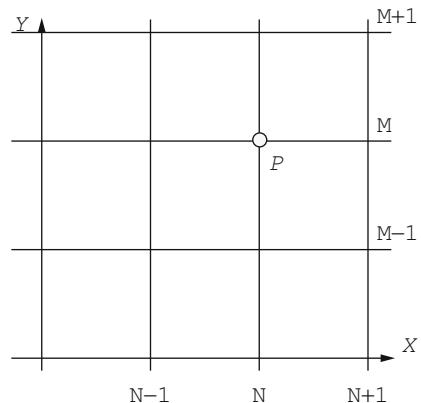
We consider only structured grids that may or may not be orthogonal. Portions of such grids will be represented here as in Fig. 1 because we are only concerned with the topology of the shocks. In other words, our considerations apply to any type of structured grids, not only to Cartesian grids as shown in the figure. Let grid lines of one family (the horizontal ones in Fig. 1) be called *x*-lines, and the others *y*-lines; *x*-lines are numbered using the index *m*, *y*-lines using the index *n* (*M* and *N* in FORTRAN codes, respectively). Thus, grid points on a given *x*-line are ordered in increasing sequences of values of *N*; grid points on a given *y*-line are ordered in increasing sequences of values of *M*. Every grid point can be identified by its coordinates (*x*, *y*) or by its indices (*N*, *M*).

A shock front is described by its intersections with grid lines. The intersection of a shock with an *x*-line is called an *x*-point and its intersection with a *y*-line is called a *y*-point. We will stipulate not to consider *x*-points where the normal to the shock forms an angle of more than 45° with the *x*-line, and not to consider *y*-points where the normal to the shock forms an angle of more than 45° with the *y*-line.

All information pertinent to shock points is stored in single arrays, depending on a counter, *J*. No attempt is made to organize the arrays; the shock points are stored at random, regardless of their relative position in the physical plane. Cross-referencing between the grid points and the shock points is necessary. To this effect, we use the following indices:

NS(J) is the value of *N* to the left of an *x*-point or the value of *N* itself for a *y*-point (Fig. 2),

MS(J) is the value of *M* for an *x*-point or the value of *M* below a *y*-point,

Fig. 1 Computational grid**Fig. 2** Shock points in a grid

$JSX(N,M)$ is the value of J for the x -point lying on the m -th x -line to the left of the n -th y -line, and it is zero elsewhere,

$JSY(N,M)$ is the value of J for the y -point lying on the n -th y -line below the m -th x -line, and it is zero elsewhere,

$INDX(N,M)$ equals zero where $JSX(N,M)$ equals zero, and 1 where JSX is not zero,

$INDY(N,M)$ equals zero where $JSY(N,M)$ equals zero, and 1 where JSY is not zero,

$JI(J)$ equals zero for an x -point with the high pressure on the left, 1 for an x -point with the high pressure on the right, 2 for a y -point with the high pressure below, and 3 for a y -point with the high pressure above.

In addition, the following arrays are defined:

XS(J), YS(J) are the x - and y -coordinates of any shock point,
 ANGLE(J), for an x -shock, is the angle, α , between the normal N to the shock at any shock point, and the x -line ($-\pi/2 \leq \alpha \leq \pi/2$),
 ANGLE(J), for a y -shock, is the angle, α , between the normal N to the shock at any shock point, and the y -line ($-\pi/2 \leq \alpha \leq \pi/2$),
 W(J) is the normal shock velocity at any shock point,
 TMA(J) is the absolute value of the normal relative Mach number at any shock point.

4 Brief Recall of λ -Scheme Procedures and Notation

Obviously, the shock-fitting technique to be explained in what follows can be used in connection with any integration scheme for ordinary points. The most natural scheme, though, is the λ -scheme because it propagates signals along bi-characteristics in a way that easily satisfies the basic requirements for the environment of a shock. Therefore, I will consider the λ -scheme as the updating technique for all ordinary points and briefly recall the pertinent notation, as needed in the application to shock-fitting. If anyone wants to attempt shock-fitting on top of other integration schemes, these should be properly modified in the vicinity of the shock to mimic the λ -scheme features. Here we consider only the simplest formulation of the λ -scheme, that is, the one based on Cartesian grids. This is done for simplicity and clarity, in an attempt to focus the attention on the treatment of the shock itself.

As explained in [2], the λ -formulation of Euler's equations in a Cartesian grid leads to the following equations:

$$\begin{aligned} S_t &= f_4^x + f_4^y \\ a_t &= 0.5\delta [f_1^x + f_2^x + f_1^y + f_2^y - aS_t] \\ u_t &= 0.5 [f_1^x - f_2^x + f_3^y] \\ v_t &= 0.5 [f_1^y - f_2^y + f_3^x] \end{aligned} \quad (1)$$

with $\delta = (\gamma - 1)/2$,

$$\begin{aligned} \lambda_i^x &= u \pm a, \quad \lambda_i^y = v \pm a, \quad R_i^x = a/\delta \pm u, \quad R_i^y = a/\delta \pm v. \\ f_i^x &= -\lambda_i^x [(R_i^x)_x - aS_x], \quad f_i^y = -\lambda_i^y [(R_i^y)_y - aS_y] \\ f_3^x &= -uv_x, \quad f_3^y = -vu_y, \quad f_4^x = -uS_x, \quad f_4^y = -vS_y \end{aligned}$$

where the symbol $(\cdot)_x$ means a left or right derivative according to the corresponding coefficient, λ_i^x or u being positive or negative and the symbol $(\cdot)_y$ means a derivative from below or from above according to the corresponding coefficient, λ_i^y or v

being positive or negative. I generally use a second-order (predictor-corrector) integration scheme. At any level, the derivatives are approximated by two-point differences. The coefficients of each f -term are expressed by their two-point averages over the same interval on which the pertinent difference is taken. In principle, this procedure maintains second-order accuracy in a nonlinear problem.

At the predictor level, each f -term is divided by 2 and multiplied by Δt to integrate (1) over half a step. Each f -term is also stored in f^0 -arrays. At the corrector step, each f -term is recomputed and then corrected by subtracting the stored f^0 previously computed at the grid point next to the point to be updated, once more on the left (or from below) if the coefficient is positive, on the right (or from above) if it is negative.

5 The NEIGHBOR Subroutine

Grid points in the vicinity of shocks must be given a special treatment, essentially to avoid differentiating across a discontinuity. Such points are detected by examining the neighborhood of each shock point. By so doing, only all values of J are scanned, not the doubly-dimensioned set of grid points. First, we describe the procedure to detect all points where one or more derivatives must be approximated in a special way (Sect. 5.1), then we detail the special approximations being used (Sect. 5.2) and, finally, in Sect. 5.3, we justify the approximations by interpreting them physically.

5.1 Detection of Grid Points Neighboring Shocks

To describe the procedure we follow the computational code step by step. First, the two-dimensional index NMOUT(N,M) is set to zero at all grid points. We will see that NMOUT is set equal to 1 every time a grid point has been treated by the subroutine; this avoids going twice over the same point.

Then, a loop over all values of J begins:

```
D0 11 J=1,JA
N=NS (J)
M=MS (J)
```

The grid point next to the shock point is thus detected; it is the one marked with a P in Fig. 2.

We choose now to consider x -shocks first:

```
IF (JI (J) .LT. 2) THEN
```

Point P is taken into consideration, unless already treated:

```
IF (NMOUT (N, M) .EQ. 1) GO TO 2
```

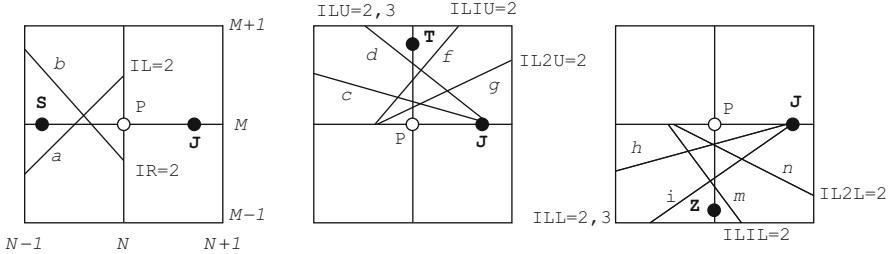


Fig. 3 Neighborhood of an x -shock point (left region)

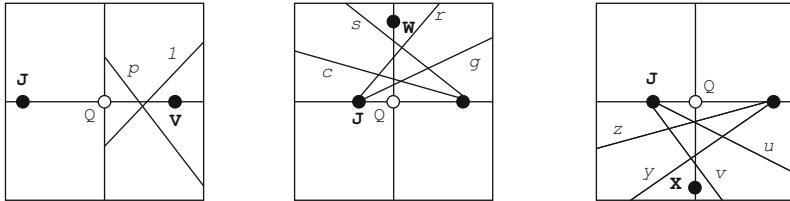


Fig. 4 Neighborhood of an x -shock point (right region)

(2 is the statement number for the next analysis). Eight indices are defined:

$$\begin{aligned}
 \text{IL} &= \text{INDY}(N, M+1) + (\text{INDY}(N-1, M) \\
 \text{IR} &= \text{INDY}(N, M) + \text{INDY}(N-1, M+1) \\
 \text{ILU} &= \text{INDY}(N-1, M+1) + \text{INDX}(N, M+1) + 1 \\
 \text{IL1U} &= \text{INDX}(N, M) + \text{INDX}(N+1, M+1) \\
 \text{IL2U} &= \text{INDX}(N, M) + \text{INDY}(N+1, M+1) \\
 \text{ILL} &= \text{INDY}(N-1, M) + \text{INDX}(N, M-1) + 1 \\
 \text{IL1L} &= \text{INDX}(N, M) + \text{INDX}(N+1, M-1) \\
 \text{IL2L} &= \text{INDX}(N, M) + \text{INDY}(N+1, M)
 \end{aligned}$$

A patient study of Fig. 3 reveals that $\text{IL}=2$ or $\text{IR}=2$ if two y -shocks define a shock line a or b , respectively; $\text{ILU}=2$ or 3, or $\text{ILL}=2$ or 3 if an additional x - or y -shock, together with shock J , define possible shock lines c , d , or h , i , respectively; $\text{IL1U}=2$ or $\text{IL2U}=2$ if an x -shock to the left of P and an x -shock or y -shock define possible shock lines f or g ; and, finally, a similar consideration can be made for $\text{IL1L}=2$ or $\text{IL2L}=2$.

5.2 Special Computations at Points Neighboring Shocks

The above indices, used in a calling sequence as the CALL FXM below, allow us to call some of the four subroutines:

```

FXM(N, M, L, IA, IB, IC)
FXP(N, M, L, IA, IB, IC)

```

```
FYM (N, M, L, IA, IB, IC)
FYP (N, M, L, IA, IB, IC)
```

for redefining some of the f -terms at point P properly. Let us consider FXM in detail. In it, all f^x -terms are set equal to zero if L or IC are different from zero, or if IA or IB are greater than 1. Otherwise, all f_i^x affected by negative λ_i^x or negative u are computed using left-differences. The same can be said for FXP, provided that *positive* is substituted to *negative* and *right* to *left*.

Similarly, in FYM all f^y -terms are set equal to zero if L or IC are not zero, or if IA or IB are greater than 1. Otherwise, all f_i^y affected by negative λ_i^y or negative v are computed using downward differences. The same can be said for FYP, substituting *positive* for *negative* and *upward* for *downward*.

Thus, the following statement:

```
CALL FXM (N, M, JSX (N, M) , IL, IR, 0)
```

takes care of the f^x -terms at P , setting all of them to zero if either S or a or b of Fig. 3 (left) are present. The next statements:

```
IF (JSY (N, M+1) .NE. 0 .OR. ILU.GT.1 .OR. IL1U.EQ.2 .OR. IL2U
    .EQ.2)
1     CALL FYM (N, M, ILL-1, IL1L, IL2L, JSY (N, M) )
IF (JSY (N, M) .NE. 0 .OR. ILL.GT.1 .OR. IL1L.EQ.2 .OR. IL2L.
    EQ.2)
1     CALL FYP (N, M, ILU-1, ILIU, IL2U, JSY (N, M+1) )
```

take care of the f^y -terms at P . The first asks the question: Does any obstacle exist above P ? If the answer is no, upward differences can be used and we proceed to the next question. Otherwise, we must take all differences from below, except if there are obstacles as well, as in the last sketch of Fig. 3, in which case all f^y -terms are set to zero (this is done by calling FYN). The second statement reverses the issues between *above* and *below*.

At this point, all f -terms have been taken care of at P and NMOUT(N,M) is set equal to 1. Then we proceed to point Q , on the other side of J (Fig. 4) where V , e , and p play the roles played before by S , a , and b . The index N is bumped to N+1 and NMOUT is tested:

```
2 N=N+1
IF (NMOUT (N, M) .EQ.1) GO TO 11
```

(11 is the end of the loop). Again, we define 8 indices:

```
IL=INDY (N, M) +INDY (N+1, M+1)
IR=INDY (N, M+1) +INDY (N+1, M)
ILU=INDY (N+1, M+1) +INDX (N+1, M+1) +1
IL1U=INDX (N+1, M) +INDX (N, M)
IL2U=INDX (N+1, M) +INDY (N-1, M+1)
```

```

ILL=INDY(N+1,M)+INDX(N+1,M-1)+1
IL1L=INDX(N+1,M)+INDX(N,M-1)
IL2L=INDX(N+1,M)+INDY(N-1,M)

```

that are explained in Fig. 4. Similarly to what we did before, we use the following statements:

```

CALL FXP(N,M,JSX(N+1,M),IL,IR,0)
IF(JSY(M,M+1).NE.0.OR.ILU.GT.1.OR.IL1U.EQ.2.OR.
   IL2U.EQ.2)
1 CALL FYM(N,M,ILL-1,IL1L,IL2L,JSY(N,M))
IF(JSY(N,M).NE.0.OR.ILL.GT.1.OR.IL1L.EQ.2.OR.
   IL2L.EQ.2)
1 CALL FYP(N,M,ILU-1,IL1U,IL2U,JSY(N,M+1))

```

Then NMOUT(N,M) is set equal to 1 and the x -shock analysis is closed by an ELSE statement.

The same analysis, with obvious changes, is repeated for the y -shocks (the entire procedure is easily modified by interchanging x and y , and N and M).

5.3 Physical Relevance of the Technique

Let us discuss the legitimacy of the redefinition of f -terms introduced in the NEIGHBOR subroutine. First, let us consider an x -shock, as in Fig. 5, and the two points, A and B , that are affected by the subroutine. In Fig. 5a the low-pressure region is to the left of the shock. The opposite happens in Fig. 5b. The flow may be supersonic at A . In the case of Fig. 5a, the flow must run from left to right; in the case of Fig. 5b, from right to left. In either case, the values in A would be computed correctly (both characteristics and the velocity vector would point toward the shock from the same direction (Fig. 6)). If the flow is subsonic at B , the $(u-a)$ -characteristic points towards the shock in Fig. 6a and the $(u+a)$ -characteristic points toward the shock in Fig. 6b. Therefore R_2^x and R_1^x are computed correctly, the former in case (a) and the latter in case (b). These are the elements needed to evaluate the shock through the function Σ , to be defined in Sect. 6.

If the flow is subsonic at A , that means that the shock moves very rapidly into the region containing A (that is, to the left in Fig. 5a and to the right in Fig. 5b). Relatively to the moving shock, the flow is still supersonic at A . Therefore, in such a frame all the f^x -terms must be approximated using up-wind differences, as the NEIGHBOR subroutine prescribes. On the other hand, if all differences are taken from the same side, certain simplifications occur and the following chains of identities result:

$$\begin{aligned}
a_t &= 0.5\delta(f_1^x + f_2^x + \dots) \\
&= -0.5\delta[\lambda_1^x(R_1^x)_x + \lambda_2^x(R_2^x)_x + \dots]
\end{aligned}$$

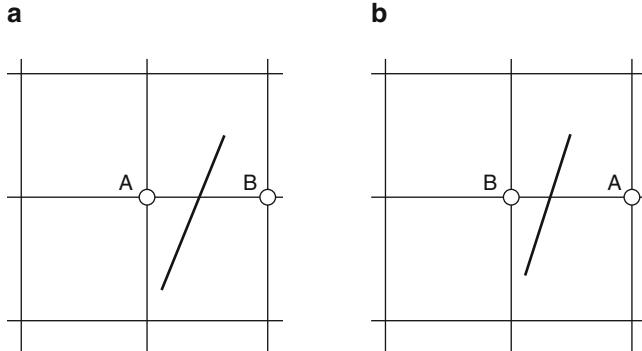


Fig. 5 Low- and high-pressure sides of an x -shock

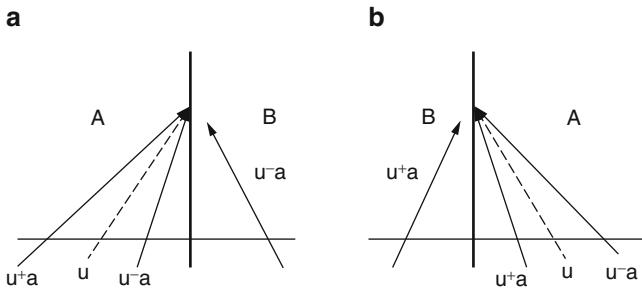


Fig. 6 Characteristics relative to a shock

$$\begin{aligned} &= -0.5\delta [(\lambda_1 + W)(R_1^x)_x + (\lambda_2^x + W)(R_2^x)_x + \dots - 2Wa_x/\delta] \\ &= a_T - Wa_x \end{aligned}$$

$$\begin{aligned} u_t &= 0.5\delta(f_1^x - f_2^x + \dots) \\ &= -0.5\delta [\lambda_1^x(R_1^x)_x - \lambda_2^x(R_2^x)_x + \dots] \\ &= -0.5\delta [(\lambda_1 + W)(R_1^x)_x - (\lambda_2^x + W)(R_2^x)_x + \dots - 2Wu_x] \\ &= u_T - Wu_x \end{aligned}$$

where W is the shock velocity, a_t , u_t and a_T , u_T are the derivatives of a and u in the fixed frame, and the moving frame respectively. Since $a_t = a_T - Wa_x$ and $u_t = u_T - Wu_x$, formally, the above results show that a_t and u_t are evaluated correctly if the f -terms are approximated as prescribed in the NEIGHBOR subroutine. A similar argument legitimizes the use of the subroutine to compute the grid values at B with the only objective, here, of getting a value for R_2^x (in case a) or R_1^x (in case b). The whole discussion can be repeated for points A and B in the case of y -shocks (Fig. 7).

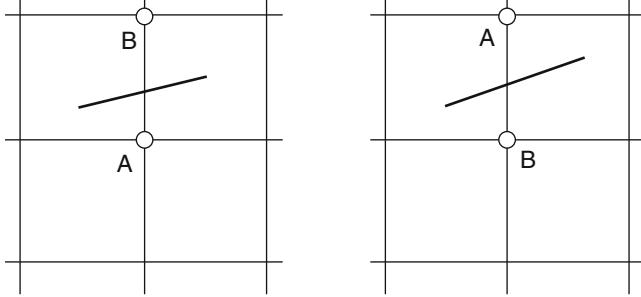


Fig. 7 Low and high pressure sides of a y-shock

6 Calculation at Shock Points

We stipulate to update all grid points, both in predictor and corrector, using the λ -scheme (or any other scheme) and the NEIGHBOR subroutine. In so doing, all points are updated correctly, except the points on the high-pressure side of a shock. At such points, however, if the grid point is next to an x -shock, we assume that one of the combinations $R_i^x = a/\delta \pm u$ is correct (the one with $i=1$ if $Ji=0$, and the one with $i=2$ if $Ji=1$; note also that the point has $\text{INDX}(N,M)=1$ if $Ji=1$ and $\text{INDX}(N+1,M)=1$ if $Ji=0$).

Let A and B be the low-pressure and high-pressure sides of the shock, respectively. At this stage, we do not make any distinction between the two points located exactly on what we called the shock point J , and the grid nodes bracketing the shock (on the same x -line for an x -shock, on the same y -line for a y -shock). We will also assume that the direction of the normal N to the shock is known, and we will denote by \tilde{u} and \tilde{v} the two components of the velocity vector, \mathbf{q} , in the direction of the normal and along the shock, respectively; therefore, if \mathbf{i} and \mathbf{j} are the unit vectors of the x - and y -lines at the shock point, and $N = N_1\mathbf{i} + N_2\mathbf{j}$, it is also $q = u\mathbf{i} + v\mathbf{j}$ and $\tilde{u} = uN_1 + vN_2$, $\tilde{v} = -uN_2 + vN_1$.

We define the following three parameters:

$$\Sigma^x = (a_B + \delta |u_B - u_A|)/a_A \quad (2)$$

$$\Sigma^y = (a_B + \delta |v_B - v_A|)/a_A \quad (3)$$

$$\Sigma = (a_B + \delta |\tilde{u}_B - \tilde{u}_A|)/a_A \quad (4)$$

If an x -shock is normal to the x -line, Σ^x is computed without errors (note that the absolute value takes care of switching R_1^x and R_2^x consistently with the relative position of A and B); this is because the contributions of the “wrong” f -terms cancel with each other, and v vanishes identically. The same can be said of Σ^y for a y -shock normal to the y -line. If an x -shock is not normal to the x -line but the shock does not

cross the y -line in the immediate vicinity of the shock point, or if a y -shock is not normal to the y -line but the shock does not cross the x -line in the immediate vicinity of the shock point, Σ^x or, respectively, Σ^y are still correct. Minor errors appear in Σ^x or Σ^y if the shocks cross the line of opposite name, because f_3^y or f_3^x may be computed from the wrong side. Such errors are negligible when compared with the relevant contributions to Σ^x and Σ^y .

For these reason, one is prompted to use Σ^x or Σ^y in dealing with x -shocks or y -shocks, respectively. The application is easy. Beginning again with an x -shock, note that, if $[]$ denotes a jump across a shock,

$$[u] = [\tilde{u}] N_1$$

since $[u] = [\tilde{u}N_1 - \tilde{v}N_2]$ but $[\tilde{v}] = 0$. The normal relative Mach number of the shock is

$$M = (\tilde{u}_A - W)/a_A \quad (5)$$

The Rankine-Hugoniot conditions are:

$$\begin{aligned} a_B &= a_A \frac{[(\gamma M^2 - \delta)(1 + \delta M^2)]^{1/2}}{(1 + \delta)M} \\ \tilde{u}_B &= \tilde{u}_A + a_A \frac{1 - M^2}{(1 + \delta)M} \\ \tilde{v}_B &= \tilde{v}_A \\ S_B &= S_A + \frac{1}{2\delta\gamma} \ln \frac{\gamma M^2 - \delta}{1 + \delta} - \gamma \ln \frac{(1 + \delta)M^2}{1 + \delta M^2} \end{aligned} \quad (6)$$

From these equations and (2) we obtain

$$\Sigma^x = \frac{1}{(1 + \delta)M} [\sqrt{(\gamma M^2 - \delta)(1 + \delta M^2)} + \delta N_1(M^2 - 1)] \quad (7)$$

which yields the basic relationship between Σ^x and M ; consequently, if Σ^x is known, M can be computed from (7). Similarly, for a y -shock, (3) and (6) can be used to get M from Σ^y .

When M is close to 1, the inversion of (7) by iterations may present some difficulty (during the process, M may occasionally become less than $\sqrt{\delta/\gamma}$, causing the radicand in (7) to become negative). It is safer to store the values of Σ^x and Σ^y at the end of the previous step into an array, SIGMA(J), and then to recompute (2) and (3) and store them into another array, SIG(J), as we start the calculation of the shock. Meanwhile, also the old values of N_1 must have been stored in one array, EN1O(J), and the new values (see Sect. 8) must be stored in another array, EN1(J). It is then possible for each shock point, to calculate the increment $\Delta\Sigma^x$ or $\Delta\Sigma^y$, and ΔN_1 .

Then, from (7) we obtain

$$\Delta M = \frac{(1 + \delta)M \Delta \Sigma^x - \delta(M^2 - 1) \Delta N_1}{\rho M - (1 + \delta)\Sigma^x + 2\delta M N_1} \quad (8)$$

where

$$\rho = \frac{\gamma - \delta^2 + 2\gamma\delta M^2}{\sqrt{(\gamma M^2 - \delta)(1 + \delta M^2)}}$$

If we stipulate that the variation of the shock slope in a single step is not as important as the variation of Σ^x and M , the following, simpler formula can be used:

$$\Delta M = \frac{(1 + \delta)}{\rho - (1 + \delta)\Sigma^x/M + 2\delta N_1} \Delta \Sigma^x \quad (9)$$

In this way, M at the end of the step is obtained. A parallel procedure allows M to be updated at all y -shocks.

The displacement of a shock point is obtained as follows. First, the normal shock velocity, W is computed from (8):

$$W = \tilde{u}_A - \tilde{a}_A M \quad (10)$$

For an x -shock (J in Fig. 8a), its new position on the x -line, J' , is obtained from

$$\begin{aligned} x_{J'} &= x_J + \Delta s \\ &= x_J + W \Delta t N_1 \end{aligned} \quad (11)$$

(note that Δs is the hypotenuse, not the cathetus of a right triangle). For a y -shock, Fig. 8b holds.

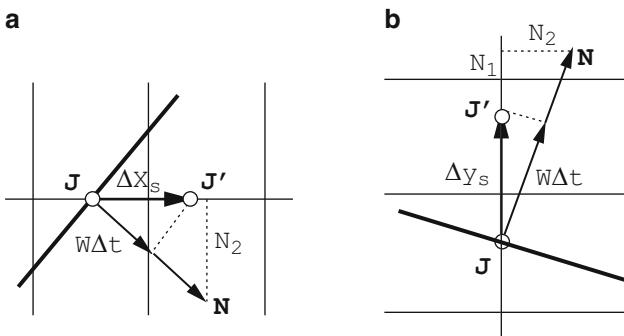


Fig. 8 Displacement of a shock point

The shock calculation is now completed by a correct updating of the values at point B , obtained using the Rankine-Hugoniot conditions (6).

Comments and further refinements of this subroutine [including the use of Σ as defined in (4)] will be found in Sect. 10.

7 Shock Detection

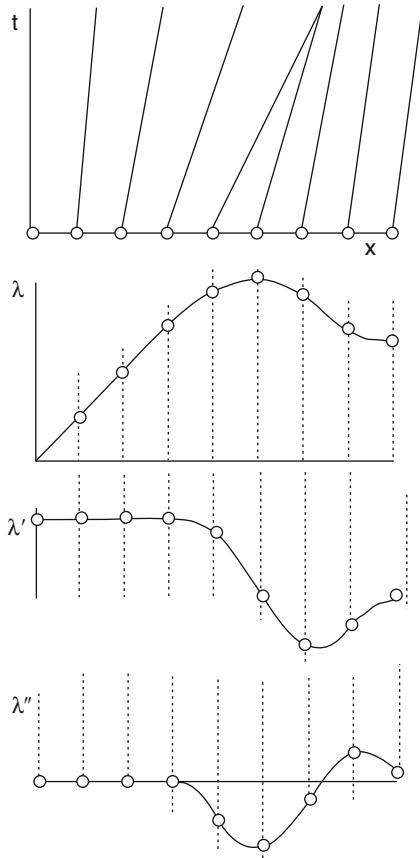
The detection of a shock as a rapid transition in some significant parameter (for example, pressure) in a compressive mode (that is, with pressure increasing in the direction of the flow) seems to be a major concern for most numerical analysts. Having applied the procedure described below to a large number of problems, however, I never found any difficulty.

7.1 Shocks Generated by Coalescence of Waves

As mentioned in Sect. 2, shocks are generated either by a gradual coalescence of compression waves or by an abrupt variation in the slope of a rigid wall. Considering the first case first. We see from Fig. 9 how coalescence of characteristics preludes to a possible formation of a shock. The upper part of the figure shows some $(u - a)$ -characteristics in the process of coalescing in time; the lower graphs represent λ_2^x and its first and second derivatives as functions of x ; at the initial time. Clearly, the inflection point of $\lambda_2^x(x)$ is the point to be monitored because it will eventually become the origin of a shock. Working on a smooth distribution of data, it is easy to evaluate an approximation to $d\lambda_2^x/dx$ by two-point differences and then again an approximation to $d^2\lambda_2^x/dx^2$ by two-point differences. The possible location of a shock point can be found by interpolation over the interval having a negative second derivative on the left and a positive second derivatives on the right. Note that coalescence will occur anyhow, regardless of the flow being subsonic or supersonic on either side; in other words, the shape of the curve, not its vertical position with respect to the horizontal axis is our subject of concern.

In the numerical handling of complicated flows, minor oscillations may occur that, although irrelevant for all practical purposes, may confuse the issues in the search for the origin of a shock. The situation, almost paradoxically, is most critical in the simplest cases. Think of a theoretically constant distribution, represented numerically by data with alternate positive and negative errors of the order of 0.00001. Obviously, a second derivative approximated as said above is also alternatively positive and negative, so that possible loci of incipient shocks would be detected at every second node. The additional testing on the shock strength, to be described below, would definitively eliminate all such accidental guesses; nevertheless, some precaution can be taken in defining the approximations to the first and second derivatives of λ_2^x . Let us simplify the notation by eliminating the

Fig. 9 Coalescence of characteristics



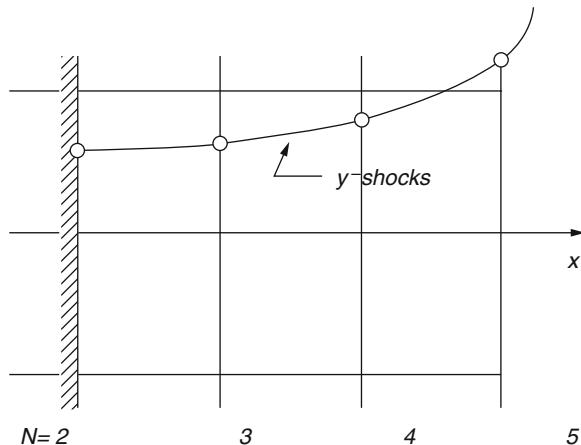
subscript and the superscript in λ_2^x and using instead a subscript to denote the nodal point where λ is defined. Imagine having replaced λ_n with its mean-square value obtained by a linear approximation over five points; the first difference, $\Delta\lambda$ is thus the slope of the approximating straight line, given by:

$$\Delta\lambda = \lambda_{n+1} - \lambda_{n-1} + 2(\lambda_{n+2} - \lambda_{n-2})$$

and the second difference, $\Delta_2\lambda$ can be obtained by subtracting $\Delta\lambda_{n-1}$ from $\Delta\lambda_n$:

$$\Delta_2\lambda|_{n-1/2} = -\lambda_{n+1} - \lambda_{n-2} - \lambda_{n-1} - \lambda_n + 2(\lambda_{n+2} + \lambda_{n-3})$$

Naturally, the test can only begin at a value of n that leaves $n-3$ still at the boundary. If the boundary is denoted by $N = 2$, the first point we can test is $N = 5$. Similarly, the test must end at point $N = NC - 3$ if the last point on the axis is denoted by NC . Testing for shocks near the boundaries, however, is not necessary. If any shock crosses such lines, most likely it is a shock of the other family (Fig. 10).

Fig. 10 Shock near a wall

The cells where the inflection points are located are found, as said above, where $\Delta_2 \lambda_n < 0$ and $\Delta_2 \lambda_{n+1} \geq 0$. We should also make sure that the pressure (or, equivalently in this as yet isentropic environment, the speed of sound) increases in the right direction. To this effect, it is convenient to determine Δa using the same mean-square linear approximation mentioned above for $\Delta \lambda$. A cell containing an inflection point is temporarily marked, and a tentative location for the incipient shock is defined:

$$x_s = x_n - \frac{\Delta_2 \lambda_n}{\Delta_2 \lambda_{n+1} - \Delta_2 \lambda_n} \Delta x$$

The point is not accepted as a shock point under three circumstances:

- (1) Σ^x , as defined by (2), is less than a prescribed tolerance, or
- (2) the pressure jump is too low, or
- (3) there is a shock point already in the cell or in any of the two adjoining cells.

The first criterion is used to discard “shocks” with too low a relative Mach number; $\Sigma^x > 1.02$ is a reasonable criterion, eliminating shocks with $M < 1.03$. The second criterion is necessary to discriminate between shocks and contact discontinuities (as in the breaking of a diaphragm in a cylindrical pipe). For a shock generated by coalescence of characteristics in a still isentropic environment, a test on a is equivalent to a test on p since p is proportional to a power of a ; therefore, the second criterion, if applied with the proper tolerance, is a redundant repeat of the first. In the case of an original contact discontinuity, instead, the speed of sound may change strongly in one cell, with pressure and velocity remaining practically the same. Thus, the value of Σ may largely exceed the prescribed tolerance and make a contact discontinuity be accepted as a shock and computed accordingly. The third

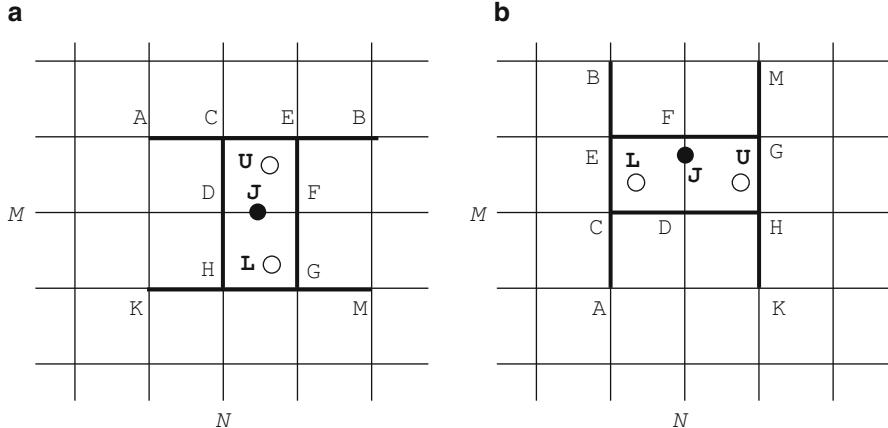


Fig. 11 Neighborhood of a shock point

criterion is simply meant to avoid overcrowding of shock points. It is interesting to note that it can be easily written in the form:

```
IF (JSX (N, M) +JSX (N+1, M) +JSX (N+2 , M) .NE .0) GO TO 2
```

where 2 is a statement outside and beyond the introduction of a new shock point.

If the new shock point is accepted, J_1 is set equal to 0 or 1 according to the rule mentioned in Sect. 3. Then, a preliminary estimate of the shock slope at that point is made. With reference to Fig. 11a, we define two neighborhoods of the new shock, J ; the upper neighborhood, formed by the segments AB , CD , and EF , and the lower neighborhood, formed by the segments FG , DH , and KM . We search for other shock points on the upper neighborhood. The index, $INDP$, defined by

$$\begin{aligned} INDP = & \text{INDX}(N, M+1) + \text{INDX}(N+1, M+1) \\ & + \text{INDX}(N+2, M+1) + \text{INDY}(N, M+1) + \text{INDY}(N+1, M+1) \end{aligned}$$

equals the number of such shocks. If $INDP$ differs from zero, the averages of the x - and y -coordinates of the shocks are evaluated and define a single point, U , as the upper neighbor of J . The same procedure generates a lower neighbor, L , if any shock point exists in the lower neighborhood of J . If neither U nor L are found, the new shock point, J , is still isolated and it is dropped from consideration.

There is a possibility (particularly when the grid is very fine) to have no shocks in the region explored as shown in Fig. 11, but still a significant shock located, for example, as U in Fig. 12. In such a case, it is convenient to extend the search to a line above $m + 1$ and a line below $m - 1$ (for an x -shock), limitedly to shocks of the same family.

If both U and L exist, the abscissa of J is corrected to be a weighted average between the abscissas of U and L . If only L exists, U is made to coincide with J ; if only U exists, L is made to coincide with J . If at least one neighbor exists, the

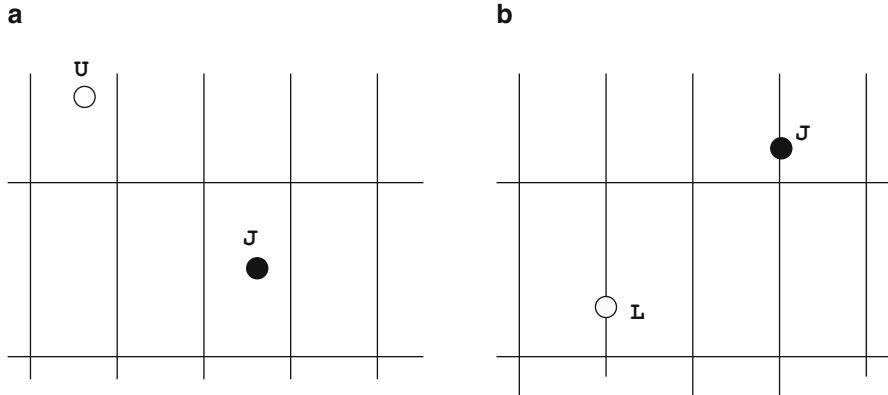


Fig. 12 Neighborhood of a shock point

angle α is defined using the coordinates of U and L , as the angle between UL and the x -line:

$$\alpha = \arctan[(x_U - x_L)/(y_U - y_L)] \quad (12)$$

Please take account of some additional comment on the definition of α in Sect. 10.

The same procedure is repeated for y -shocks, working along y -lines. The above outline can be used, replacing $u \pm a$ with $v \pm a$, x with y , JSX with JSY, increments and decrements of N with increments and decrements of M , and switching INDX and INDY with one another. The “upper” and “lower” points are now points to the right and to the left of the y -line, respectively. The index, J_1 , has to be set equal to 2 or 3 according to Δa being negative or positive. Figure 11a is replaced by Fig. 11b and Fig. 12a by Fig. 12b. Equation (12) is replaced by

$$\alpha = \arctan[(y_U - y_L)/(x_U - x_L)] \quad (13)$$

7.2 Shocks Produced by Abrupt Changes in Geometry

If, somewhere in a supersonic flow, the slope of a rigid wall changes abruptly, locally producing a sudden, non-isentropic jump in the flow parameters, two possibilities must be considered:

1. The perturbation is large; a local subsonic region is formed, preceded by a normal shock, or
2. The perturbation is sufficiently small; the flow is deflected abruptly across an oblique shock, attached to the corner in the wall.

Both situations may occur under different circumstances:

- (a) in a fixed geometry, by an impulsive start of the calculation with supersonic flow prescribed.
- (b) in a variable geometry, when a wedge or a bump develops above a previously straight wall.

In any event, the same procedure can be followed, that consists of freezing the environment of the corner during the calculation and looking at the possible existence of a steady, attached shock. The scale of the flow-field in the immediate vicinity of the corner is indeed very small and an unsteady flow may be treated as a succession of quasi-steady fields.

The input data are: the velocity vector at the wall node immediately ahead of the corner, and the deflection angle produced by the wall, ϵ . The angle θ between an attached shock and the impinging flow is defined by:

$$\zeta^3 + b\zeta^2 + c\zeta + d = 0 \quad (14)$$

where

$$\begin{aligned} \zeta &= \sin^2 \theta, \quad b = -2M_1^2 - 1 - \gamma \sin^2 \epsilon, \\ c &= (2M_1^2 + 1)/M_1^4 + [(1 + \delta)^2 + 2\delta/M_1^2] \sin^2 \epsilon, \quad d = -(\cos^2 \epsilon)/M_1^4 \end{aligned}$$

and $M_1 = q/a$ at the node on front of the corner.

A subroutine to solve (14) is needed. We can make it as a function defining θ by the code name SHANG, as follows:

```
FUNCTION SHANG (SM, SDO, GAMMA, PI, KUT)
SHANG=0.
GD=.5*(GAMMA-1.)
KUT=1
B=-1.-2./SM-GAMMA*SDO
C=2./SM+1./SM**2+((1.+GD)**2+2.*GD/SM)*SDO
D=-1./SM**2*(1.-SDO)
P=(3.*C-B**2)/9.
Q=(2.*B**3-9.*B*C+27.*D)/54.
IF (Q**2+P**3.GT.0.) RETURN
KUT=0
SQRP=SQRT(-P)
ANG=ASIN(-Q/(P*SQRP))
SIG1=SIN(ANG/3.)
SIG2=SIN((2.*PI+ANG)/3.)
SIG3=SIN((4.*PI+ANG)/3.)
SIGMAX=AMAX1(SIG1,SIG2,SIG3)
SIGMIN=AMIN1(SIG1,SIG2,SIG3)
```

```

    IF(SIG1.EQ.SIGMIM.OR.SIG1.EQ.SIGMAX) GO T0 1
    SHANG=ASIN(SQRT(2.*SQRP*SIG1-B/3.))
    RETURN
1   IF(SIG2.EQ.SIGMIN.OR.SIG2.EQ.SIGMAX) GO T0 2
    SHANG=ASIN(SQRT(2.*SQRP*SIG2-B/3.))
    RETURN
2   SHANG=ASIN(SQRT(2.*SQRP*SIG3-B/3.))
    RETURN
END

```

In the calling sequence, SM is the square of q/a in front of the corner, SDO is the square of $\sin \epsilon$, GAMMA is the ratio of specific heats, PI is π , and KUT its a flag defined by the function itself. We recall that, if (14) has a single real root, the shock cannot be attached; in such a case, the functions return with KUT=1 to let the code proceed without attempting to force an attached shock at the corner (the detached shock will then be detected as explained above in (A) and it will be forced to be normal to the wall). If (14) has three real roots, the one of interest is the middle one (the other two define a strong attached shock and an expansion shock, respectively). The function is thus defined, and KUT set equal to 0.

8 Shock Tracking; Elimination of Redundant Shocks

When the detection of new shocks is completed, the arrays providing the shock information may contain data pertaining to shock points already existing at the previous computational step, as well as data pertaining to newly detected points.

8.1 Elimination of Shocks

The first question to be answered is. “Does any mesh interval contain more than one shock?” To answer the question, the entire array of shock points is checked. For each value of J, we test, in the case of an x -shock, whether $\text{JSX}(\text{NS}(J)+1,\text{MS}(J))$ equals J or not. If it does not, that means that there are two shocks, one defined by the index J and another by the index $\text{JSX}(\text{NS}(J)+1,\text{MS}(J))$, in the same interval. The shock that has the smaller Mach number (TMA) is dropped. We proceed similarly if J denotes a y -shock and $\text{JSY}(\text{NS}(J),\text{MS}(J)+1)$ is not equal to J. Another reason for eliminating a shock is its progressive weakening produced by interactions with expansion waves; this is accomplished by testing whether Σ^x (or Σ^y) are below the prescribed tolerance and dropping the corresponding shock accordingly.

A shock point cannot be eliminated within a DO-loop ranging over the total number of shock points (JA), because JA itself decreases by 1 any time a shock is dropped. Keeping this in mind, we build an explicit loop in the form:

```

J=0
1   J=J+1
2   IF (J.GT.JA) GO TO 3

```

Here we test for possible causes of shock elimination. If any of them is found, the code is sent to statement 4. Otherwise, it continues as follows:

```

GO TO 1
4   CALL MOVE (J)
GO TO 2
3   CONTINUE

```

The subroutine MOVE restores JSX to zero at NS(J)+1, MS(J) or JSY to zero at NS(J), MS(J)+1, according to J being an x -shock or a y -shock, for all the shock points. Then, it moves the last point in the shock array (JA) to the position J, decreases JA by 1, and redefines all JSX and JSY for all the shock points.

8.2 Shocks Crossing Mesh Lines

Next, the question to be answered is: “Did any point, in its motion along an x -line or a y -line, cross over a line of the opposite family?” To answer the question in the case of x -shocks, a new value of NS, called NSN, is computed from the new value of XS for each point. If NSN and NS differ, not only NS is updated but some field values must be changed. Using Fig. 13, assume that the shock point, P has moved to a new position, Q . The flow values at A , that originally pertained to one side of the shock, now pertain to the opposite side; consequently, the code replaces them with the new values at B . Opposite action has to be taken if the shock point moves from Q to P ; the values at A must be replaced with the new values at C . One proceeds in a similar way for y -shocks. Obviously, after checking the crossings, the distributions of JSX, JSY and INDX, INDY are updated. Note: if the shock moves from P to Q , point C must acquire the values of JSX and INDX previously attributed to point A , and JSX, INDX at A must be made equal to zero. Similarly, if the shock moves from Q to P , the values of JSX, INDX formerly at C must be moved to A , and they must be set equal to zero at C .

Fig. 13 Shock point moving across mesh lines

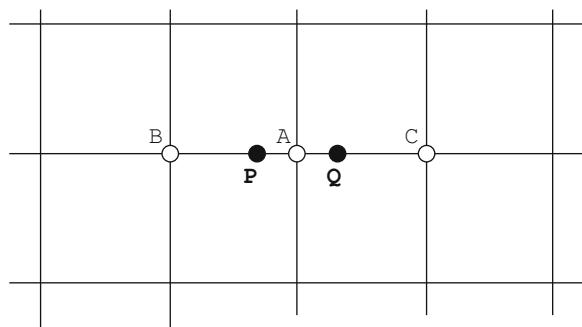
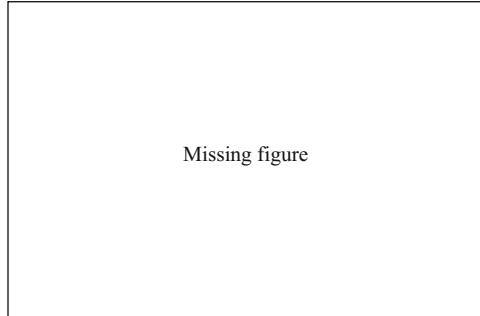


Fig. 14 Illustrating a need for elimination of shocks



8.3 Second Process of Elimination

After analyzing the crossings, the entire procedure described in Sect. 8.1 is repeated. The apparent redundancy can be justified with the help of Fig. 14. In Fig. 14a, we see two shocks, J and K , in the same cell, prior to the analysis of crossings. After the analysis, J remained in the same cell but K went to the next. One of these two shocks should have been eliminated before letting K cross over. In fact, to all practical effects, J should have been equal to K ; their velocities should have been the same and they should have crossed over together. If this did not happen, most probably it is because J is a newly found shock point, having a poorly defined velocity. If we do not kill J before K crosses over to the next cell, a test for two shocks in one cell made after the crossing fails and J is kept alive in the next computational step. In principle, this should not be a cause of concern; one of the two shocks should have a vanishing strength and be dropped at the end of the step. Unfortunately, according to the sequence of events in the calculation, the “good” shock could be dropped, leaving the “bad” one in a wrong position with a wrong strength; this may trigger a series of catastrophic events or, at least, produce a perturbation that may require many computational steps to be ironed out. Figure 14b shows the opposite case, where two shocks in two different cells fall into the same cell after crossing; this requires the test for two shocks in one cell to be taken after the crossing.

Then, a search for neighbors of all shock points is performed again, as described in Sect. 7. All isolated shocks are dropped. For the remaining shocks, the values of α are defined by (12) or (13). Shock points whose $|\alpha|$ is larger than 50° are dropped (a shock of the opposite family should appear in their immediate neighborhood as more representative). See also, in Sect. 10, how to drop points that may have been detected as belonging to oblique shocks but are, instead, on an oblique contact discontinuity.

At this stage, we have all the ingredients needed to define Σ^x or Σ^y , according to (2) or (3), at all shock points. All points where Σ^x or Σ^y is less than 1.02 are dropped. The normal, relative Mach number, M is evaluated according to (7) for x -shocks or its counterpart for y -shocks. The iterations needed to get M are, at this stage, quite safe. The values of Σ^x or Σ^y , N_1 and M will be used at the next computational step as initial values for the shock computation.

Here is the function TMD that I use to define $M(\Sigma, N_1)$:

```

FUNCTION TMD(SIG, EN1, GI, GL, GAMMA)
GD=.5*(GAMMA-1.)
GE=1.+GD
G1=3./ (GI+GL*EN1-1.)
TMD=1.+G1*(SIG-1.)
KIP=1
1  SM=TM**2
DSIG= (SQRT( (GAMMA*SM-GD)*(1+GD*SM) ) +GD*(SM-1)*EN1)
      / (GE*TM)
SIG1=SIG-DSIG
TMD=TMD+G1*SIG1
KIP=KIP+1
IF (ABS(SIG1).LT.1.E-8) RETURN
IF (KIP.LT.30) GO TO 1
WRITE(6,*) FAILURE AT TMD
RETURN
END

```

9 Finding a Robust Normal to a Shock

In Sect. 6, we saw that two parameters are necessary to determine the shock normal relative Mach number, Σ^x (or Σ^y , or Σ , as the case may be) and N_1 . The former is easily and safely computed using information from the nodes (see also Sect. 11 for more details). The second, instead, requires a knowledge of the shock geometry and it is the most difficult parameter to determine for shock-fitting. There are several sources of difficulties. Firstly, the geometric pattern of all shocks as a whole is not a basic information of the code. We should keep in mind that the shock pattern is two-dimensional, but the computer (at least, so far) works essentially in a one-dimensional mode. The shock information is stored at random, without any reference to the structure of the shocks in the computational plane; this makes creation, manipulation, and suppression of shock points very easy and the related subroutines independent of the physical problem under scrutiny. To determine the normal to the shock at any of its points, however, we must know which points lie on each side of it along the same shock line. We have seen in Sect. 7 how such points can be detected; here we want to examine a few typical cases, to evaluate the efficiency and correctness of the technique.

If complex shock patterns develop, care must be taken not to let the orientation of the normal at any given point change abruptly from step to step or, even worse, oscillate back and forth between two widely different values. This is, most probably, a non-physical behavior; the flow downstream is forced to undergo violent changes; the instability may not remain confined to the immediate vicinity of the shock point and catastrophic instabilities may even be triggered. Situations of this kind are likely

to appear at triple points. See, for example, in Fig. 15 a possible evolution of shocks at a typical Mach reflection. Point A, the triple point, is not detected as a shock point. During the evolution, shock points appear and disappear, as shown in the figure.

In this case, all shock points are x -points. From the figure, the slope of the shocks at each point is clear, but we must assure that our technique for detecting “upper” and “lower” points provides a reasonable approximation to such slopes, and that changes occur progressively and with a reasonable smoothness (taking into account that the advances in time are themselves discretized). The three crucial points in this figure are B , D , and E . Figures 16, 17, 18 show what the “upper” and “lower” points

Fig. 15 Moving triple point

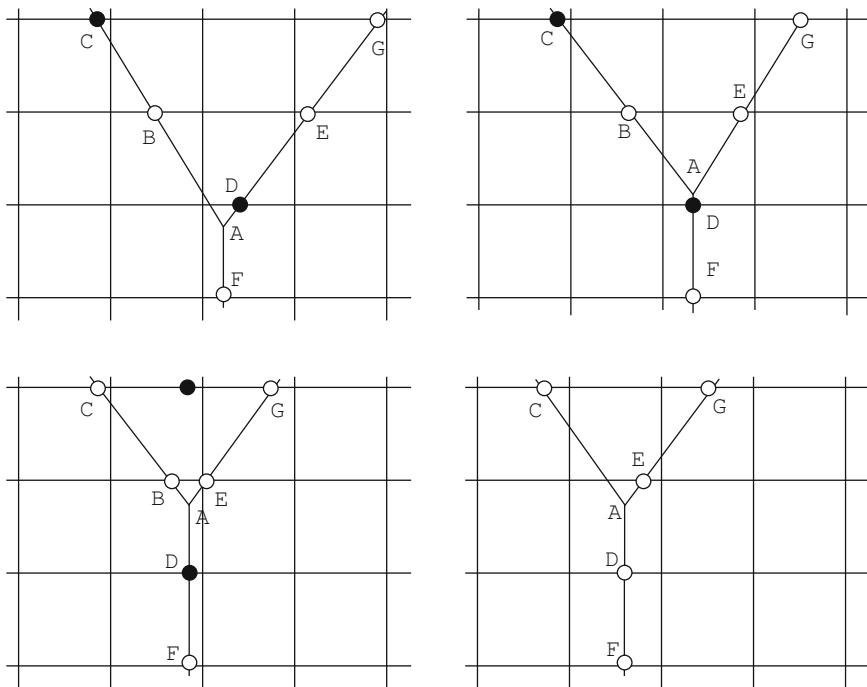
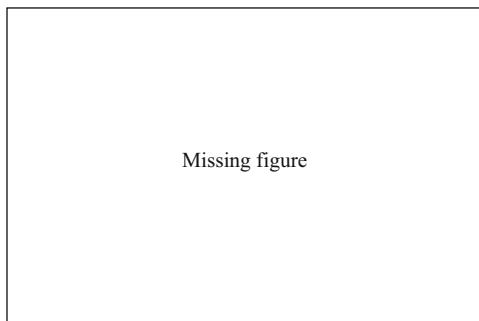


Fig. 16 Definition of slopes at B

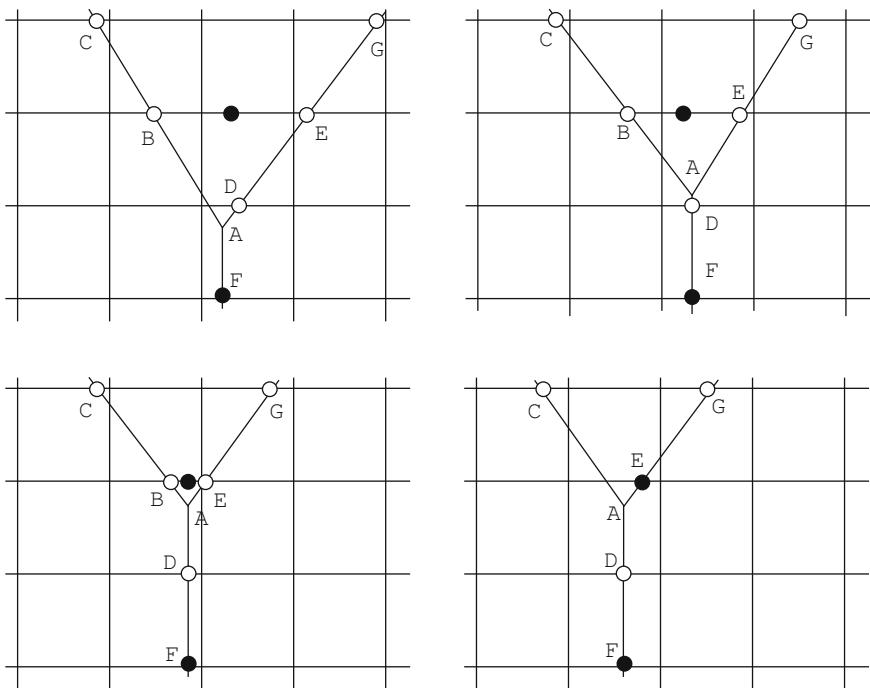


Fig. 17 Definition of slope at D

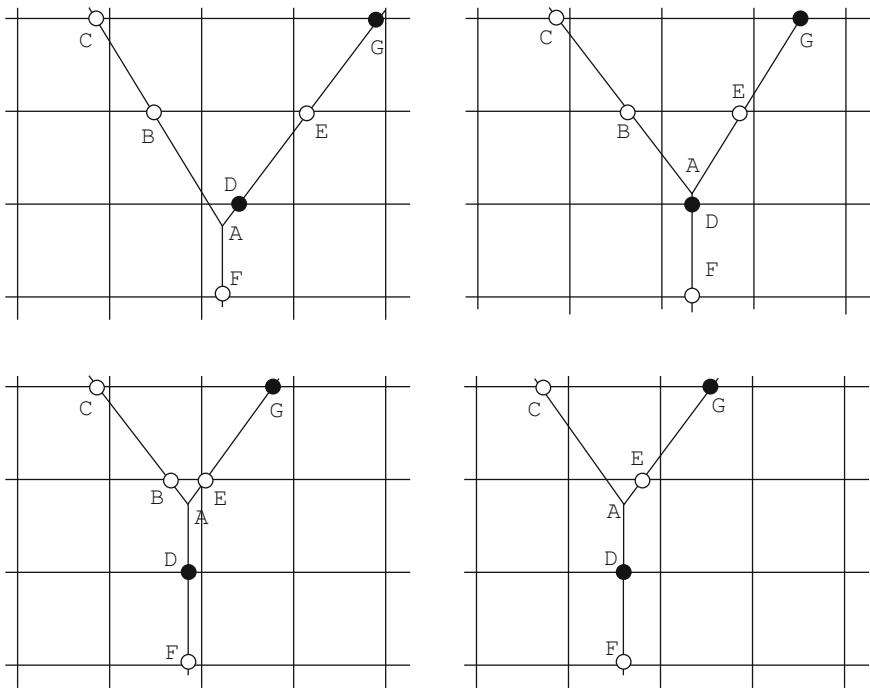
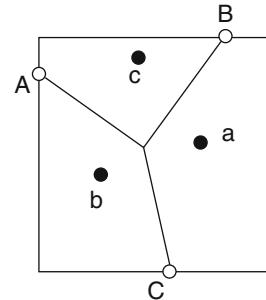


Fig. 18 Definition of slope at E

Fig. 19 Definition of slopes at a triple point



are, for B , D and E , respectively; such points are marked by black dots. In the first two diagrams of Fig. 16 the slope, as determined by the black dots, is very close to reality; in the third, it seems to depart substantially. Later, we will come back to this pattern and discuss a possible variant. Now, we limit ourselves to observe that the apparently abnormal definition of the slope at B is simply an anticipation of the fact that B itself is about to be swallowed by the vertical stem of the shock.

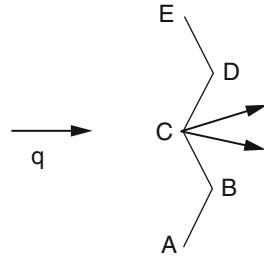
The numerical definition of the slope at D does not create any problem in the present case; D is, indeed, always considered as belonging to the vertical stem. For point E , considerations similar to the ones exposed above for B are in order. As the vertical stem lengthens and swallows more and more of the left oblique shock, the slope at E tends to become vertical, as it should be since at the next time step (not shown in the graphs) point E becomes itself a part of the vertical stem.

Many other patterns, involving y -shocks as well as x -shocks, can appear in the course of a calculation but, as far as we know, the substantial facts remain the same. I would like to conclude the presentation of examples by showing another typical shock interaction pattern (Fig. 19) in which three shock points (in this case, two of the x -type and one of the y -type) appear on three sides of the same cell. The shock points are labeled A , B , and C ; the black dots, labeled a , b , and c , are the points to be used to determine the slopes at A , B , and C , respectively, according to the technique described above. Again, we may see that the slope thus determined is quite acceptable.

9.1 Eliminating Wrinkles on a Shock

Another problem in shock-fitting is the stability of a shock line itself, regardless of interactions. We must consider two cases.

The first is the occasional wrinkling of a shock line. A real shock is a taut surface. Consider, indeed, a wrinkled shock ($ABCDE$), as shown in Fig. 20. Assuming that the impinging flow is supersonic, uniform and its velocity vector, \mathbf{q} , is horizontal, the flow downstream of BC would be tilted downwards, and the flow downstream of CD upwards. Such a pattern would be physically impossible; at B a Mach reflection

Fig. 20 Wrinkled shock

would appear producing a local strengthening of the shock and, consequently, pushing it to the left; at C an expansion would be produced, weakening the original shock and pulling it to the right. Numerically, wrinkles may appear if the slope of the shock is computed as the slope of the straight line joining the “lower” and “upper” points. Another glance at Fig. 20 explains why. Suppose that the solution to the problem were a vertical, steady shock, separating a supersonic flow on the left from a subsonic flow on the right; and suppose that, for any numerical accident (even a different round-off in the calculation of two adjacent shock points), a minor wrinkle appeared (grossly amplified in Fig. 20). In evaluating the slope at B as the slope of the line between A and C , and the slope at C as the slope of the line between B and D , both lines would still be vertical; no correction would be made to the shock Mach numbers and the original error in their velocities W , that had produced the original wrinkle, would not be corrected. The wrinkle would not be ironed out and it could worsen, if the original error persisted. If the shock is oblique and the flow is supersonic on both sides of it, the remedy is very simple: it suffices to evaluate the slope of the shock using one-sided differences (that is, using the coordinates of the point itself and of either the “upper” or the “lower” point). By so doing, the actual slope between two adjacent shock points will be used. How should one choose between one side or another? For oblique shocks embedded in supersonic flows, the slope should be computed from the side opposite to the deflection of the flow (Fig. 21). In practice, we will use the “lower” point for x -shocks having $u > 0$, $N_2 < 0$ or $u < 0$, $N_2 > 0$ and for y -shocks having $N_2 < 0$.

For shocks followed by a subsonic flow it is more convenient to use two-sided differences to give equal weight to all points in the subsonic environment. In this case a slight correction on the value of Σ defining the shock Mach number should be used. Without attempting a sophisticated but complicated evaluation of the correction factor, we can argue as follows: The correction, as Fig. 20 shows, is related to the local curvature of the shock curve. For an x -shock, if the second difference of the x -coordinates is negative (as at point B in Fig. 20) and $u > 0$, Σ must be increased to make the shock stronger; if $u < 0$, Σ must be decreased. The same can be said for a y -shock, provided that y and v are used in lieu of x and u . The opposite should be done when the second differences are positive. If we write the correction to E in the form:

$$\Sigma(1 \pm \epsilon \Delta_2 x)$$

the value of ϵ should be of the order of 0.0001.

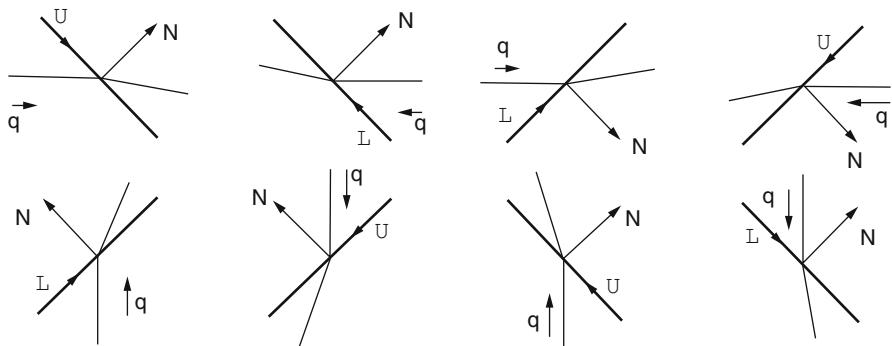


Fig. 21 One-sided differences on shocks

10 Missing Section

.....

11 Missing Section

.....

12 Missing References

.....

Steady and Unsteady Shock Interactions by Shock Fitting Approach

Francesco Nasuti and Marcello Onofri

1 Introduction

In the last decades computational fluid dynamics (CFD) has grown its predicting capabilities becoming more and more an indispensable design tool. The most successful numerical approaches have undergone both a refining process, eventually leading to robust codes able to treat a wide range of complex problems, and an analysis of the limits connected to their own nature, that continues to encourage the study of less popular approaches. One of them is the shock fitting technique, left aside after its success in the sixties and seventies [8], and being still considered as an option by a few researchers [14, 24, 25]. The main merit of shock fitting is its capability of considering shocks as discontinuities, that is as they actually are in all scales larger than the molecular scale. Moreover, it allows discretizing the equations without passing through the formulation in “conservation” variables. The latter merit greatly reduces the numerical error because of the “quasi-linear” form of the equations and especially because of the possibility of naturally describing wave propagation phenomena, respecting the domains of dependence. These merits yielded the solution of the inviscid blunt body flows by just a few dozens of cells [10].

While shock fitting was considered the best approach to solve blunt body flows, where the shock was taken as a boundary of the computational domain, it enjoyed less success for complex problems featuring multiple shocks. Nevertheless, attempts were done to make the approach as general as possible for two-dimensional problems including inviscid and viscous flows in both steady-state and unsteady regimes, and also to solve specific three-dimensional problems [2]. The most general

F. Nasuti (✉) • M. Onofri

Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome,
via Eudossiana 18, 00184 Roma, Italy

e-mail: francesco.nasuti@uniroma1.it; marcello.onofri@uniroma1.it

technique was that of letting shocks move over a fixed grid: it was referred to as floating shock fitting [6]. In this paper results obtained by using the floating shock fitting technique for steady and unsteady shock interactions are presented. The technique is coupled with a solver of governing equations for the “continuous” regions of the flow. The solver is an extension to viscous flow of the “lambda-scheme” introduced by Moretti [4]. Further extension to more complex problems has been made on the basis of a multi-block procedure introduced in [15].

2 Governing Equations

The governing equations for the present study of steady and unsteady turbulent compressible flows are the two-dimensional (linear and axisymmetric) Reynolds-Averaged Navier-Stokes equations (RANSE), written in nonconservative form. They are integrated by an explicit second-order time and space accurate scheme that follows Moretti’s “lambda” formulation for the convective terms, and uses central differencing for the viscous terms [13]. Turbulence is treated according to the one-equation model of Spalart and Allmaras, including a correction that takes into account the effect of compressibility in shear-layers [23, 26]. The present technique features less nonlinearity than the conservative form and thus it yields the same error with less cells in the inviscid regions of the flow field. Moreover, it does not require any flux limiter like those needed in conservative approaches to capture shocks, provided that shocks are treated as discontinuities by shock-fitting.

Following the approach proposed in [3], which analyzes the mathematical properties of the non-conservative form of Navier-Stokes equations written in terms of speed of sound a , velocity \mathbf{v} , and entropy s , the equation set is considered here in nondimensional variables as:

$$\frac{1}{\delta} \frac{D a}{D t} + a \nabla \cdot \mathbf{v} - a \frac{D s}{D t} = 2\delta a V_s \quad (1)$$

$$\frac{D \mathbf{v}}{D t} + \frac{a}{\delta} \nabla a - a^2 \nabla s = \mathbf{V}_m \quad (2)$$

$$\frac{D s}{D t} = V_s \quad (3)$$

where t denotes time, γ the ratio of specific heats, $\delta = (\gamma - 1)/2$, and the nondimensional viscous terms V_s , \mathbf{V}_m are:

$$V_s = (\nabla \cdot \mathbf{q} + \nabla \mathbf{v} : \mathbf{T})/(\gamma p) \quad (4)$$

$$\mathbf{V}_m = (\nabla \cdot \mathbf{T})/(\rho) \quad (5)$$

where p denotes pressure, ρ density, \mathbf{T} the stress tensor and \mathbf{q} the heat flux.

The numerical integration of (1)–(3) is based on the separation of the role of the viscous terms, whose derivatives are discretized by central differencing [13], from the convective terms, which are treated following the λ -scheme [6] by upwind differences, to emphasize the effect of the propagation of signals.

As a consequence, the left-hand sides of the Eqs.(1)–(3) are reformulated following [6]. The propagation phenomenon is simulated by signals running along four bicharacteristic lines and the streamline direction. In particular, assuming an orthogonal grid and two unit vectors \mathbf{i}, \mathbf{j} , orthogonal to each other and parallel to the coordinate lines at each point, the equations are recast, using the two velocity components $u = \mathbf{v} \cdot \mathbf{i}$, and $v = \mathbf{v} \cdot \mathbf{j}$, as:

$$\begin{aligned} a_t &= \delta(f_1^x + f_2^x + f_1^y + f_2^y + 2\delta a V_s + a s_t) \\ u_t &= f_1^x - f_2^x + f_3^y + \mathbf{V}_m \cdot \mathbf{i} \\ v_t &= f_1^y - f_2^y + f_3^x + \mathbf{V}_m \cdot \mathbf{j} \\ s_t &= f_4^x + f_4^y + V_s \end{aligned} \quad (6)$$

where the subscript (t) denotes the time derivative and the terms f_p^q indicate the convective contributions along each bicharacteristic line. For instance, f_1^x is the contribution carried along the bicharacteristic λ_1^x to the propagating signal, f_2^x is the contribution along the bicharacteristic λ_2^x , and so on. In particular, they are:

$$\begin{aligned} f_1^x &= -0.5\lambda_1^x[(R_1^x)_X - v\alpha_X - a s_X] \\ f_2^x &= -0.5\lambda_2^x[(R_2^x)_X + v\alpha_X - a s_X] \\ f_3^x &= -\lambda_3^x(v_X + u\alpha_X) \\ f_4^x &= -\lambda_3^x s_X \end{aligned} \quad (7)$$

$$\begin{aligned} f_1^y &= -0.5\lambda_1^y[(R_1^y)_Y + u\alpha_Y - a s_Y] \\ f_2^y &= -0.5\lambda_2^y[(R_2^y)_Y - u\alpha_Y - a s_Y] \\ f_3^y &= -\lambda_3^y(u_Y - v\alpha_Y) \\ f_4^y &= -\lambda_3^y s_Y \end{aligned} \quad (8)$$

where the subscripts (X) and (Y) denote space derivatives in the X and Y directions, respectively, and α in the above equations and G in the following are the metric coefficients of the mapping between the computational (X, Y) and the physical (x, y) planes, as defined in [6]. The other terms appearing in (7)–(8) are defined as:

$$\lambda_1^x = G(u + a), \quad \lambda_2^x = G(u - a), \quad \lambda_3^x = Gu \quad (9)$$

$$\lambda_1^y = G(v + a), \quad \lambda_2^y = G(v - a), \quad \lambda_3^y = Gv \quad (10)$$

$$R_1^x = a/\delta + u, \quad R_2^x = a/\delta - u \quad (11)$$

$$R_1^y = a/\delta + v, \quad R_2^y = a/\delta - v \quad (12)$$

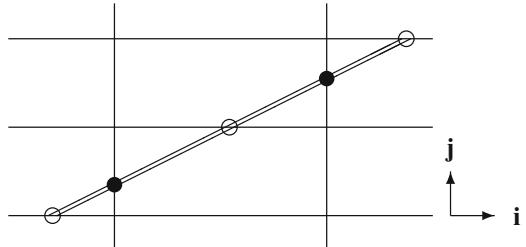
The integration of (6) is performed following the two level explicit scheme, having second-order accuracy, described in detail in [6], and extended to viscous flows in [13].

3 Floating Shock-Fitting

In the framework of shock-fitting, an important step towards a technique capable of handling complex flow configurations, including shock interactions, was undertaken by Moretti with the development of the floating shock-fitting, which led to the outstanding results presented in [7] and relevant to typical shock interaction test cases. Some details of the technique were only published in [5], whereas more details are presented now in [9]. In the floating shock-fitting the discontinuities are allowed to move (float) freely over a fixed background structured grid. A shock front is described by its intersections with grid lines, which give rise to shock points, as shown in Fig. 1.

Moretti's claim was that shock fitting codes were simple and provided fast and accurate solutions if coupled with a suitable solver of Euler equations, as the λ -scheme proposed by himself. He started from the assumption that shock-fitting simplicity was obvious in 1D problems, where the shock depends on its environment through a Riemann variable which can be correctly computed on both sides of the shock. In a one-dimensional problem, the values of the variables in the mesh node on the low pressure side of the shock (λ_A) and the value of one Riemann variable ($R = 2a/(\gamma - 1) \pm u$, where a is the local speed of sound and u the local velocity) in the mesh node on the high pressure side are correctly computed by λ -scheme [6]. On the other hand, the values on the high pressure side of the shock (λ_B) will be

Fig. 1 Shock points: (open circle) x -shock, (filled circle) y -shock



updated by means of the Rankine-Hugoniot relations once the shock strength M_s , defined as:

$$M_s = \frac{|u_A - W_s|}{a_A} \quad (13)$$

has been computed. To evaluate M_s , a new variable Σ is introduced [11], computed from the variables on the low pressure side and the Riemann variable on the high pressure side of the shock as:

$$\Sigma = (a_B + \delta|u_A - u_B|)/a_A \quad (14)$$

Since Σ can also be expressed as a monotonic increasing function of M_s :

$$\Sigma = \frac{\left[\sqrt{(\gamma M_s^2 - \delta)(1 + \delta M_s^2)} + \delta(M_s^2 - 1) \right]}{(1 + \delta)M_s} \quad (15)$$

the shock strength M_s is obtained by inverting (15).

According to Moretti, simplicity is not lost in two dimensions, where the equations are only slightly altered by a slope factor and one more Rankine Hugoniot condition. The topological problem of evaluation of shock slope was solved by suitably analyzing the shock point neighborhood. Before looking to neighbors let us consider first with a closer attention what is a shock point and how it can be classified and used in numerical simulations. When a shock occurs in a two-dimensional field it can cross one or both families of coordinate lines. Therefore we can identify x and y shocks accordingly. Moreover, for each of them one can identify the high pressure side of the shock. The solution of the topological problem of shock slope evaluation was found storing information relevant to shock points in single arrays without ordering them in any particular way. Shock slope was therefore obtained looking to the immediate neighborhood of the shock point. Let us consider, for instance, the x shock point J sketched in Fig. 2. Its neighborhood is made by the two cells surrounding it. The task of identifying which is a neighboring shock point of a given shock point as J in Fig. 2, if any exist, is carried out as follows. In case of ordinary shock points, which are defined as shock points J with no more than two shock points in their neighborhood, one (J_i , $i = 1, 3, 5$) in the mesh above (right) and/or one (J_j , $j = 2, 4, 6$) in the mesh below (left). In the general case, the connection of the shock points around J used to compute θ_s (the shock angle with respect to coordinate line directions) has to be carried out according to a physical criterion based on the domain of dependence. From this point of view, two cases are possible:

1. *sup-shock* when the velocity component along the shock direction (\tilde{v}) is such that $|\tilde{v}_A| > a_A$ and $|\tilde{v}_B| > a_B$ (note that $\tilde{v}_A = \tilde{v}_B$). In this case no signal can propagate upstream in the direction tangential to the shock, being the flow supersonic in that direction at both sides of the shock. As a consequence, no J_i which is downstream

Fig. 2 Possible shock points around: (a) an x -shock; (b) a y -shock

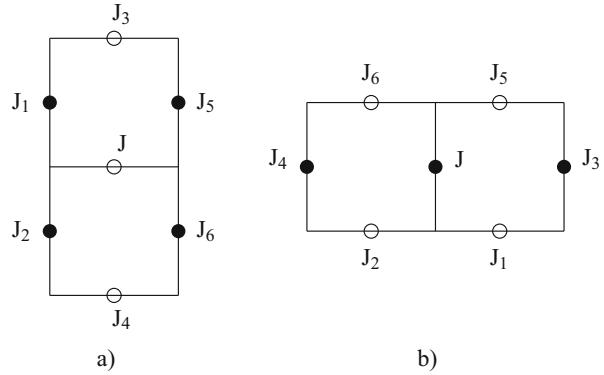
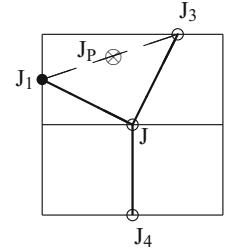


Fig. 3 Triple shock point



of J along the shock can affect the computation of θ_s in J . Therefore, only the shock point located upstream of J is used to compute θ_s , and the direction $J-J_i$ is taken as the shock direction.

2. *sub-shock* when $|\tilde{v}_A| < a_A$ or $|\tilde{v}_B| < a_B$ ($\tilde{v}_A = \tilde{v}_B$). In this case θ_s is computed taking the direction J_i-J_j .

The technique is also able to deal with shock interaction so also triple or multiple shock points may occur. A shock point is defined as a triple shock point if either two of J_1 , J_3 and J_5 or two of J_2 , J_4 and J_6 exist (Fig. 2). Let us consider the example of an x -shock in Fig. 3. If J , J_4 , and J_3 have the high pressure side on the right, and J_1 on the upper side, then J is actually a triple point and requires a specific treatment, whereas J_1 and J_3 are ordinary shock points. θ_s is computed taking the direction J_P-J_4 as the shock direction, where J_P is the middle point of the line J_1-J_3 . During the computation, particularly in the case of transient flows, it may happen that many shock points are close to each other. If there are both two or three compatible shock points above J and two or three compatible shock points below J (Fig. 2a), θ_s is obtained with a procedure similar to that used for triple points, considering a suitable averaging of compatible shock point positions. The floating shock-fitting technique is completed by specific procedures able to move shock points along the grid according to computed shock velocity, detect the formation of new shocks [12, 17] and the possible weakening of shocks up to disappear.

The procedure can also include fitting of contact-discontinuities as can be of interest for specific problems. Procedure is not much different than for shocks. Contact discontinuities (CD) are fitted following a procedure similar to the shock fitting, but only those existing at the beginning of the computation have been considered in this floating shock fitting approach, without performing any detection. A set of indices and variables identical to that for a shock point is introduced for a CD point, and the calculation of the local slope, the displacement, and the introduction of connecting points follow the same procedure as for the shocks. The main difference concerns the enforcement of the relations across the CD. Only two kinds of CD are defined: x -CD, and y -CD. Moreover, $(\cdot)_A$ denotes the left (lower) side and $(\cdot)_B$ the right (upper) side.

The value of variables at the mesh nodes near a CD computed in the first integration step are not correct on either side of the discontinuity. But, in a frame of reference moving with the CD, the variables R_1^x in the mesh node at the left side of an x -CD and R_2^x in the mesh node at its right side (see Eq. 11) are computed correctly. Let us now introduce the components of flow velocity in the direction normal to local CD direction, \tilde{u} , and in the direction parallel to local CD direction, \tilde{v} . The CD relations

$$\tilde{u}_A = \tilde{u}_B = W_c \quad (16)$$

$$p_A = p_B \quad (17)$$

along with the definitions

$$a_A + \delta \tilde{u}_A = \delta(R_1^x)_A \quad (18)$$

$$a_B - \delta \tilde{u}_B = \delta(R_2^x)_B \quad (19)$$

provide the equations for the calculation of the unknowns \tilde{u}_A , \tilde{u}_B , a_A , a_B , and W_c . The mesh nodes on the left and right side of the discontinuity are updated with the values of variables resulting from interpolation between their values in $(\cdot)_A$ and $(\cdot)_B$, respectively, and their values at the next nodes (for instance, interpolation for node N , on the left side of an x -CD lying between N and $N + 1$ is made using the values in node $N - 1$ and the values in $(\cdot)_A$). The values \tilde{v}_A , s_A , and \tilde{v}_B , s_B are independent of each other. They are correctly computed in the first step of the integration, since they are not affected by signals propagating across the CD.

Summarizing, each time iteration of the integration process can be organized in six main steps to be considered in any computations that may include shocks:

1. Integration of the equations over all the mesh nodes, according to the λ -scheme.
2. Detection of the formation of new shocks.
3. Calculation of the local slope of the shocks.
4. Enforcement of the Rankine-Hugoniot jump conditions between the two sides of each shock point.

5. Displacement of the shock points, according to the velocity of propagation of the shock front.
6. Check of crossing of shock and grid lines introducing markers [5, 7, 9] or connecting shock points [17].

Four more steps have to be added to the above six steps of the integration technique if also contact discontinuities are fitted:

7. Calculation of the local slope of the CD.
8. Enforcement of the CD relations across the discontinuity.
9. Displacement of the CD.
10. Introduction of connecting CD points.

4 Multi-Block Approach

The approach described in the foregoing sections is extended to overlapping multi-block mesh structures, that permit a suitable distribution of cells throughout the computational domain while preserving a structured distribution and reasonable cell aspect ratio. The equations are treated at the block boundary as in the inner nodes by introducing ghost nodes and by using bilinear interpolation, whereas a peculiar interpolation procedure is introduced to correctly treat shocks.

4.1 Governing Equations

At boundaries some of the spatial derivatives cannot be computed by differences among values at nodes inside the block, because some signal will propagate from outside the block, except for the case of supersonic outflow. In general, signals propagating from outside can be obtained by enforcing the proper boundary condition, whereas in case of boundary between blocks the missing derivatives must be inferred from the solution in the neighboring block. An approach that preserves the accuracy of the method, while maintaining a general easy-to-implement coding, is that of using two rows of ghost nodes outside the block. In general, considering all of the boundaries of the block, two rings of ghost nodes are defined as shown in Fig. 4, whose position is obtained by smoothly prolonging outside of the block the inner node distribution.

To make it possible that the solution at the ghost nodes is interpolated from that of the neighboring block, such nodes are required to lie inside the neighboring block, or in other words, they have to be such that there are no mesh-size holes between neighboring blocks. A possible configuration is drawn in Fig. 5, where only the ghost nodes of the left block (block 1) are displayed for the sake of clarity. The generic ghost point A lies inside a real cell BCED of the neighboring block (block 2) and the solution of the generic function \mathcal{F}_A at the point A is obtained by the bilinear interpolation

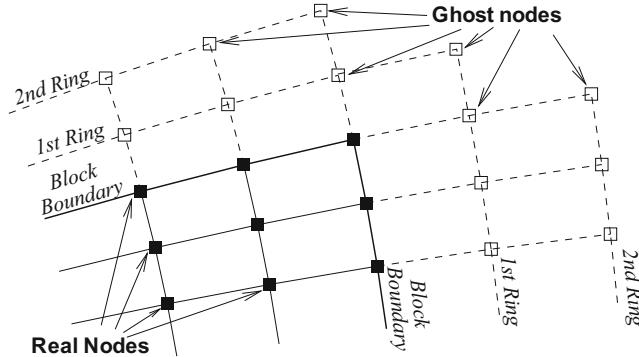
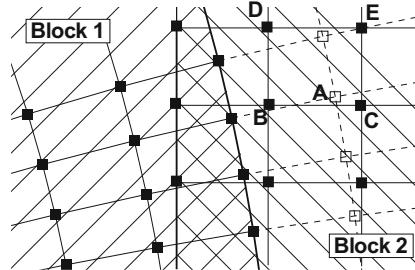


Fig. 4 Ghost points

Fig. 5 Interpolation at ghost points



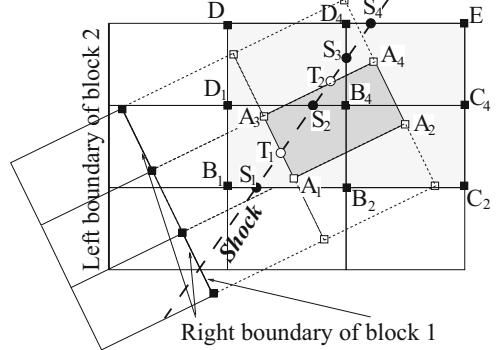
$$\mathcal{F}_A = \mathcal{A} + \mathcal{B}\hat{x}_A + \mathcal{C}\hat{y}_A + \mathcal{D}\hat{x}_A\hat{y}_A \quad (20)$$

where the coordinates of the node A in the frame of reference (\hat{x}, \hat{y}) having origin in B and axes directed as BC and BD are known, whereas \mathcal{A} , \mathcal{B} , \mathcal{C} , and \mathcal{D} are computed by solving at each step and for each variable the 4×4 linear system obtained applying (20) in B, C, D, and E, respectively.¹ Then, at the next iteration, the solution in the boundary nodes can be updated as done in the inner nodes because the missing derivatives are obtained from the solution in the inner and ghost nodes, the latter bringing information from the neighboring blocks.

4.2 Shock-Fitting

The use of ghost nodes allows introducing also ghost shock points needed to follow the general shock fitting procedure to transfer information between blocks. In particular, the introduction of ghost shock points is necessary to compute the local

¹Bilinear interpolation is second order accurate as can be easily shown.

Fig. 6 Ghost shock points

shock slope of real shock points at the block boundary and to avoid interpolating across shocks. The procedure used to find ghost shock points is illustrated by the example shown in Fig. 6, where the problem is to find the ghost shock points in the prolongation of block 1, when the shock position is known by the real shock points in block 2. A possible approach is to consider for each ghost cell $A_1A_2A_3A_4$ a quadrilateral B_1C_2ED , that is the smallest group of cells large enough to include the whole ghost cell. Once the quadrilateral is defined, it is possible to look for shock points along its boundaries, and, in case some are found, connect them by straight lines and find the intersection of these lines with the boundaries of the ghost cell. In the example of Fig. 6 the procedure finds the shock points S_1 and S_4 , and the intersection of S_1S_4 with A_1A_3 and A_3A_4 provides the ghost shock points T_1 and T_2 , respectively.

5 Steady Interactions

The approach described in the foregoing sections is suited to time-accurate computations of unsteady problems. However, it can also be used to solve problems having a steady state solution through a numerical transient in time. A bunch of solutions for steady state problems is therefore reported first to show the capability of the technique to solve problem with shock reflections and interactions.

5.1 Regular Reflection

The first case of shock interaction that is analyzed here is the regular reflections occurring in a planar converging channel. A supersonic flow with a Mach number $M = 3$ enters a 2D plane duct, whose geometry displays two regions with constant area sections joined through a ramp of 10° slope, and having a contraction ratio of

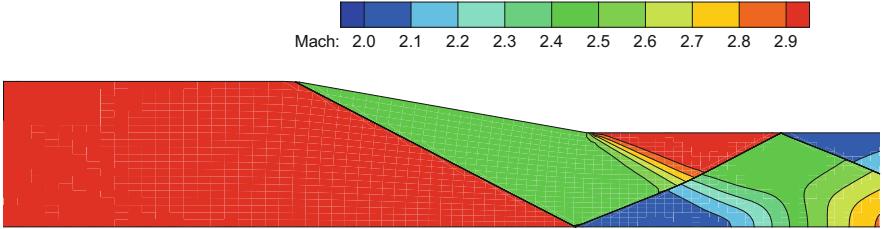


Fig. 7 Steady inviscid regular reflection of an oblique shock computed by floating shock fitting

0.65. Mach number flowfield computed with a 70×20 grid is shown in Fig. 7. The straight shock wave originated at the first concave corner reaches the lower wall. A suitable boundary condition has to be considered for shock points moving along walls, as can occur for inviscid problems, or also on symmetry lines of viscous problems. This boundary condition must allow the shock to be such to keep the flow tangential to the wall. Two possible options exist: the shock is perpendicular to the wall or a regular reflection occurs. Basing on shock relations it can be identified which of the two solutions takes place during the numerical transient, up to the final steady solution. In the test case shown in Fig. 7, regular reflection occurs. Then, the reflected shock propagates back towards the upper wall, is bent while crossing the expansion fan generated at the convex corner, and eventually reaches the upper wall where it experiences a second regular reflection.

Another example of regular reflection computation is that relevant to a confined jet as shown in Fig. 8. A supersonic plane jet at $M = 1.5$ expands from a nozzle in a wider channel. The supersonic jet is uniform and the expansion pressure ratio (jet total pressure over ambient pressure) is $PR = 7.82$, that yields an expansion from $M = 1.5$ to $M = 2$. The shear layer between jet and external ambient is simulated under the inviscid hypothesis by a contact discontinuity line. As shown in Fig. 8, the jet expands by a fan centered at the nozzle lip, and the flow takes the outward direction. As a consequence, the flow impinges on the channel wall at an angle, and at that location an oblique shock takes place. Figure 8a shows the Mach number contour lines of the computed flowfield: the expansion waves are reflected on the symmetry axis as expansion waves, that turn the flow toward the axial direction. More downstream, the reflected expansion waves and the shock interact with each other, and are then both reflected on the upper and lower boundaries respectively, as waves of the same family. The computed flowfield includes also a second shock reflection on the upper wall.

From the point of view of the numerical implementation it is worth to note that because the nozzle lip is a singular point, a specific treatment is needed to handle the discretized form of a finite-differences method. This is achieved here by enforcing the solution of the two-dimensional Riemann problem at that point and at its three closer neighboring points.

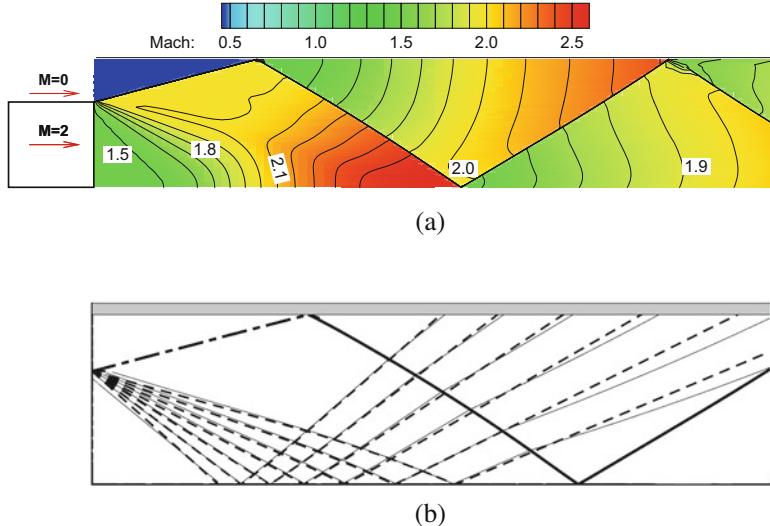


Fig. 8 $M = 1.5$ confined jet test case simulation. (a) Numerical solution of the Mach number flowfield. (b) Numerical (line) and semi-analytical (spaced line) characteristic lines, and numerical slip line (line with dot) and shocks (thick line)

The precision of the solution can be assessed close to the nozzle exit section, where the semi-analytical solution of the expansion fan and of its reflection can be easily achieved through a characteristic based computation. Figure 8b shows the comparison of the characteristic lines obtained by the numerical and semi-analytical solutions.

5.2 Mach Reflection

A steady Mach reflection test case has been obtained by simulating a $M = 5$ flow over a ramp followed by a convex corner. This result is part of a study made to understand the capability of the shock fitting approach to reproduce the hysteresis phenomenon occurring when the free stream Mach number is changed cyclically. In fact, the Mach number for transition from regular to Mach reflection changes depending on the direction of Mach number change: it is higher when Mach number is increasing, lower when transition occurs for decreasing Mach number [22]. More specifically, results reported in [22] are referred to a fluid featuring ratio of specific heats $\gamma = 1.4$, to a ramp angle is $\theta = 26.565$, and to a cyclic variation of free stream Mach number between 2.7 and 12.3. The solution obtained at $M = 5$ shows an example of Mach reflection and the capability to handle this kind of reflection and the triple point by the floating shock fitting approach (Fig. 9).

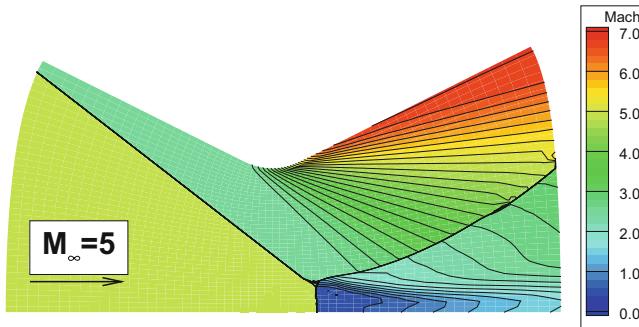


Fig. 9 Steady inviscid Mach reflection of an oblique shock computed by floating shock fitting

5.3 Overexpanded Jet

A peculiar test case is that of the exhaust of an overexpanded supersonic jet in a quiescent ambient, because of the particular interaction between the shock and the jet boundary, computed as a contact discontinuity. The present test case concerns the exhaust of a supersonic jet at $M = 2$ in a quiescent ambient. The stagnation to ambient pressure ratio is $PR = 5$, and therefore the jet to ambient pressure ratio is $PR = 0.634$. Also in this case the two-dimensional Riemann problem is used to solve the singularity at the nozzle lip, from where a shock and the contact discontinuity introduced to simulate the jet boundary, emanate.

The computed flowfield is shown in Fig. 10 by the Mach number and pressure contour lines. The dashed line indicates the position of the jet boundary. It is worth to remark that the solution of both the shock emanating from the nozzle lip and the reflected shock is superimposed to the analytical solution. It is also interesting to notice the accurate solution of the interaction between shock and contact discontinuity and between expansion waves and contact discontinuity, and the resulting sequence of expansions and compressions due to the external constant pressure ambient.

5.4 Interaction of Two Supersonic Jets

The next test aims to show the result of the interaction between a nozzle jet and a supersonic external flow. Two parallel uniform supersonic jets with different Mach numbers and static pressures interacting with each other are considered. The lower (“nozzle”) jet exhausts at $M = 2$ and at a static pressure lower than the external free jet, that features a flow with $M = 4$. The internal to external jet pressure ratio is $PR = 0.383$. The upper boundary features conditions of an unconfined jet, whereas a symmetry line is enforced at the lower boundary. From the nozzle lip a shock,

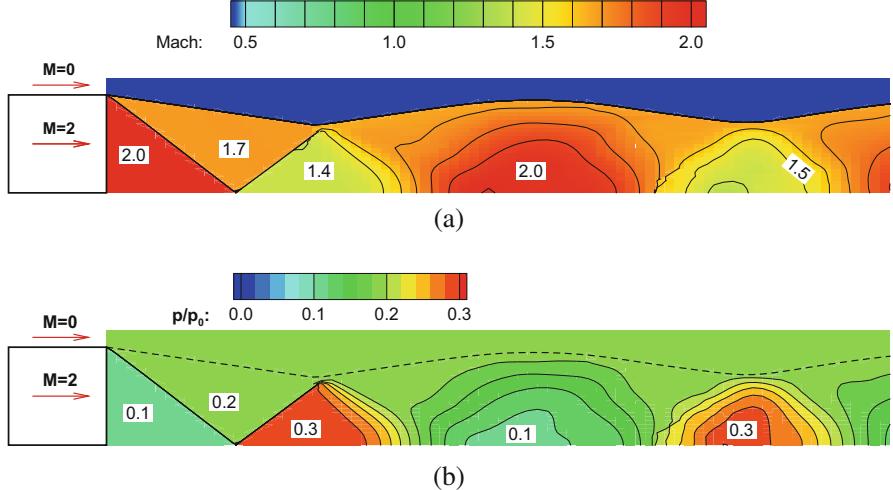


Fig. 10 $M = 2$ overexpanded jet test case. (a) Mach number flowfield. (b) Pressure flowfield and slip line (dashed line)

a contact discontinuity, and an expansion fan emanate, and therefore this singular point is treated according to the analytical approach illustrated for the foregoing test cases.

The solution of this test case is shown in Fig. 11, and should be compared with the previous one (Fig. 10). In fact, when compared to the preceding test case, the interaction of the reflected shock wave with the contact discontinuity line shows a different effect on the evolution of the jet structure. The higher external pressure deviates both the streams towards the lower wall, increasing the lower jet pressure through a shock and decreasing that of the upper flow through a centered expansion. The shock is reflected on the lower wall, and the reflected shock deviates the jet streamlines upwards, in direction parallel to the lower wall. As this reflected shock reaches the jet boundary a two-dimensional Riemann problem takes place. In particular, downstream the interaction point the contact discontinuity, even if it is still directed towards the lower wall, deviates upwards such to yield a weaker shock in the upper flow and a reflected shock in the lower jet.

More downstream, when this new reflected shock—further reflected at the lower wall—reaches again the jet boundary, the phenomenon will repeat itself with lower intensity. The final consequence is that the presence of the external flow changes the characteristics of the internal jet, that does not display the typical sequence of compression and expansion waves shown in the foregoing test case. On the contrary the nozzle jet experiences a sequence of compressions of decreasing intensity, that gradually increase its pressure value, until a uniform pressure is achieved in both flows. The profile of the jet boundary behaves accordingly, with the height decreasing by steps, but practically constant downstream the first two reflected compressions.

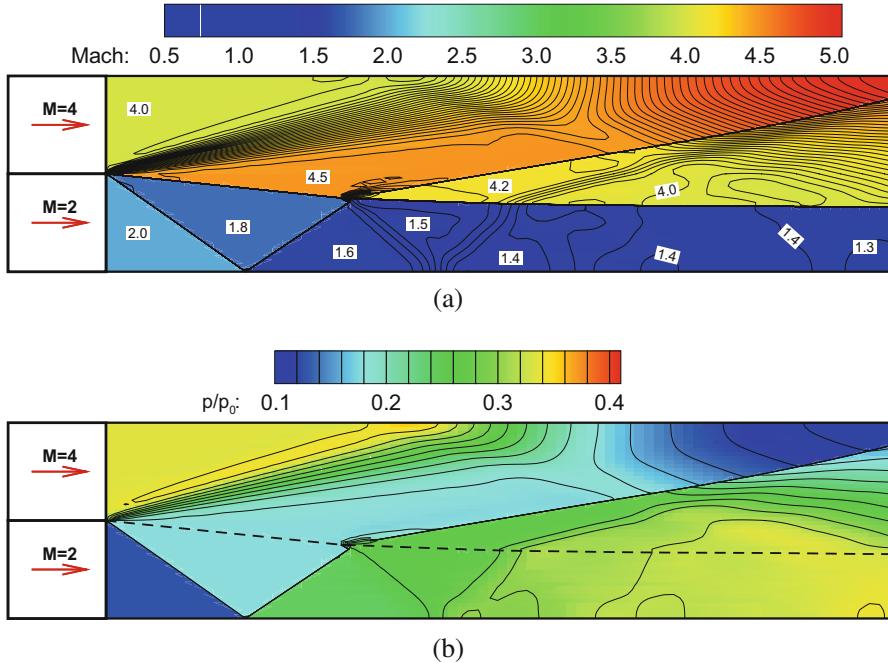


Fig. 11 Interaction of two supersonic jets. **(a)** Mach number flowfield. **(b)** Pressure flowfield and slip line (dashed line)

5.5 Shock Boundary Layer Interaction

If shocks are able to reach walls and reflect in the inviscid flow theory, this is no longer true in case of viscous flows where a boundary layer develops along the walls. Basic studies of shock boundary layer interaction solutions obtained by floating shock fitting were carried out in [13]. As an example, the supersonic flow over a flat plate followed by a wedge is considered (Fig. 12a). The Mach number of the supersonic stream is $M_\infty = 3$ and the wedge has a 20° slope. Figure 12a shows the generation of three shocks. The fitting technique allows to solve all of them as shown by the shock points in the figure. Note that the first shock, originating from the flat plate leading edge, starts in a singular point of the computation and its position is known a priori providing a boundary condition for the shock. Inviscid flow solution would show a single shock generated by the concave corner. In the viscous case, besides the flat plate leading edge shock, the concave corner should generate a shock similarly to that of the inviscid case. However, this shock generates an adverse pressure gradient in the boundary layer that consequently separates from the wall. The result is that the boundary layer separation generates a shock before the wedge. Flow reattachment follows with a further change of flow direction and a third shock. Note that, theoretically the two shocks originated at the wedge are slightly

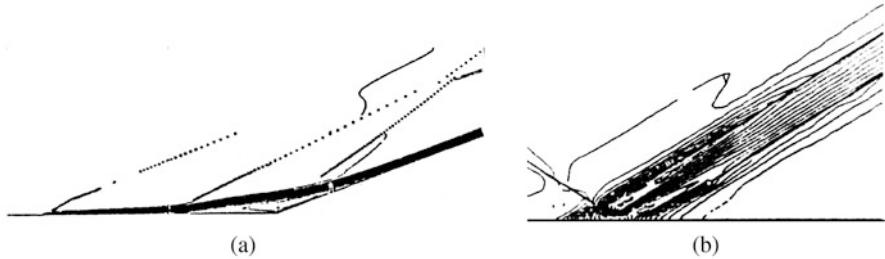


Fig. 12 Shock boundary layer interactions in supersonic flows (adapted from [13]). **(a)** Flat plate followed by a wedge. **(b)** Shock impinging on a flat plate

converging such to lead, far enough from the wall to the single shock solution foreseen in the inviscid case. This is correctly obtained and shown in Fig. 12a with the present shock fitting approach.

Another example is the reflection of an oblique shock from a rigid wall (Fig. 12b). The free-stream Mach number is $M_\infty = 2$ and the impinging shock is a shock produced by a 3° deflection. The same mechanism generating two shocks instead of a single one at the wedge occurs in this case. The adverse pressure gradient caused by the impinging shock thickens the boundary layer before the impinging point and yielding a reflected shock anticipating shock impingement. Then, downstream the flow direction is redirected towards the wall by a centered expansion and eventually realigns with the horizontal direction through a second shock wave. Again, the two reflected shocks will eventually merge far from the wall to yield the inviscid solution. Figure 12b shows the fitting of the impinging and of the two reflected shocks that can be easily identified as discontinuities in the flowfield rather than as thick change of property regions that would be shown by shock capturing approaches (unless order of magnitude higher number of cells is considered). These two results show the capabilities of shock fitting to manage the challenging transition from the shock discontinuity, typical of the inviscid region, to the smooth pressure variation in the viscous region where the shock discontinuity vanishes.

5.6 Examples of Application

Steady state solutions of flows with shocks and contact discontinuities have been obtained for different applications. A few examples are reported here. The first one is the case of an unstarted supersonic air intake. The case shown in Fig. 13 is relevant to a mixed external-internal compression air intake operating at $M_\infty = 2.25$ and whose behavior is discussed in detail in [27]. Being unstarted, some of the incoming mass flow has to be spilled out before entering the air intake duct. This can only occur because of the presence of a strong shock ahead of the intake. A complex shock structure results, as shown in Fig. 13, with the shock fitting technique able to

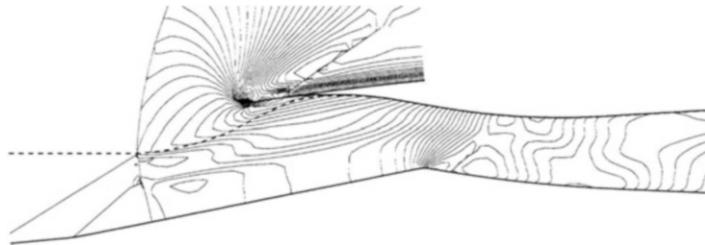


Fig. 13 Shock interaction ahead of an unstarted supersonic air intake

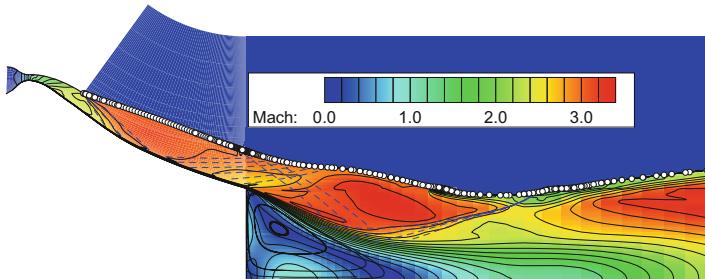


Fig. 14 Fitted contact discontinuity in the simulation of overexpanded jet in a plug nozzle

correctly compute both oblique shocks starting at concave corners of the external compression ramp, the strong shock, and the interactions between each oblique shock and the strong shock.

Another application is that of the exhaust jet from a plug nozzle operating in quiescent air. In this case it is advantageous, even in a RANSE simulation, to consider the jet boundary (where mixing with the quiescent air occurs) as a slip line that is fitted as a contact discontinuity. An example of the solution obtained for overexpanded operation is reported in Fig. 14. More examples and details are presented in [20].

6 Unsteady Shock Interactions

One of the most important properties of shock fitting is its capability to deal with transient flows without requiring local grid refinements moving in time. Examples are reported in the following.

6.1 Unsteady Mach Reflection

An example of computation of unsteady inviscid flows, with shock interaction is that of the unsteady Mach reflection that occurs when a planar moving shock reaches a concave corner. Figure 15 shows the results obtained considering a plane shock ($M_s = 6.69$) followed by an inviscid supersonic flow ($M = 1.75$) which moves from the left in quiescent air towards a concave corner of 10° . As the shock enters the ramp a simple Mach reflection occurs. The solution is pseudo-stationary, i.e. similar to itself in time, with the triple point moving along a straight line. The shock evolution in time is shown in Fig. 15. The straight line overlapping the computed triple point trajectory is drawn according to the experimental measure fit reported in [1]. Details of the position of computed shock points at a given time are shown in Fig. 16.

Fig. 15 Unsteady inviscid Mach reflection: comparison of experimental data fit line for triple point evolution and computed shock at different times (shock lines are obtained connecting shock points)

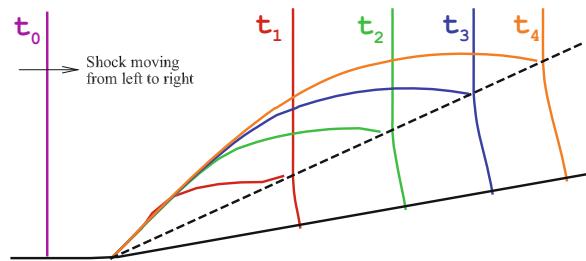
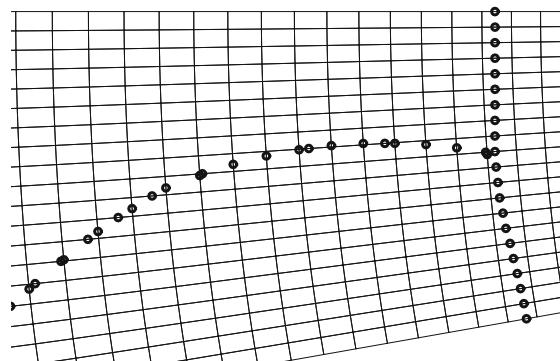


Fig. 16 Unsteady inviscid Mach reflection: computed shock points and the underlying grid lines



6.2 Inviscid Nozzle Flow Transient

The impulsive startup of an overexpanded nozzle has been computed assuming inviscid flow [16]. At the diaphragm rupture a shock wave followed by a contact discontinuity moves inside the converging diverging nozzle. At the first time shown in Fig. 17a three different discontinuities can be identified. From right to left the first one is the front shock which has already propagated from the inlet to the diverging section. It appears as a discontinuity extended from the axis to the wall. Because of the converging-diverging channel it has changed its shape, and its local and average intensity, with respect to the planar uniform shape it had at starting time (when it was placed at the left inlet). The second discontinuity also appears in the divergent section, and it is a shock of the other family. It is only extended a few cells from the upper wall and occurs because of the overexpansion generated by the convex wall. Finally, the contact discontinuity generated at the diaphragm rupture is still in the converging section, and is characterized by a lower propagation speed than the front shock. Note CD bending due to the different flow velocity if one moves from the wall to the axis. The next three time instants (Fig. 17b-d) show the slow propagation and the effect of the overexpansion shock, which now covers the whole cross section. More specifically, in Fig. 17b, the contact discontinuity has reached the divergent section and has passed the overexpansion shock, continuing to follow the front shock which is approaching the divergent section exit. The next two times show that vorticity is being generated behind the quasi-steady overexpansion shock. This vorticity is affecting the contact discontinuity shape that is no longer smooth, and its central part is being significantly slowed down by the peculiar flow generated downstream the overexpansion shock. This solution shows the reliability of fitting both shock and contact discontinuities in impulsive startup problems, provided

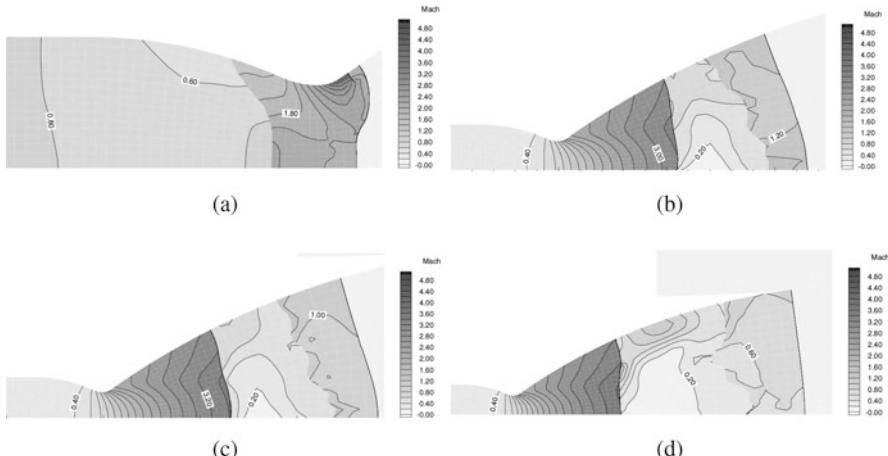


Fig. 17 Impulsive startup of an inviscid overexpanded converging-diverging nozzle

that this assumption is justified by the problem physics. In fact, when a vortical region affects the contact discontinuity, mixing becomes no longer negligible and, accordingly, the assumption of modeling the original contact discontinuity with a discontinuous front is no longer valid.

6.3 Viscous Nozzle Flow Transient

The viscous nozzle startup of Vulcain engine was one of the most successful applications of the technique from an application point of view, leading to the identification of the phenomenon referred to as “inviscid flow separation”[18] with a modest computational cost as compared to the other attempts made in the 1990s to compute the same phenomenon with software based on shock-capturing schemes. The results shown in the following are relevant to a computational campaign made after the study published in [19]. More specifically, a series of steady-state simulations was carried out because of the slow variation of combustion chamber condition in time, that allowed to neglect time-dependent phenomena. Solutions are relevant to non-reacting RANSE, for prescribed chamber (total pressure and temperature) and ambient (ambient pressure) conditions. Each simulation is identified by the corresponding chamber to ambient pressure ratio (PR). This quasi-steady simulation is started from the less overexpanded conditions (referred to as in the figure as $PR = 130$, note that the adaptation condition is about $PR = 500$) and then the solution for $PR = 100$ is obtained starting from that at $PR = 130$ and changing the chamber conditions. The same procedure is repeated for the other chamber conditions, that is for $PR = 40$ the initial condition is the steady state solution for $PR = 100$, for $PR = 30$ the initial condition is the steady state solution for $PR = 40$ and so on. After reaching the minimum $PR = 10$ the pressure ratio is increased again to show the possible occurrence of hysteresis (i.e., in this case solutions which depend on the initial conditions). Results show that starting from an operating condition with a conventional Mach reflection generated by supersonic nozzle overexpansion, increasing the degree of overexpansion moves towards a peculiar shock structure that the floating shock-fitting approach is able to correctly capture despite the different shock interactions and vortical structures that take place and while keeping always the same grid. The most peculiar shape is that leading to the flow structure known as “restricted-shock-separation” (Fig. 18c–e). In this case an upstream bent shock generates a big vortex which confines the jet in a narrow region close to the wall where a shock boundary layer interaction similar to that shown in Fig. 12b occurs [21]. Note that different solutions can occur at the same PR (see $PR = 20$) depending on the initial conditions, or in other words on the direction of PR change. This hysteresis phenomenon is confirmed experimentally.

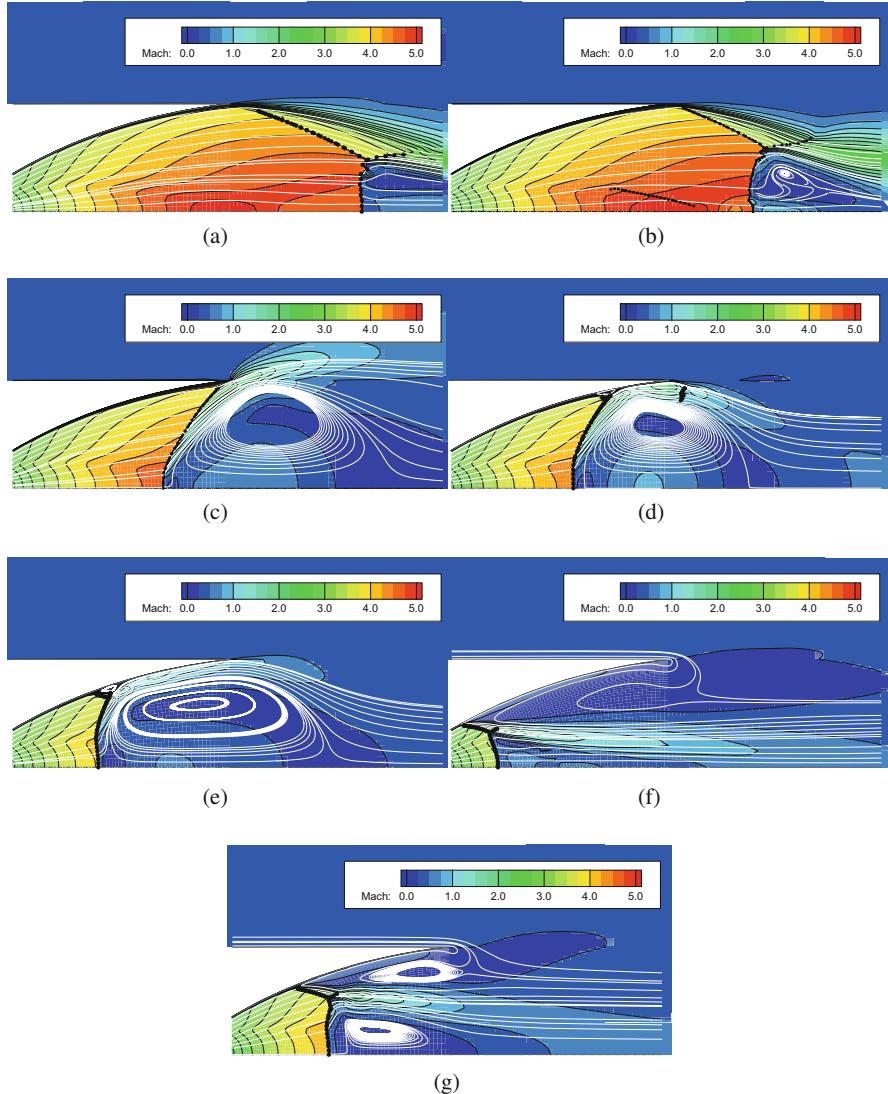


Fig. 18 Mach number flowfield, streamlines, and shock points (*filled circle*) for the steady-state solutions at varying PR: each solution is taken as initial condition for the next. (a) PR=130. (b) PR=100. (c) PR= 40. (d) PR= 30. (e) PR= 20. (f) PR= 10. (g) PR= 20

7 Conclusions

A floating shock fitting technique following the approach introduced by Moretti has been used to compute different flowfield including fitting of shocks and contact discontinuities as well as shock interactions. Fitting of discontinuities in

two-dimensional flows showed to be successful for a high number of problems, especially because of its efficiency in terms of computational cost for the study of flow transients. Some of the results obtained by the authors in a couple of decades have demonstrated the versatility and reliability of the technique.

Acknowledgements This paper has been prepared in the memory of Gino Moretti, who has been the mentor of authors in studying, developing, and using CFD techniques. The authors are indebted to him for the unique lessons he left to them.

References

1. Ben-Dor, G.: Shock Wave Reflection Phenomena. Springer, New York (1992)
2. Marconi, F., Salas, M.: Computation of three dimensional flows about aircraft configurations. *Comput. Fluids* **1**, 185–195 (1973)
3. Moretti, G.: Thoughts and afterthoughts about shock computations. PIBAL Report 72–37 (1972)
4. Moretti, G.: The λ -scheme. *Comput. Fluids* **7**, 191–205 (1979)
5. Moretti, G.: Efficient calculation of two-dimensional, unsteady, compressible flows. In: Advanced in Computer Methods for Partial Differential Equations – VI. IMACS (1987)
6. Moretti, G.: A technique for integrating two-dimensional euler equations. *Comput. Fluids* **15**(1), 59–75 (1987)
7. Moretti, G.: Efficient Euler solver with many applications. *AIAA J.* **26**(6), 655–660 (1988). <https://doi.org/10.2514/3.9950>
8. Moretti, G.: Thirty-six years of shock fitting. *Comput. Fluids* **31**(4–7), 719–723 (2002)
9. Moretti, G.: Shock-fitting analysis. In: Onofri, M., Paciorri, R. (eds.) Shock fitting: classical techniques, recent developments, and memoirs of Gino Moretti, pp. 3–31. Springer, New York (2017). https://doi.org/10.1007/978-3-319-68427-7_11
10. Moretti, G., Abbott, M.: A time-dependent computational method for blunt body flows. *AIAA J.* **4**(12), 2136–2141 (1966)
11. Moretti, G., Di Piano, M.T.: An improved lambda-scheme for one-dimensional flows. Nasa-cr-3712 (1983)
12. Moretti, G., Valorani, M.: Detection and fitting of two-dimensional shocks. *Notes Numer. Fluid Mech.* **20**, 239–246 (1988)
13. Moretti, G., Marconi, F., Onofri, M.: Shock boundary layer interactions computed by a shock fitting technique. In: Lecture Notes in Physics, vol. 414, pp. 345–350. Springer, Berlin (1993)
14. Najafi, M., Hejranfar, K., Esfahanian, V.: Application of a shock-fitted spectral collocation method for computing transient high-speed inviscid flows over a blunt nose. *J. Comput. Phys.* **257**, 954–980 (2014). <https://doi.org/10.1016/j.jcp.2013.09.037>
15. Nasuti, F.: A multi-block shock-fitting technique to solve steady and unsteady compressible flows. In: Armfield, S., Morgan, P., Srinivas, K. (eds.) Computational Fluid Dynamics 2002, pp. 217–222. Springer, Berlin (2003)
16. Nasuti, F., Onofri, M.: Transient flow analysis of nozzle start-up by a shock-fitting technique. In: Unsteady Flows in Aeropropulsion, vol. 40, pp. 127–135. ASME, New York (1994)
17. Nasuti, F., Onofri, M.: Analysis of unsteady supersonic viscous flows by a shock-fitting technique. *AIAA J.* **34**, 1428–1434 (1996)
18. Nasuti, F., Onofri, M.: Viscous and inviscid vortex generation during nozzle flow transients. In: 34th AIAA Aerospace Sciences Meeting, Reno, NV. AIAA Paper 96-0076. AIAA, Washington, DC, USA (1996).
19. Nasuti, F., Onofri, M.: Viscous and inviscid vortex generation during startup of rocket nozzles. *AIAA J.* **36**, 809–815 (1998)

20. Nasuti, F., Onofri, M.: Analysis of in-flight behavior of truncated plug nozzles. *J. Propuls. Power* **17**(4), 809–817 (2001)
21. Nasuti, F., Onofri, M.: Shock structure in separated nozzle flows. *Shock Waves* **19**, 229–237 (2009). <https://doi.org/10.1007/s00193-008-0173-7>
22. Onofri, M., Nasuti, F.: Theoretical considerations on shock reflections and their implications on the evaluations of air intake performance. *Shock Waves* **11**(2), 151–156 (2001). <https://doi.org/10.1007/PL00004067>
23. Paciorri, R., Sabetta, F.: Compressibility correction for the Spalart-Allmaras model in free-shear flows. *J. Spacecr. Rocket.* **40**(3), 326–331 (2003)
24. Paciorri, R., Bonfiglioli, A.: A shock-fitting technique for 2D unstructured grids. *Comput. Fluids* **38**(3), 715–726 (2009). <https://doi.org/10.1016/j.compfluid.2008.07.007>.
25. Rawat, P., Zhong, X.: Numerical simulations of strong shock and disturbance interactions using high-order shock-fitting algorithms (2008). In: 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada. <https://doi.org/10.2514/6.2008-746>
26. Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale* **1**, 5–23 (1994)
27. Valorani, M., Nasuti, F., Onofri, M., Buongiorno, C.: Optimal supersonic intake design for air collection engines (ace). *Acta Astronaut.* **45**(12), 729–745 (1999). [https://doi.org/10.1016/S0094-5765\(99\)00185-X](https://doi.org/10.1016/S0094-5765(99)00185-X)

Part II

**Recent Developments in Shock-Fitting
Techniques for Unstructured Grids**

Basic Elements of Unstructured Shock-Fitting: Results Achieved and Future Developments

Renato Paciorri and Aldo Bonfiglioli

1 Introduction

At the CFD dawn, two alternative approaches were introduced to simulate flows with shocks: shock-fitting and shock-capturing. The former was able to provide very accurate solutions and to be very efficient in terms of computational cost. Unfortunately it also had clear limitations in simulating three-dimensional flow-fields and complex shock patterns.

Shock-capturing discretizations lay their foundations in the mathematical theory of weak solutions, which allows the computation of all flow types, including those with shocks, using the same discretization of the conservation-law-form of the governing equations at all grid cells. This yields obvious consequences in terms of coding simplicity, since a single numerical scheme is used and the same set of operations is repeated within all control volumes of the mesh, no matter how complicated the flow might be.

As a consequence, nowadays shock-capturing schemes are the most widespread choice for practical fluid-dynamics simulations involving shock waves.

Coding simplicity does not come for free, however, and shock-capturing solutions of flows featuring strong shock waves are often characterized by the appearance of bizarre anomalies and sometimes by large numerical errors. The deficiencies exhibited by state-of-the-art shock capturing discretizations are in some cases

R. Paciorri (✉)

Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome,
Via Eudossiana 18, 00184 Rome, Italy
e-mail: renato.paciorri@uniroma1.it

A. Bonfiglioli

Scuola di Ingegneria, Università degli Studi della Basilicata, Viale dell'Ateneo Lucano 10,
85100 Potenza, Italy
e-mail: aldo.bonfiglioli@unibas.it

sufficient to lead the author of a recent review paper [49] on numerical methods for high-speed flows to conclude that “these limitations, related to the misrepresentation of discontinuities on a mesh with finite spacing, can only be overcome by some form of shock-fitting.”

Shock-fitting consists first in locating and then tracking the discontinuities in the flow-field. These are treated as boundaries between regions where a smooth solution to the governing partial differential equations (PDEs) exists. The flow variables on the two sides of the discontinuities are evaluated analytically by enforcing the Rankine-Hugoniot (R-H) jump relations, which are simple algebraic equations connecting the states on both sides of the discontinuity and its local speed. Then, this solution is used to compute the space-time evolution of the discontinuity, that is, to track its motion.

Shock-fitting methods enjoyed a remarkable popularity in the early CFD era, thanks to the efforts of Gino Moretti and his collaborators, but Moretti’s studies were only carried out on structured grids. In fact he has been active over a time frame, ranging from the mid 1960s to the late 1980s, when almost exclusively structured-grid discretizations were used. Within the structured-grid framework, shock-fitting methods took two different approaches: boundary shock-fitting and floating shock-fitting.

In the boundary shock-fitting approach the shock is made to coincide with one of the boundaries of the computational domain so that the treatment of the R-H relations is confined to the boundary points. This greatly simplifies the coding, but makes the treatment of embedded and/or interacting shocks a “hard bone to chew” [33]. The reason is primarily topological. Indeed, within the boundary shock-fitting framework the embedded shocks are interior boundaries that separate different blocks of a multi-block grid setting. Since shocks move and eventually interact, handling the motion and deformation of the various blocks soon becomes a “topological nightmare” [35].

In the floating version of the shock-fitting technique, discontinuities are allowed to move (float) freely over a fixed background structured grid and the shock front is described by its intersections with grid lines. The main features of the methodology are the same as those of the boundary fitting technique, except for the need of a special treatment for grids nodes neighboring shocks. This is because approximating derivatives by differences between nodes located on opposite sides of a shock must be avoided and therefore, ad hoc finite difference formulae have to be used in this case.

When Moretti retired, by the end of the 1980s, unstructured-grid schemes started to be applied to CFD problems, unveiling new unpredictable opportunities for building more robust and versatile shock-fitting schemes. The continuous advances in computer power and computational science over the last decades have made the use of unstructured grids computationally affordable and algorithmically simpler. Taking advantage of this opportunity, a new shock-fitting technique for unstructured meshes has been developed in Italy by second generation of Moretti’s disciples [9, 10, 41, 42].

The unstructured shock-fitting technique to be described in this article made its first appearance on a journal in 2009 [41]; at that time the algorithm was capable of simulating steady, two-dimensional flows featuring an isolated shock-wave. Multiple shocks could also be handled using a hybrid approach, whereby only one shock was fitted and all other discontinuities and their interactions were captured. Later developments [21, 42] made the algorithm capable of fitting contact discontinuities (beside shocks), shock–shock and shock–wall interactions. An order-of-accuracy analysis of the steady, 2D algorithm has been conducted in [7]: it has shown that shock-fitting allows to preserve the design order of the spatial discretization scheme within the entire shock-downstream regions, whereas shock-capturing falls below first order. The present unstructured, shock-fitting algorithm has also been used to simulate real-gas effects in hypersonic, two-dimensional steady flows, both by the authors [46, 48] and at the VKI, see the companion paper [25]. Time accurate simulations of two-dimensional, un-steady flows have only recently been reported [10], see also the contribution [11] within this volume. The three-dimensional version of the algorithm has appeared in [9]: at present it is capable of dealing with steady flows and multiple shocks; shock–shock interactions are however captured, rather than fitted, for reasons that will be explained in Sect. 5.2.

The article is organized as follows. The unstructured shock-fitting algorithm is described in Sect. 2 and details concerning its coding are given in Sect. 3. A selected set of representative numerical results are given in Sect. 4, whereas Sect. 5 addresses some currently unsolved issues and presents ongoing work aimed at overcoming these limitations.

2 Unstructured Shock-Fitting: Algorithmic Features

The approach is inherently time-dependent: both the solution and the grid change with time, due to the displacement of the fitted discontinuities. When a steady solution exists, the shock speed will asymptotically vanish and the tessellation of the flow domain will not any longer change.

At time t the set of dependent variables and grid velocity are available within all grid-points of a tessellation (made of triangles in 2D and tetrahedra in 3D) that covers the entire computational domain; this is what we call the “background” mesh. In addition to the background mesh, the fitted discontinuities (either shocks or contact discontinuities) are discretized using a collection of points which are mutually joined to form a connected series of line segments, as shown in Fig. 1a for the 2D case, or a triangulated surface in 3D, as shown in Fig. 1b. For example, a thick solid (yellow) line is used in Fig. 1a to mark the various fitted discontinuities that arise due to the interaction between two shocks of the same family: the two incident shocks, the resulting shock, a weak compression (or expansion) wave, and the contact-discontinuity located between the former two. Figure 1b, which refers to the three-dimensional, supersonic flow past a blunt-nosed object, shows

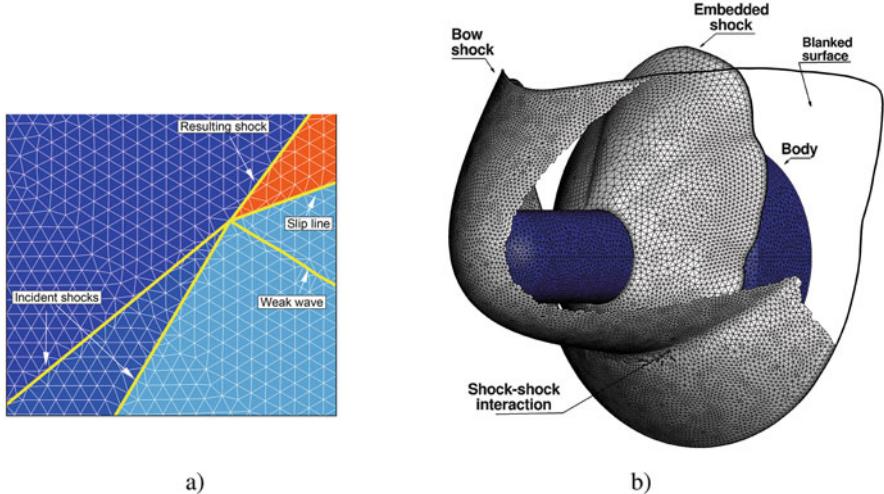


Fig. 1 Examples of fitted discontinuities on unstructured meshes. **(a)** Interaction of shocks belonging to the same family. **(b)** Supersonic flow over a blunt-nosed body

the triangulated surfaces used to fit the bow shock and the embedded shock that arises at the cylinder-flare junction. Although it may not be evident from Fig. 1, each fitted discontinuity is a double-sided internal *boundary* of zero thickness; this is sketched in Fig. 2d, which refers to the case of a shock wave. Because the width of the discontinuity is negligible (it has been amplified in Fig. 2d to improve visibility), its two sides are discretized using the same polygonal curve or triangulated surface; each pair of nodes that face each other on the two sides of the discontinuity share the same geometrical location, but store different values of the dependent variables, one corresponding to the upstream state and the other to the downstream one. Moreover, a velocity vector normal to the discontinuity is assigned to each pair of grid-points on the fitted discontinuity, representing its displacement velocity. As shown in Fig. 2a, the spatial location of the fitted discontinuities is independent of the location of the grid-points that make up the background grid.

The process that leads from the available mesh and solution at time t to an updated mesh and solution at time $t + \Delta t$ can be split into seven steps that will be described in the following paragraphs.

2.1 Cell Removal Around the Shock Front

In this first step, the fitted discontinuities are laid on top of the background mesh, as shown in Fig. 2a. All those cells that are crossed by the fitted discontinuities and those mesh points that are located too close to it are temporarily removed from the background mesh, as shown in Fig. 2b. We call “phantom” those grid-points of the

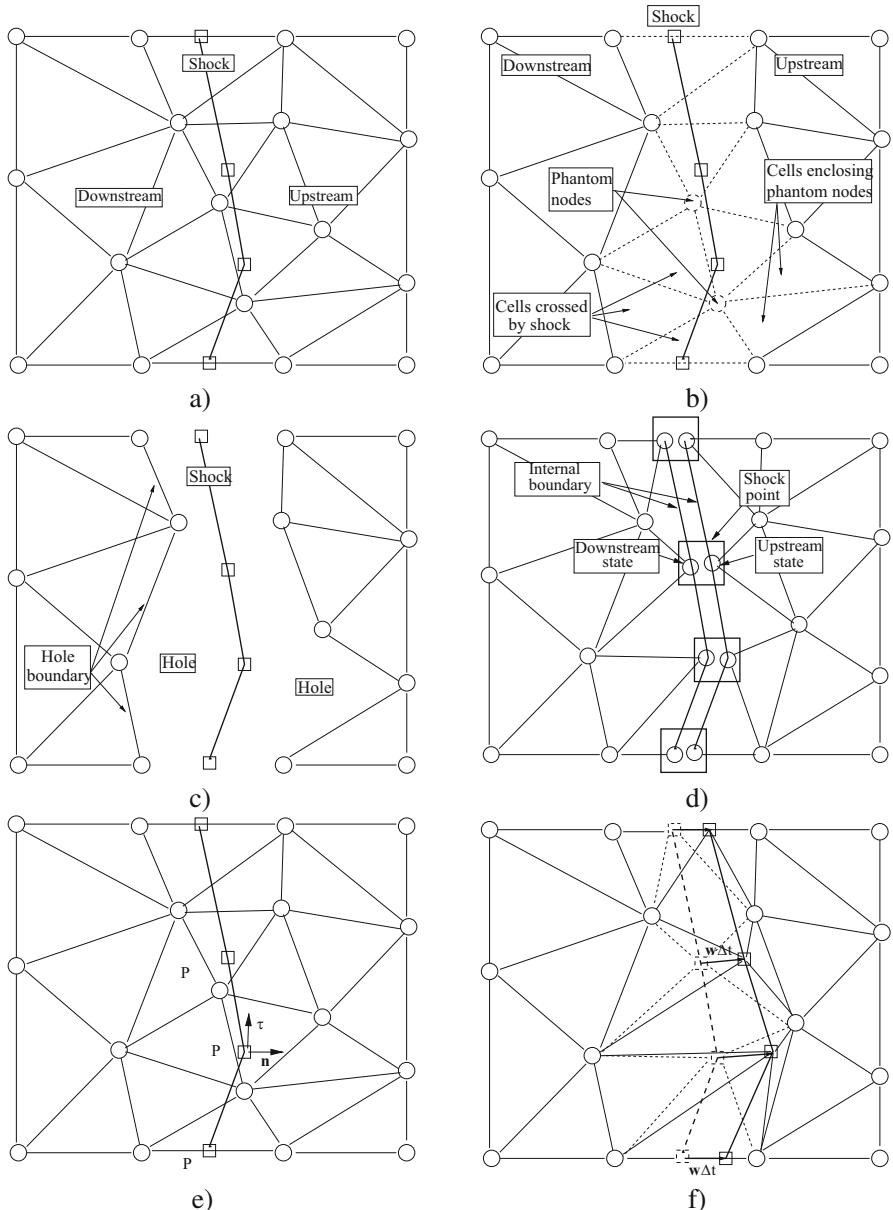


Fig. 2 Unstructured shock-fitting. **(a)** Shock front moving over the background triangular mesh at time t . **(b)** Dashed lines mark the cells to be removed; dashed circles denote the phantom nodes. **(c)** The background mesh is split into disjoint sub-domains by a hole which encloses the shock. **(d)** The triangulation around the shock has been rebuilt. **(e)** Calculation of the shock-tangent and shock-normal unit vectors. **(f)** The shock displacement induces mesh deformation

background mesh (shown using dashed circles in Fig. 2b) that have been temporarily removed. All cells having at least one phantom node among their vertices are also removed from the background triangulation; these are the cells shown using dashed edges in Fig. 2b. Further details concerning the criteria used to identify and remove the phantom nodes can be found in [9, 41].

2.2 Local Re-Meshing Around the Shock Front

Following the cell removal step, the background triangulation has been split into two or more disjoint sub-domains, as shown in Fig. 2c. The hole dug by the fitted front is then re-meshed using a constrained Delaunay tessellation (CDT): the edges (triangles in 3D) that make up the fitted discontinuity and the boundary of the hole are both constrained to be part of the final tessellation; this is illustrated in Fig. 2d. Observe that re-meshing is localized around the discontinuities and, therefore, does not overload the algorithm in terms of CPU cost.

Upon completion of this stage, the computational domain is discretized using what we call the “shock-fitting” mesh, which differs from the background mesh only in the neighborhood of the fitted discontinuities.

Further details concerning the software used to construct the CDT will be given in Sect. 3.

2.3 Calculation of the Unit Vectors Normal to the Shock Front

In order to apply the jump relations, normal (\mathbf{n}) and tangent ($\boldsymbol{\tau}$) unit vectors are needed within each pair of grid-points located along the discontinuities, see Fig. 2e. These unit vectors are computed using finite-difference (FD) formulae which involve the coordinates of the shock-point itself and those of its neighboring shock-points. Depending on the local flow regime, it may be necessary to use upwind-biased formulae to avoid geometrical instabilities along the fitted discontinuity. Full details describing how to compute the normals to the discontinuity can be found in [41, 52] for the 2D case and in [9] for the 3D case.

2.4 Solution Update Using the Shock-Capturing Code

Using the shock-fitting mesh as input, a single time step calculation is performed using an unstructured, vertex-centered shock-capturing solver which returns updated nodal values at time $t + \Delta t$. Since the discontinuities are seen by the shock-capturing code as internal boundaries (of zero thickness) moving with the velocity of the discontinuity, there is no need to modify the spatial discretization scheme

already implemented in the PDEs solver to account for the presence of the fitted discontinuities. In practice, the shock-capturing solver is used as a black-box: it receives as input the shock-fitting grid, the nodal values of the solution and grid velocity at time t and returns the updated solution at time $t + \Delta t$.

The solution returned by the shock-capturing solver at time $t + \Delta t$ is however missing some boundary conditions on one or both sides of each discontinuity, depending on whether it is a shock or a contact. These missing pieces of information will be determined in the next step.

2.5 Enforcement of the Jump Relations

The missing pieces of information that are needed to correctly update the solution within all pairs of grid-points located on the discontinuities are obtained by enforcing the R-H jump relations; this also provides the local velocity of the discontinuity along its normal.

The R-H jump relations are a set of non-linear algebraic equations that can be solved within each pair of grid-points located along the discontinuities by means of Newton-Raphson's algorithm. In order to match the number of unknowns with the available equations, one or more additional pieces of information are required within both or either of the two sides of the fitted discontinuity, depending on whether this is a shock or a contact discontinuity. These additional pieces of information are obtained from the characteristic formulation of the Euler equations and correspond to those characteristic quantities that are convected towards the discontinuity from the sub-domain that is attached to that side of the discontinuity. Using an upwind-biased discretization within the shock-capturing solver, one can reasonably assume that the spatial and temporal evolution of these characteristic quantities has been correctly computed. Full algorithmic details concerning the practical implementation of the jump relations for shocks and contact discontinuities are reported elsewhere and will not be repeated here: see [21, 41, 42] for the 2D case and [9] for the 3D case.

An ad-hoc treatment is moreover required within those special points where different discontinuities interact; this is the case of triple and quadruple points where an impinging shock is reflected from a solid surface, etc. The algorithmic details are described in [21, 42] for the 2D case, whereas the interaction among fitted discontinuities has not yet been dealt with in 3D. This specific issue will be further addressed in Sect. 5.2.

2.6 Shock Displacement

The enforcement of the jump relations provides the speed (w) at which each pair of grid-points located on the discontinuity move along its local normal vector, \mathbf{n} . The position of the discontinuity at time $t + \Delta t$ is computed in a Lagrangian manner by

displacing all its grid-points, as shown in Fig. 2f where the dashed and solid lines represent the discontinuity at time t , resp. $t + \Delta t$. When simulating steady flows, this can be accomplished using the following first-order-accurate (in time) integration formula:

$$\mathbf{P}_i^{t+\Delta t} = \mathbf{P}_i^t + w_i^t \mathbf{n}_i^t \Delta t \quad (1)$$

which returns the spatial coordinates of the i -th grid-point at time $t + \Delta t$. The low temporal accuracy of Eq. (1) does not affect the spatial accuracy of the steady state solution which only depends on the spatial accuracy of the gas-dynamics solver and that of the tangent and normal unit vectors. On the contrary, when dealing with unsteady flows, the temporal accuracy of the shock motion has to be the same as that of the shock-capturing solver, i.e. second-order-accurate in our case. This can be accomplished using a predictor-corrector type temporal integration scheme, as described in [10].

Figure 2f shows that even when the background mesh is fixed in space, the triangular cells that abut on the discontinuity have one of their edges that moves with the discontinuity, thus deforming the cell. This implies that the shock capturing solver used in Step 2.4 must be capable of handling moving meshes, i.e. it must be capable of solving the governing PDEs written using an Arbitrary Eulerian Lagrangian (ALE) formulation.

Finally, the time step Δt to be used in Eq. (1) to move the shock is chosen in such a way that during the time interval $[t, t + \Delta t]$ the shock will remain within the hole that it has dug in the background mesh. By doing so, none of the grid-points of the shock-fitting mesh will be overcome by the moving discontinuity, as shown in Fig. 2f.

2.7 Interpolation of the Phantom Nodes

Upon completion of the previous steps, all nodes of the shock-fitting mesh have been updated at time $t + \Delta t$. The shock-fitting mesh is made up of all the grid-points belonging to the fitted discontinuities and all nodes of the background mesh, except those that have been declared “phantom”. Therefore, the nodal values within the phantom nodes have not been updated to time $t + \Delta t$. However, during the current time step, the discontinuity might have moved sufficiently far away from its previous position, that some of the phantom nodes may re-appear in the shock-fitting mesh at the next time step. It follows that also the nodal values within the phantom nodes need to be updated to time $t + \Delta t$. This is easily accomplished by transferring the available solution at time $t + \Delta t$ from the current shock-fitting mesh to the grid-points of the background one, using linear interpolation. Once the phantom nodes have been updated, the shock-fitting mesh used in the current time interval has completed its task and can be removed.

At this stage the numerical solution has correctly been updated at time $t + \Delta t$ within all grid-points of the background tessellation and within all pairs of grid-points belonging to the fitted discontinuities. The next time interval can be computed re-starting from the first step 2.1 of the algorithm.

3 Intrinsic Modularity and Modular Implementation

The algorithm that we have just described is intrinsically modular and we have taken advantage of this feature by splitting the code into different, independent modules.

Three key software components (or modules) can be identified in the unstructured shock-fitting procedure described in Sect. 2: (1) the *shock-fitting module*, (2) the *gas-dynamic solver*, and (3) the *meshing software*.

The role played by these three different pieces of software can be better highlighted if we summarize the seven steps of the algorithm described in Sect. 2 into the following three:

1. the *shock-fitting module* identifies the grid-points and constrained edges (triangles in 3D) that make up the shock-fitting mesh to be used over the current time interval (Sect. 2.1) and calls the *meshing software* that builds the shock-fitting mesh using a CDT (Sect. 2.2);
2. the solution is updated by the *gas-dynamic solver* within all grid-points of the shock-fitting mesh (Sect. 2.4); while doing so, no boundary conditions¹ are applied by the gas-dynamic solver within the pair of grid-points located along the fitted discontinuities;
3. the *shock-fitting module* sets the correct boundary conditions by imposing the jump relations across the discontinuities (Sect. 2.5) and moves the shocks (Sect. 2.6).

Modularity stems from the fact that these three different components communicate through ad-hoc interfaces. This programming approach may not be the most efficient from the standpoint of computational speed, because, for instance, one has to switch, at each time step, among the different data-structures used by the three different modules. However, the approach is very convenient, since it allows us to use “off the shelf” gas-dynamics solvers and mesh generation tools that are treated as black boxes and can be replaced by similar ones only by changing the interfaces, with a modest coding effort.

In the following we shall briefly describe the various modules that have been used so far.

Concerning the *shock-fitting module*, two different FORTRAN versions have been developed by the authors. The original F77 code is capable of dealing with steady and un-steady two-dimensional flows, featuring both shocks and contact

¹A different approach for dealing with the shock-downstream boundary is described in [12, 25].

discontinuities as well as their mutual interactions or their interaction with solid surfaces. This is the code that has been used to compute the results presented in Sects. 4.1 and 4.2 and in the contribution [11] within this book. A different F90 code is capable of dealing with steady three-dimensional flows featuring multiple, but non-interacting shocks; it has been used to compute the results presented in Sect. 4.3. A C++ implementation of the *shock-fitting module*, capable of handling both two- and three-dimensional flows, has been developed at the Von Karman Institute; it is described in this book [25] and it has been made publicly accessible: <https://github.com/andrealani/ShockFitting>.

Concerning the *gas-dynamic solvers*, three different in-house codes have been used so far: `eulfs`, which has been developed by the authors [5, 6], the VKI solver `CoolFluid` [26, 27] and `NEO`, developed at INRIA [4, 50, 51]. All three codes share the same vertex-centered, Fluctuation Splitting [1, 13] discretization of the governing PDEs on linear triangles and tetrahedra. However, virtually any vertex-centered FE or FV solver featuring a linear representation of the dependent variables can be used as gas-dynamic solver within the shock-fitting procedure. The use of cell-centered solvers in conjunction with the algorithm described in Sect. 2 has been attempted in [12], with only partial success. The reader is referred to [61] and the contribution [28] in this book for the description of a different unstructured shock-fitting technique coupled to a cell-centered FV solver.

Concerning the *meshing software*, both the FORTRAN and C++ implementations of the algorithm use the public domain codes `triangle` [53] and `tetgen` [54] to build the CDT in 2D, resp. 3D, whereas `Yams` [15] or `CGAL` [55] has been used to construct the triangulation over the shock surfaces.

The interested reader will find a much more in-depth discussion of the algorithmic and programming details of the unstructured shock-fitting algorithm in the contribution [25] in this book.

4 Unstructured Shock-Fitting: Applications

The current capabilities of our unstructured, shock-fitting algorithm will be hereafter demonstrated by reference to a limited number of test-cases, including steady two- and three-dimensional flows. For all sets of simulations we have used the `eulfs` [5, 6] gas-dynamic solver.

4.1 External, Transonic Flow Past a NACA 0012 Airfoil

The superior accuracy that fitted shock-waves deliver over captured ones is illustrated by reference to a well-documented external flow test-case, namely the two-dimensional, inviscid, transonic flow past the NACA 0012 airfoil at $\alpha_\infty = 1^\circ$ angle of incidence and free-stream Mach number equal to $M_\infty = 0.85$.

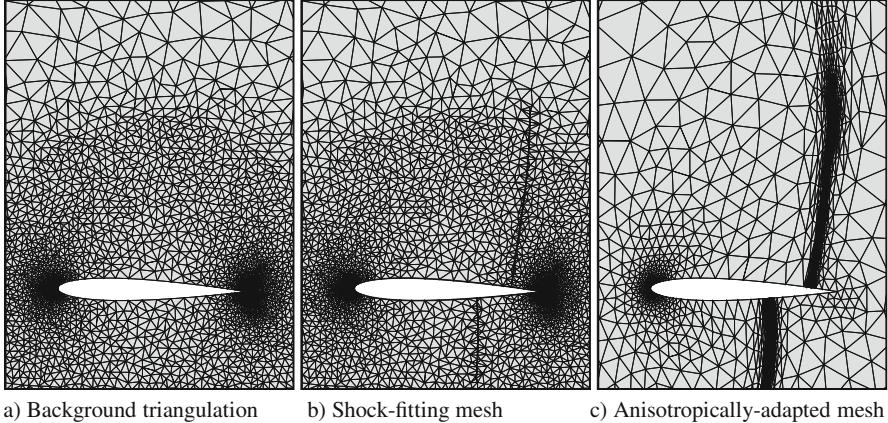


Fig. 3 NACA 0012 airfoil, $\alpha_\infty = 1^\circ$, $M_\infty = 0.85$. Triangular meshes used in the shock-capturing (**a** and **c**) and shock-fitting (**b**) calculations

Three different calculations have been performed: shock-capturing using both an un-adapted and an anisotropically adapted mesh and shock-fitting; the corresponding grids are shown in Fig. 3. One of the two shock-capturing calculations uses the same grid also used as background triangulation in the shock-fitting simulation; this grid, which is shown in Fig. 3a is made of 9912 triangles and 5024 mesh-points, 78 of which are placed along the airfoil's profile. The shock-fitting mesh at steady state is shown in Fig. 3b: this mesh has 9938 triangles and 5072 grid-points and it only differs from the background triangulation in the neighborhood of the two shocks. The anisotropically adapted mesh is shown in Fig. 3c: it has been obtained using the software described in [14] after four adaptation cycles, starting from the shock-capturing solution computed on the background triangulation. When comparing the solution-adapted mesh with the other two, the following observations are in order. First, the solution-adapted mesh has almost twice as many cells and gridpoints than the other two grids, since it is made of 17,895 triangular elements and 9123 grid-points, with 101 grid-points located along the airfoil's profile and clustered near the two shocks. Second, the repeated application of the mesh-adaptation technique led to an adapted mesh which is very different from the background triangulation, not only close to the shocks, but also in the smooth-flow regions where the adapted mesh is significantly coarser than the background triangulation. This fact has a negative effect on the quality of the solution, as will be explained hereafter.

Figure 4 shows the Mach number field in the vicinity of the profile, computed using shock-capturing on the two different grids (frames a and c) and shock-fitting (frame b). A more precise comparison is made possible in Fig. 5, where the pressure coefficient C_p distribution along the profile using the two different modeling practices is displayed; the smaller frame shows a detail of the point of the suction side where the shock impinges on the solid surface.

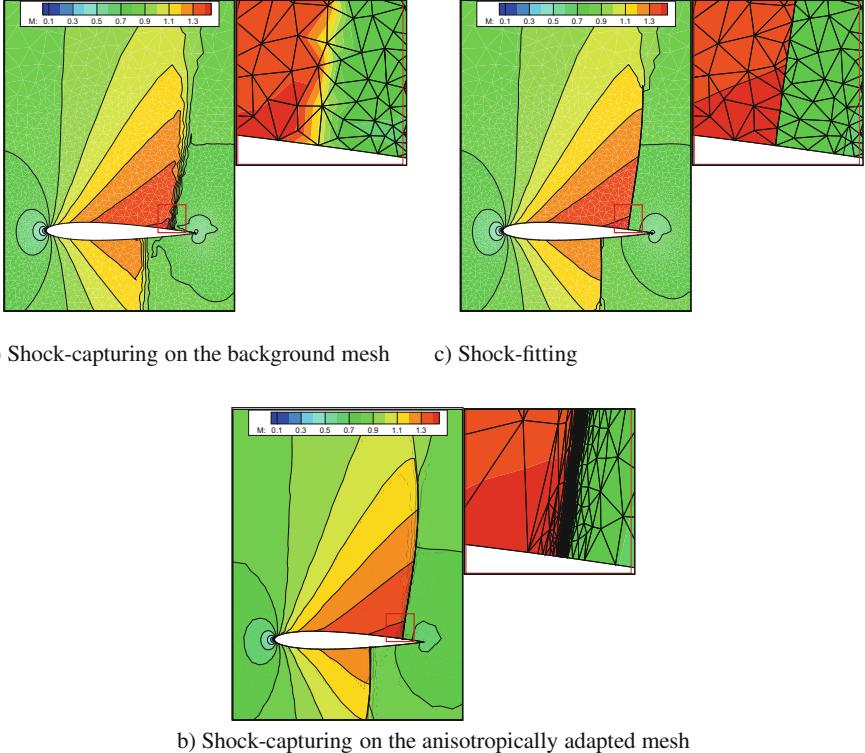


Fig. 4 NACA 0012 airfoil, $\alpha_\infty = 1^\circ$, $M_\infty = 0.85$. Mach number field: (a) Shock-capturing on the un-adapted mesh, (b) Shock-capturing on the anisotropically adapted mesh, (c) Shock-fitting

When comparing shock-capturing on the un-adapted mesh against shock-fitting (see Fig. 4a, b), it is clear that the shock-fitting approach returns a much more realistic shock thickness than shock-capturing. Not only the shock-capturing calculation predicts an un-physically large shock-thickness, it also completely misses the so-called Zierep [60] singularity, visible in the small frame of Fig. 5, that occurs on the downstream foot of the shock which, by contrast, is picked up when the shock is fitted or adaptation is used.

Close inspection of the C_p distribution reveals that the shock positions predicted by shock-fitting and shock-capturing on the adapted mesh are different, even if the difference is smaller than the shock-thickness captured on the un-adapted mesh. This tiny difference could be rooted to the fact that the adapted mesh is significantly coarser than the other two in the smooth-flow-regions and, in particular, around the trailing edge, see Fig. 3.

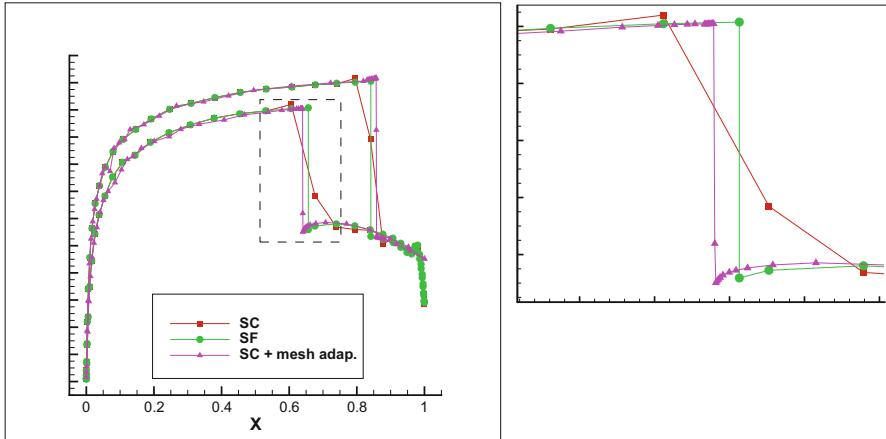


Fig. 5 NACA 0012 airfoil, $\alpha_\infty = 1^\circ$, $M_\infty = 0.85$: pressure coefficient distribution along the profile. Squares: shock-capturing on the un-adapted mesh; triangles: shock-capturing on the anisotropically adapted mesh; circles: shock-fitting

4.2 Internal, Transonic Flow Past the VKI LS-59 GT Rotor Blade

The Von Karman Institute gas turbine rotor blade (VKI LS-59) is a high loaded blade with a thick, rounded trailing edge; this blade has been extensively tested, both experimentally [24] and numerically, see, e.g., [2, 3].

Computations relative to an isentropic exit Mach number, $M_{is} = 1.2$, are shown in Figs. 6 and 7. Figure 6 compares the density iso-contour lines computed in the present unstructured shock-capturing and shock-fitting calculations (shown, respectively, in Fig. 6b, c) with the result obtained by Arnone et al. [3] using a 449×17 non-periodic C-type structured grid.

Concerning the two shock-capturing calculations, Fig. 6a, b, there are noticeable differences in the shock structure: the shock reflected off the suction side is better captured in the unstructured calculation and also the trailing edge shock patterns are different. This is likely due to differences in the mesh resolution between the structured and unstructured grids. However, also the shock-capturing and shock-fitting calculations on the unstructured grid show noticeable differences in the trailing edge shock structure, even if the two triangular grids are nearly identical. The aforementioned differences can be better seen in Fig. 7, which shows an enlarged view of the density iso-contour lines in the trailing edge region for both unstructured-grid calculations: shock-capturing in Fig. 7a and shock-fitting in Fig. 7b. The key difference between these two solutions is the much more pronounced unsteadiness of the shock-fitting solution, which suggests that it features a reduced level of numerical viscosity, compared to the shock-capturing one. The entire shock topology downstream of the trailing edge is consequently

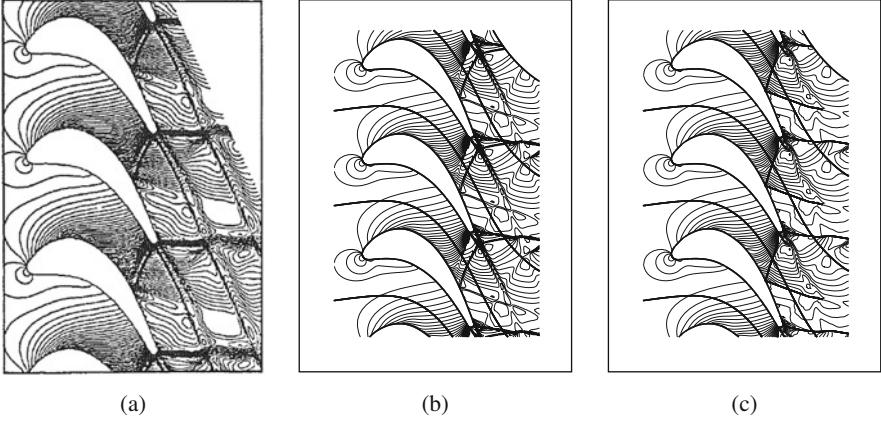
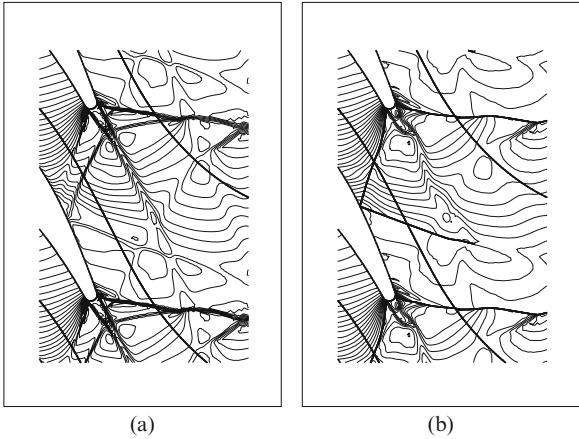


Fig. 6 VKI LS-59 GT rotor blade ($M_{is} = 1.2$): density iso-contours. (a) Re-printed from [3]. (b) Present shock-capturing calculation. (c) Present shock-fitting calculation

Fig. 7 VKI LS-59 GT rotor blade ($M_{is} = 1.2$): density iso-contours revealing the shock structure in the trailing edge region.

- (a) Shock-capturing.
- (b) Shock-fitting



affected and it is therefore different in the two sets of unstructured-grid calculations. It is also worth mentioning that those discontinuities that are not fitted in the shock-fitting calculation can however be captured thanks to the use of a shock-capturing discretization away from the fitted discontinuities. This is the case of the λ -shock structure that is visible in both sets of calculation shown in Fig. 7. Further details concerning the present calculation can be found in [8].

Finally, we observe that, when dealing with inviscid flows, shock waves reach the impermeable solid surface from which they are eventually reflected: this is what happens, for instance, to the oblique shock that impinges on the suction surface in Fig. 7b. However, when viscous effects are accounted for, the shock cannot reach the wetted surface and a shock/boundary-layer interaction takes place. This phenomenon has not yet been modeled in the present unstructured shock-fitting

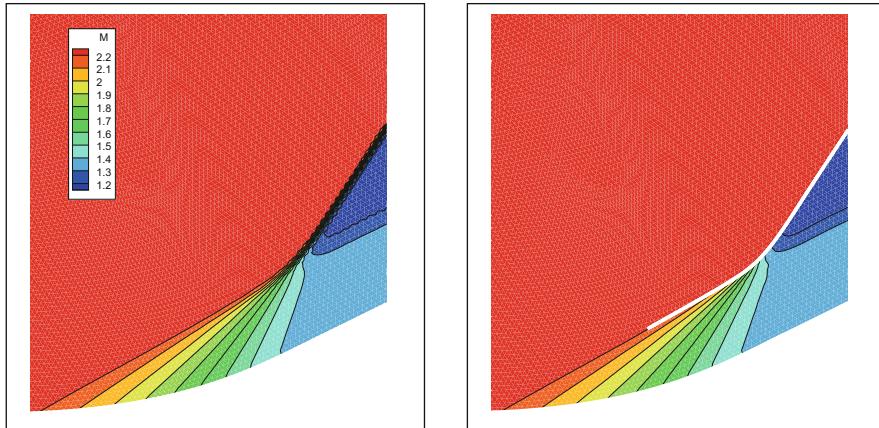


Fig. 8 Shock-wave formation within a supersonic, inviscid stream. Mach number field; *left*: shock-capturing; *right*: shock-fitting

algorithm, although shock/boundary–layer interactions have already been addressed in the shock-fitting community: examples include the early work by Marconi and Moretti [31] and the more recent work by Nasuti and Onofri [39, 40].

Preliminary results obtained using the unstructured shock-fitting algorithm are shown in Fig. 8, which displays a uniform supersonic stream being deflected while it flows over a smooth concave wall and the formation of a shock at a certain distance above the wall. The reader should not be surprised by the fact that the fitted-shock has been introduced upstream of the point where the characteristics merge because, in Moretti's words [32]:

Premature fitting of the shock in the region where compression waves tend to coalesce is not harmful at all, provided that the shock behaves as one of the characteristic surfaces coalescing into a finite discontinuity.

Even though the phenomenon that has been simulated is inviscid, the mechanism that leads to shock-formation, i.e. the coalescence of characteristic curves of the same family is representative of the situation that would be encountered in shock/boundary–layer interactions.

4.3 Hypersonic and Supersonic Three-Dimensional Blunt-Body Flows

The shock-fitting technique for unstructured grids has been successfully extended to the three-dimensional case [9]. Steady simulations of supersonic and hypersonic flows past three-dimensional bodies are shown in Fig. 9. Figure 9a shows pressure iso-contours within three cross-flow planes and over the body of the European

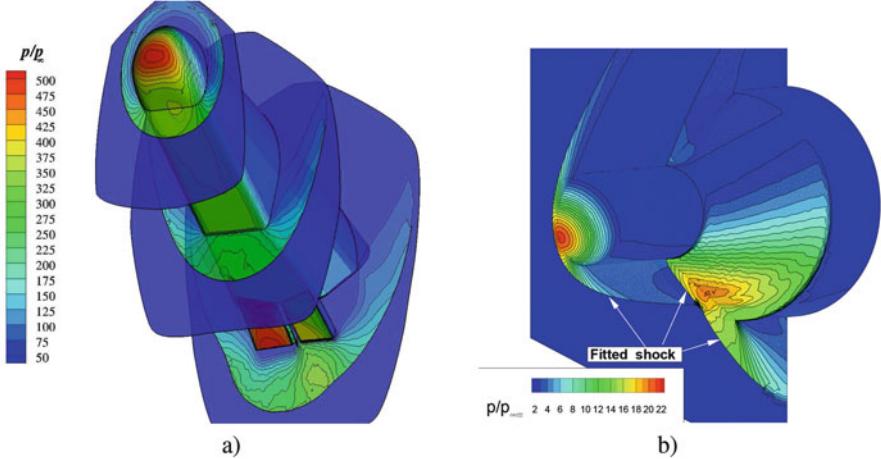


Fig. 9 Hypersonic ($M_\infty = 24$) and supersonic ($M_\infty = 4.04$) flow past three-dimensional, blunt-nosed objects. **(a)** ESA's Intermediate eXperimental Vehicle (IXV). **(b)** Cylinder with an hemispherical nose and a conical flare

Agency IXV vehicle, flying at $M_\infty = 24$ and 45° angle of attack. In this simulation only the bow shock has been fitted, whereas it has been left to the shock-capturing solver to capture the embedded shocks that arise at the flap-body junction and their interaction with the fitted bow shock.

Figure 9b shows pressure iso-contours within the symmetry plane and on the surface of a cylinder with a hemispherical nose and a conical flare. Free-stream conditions are: $M_\infty = 4.04$ and 20° angle of attack. In this calculation not only the bow shock, but also the embedded shock that arises at the cylinder-cone junction has been fitted. In contrast to the two-dimensional case, whereby we are capable of fitting various kinds of shock–shock interactions, this capability is currently unavailable in three dimensions; therefore, in the simulation shown in Fig. 9b, the interaction between the embedded and the bow shocks has been captured, rather than fitted. We will further address this specific issue in Sect. 5.2.

In all the numerical examples presented so far, the unstructured shock-fitting technique has been used to simulate inviscid flows. However, the algorithm has also been used to study the heat-flux anomalies that unstructured-grid codes exhibit around the stagnation point in hypersonic, viscous flows [16–19]. Even though shock-fitting did not provide conclusive answers in this context, it definitively performed better than shock-capturing: see the contribution [25] in this volume.

5 Future Algorithmic Improvements

A careful and honest analysis of the results obtained using the unstructured, shock-fitting technique developed by the authors reveals both its strengths and current limitations.

The upside of the shock-fitting method is that it allows to obtain accurate results on coarse meshes or, in Moretti's words [35]:

the more discontinuities are fitted, the fewer grid points are needed.

In this respect, the use of unstructured-grids has added nothing new to what the technique was capable of doing in the 1960s; however, the geometrical flexibility offered by the use of unstructured meshes allowed us to take advantage of the useful features of both the *boundary* and *floating* variants of the technique, without incurring into their limitations. Even so, the unstructured-grid shock-fitting technique that has been developed so far faces a number of unsolved issues that prevent it from being a general-purpose simulation tool. Although the following list may not be exhaustive, in Sects. 5.1 and 5.2 we will focus on the following two, currently unsolved issues:

- the occurrence of topological changes in the shock-pattern when dealing with un-steady flows;
- the interaction among different fitted discontinuities in the three-dimensional space.

It is important to underline that these issues are not due to intrinsic limitations of the shock-fitting technique, but they originate from either a lack of algorithmic development or the use of mesh generation tools that had been developed for different applications. Indeed, topological changes in un-steady flows had already been successfully addressed by Moretti in the late 1980s. For example, Fig. 10, which has been re-printed from [34], shows the simulation of a shock moving through an elbow performed by Moretti using the floating shock-fitting technique: it is evident that the technique is capable of automatically identifying the formation of a Mach reflection, the growth of the Mach stem, as well as other time-dependent features.

5.1 Topological Changes

The capability to recognize and deal with the topological changes that often occur in unsteady flows is a fundamental ingredient of the unstructured shock-fitting technique which we are currently developing. The task is certainly complex, but fortunately, as demonstrated in Fig. 10, the subject of automatic shock-detection has already been addressed by the shock-fitting community, see also [37, 38] for additional references.

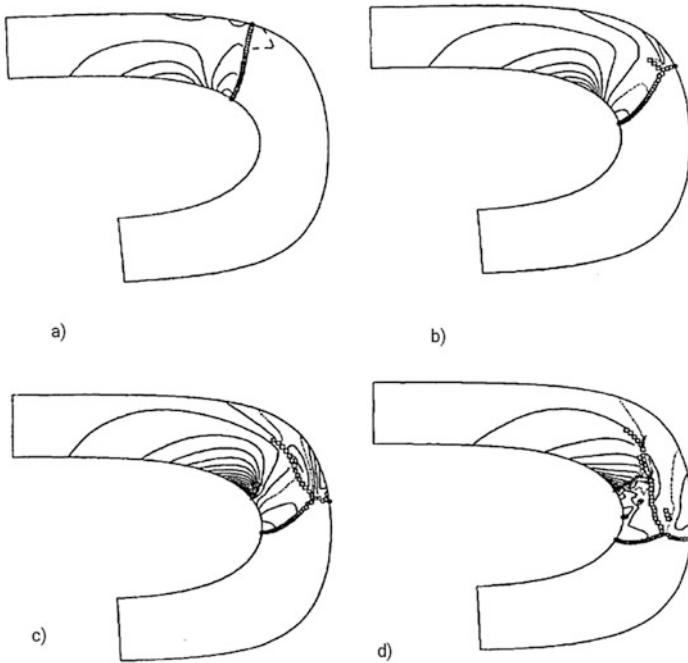


Fig. 10 Transonic flow in a duct with a 180° bend, re-printed from [34]: pressure iso-contours at four subsequent time instants

Topological changes may occur in a compressible flow in a variety of situations; for instance:

- A shock becomes so weak that it disappears, either completely or partly. In this latter case, the original shock is broken up into one or more shocks.
- The coalescence of compression waves leads to the formation of a shock-wave within the flow-field; as time elapses, the shock front may stretch over an increasingly wider area. For example, both the Mach stem and the reflected shock in Fig. 10 are seen to increase their length over time.
- Two distinct shocks interact, giving rise to a shock–shock interaction. Such an example can be seen in Fig. 10d: the re-compression shock that terminates the strong supersonic expansion along the inner wall interferes with the reflected shock.
- When a shock impinges on a wall, a regular or a Mach reflection arises. In the former case, a reflected shock appears, whereas in latter the situation is more complex because, in addition to the reflected shock, a Mach stem and a slip line also form and these are mutually connected with the impinging shock in a triple point. For instance, Fig. 10b shows the formation of a Mach reflection.

- A normal shock moving along a wall encounters a sudden variation of the wall slope (a ramp) which causes different kinds of complex interactions and these, in turn, produce topological changes.

This list, which is not exhaustive, shows that the situations that give rise to topological changes are numerous and varied. Therefore, it is by no means a trivial task to write an algorithm capable of dealing correctly with all possible situations. First of all, it is necessary to code rules capable of recognizing the spatial and temporal location where the topological change takes place and then discriminate among the various possible scenarios. This algorithmic development is still in its infancy and it is expected to take a long time. Only some of the aforementioned causes of topological changes have been up to now taken into account and translated into computer code; the remaining ones will be added at a later stage, thus making the algorithm capable of dealing with other types of topological changes.

In Sects. 5.1.1, 5.1.2, and 5.1.3 we shall briefly describe the algorithmic ingredients that have been developed so far in order to make the algorithm capable of dealing with changing shock-topologies. A preliminary example will be presented in Sect. 5.1.4.

5.1.1 Shock-Point Removal and Shock-Front Splitting

When part of a shock weakens, some of its shock-points may have to be removed. This can either shorten the shock front, if the shock-points to be removed are the end-points or lead to the formation of two or more smaller shock-branches, if only the inner shock-points are removed. This latter case clearly represents a topological change, because the number of shocks within the flow-field increases. The shock-strength within a shock-point can be assessed using the shock-upstream relative Mach number, $M_{n,u}$:

$$M_{n,u} = |w - \mathbf{u}_u \cdot \mathbf{n}| / a_u \quad (2)$$

where a_u and \mathbf{u}_u are, respectively, the sound and flow speed within the upstream shock-point. In [34] Moretti suggested that a shock-point should be removed if the following condition is met:

$$M_{n,u} < 1 + \epsilon \quad (3)$$

The parameter ϵ is an arbitrary threshold that we have set equal to $\epsilon = 0.05$.

5.1.2 Detection of New Shock Points

As already mentioned, new shocks may appear in an un-steady flow and/or the existing shocks may increase their length due to the coalescence of the characteristic curves. Therefore, at each time step, it is necessary to check for the appearance

of new shock-points within flow-field, in addition to those that belong to the discontinuities that have already been fitted.

Different techniques for detecting shock-waves have been documented in literature: a recent review can be found in [56]. Ma et al. [30] and Pagendarm and Seitz [45] proposed a criterion based on density derivatives. Lovely and Haimes [29] suggested a different criterion based on the scalar product between the local velocity vector and the pressure gradient. More recently, see also the contribution [23] in this book, Kanamori and Suzuki [22] used the theory of characteristics to identify the presence of new shock points. Characteristics theory had also been used by Moretti [36] to detect shock-waves in his floating shock-fitting technique on structured meshes. When dealing with structured meshes, however, a one-dimensional analysis can be applied separately along each coordinate direction, which is somewhat different from the technique suggested by Kanamori e Sukuzi, based on a truly multidimensional analysis.

In our tests we have used the simple criterion by Lovely and Haimes, but it is our opinion that the most suitable technique is the one proposed by Kanamori and Suzuki. Indeed, while all other techniques detect the shock only when it has already formed and its intensity is strong enough, the technique based on the characteristic curves is able to detect shocks in the early stages of their development when their strength is still very weak.

Unfortunately, it is not sufficient to apply a detection technique to complete the shock-detection task. Indeed, all the aforementioned techniques are able to detect new shock-points, but these have either to be connected to the shocks that have already been fitted or to be grouped together to form a new fitted-shock.

At present, the new shock-points are connected to the existing shocks or mutually grouped together with the user's intervention; in the future this process will be made automatic using an approach similar to that proposed by the authors in [43, 44], which takes advantage of fuzzy logic and algorithms devised for image processing.

5.1.3 Shock-Shock Interactions: Triple and Quadruple Point Insertion

In an unsteady flow, two different shocks can start to interact at a specific time in a specific point in space. The interaction point can be either a triple or a quadruple point in which case two new shocks form, along with a contact discontinuity. In order to detect the spatial location where new shock–shock interactions take place, it is necessary to check whether the end-point of an existing shock approaches a different shock. If this is the case, a triple (or a quadruple) point is inserted on the shock, which is split into two distinct shock-branches; moreover, a new fitted-shock and a fitted-slip-line must also be inserted. Whether the interaction point is a triple or a quadruple point, the features of the new discontinuities departing from the interaction point have to be inferred by analyzing the flow-field surrounding the interaction point. At present, this analysis has not yet been coded in a general-purpose algorithm.

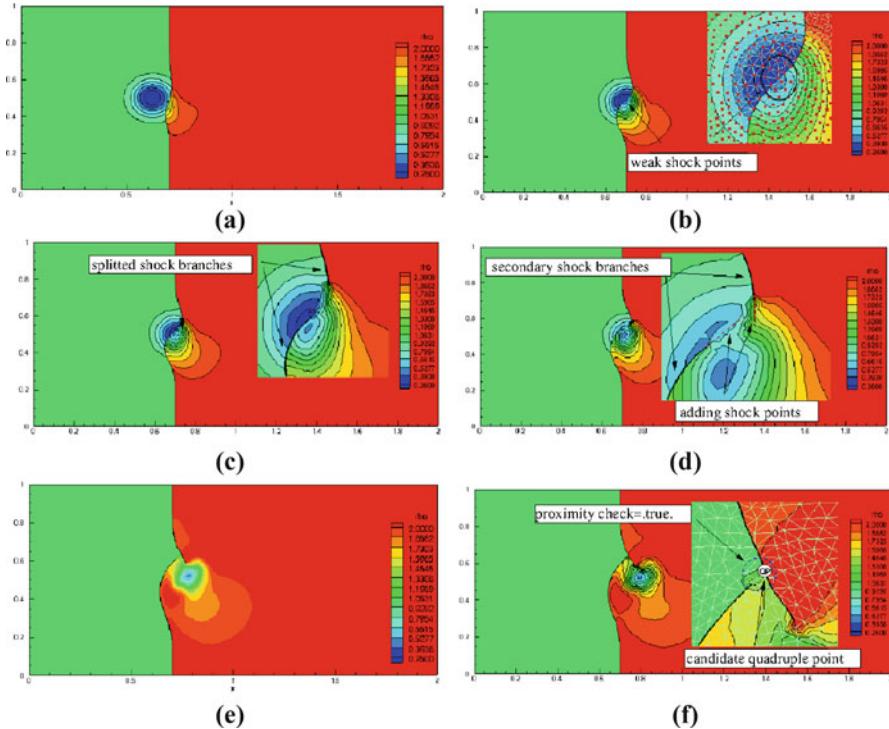


Fig. 11 Shock–vortex interaction: temporal evolution of flow topology in the region where the interaction occurs

5.1.4 A First Attempt to Simulate a Flow with Topological Changes

In order to explore the capability to treat the topological changes in an unsteady flow and to tune the algorithm and its coding, a shock–vortex interaction has been simulated trying to fit the topological changes using the algorithmic ingredients described in Sects. 5.1.1, 5.1.2, and 5.1.3.

In the early stages of the shock–vortex interaction, the part of the shock that has been hit by the vortex becomes so weak, see Fig. 11a, b, that Eq. (3) is verified in some of its shock-points, which are therefore removed. As a consequence, see Fig. 11c, the original shock is split into two different branches.

Once the shock has been split, the evolution of the flow conditions in the interaction region leads to the strengthening of the shock branches and the detection of some new shock-points. These new shock-points are connected to the end points of the two shock-branches which start to grow in length; this can be seen in Fig. 11d. The upstream and downstream states within these new shock-points are extrapolated from the neighboring shock-points.

At a certain time, see Fig. 11e, the two shock branches are sufficiently close to each other to start interacting. In the case presented here we know in advance that,

for the chosen pair of shock and vortex Mach numbers, the shock–vortex interaction gives rise to a strong, regular reflection, according to the taxonomy proposed in [20]: the two incident shocks merge in a quadruple point, as shown in Fig. 11f.

Even the rudimentary and handcrafted example reported in Fig. 11 reveals that the technique is capable of following the finest flow details, but requires careful coding in order to be of general-purpose applicability.

5.2 Mesh Generation and Re-Meshing Capability Improvements

As already mentioned in Sect. 3, the modular programming approach that we have used to develop the unstructured, shock-fitting algorithm allowed us to keep the overall coding effort within acceptable limits. This holds particularly true for the public domain software libraries (which we have referred to in Sect. 5) that we used for volumetric re-meshing around the shock-front and for triangulating the shock-surface. When using existing software, however, one is bound to the functionality provided, which might not address all the user’s needs. The use of meshing software tailored to our specific application may therefore prove essential to solve the second unsolved issue that we have mentioned in Sect. 5. Indeed, the meshing libraries that we have tested so far do not allow us to deal with three-dimensional shock–shock interactions, such as the one depicted in Fig. 9b. Even in the relatively simple case of Fig. 9b, where only two shocks are present, the embedded and the bow-shock interact along a three-dimensional curve. During their motion, the shock surfaces, and therefore the curve along which they intersect, move in space, eventually changing their shape and have to be re-triangulated, at least from time to time; it is important that this is done with minimal topological changes, so as to avoid interpolation errors, which could be particularly detrimental in un-steady simulations.

An algorithm that inserts curves or surfaces (in the 2D, resp. 3D space) into an existing triangular, resp. tetrahedral, mesh has been recently developed by Zaide and Gooch [57, 58]. Although originally developed for different applications, the algorithm turns out to be particularly well suited to perform the local CDT inside the gap dug by the shock-front in the background mesh, as described in Sec. 2.2. This is because in Zaide and Gooch’s algorithm [57] insertion of the curve/surface is done with minimal adjustment to the original topology while maintaining the original sizing of the mesh. Preliminary results obtained by applying the aforementioned algorithm to the shock-insertion problem are presented in the companion paper [59] in this book. We are hopeful that the ongoing collaboration [47] with Prof. Gooch’s research group will allow us, not only to use their algorithm to perform the shock-front insertion, but also to tackle the shock–shock interaction issue we have described above.

6 Conclusions

In this article the authors have summarized the state-of-the-art of the 10-years-long development of a shock-fitting technique for unstructured meshes, highlighting its strengths and also its current weaknesses. The analysis of a few selected applications allowed to highlight the advantages that fitted shocks offer over captured ones, when unstructured grids are used. It has also been shown that the use of unstructured grids allows to relieve some of the shortcomings that plagued the *classical* shock-fitting technique, originally developed within the structured-grid environment. Adequate room has also been given in this article to the current limitations of the technique, underlining that these are not intrinsic to the methodology, and presenting some preliminary attempts aimed at their overcoming.

At the end of this analysis we can draw an overall assessment of these 10 years of development of the unstructured shock-fitting technique and of its future perspectives. In these 10 years the technique has been successfully developed in order to demonstrate its applicability to steady, unsteady, two-dimensional and three-dimensional flows. The effort, however, which is necessary to complete the development of the technology and bring it to final maturity, is not modest and requires the merging of different skills, some of which belong to fields well outside CFD. In order to be successful, it will be necessary to broaden the audience of the researchers and expertise involved in the development of the shock-fitting technique.

Our hope is that over the next 10 years the unstructured shock-fitting can reach full maturity and be considered as a valid alternative to shock-capturing techniques, thereby achieving the goal that has been so actively pursued by Gino Moretti during his scientific life.

References

1. Abgrall, R.: Residual distribution schemes: current status and future trends. *Comput. Fluids* **35**(7), 641–669 (2006). <https://doi.org/10.1016/j.compfluid.2005.01.007>. Special Issue Dedicated to Professor Stanley G. Rubin on the Occasion of his 65th Birthday
2. Arnone, A., Swanson, R.C.: A Navier-Stokes solver for turbomachinery applications. *J. Turbomach.* **115**(2), 305–313 (1993). <https://doi.org/10.1115/1.2929236>
3. Arnone, A., Liou, M.S., Povinelli, L.A.: Transonic cascade flow calculations using non-periodic c-type grids. Technical Report 19910011758, NASA Lewis Research Center, Cleveland, OH, United States (1991)
4. Arpaia, L., Ricchiuto, M., Abgrall, R.: An ale formulation for explicit runge-kutta residual distribution. *J. Sci. Comput.* **190**(34), 1467–1482 (2014)
5. Bonfiglioli, A.: Fluctuation Splitting schemes for the compressible and incompressible Euler and Navier-Stokes equations. *Int. J. Comput. Fluid Dyn.* **14**(1), 21–39 (2000). <https://doi.org/10.1080/10618560008940713>
6. Bonfiglioli, A., Paciorri, R.: A mass-matrix formulation of unsteady fluctuation splitting schemes consistent with Roe's parameter vector. *Int. J. Comput. Fluid Dyn.* **27**(4–5), 210–227 (2013). <https://doi.org/10.1080/10618562.2013.813491>

7. Bonfiglioli, A., Paciorri, R.: Convergence analysis of shock-capturing and shock-fitting solutions on unstructured grids. *AIAA J.* **52**(7), 1404–1416 (2014). <https://doi.org/10.2514/1.J052567>
8. Bonfiglioli, A., Paciorri, R., Campoli, L.: Unstructured shock-fitting calculations of transonic turbo-machinery flows. In: 11th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, ETC (2015). <http://www.euroturbo.eu/paper/ETC2015-100.pdf>
9. Bonfiglioli, A., Grottadaurea, M., Paciorri, R., Sabetta, F.: An unstructured, three-dimensional, shock-fitting solver for hypersonic flows. *Comput. Fluids* **73**(0), 162–174 (2013). <https://doi.org/10.1016/j.compfluid.2012.12.022>
10. Bonfiglioli, A., Paciorri, R., Campoli, L.: Unsteady shock-fitting for unstructured grids. *Int. J. Numer. Methods Fluids* **81**(4), 245–261 (2016). <https://doi.org/10.1002/fld.4183>, Fld.4183
11. Campoli, L., Quemar, P., Bonfiglioli, A., Ricchiuto, M.: Shock-fitting and predictor-corrector explicit ALE residual distribution. In: Onofri, M., Paciorri, R. (eds.) *Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti*, pp. 113–129. Springer International Publishing, Berlin (2017)
12. De Amicis, V.: An unstructured, two-dimensional, shock-fitting solver for hypersonic flows. Project report, Von Karman Institute, Chaussée de Waterloo, 72 B-1640 Rhode-St-Genèse Belgium (2015). Supervisors: Dr. Andrea Lani and Prof. Herman Deconinck
13. Deconinck, H., Paillère, H., Struijs, R., Roe, P.L.: Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Comput. Mech.* **11**, 323–340 (1993). <https://doi.org/10.1007/BF00350091>
14. Dolejší, V.: Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Comput. Vis. Sci.* **1**(3), 165–178 (1998). <https://doi.org/10.1007/s007910050015>
15. Frey, P.J.: Yams: a fully automatic adaptive isotropic surface remeshing procedure. Rapport technique n° 0252, Institut National de Recherche en Informatique et en Automatique (I.N.R.I.A.), INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France) (2001). ISSN 0249-0803
16. Garicano-Mena, J., Pepe, R., Lani, A., Deconinck, H.: Assessment of heat flux prediction capabilities of residual distribution method: application to atmospheric entry problems. *Commun. Comput. Phys.* **17**(3), 682–702 (2015). <https://doi.org/10.4208/cicp.070414.211114a>
17. Gnoffo, P.: Simulation of stagnation region heating in hypersonic flow on tetrahedral grids (invited). In: *Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics, Reston (2007). <https://doi.org/10.2514/6.2007-3960>. AIAA 2007-3960
18. Gnoffo, P.: Multi-dimensional inviscid flux reconstruction for simulation of hypersonic heating on tetrahedral grids. In: *Aerospace Sciences Meetings*. American Institute of Aeronautics and Astronautics, Reston (2009). <https://doi.org/10.2514/6.2009-599>
19. Gnoffo, P., White, J.A.: Computational aerothermodynamic simulation issues on unstructured grids. In: *37th AIAA Thermophysics Conference*. American Institute of Aeronautics and Astronautics (AIAA), Reston (2004). <https://doi.org/10.2514/6.2004-2371>. AIAA-2004-2371
20. Grasso, F., Pirozzoli, S.: Shock-wave-vortex interactions: shock and vortex deformations, and sound production. *Theor. Comput. Fluid Dyn.* **13**(6), 421–456 (2000). <https://doi.org/10.1007/s001620050121>
21. Ivanov, M.S., Bonfiglioli, A., Paciorri, R., Sabetta, F.: Computation of weak steady shock reflections by means of an unstructured shock-fitting solver. *Shock Waves* **20**(4), 271–284 (2010). <https://doi.org/10.1007/s00193-010-0266-y>
22. Kanamori, M., Suzuki, K.: Shock waves detection in two-dimensional flow based on the theory of characteristics from CFD data. *J. Comput. Phys.* **230**(8), 3985–3082 (2011)
23. Kanamori, M., Suzuki, K.: Shock wave detection based on the theory of characteristics. In: Onofri, M., Paciorri, R. (eds.) *Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti*, pp. 171–189. Springer International Publishing, Berlin (2017)
24. Kiocok, R., Leithaus, F., Baines, N.C., Sieverding, C.H.: The transonic flow through a plane turbine cascade as measured in four european wind tunnels. *J. Eng. Gas Turbines Power* **108**(2), 277–284 (1986). <https://doi.org/10.1115/1.3239900>

25. Lani, A., De Amicis, V.: SF: An open source object-oriented platform for unstructured shock-fitting methods. In: Onofri, M., Paciorri, R. (eds.) Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti, pp. 85–112. Springer International Publishing, Berlin (2017)
26. Lani, A., Quintino, T., Kimpe, D., Deconinck, H.: The COOLFluiD framework - design solutions for high-performance object oriented scientific computing software. In: International Conference Computational Science 2005, vol. 1, pp. 281–286. Springer-Verlag, Atlanta (2005)
27. Lani, A., et al.: COOLFluiD Wiki page. <https://github.com/andrealani/COOLFluiD/wiki> (2016)
28. Liu, J., Zou, D.: A shock-fitting technique for ALE finite-volume methods on unstructured dynamic meshes. In: Onofri, M., Paciorri, R. (eds.) Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti, pp. 131–149. Springer International Publishing, Berlin (2017)
29. Lovely, D., Haimes, R.: Shock detection from computational fluid dynamics results. AIAA-99-3285 (1999)
30. Ma, K.L., Rosendale, J.V., Vermeer, W.: 3D shock wave visualization on unstructured grids. In: Proceedings of 1996 Symposium on Volume Visualization, pp. 87–94, 104 (1996). <https://doi.org/10.1109/SVV.1996.558049>
31. Marconi, F., Moretti, G.: Highly accurate prediction of unsteady viscous flows. Technical report, DTIC Document (1992)
32. Moretti, G.: On the matter of shock fitting. In: Richtmyer, R.D. (ed.) Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics. Lecture Notes in Physics, vol. 35, pp. 287–292. Springer, Berlin/Heidelberg (1974). <https://doi.org/10.1007/bfb0019764>
33. Moretti, G.: Computation of flows with shocks. *Annu. Rev. Fluid Mech.* **19**(1), 313–337 (1987). <https://doi.org/10.1146/annurev.fl.19.010187.001525>
34. Moretti, G.: A general-purpose technique for two dimensional transonic flows. NASA Contractor Reports 194186, NASA (1987)
35. Moretti, G.: Thirty-six years of shock fitting. *Comput. Fluids* **31**(4–7), 719–723 (2002)
36. Moretti, G.: Shock-fitting analysis. In: Onofri, M., Paciorri, R. (eds.) Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti, pp. 3–31. Springer International Publishing, Berlin (2017)
37. Moretti, G., Valorani, M.: Detection and fitting of two-dimensional shocks. *Notes Numer. Fluid Mech.* **20**, 239–246 (1988)
38. Nasuti, F., Onofri, M.: Analysis of unsteady supersonic viscous flows by a shock-fitting technique. *AIAA J.* **34**, 1428–1434 (1996)
39. Nasuti, F., Onofri, M.: Viscous and inviscid vortex generation during startup of rocket nozzles. *AIAA J.* **36**(5), 809–815 (1998). <https://doi.org/10.2514/2.440>
40. Onofri, M., Nasuti, F.: The physical origins of side loads in rocket nozzles. In: 35th Joint Propulsion Conference and Exhibit, Joint Propulsion Conferences. American Institute of Aeronautics and Astronautics, Reston (1999). <https://doi.org/10.2514/6.1999-2587>
41. Paciorri, R., Bonfiglioli, A.: A shock-fitting technique for 2d unstructured grids. *Comput. Fluids* **38**(3), 715–726 (2009). <https://doi.org/10.1016/j.compfluid.2008.07.007>
42. Paciorri, R., Bonfiglioli, A.: Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique. *J. Comput. Phys.* **230**(8), 3155–3177 (2011). <https://doi.org/10.1016/j.jcp.2011.01.018>
43. Paciorri, R., Bonfiglioli, A.: Recognition of shock-wave patterns from shock-capturing solutions. In: Computational Modelling of Objects Represented in Images III, pp. 91–96. CRC Press, Boca Raton (2012). <https://doi.org/10.1201/b12753-17>
44. Paciorri, R., Bonfiglioli, A.: Detection of shock-waves from shock-capturing solutions. In: Spitaleri, R.M. (ed.) MASCOT11 Proceedings, IMACS Series in Computational and Applied Mathematics 17 (2013), IMACS, pp. 161–170. IMACS, Rome (2013)
45. Pagendarm, H., Seitz, B.: An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves. In: Palamidese, P. (ed.) Scientific Visualization: Advanced Software Techniques. Prentice Hall India, Delhi (1993)

46. Pepe, R., Bonfiglioli, A., D'Angola, A., Colonna, G., Paciorri, R.: Shock-fitting versus shock-capturing modeling of strong shocks in nonequilibrium plasmas. *IEEE Trans. Plasma Sci.* **42**(10), 2526–2527 (2014). <https://doi.org/10.1109/TPS.2014.2324493>
47. Pepe, R., Bonfiglioli, A., Paciorri, R., Lani, A., Mena, J.G., Ollivier-Gooch, C.F.: Towards a modular approach for unstructured shock-fitting. WCCM XI, ECCM V, ECFD VI (2014)
48. Pepe, R., Bonfiglioli, A., D'Angola, A., Colonna, G., Paciorri, R.: An unstructured shock-fitting solver for hypersonic plasma flows in chemical non-equilibrium. *Comput. Phys. Commun.* **196**, 179–193 (2015). <https://doi.org/10.1016/j.cpc.2015.06.005>
49. Pirozzoli, S.: Numerical methods for high-speed flows. *Annu. Rev. Fluid Mech.* **43**(1), 163–194 (2011). <https://doi.org/10.1146/annurev-fluid-122109-160718>
50. Ricchiuto, M.: An explicit residual based approach for shallow water flows. *J. Comput. Phys.* **80**, 306–344 (2015)
51. Ricchiuto, M., Abgrall, R.: Explicit Runge-Kutta residual distribution schemes for time dependent problems: second order case. *J. Comput. Phys.* **229**(16), 5653–5691 (2010)
52. Salas, M.: A Shock-Fitting Primer, 1st edn. CRC Applied Mathematics & Nonlinear Science. Chapman & Hall, London (2009)
53. Shewchuk, J.R.: Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In: Lin, M.C., Manocha, D. (eds.) *Applied Computational Geometry: Towards Geometric Engineering*. Lecture Notes in Computer Science, vol. 1148, pp. 203–222. Springer, Berlin/Heidelberg (1996). From the First ACM Workshop on Applied Computational Geometry
54. Si, H.: TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.* **41**(2), 11:1–11:36 (2015). <https://doi.org/10.1145/2629697>
55. The CGAL Project: CGAL User and Reference Manual, 4.9 edn. CGAL Editorial Board. <http://doc.cgal.org/4.9/Manual/packages.html> (2016)
56. Wu, Z., Xu, Y., Wang, W., Hu, R.: Review of shock wave detection method in CFD post-processing. *Chin. J. Aeronaut.* **26**, 501–513 (2013)
57. Zaide, D.W., Ollivier-Gooch, C.F.: Inserting a curve into an existing two dimensional unstructured mesh. In: Sarrate, J., Staten, M. (eds.) *Proceedings of the 22nd International Meshing Roundtable*, pp. 93–107. Springer International Publishing, Cham (2014)
58. Zaide, D.W., Ollivier-Gooch, C.F.: Inserting a surface into an existing unstructured mesh. *Int. J. Numer. Methods Eng.* **106**(6), 484–500 (2016). <https://doi.org/10.1002/nme.5132>. Nme.5132
59. Zaide, D.W., Ollivier-Gooch, C.F.: Inserting a shock surface into an existing unstructured mesh. In: Onofri, M., Paciorri, R. (eds.) *Shock Fitting: Classical Techniques, Recent Developments, and Memoirs of Gino Moretti*, pp. 151–169. Springer International Publishing, Berlin (2017)
60. Zierep, J.: New results for the normal shock in inviscid flow at a curved surface. *ZAMM - J. Appl. Math. Mech./Z. Angew. Math. Mech.* **83**(9), 603–610 (2003). <https://doi.org/10.1002/zamm.200310051>
61. Zou, D., Xu, C., Dong, H., Liu, J.: A shock-fitting technique for cell-centered finite volume methods on unstructured dynamic meshes. *J. Comput. Phys.* **345**, 866–882 (2017). <https://doi.org/10.1016/j.jcp.2017.05.047>

SF: An Open Source Object-Oriented Platform for Unstructured Shock-Fitting Methods

Andrea Lani and Valentina De Amicis

1 Introduction

A novel unstructured shock-fitting method has been developed over the last few years [6, 32, 34], combining features of both variants of the traditional structured-grid shock-fitting technique: the coding simplicity of the *boundary* version and the capability of dealing with complex flows of its *floating* counterpart. At the same time, the unstructured version of the technique is not affected by the strong topological limitations plaguing boundary shock-fitting when implemented on structured grids. Moreover, the coupling between the unstructured shock-fitting and existing CFD solvers is algorithmically much simpler than integrating the floating shock-fitting technique with a structured-grid solver. The modularity of the approach was recently demonstrated by Pepe et al. [41], where different CFD solvers have been coupled to the existing shock-fitting code used in [7, 13, 34]. The latter code, written in FORTRAN90, is capable of handling steady and unsteady flows, including shocks and/or contact discontinuities, mostly in 2D but also in more complex 3D cases [6].

This paper describes *SF*, the first (and only) open source and object-oriented shock-fitting framework which has been developed in C++ and released on <https://github.com/andrealani/ShockFitting>, aiming at:

A. Lani (✉)

Von Karman Institute for Fluid Dynamics, Chaussee de Waterloo 72, Belgium
e-mail: lani@vki.ac.be

V. De Amicis

Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Piazzale Aldo Moro 5, 00185 Rome, Italy
e-mail: valentina.deamicis91@gmail.com

1. providing most features of the original FORTRAN90 code;
2. making the development of shock-fitting more user-friendly, allowing each feature to be treated as a dynamic component with well-defined APIs;
3. fully automatizing each step of the method, in particular the construction of the initial shock profile (i.e., surface in 3D);
4. easing the integration of new algorithms and physical models (e.g., magnetohydrodynamics, reactive/multi-phase flows) which could benefit from the method;
5. easing the coupling to third party open source tools for mesh generation, geometric manipulation (e.g., surface reconstruction, smoothing), CFD simulation.

This contribution is organized as follows: in Sects. 2 and 3 respectively, an improved, totally automatic, shock-fitting method and its coupling to a state-of-the-art Residual Distribution (RD) solver (provided by the open source COOLFluiD platform) are briefly described; then, Sect. 4 focuses on the design of SF; numerical results are presented in Sect. 5; finally, Sect. 6 addresses conclusions and ongoing work.

For the purpose of this paper, we assume to deal only with 2D meshes of triangular cells. The underway extension to 3D cases, involving tetrahedral cells, is not mature enough to be discussed here but will be the subject of future publications.

2 Shock-Fitting Algorithm

The shock-fitting approach is inherently time-dependent, since both solution and grid change with time, due to the displacement of the fitted discontinuities. When a steady solution exists, the discontinuities speed will asymptotically vanish and the tessellation of the flow domain will not change any longer. At time t , the solution and grid velocity is available within all grid points of a triangulated *background* mesh covering the entire computational domain. In addition to this, the fitted discontinuity (i.e., a shock in our case) must be discretized using a collection of points which are mutually joined to form a connected series of line segments.

The following subsections provide a concise description of the main algorithmic steps leading from the available mesh/solution at time t to an updated mesh/solution at time $t + \Delta t$. Herein, the first subsection will describe in detail the automatic determination of the initial shock profile (Sect. 2.1), which is a new, important, feature that has been implemented in SF but not available in the legacy FORTRAN90 code.

2.1 Automatic Determination of the Initial Shock Profile

In order to initialize the shock-fitting algorithm, the initial shock profile can be either extracted *manually* and read in as a file (as in the original FORTRAN90 code) or automatically determined by a shock detector algorithm (as in SF). The latter

requires (1) a criterion to identify the shock points, (2) a method to represent their distribution through a curve tracing the discontinuity profile (e.g., via least square fitting techniques), and (3) the identification of the *special* points.

2.1.1 Shock Points Detection

As reported in [48], shock wave detection methods can be classified as *traditional* and *advanced*. One example of the former, accounting for the pressure contours, which cluster in the vicinity of the shock, is hardly applicable to 3D flows, producing an extracted shock which is usually not well defined. Another traditional method, considering the Mach number iso-surface, yields too many surfaces other than the desired ones. Advanced methods include *Hough transform*-based [33, 35], *Theory of characteristics* [14], *Density gradient maxima* [37], and *Normal Mach number* [25] algorithms. In particular, the *Density gradient maxima* looks for the maximum of the density gradient. The first and second density derivatives, $\frac{d\rho}{dn}$ and $\frac{d^2\rho}{dn^2}$, are computed through the velocity:

$$\begin{cases} \frac{d\rho}{dn} = \nabla \rho \cdot \frac{\mathbf{v}}{|\mathbf{v}|} \\ \frac{d^2\rho}{dn^2} = \nabla \left(\nabla \rho \cdot \frac{\mathbf{v}}{|\mathbf{v}|} \right) \cdot \frac{\mathbf{v}}{|\mathbf{v}|}. \end{cases} \quad (1)$$

The iso-surface of $\frac{d^2\rho}{dn^2} = 0$ corresponds to the maxima or minima of streamwise density gradient. $\frac{d\rho}{dn}$ is used as the filtering variable. Conditions $\frac{d\rho}{dn} > 0$ correspond to shock wave, while $\frac{d\rho}{dn} < 0$ correspond to expansion wave. The *Normal Mach number* method relies upon the calculation of the normal Mach computed through the velocity projection along the local pressure gradient direction, since the direction normal to the shock wave is aligned with the latter. The iso-surface with unit normal Mach number represents the detected shock-wave surface:

$$\mathbf{M}_n = \frac{\mathbf{M} \cdot \nabla p}{|\nabla p|} = \frac{\mathbf{v} \cdot \nabla p}{a |\nabla p|} = 1, \quad (2)$$

where \mathbf{M} is the Mach number vector in the direction of local flow velocity \mathbf{v} and a is the local sonic speed. All aforementioned methods require filtering algorithms in order to reduce the spurious shock points appearing in the smooth regions.

A simpler and more effective method is proposed by Gnoffo in [10]. It is based on the computation of the maximum pressure ratio between the nodes of a triangle (or tetrahedron in 3D). The shock point is identified when the pressure ratio between two nodes is higher than a specified value Φ_{\max} (typically set = 3):

$$\left(\frac{p_{\text{node}_i}}{p_{\text{node}_j}} \right)_{\text{cell}} |_{\max} > \Phi_{\max}. \quad (3)$$

Our early attempts to extract shock points via *Density Gradient Maxima* or *Normal Mach Number* methods resulted in spread distributions, extending up to the smooth regions. In those calculations, nodal pressure gradients were evaluated by a Green Gauss integration over a *median dual cell* [5]. Since Gnoffo's approach turned out to work effectively with strong shocks and be easier to implement, it became our criteria of choice for extracting shock points in all calculations (see Sect. 5).

2.1.2 2D Least Squares Methods for Fitting the Shock Points Distribution

After applying a shock detector, the resulting points distribution is fitted by a polynomial curve representing the shock using a Least Squares Method (LSM). To make SF as flexible as possible, three different LSM variants have been implemented.

The *Polynomial Curve Fitting* interpolates a set of N_p points by applying LSM to a polynomial with user-defined order P . Considering:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_jx^j = a_0 + \sum_{k=1}^P a_k x^k, \quad (4)$$

the curve giving the minimum error between data y_i and the fit $f(x)$ is the best one. The error function `err` can be written as follows:

$$\text{err} = (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + \dots + (y_j - f(x_P))^2 = \sum_{i=1}^{N_p} \left(y_i - \left(a_0 + \sum_{k=1}^P a_k x^k \right) \right)^2. \quad (5)$$

In order to minimize `err` in (5), the following system is obtained:

$$\begin{bmatrix} N_p & \sum_{i=1}^{N_p} x_i^2 & \dots & \sum_{i=1}^{N_p} x_i^j \\ \sum_{i=1}^{N_p} x_i & \sum_{i=1}^{N_p} x_i^2 & \dots & \sum_{i=1}^{N_p} x_i^{P+1} \\ \sum_{i=1}^{N_p} x_i^2 & \sum_{i=1}^{N_p} x_i^3 & \dots & \sum_{i=1}^{N_p} x_i^{P+2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_p} x_i^P & \sum_{i=1}^{N_p} x_i^{P+1} & \dots & \sum_{i=1}^{N_p} x_i^{P+P} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_p} y_i \\ \sum_{i=1}^{N_p} x_i y_i \\ \sum_{i=1}^{N_p} x_i^2 y_i \\ \vdots \\ \sum_{i=1}^{N_p} x_i^P y_i \end{bmatrix}, \quad (6)$$

where the derivatives with respect to the unknown coefficients a_0, \dots, a_P (that best fit $f(x)$) have been set to 0: $\left(\frac{\partial \text{err}}{\partial a_0} = 0 ; \frac{\partial \text{err}}{\partial a_1} = 0 ; \dots ; \frac{\partial \text{err}}{\partial a_P} = 0 \right)$.

The *Ellipse Curve Fitting* can fit shock points distribution shaping ellipses:

$$f(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0. \quad (7)$$

As explained in [1], the general quadratic form of a second-order curve is chosen as error function. F is set to 1 in order to avoid the trivial solution $A = B = C = D = E = F = 0$. The aggregate error is therefore given by:

$$\text{err}(x, y) = \sum_{i=1}^{N_p} (Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + 1)^2. \quad (8)$$

In order to minimize err in (8), the derivatives with respect to A, B, C, D , and E must be set to zero, leading to solve the following system:

$$\begin{bmatrix} \sum_{i=1}^{N_p} x_i^4 & \sum_{i=1}^{N_p} x_i^3 y_i & \sum_{i=1}^{N_p} x_i^2 y_i^2 & \sum_{i=1}^{N_p} x_i^3 & \sum_{i=1}^{N_p} x_i^2 y_i \\ \sum_{i=1}^{N_p} x_i^3 y_i & \sum_{i=1}^{N_p} x_i^2 y_i^2 & \sum_{i=1}^{N_p} x_i y_i^3 & \sum_{i=1}^{N_p} x_i^2 y_i & \sum_{i=1}^{N_p} x_i y_i^2 \\ \sum_{i=1}^{N_p} x_i^2 y_i^2 & \sum_{i=1}^{N_p} x_i y_i^3 & \sum_{i=1}^{N_p} y_i^4 & \sum_{i=1}^{N_p} x_i y_i^2 & \sum_{i=1}^{N_p} y_i^3 \\ \sum_{i=1}^{N_p} x_i^3 & \sum_{i=1}^{N_p} x_i^2 y_i & \sum_{i=1}^{N_p} x_i y_i^2 & \sum_{i=1}^{N_p} x_i^2 & \sum_{i=1}^{N_p} x_i y_i \\ \sum_{i=1}^{N_p} x_i^2 y_i & \sum_{i=1}^{N_p} x_i y_i^2 & \sum_{i=1}^{N_p} y_i^3 & \sum_{i=1}^{N_p} x_i y_i & \sum_{i=1}^{N_p} y_i^2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} = - \begin{bmatrix} \sum_{i=1}^{N_p} x_i^2 \\ \sum_{i=1}^{N_p} x_i y_i \\ \sum_{i=1}^{N_p} y_i^2 \\ \sum_{i=1}^{N_p} x_i \\ \sum_{i=1}^{N_p} y_i \\ \sum_{i=1}^{N_p} y_i \end{bmatrix}. \quad (9)$$

The aforementioned methods work well when the shock profiles are quite regular. Nevertheless, strong shocks in front of blunt bodies often do not trace ellipse or exact polynomial curves profiles. In order to tackle those cases, a third method, *Splitting Curves*, was implemented in SF, allowing for subdividing the shock points distribution into two (or more) smaller parts to which different polynomial orders can be assigned. In addition, a smoothing algorithm has been implemented to cure possible *oscillating* trends that can appear when high-order polynomials are used. In this case, the y-coordinates of the curve are simply averaged using the positions of the closest points according to $y_i = \frac{1}{N_s} \sum_{j=0}^{N_s-1} y_{i+j}$, where N_s is the size of the stencil chosen to compute the average.

2.1.3 Identification of *special* Junction Points

We define *special* those points that represent intersection points between the shock edges and the domain boundaries (i.e., *junction* points) or between the shock edges and other discontinuities (i.e., *interaction* points). Those points are individually treated inside the SF code in the normal vector computation, in the re-definition of the mesh pattern and when the jump relations are applied. Since, until now, the main focus has been to handle a single shock around blunt bodies, only junction points have been implemented inside SF so far. The detection of the junction points is quite simple. Strings, assigned to the boundaries of the domain, distinguish if the boundary is an *inlet*, an *outlet*, or a *wall* boundary. The rule, classifying a shock edge point as special, evaluates the distance from the boundary: if the distance is small enough (i.e., smaller than a predefined threshold, proportional to the characteristic

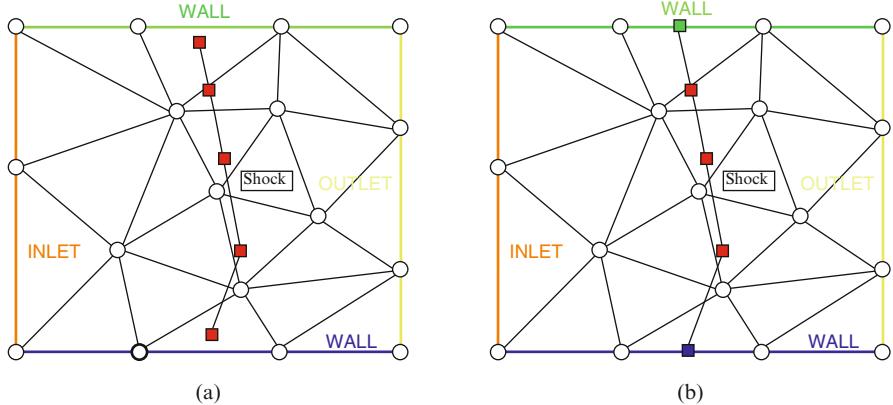


Fig. 1 Identification of the special points. **(a)** Shock profile and boundary conditions. **(b)** Assignment of the special points according to the boundary conditions

mesh size DXCELL), the special point type is assigned according to the string associated with the closest boundary edge, as schematically depicted in Fig. 1.

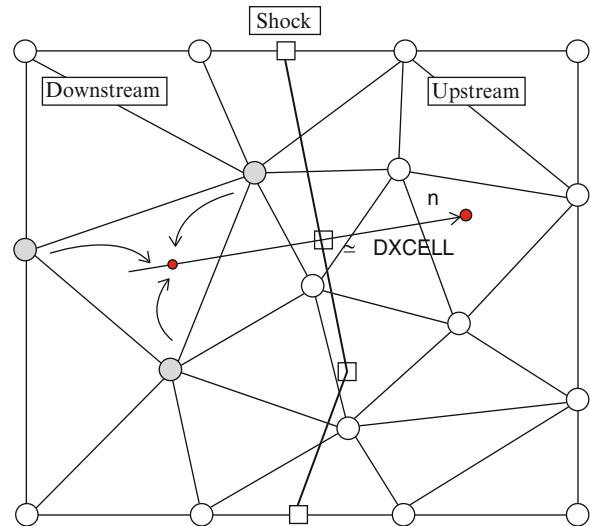
2.1.4 Assignment of the Shock Points State

Once that the shock polyline is extracted, appropriate upstream and downstream states must be assigned to the shock points. The states of the shock points are computed in the direction normal to the shock surface at a distance approximately equal to the mesh characteristic dimension DXCELL . The point, which is used to extract the downstream (or upstream) state, must be located outside from the shock thickness that is usually equal to one or two mesh sizes (it can be chosen each time). The state is then extracted by averaging the nodal states from the cell in which the extracted point is located, as shown in Fig. 2 for a downstream state.

2.2 Cell Removal Around the Shock Front

In this first step, the fitted shock is laid on top of the background mesh, as shown in Fig. 3a. All cells which are crossed by the fitted shock and those mesh points that are located too close to it are temporarily removed from the background mesh, as shown in Fig. 3b. We call *phantom* those grid-points of the background mesh (indicated with dashed circles in Fig. 3b) that have been temporarily removed. All cells having at least one phantom node among their vertexes are also removed from the background triangulation; these are the cells shown using dashed edges

Fig. 2 Extraction of the downstream and upstream state



in Fig. 3b. Further details concerning the criteria used to identify and remove the phantom nodes can be found in [6, 32].

2.3 Local Re-Meshing Around the Shock Front

Following the cell removal step, the background triangulation is split into two or more disjoint sub-domains, as shown in Fig. 3c. The hole dug by the fitted front is then re-meshed using a Constrained Delaunay Tessellation (CDT). The edges making up the fitted discontinuity and the boundary of the hole are both constrained to be part of the final tessellation, as illustrated in Fig. 3d. For the moment, the open source Triangle [45] mesh generator is used to construct the CDT, but other generators can be easily interfaced in SF. Since the re-meshing is localized around the shock, it has a very limited CPU cost. Upon completion of this stage, the computational domain is discretized using the so-called *shock-fitting* mesh, differing from the background one only in the neighborhood of the fitted shock.

2.4 Computation of the Tangent and Normal Unit Vectors

In order to apply the jump relations, tangent (τ) and normal (n) unit vectors are needed within each pair of grid-points located along the discontinuities; this is schematically illustrated in Fig. 3e. These unit vectors are computed using finite-

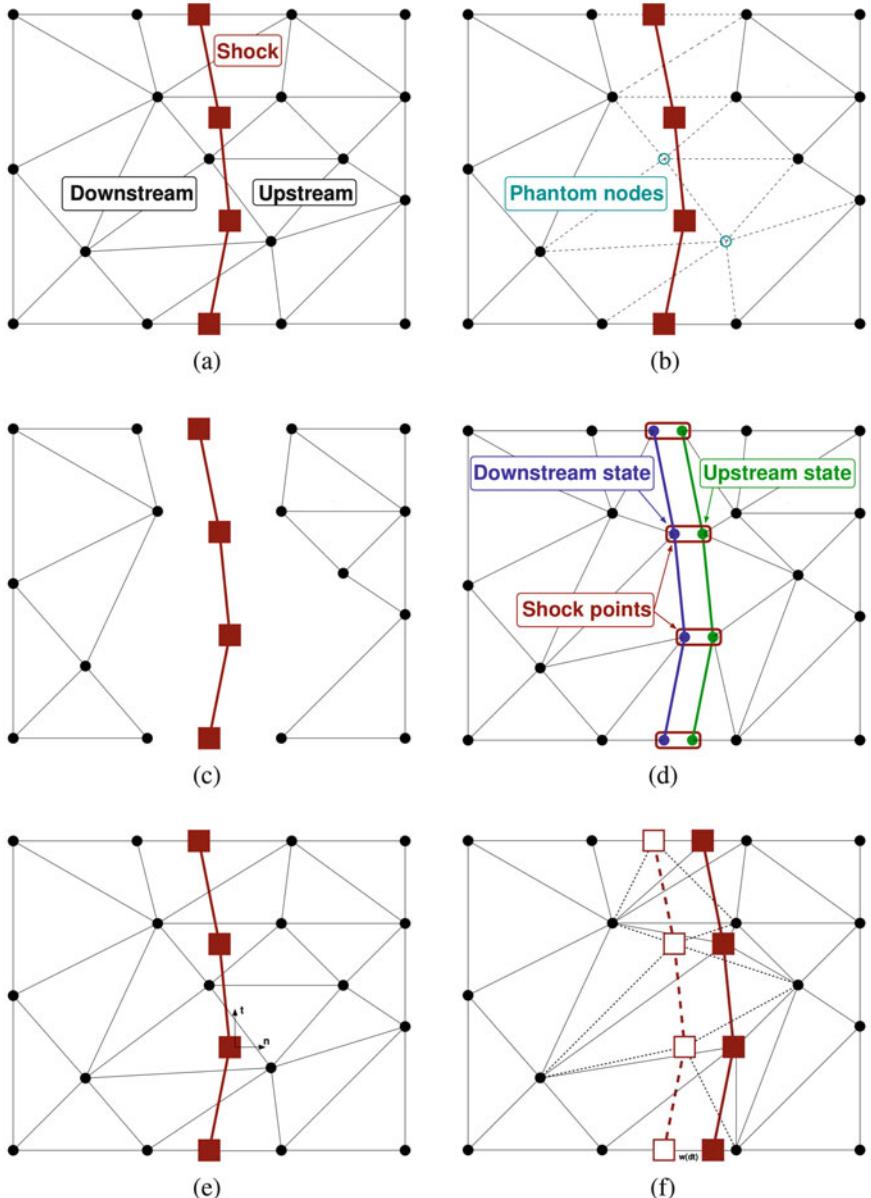


Fig. 3 Main steps of the unstructured shock-fitting algorithm. (a) Shock front moving over the background triangular mesh at time t . (b) Dashed lines mark the cells to be removed; cyan circles denote the phantom nodes. (c) The background mesh is split into disjoint sub-domains by a hole which encloses the shock. (d) The triangulation around the shock has been rebuilt. (e) Calculation of the shock-tangent and shock-normal unit vectors. (f) The shock displacement induces mesh deformation

difference (FD) formulas, involving the coordinates of the shock point itself and those of its neighboring shock points. Depending on the local flow regime, it may be necessary to use upwind-biased formulas, rather than centered ones, to avoid that geometrical instabilities arise along the fitted shock. Full details about the normal calculations in 2D are reported in [32, 43].

2.5 Solution Update Using the Shock-Capturing Code

Using the shock-fitting mesh as input, a single time step calculation is performed using an unstructured, vertex-centered shock-capturing solver which returns updated nodal values at time $t + \Delta t$. Since the fitted shock is seen by the CFD solver as an internal boundary (of zero thickness) moving at the shock speed, there is no need for modifying the spatial discretization scheme, as long as only one time step is run between two subsequent shock-fitting calculations. However, this approach is not correct if more than one time step is allowed between two subsequent shock-fitting applications. In the latter case, the CFD solver must impose a consistent boundary condition on the post-shock side of the fitted discontinuity, for instance a far field condition. This alternative strategy, which has not been considered in previous literature concerning the unstructured shock-fitting methods, allows for saving computational time whenever the shock has reached its final position but the flow simulation is still far from having reached convergence, e.g. in viscous cases.

In general, the shock-capturing solver can be always used as a black-box: it receives as input the shock-fitting grid, the nodal values of the solution and grid velocity at time t and returns the updated solution at time $t + \Delta t$.

2.6 Enforcement of the Jump Relations

The solution returned by the CFD solver at time $t + \Delta t$ is missing some boundary conditions on the downstream side of the shock. Those conditions are enforced by the Rankine-Hugoniot (R-H) jump relations as described in [13, 34] and concisely reported here just for clarity. In order to match the number of unknowns with the available equations, additional pieces of information are obtained from the characteristic formulation of the Euler equations and correspond to those characteristic quantities that are convected towards the shock from the sub-domain that is attached to that shock side. Upon completion of step 2.5, the upstream shock states provided by the shock-capturing solver at time $t + \Delta t$ are entirely correct, since the flow remains supersonic and all characteristic waves are in-going. However, the provisional values computed by the CFD code for the downstream states are inconsistent since the entropy, vorticity, and forward moving acoustic wave are convected away from the shock, requiring three boundary conditions to

be imposed (as in a subsonic inlet). The downstream flow is subsonic in the shock-normal direction so that the backward moving acoustic wave conveys the following signal:

$$R_d^{t+\Delta t} = \tilde{a}_d^{t+\Delta t} + \frac{\gamma - 1}{2} \tilde{\mathbf{u}}_d^{t+\Delta t} \cdot \mathbf{n}^t, \quad (10)$$

from the downstream region towards the downstream side of the shock. In Eq. (10), the Riemann variable $R_d^{t+\Delta t}$ is assumed to be correct even if it is calculated from $\tilde{a}_d^{t+\Delta t}$ and $\tilde{\mathbf{u}}_d^{t+\Delta t}$, i.e. the inconsistent values of the acoustic and flow velocity of the downstream state of the shock nodes which are computed by the CFD solver on the shock-fitting mesh. The correct downstream state and the shock speed component w normal to the discontinuity are then obtained by solving a system of five non-linear algebraic equations, including the following four R-H jump relations:

$$\rho_u^{t+\Delta t} \mathbf{v}_u^{t+\Delta t} \cdot \mathbf{n}^t = \rho_d^{t+\Delta t} \mathbf{v}_d^{t+\Delta t} \cdot \mathbf{n}^t \quad (11a)$$

$$p_u^{t+\Delta t} + \rho_u^{t+\Delta t} (\mathbf{v}_u^{t+\Delta t} \cdot \mathbf{n}^t)^2 = p_d^{t+\Delta t} + \rho_d^{t+\Delta t} (\mathbf{v}_d^{t+\Delta t} \cdot \mathbf{n}^t)^2 \quad (11b)$$

$$\mathbf{v}_u^{t+\Delta t} \cdot \boldsymbol{\tau}^t = \mathbf{v}_d^{t+\Delta t} \cdot \boldsymbol{\tau}^t \quad (11c)$$

$$H_u^{t+\Delta t} = H_d^{t+\Delta t} \quad (11d)$$

$$R_d^{t+\Delta t} = a_d^{t+\Delta t} + \frac{\gamma - 1}{2} \mathbf{u}_d^{t+\Delta t} \cdot \mathbf{n}^t \quad (11e)$$

plus Eq. (10). For notational convenience, in Eqs. (11), we have introduced the flow velocity relative to the shock, namely $\mathbf{v} = \mathbf{u} - \mathbf{w}$, where $\mathbf{w} = w \mathbf{n}$ is the shock speed relative to an inertial reference frame and H is the specific total enthalpy of the relative motion:

$$H = \frac{\gamma}{\gamma - 1} \frac{p}{\rho} + \frac{\mathbf{v}^2}{2}. \quad (12)$$

While all downstream (\cdot_d) variables in the RHS of Eqs. (11) are unknowns, all upstream (\cdot_u) values on the LHS (except for w) are those correctly updated by the shock-capturing solver at time $t + \Delta t$. The five unknown quantities are: $a_d^{t+\Delta t}$, and $\rho_d^{t+\Delta t}$ (from which pressure p can be derived), the two Cartesian components of the velocity $\mathbf{u}_d^{t+\Delta t}$, and the shock speed $w^{t+\Delta t}$. A Newton-Raphson's algorithm is used to solve the system of Eqs. (11) for each pair of shock points.

2.7 Shock Displacement

The enforcement of the jump relations described in Sect. 2.6 provides the speed at which each pair of grid-points located on the discontinuity moves along its local

normal vector. The position of the discontinuity at time $t + \Delta t$ is computed in a Lagrangian manner by displacing all its grid-points, as shown in Fig. 3f where the dashed and solid lines represent the discontinuity at time t and $t + \Delta t$, according to the following first-order accurate (in time) integration formula:

$$\mathbf{P}_i^{t+\Delta t} = \mathbf{P}_i^t + \mathbf{w}_i^t \Delta t, \quad (13)$$

which suffices for steady cases and returns the spatial coordinates of the i -th grid-point at time $t + \Delta t$. Herein, Δt is chosen such that during the time interval $[t, t + \Delta t]$ the shock will remain within the hole that it has dug in the background mesh. By doing so, none of the grid-points of the shock-fitting mesh will be overcome by the moving discontinuity, as shown in Fig. 3f.

2.8 Interpolation of the Phantom Nodes

Upon completion of the previous steps, all nodes of the shock-fitting mesh have been updated at time $t + \Delta t$. The shock-fitting mesh is made up of all the grid-points belonging to the fitted shock and all nodes of the background mesh, except those that have been declared *phantom*, which have not been updated yet. However, during the current time step, the discontinuity might have moved sufficiently far away from its previous position, so that some of the phantom nodes may re-appear in the shock-fitting mesh at the next time step. It follows that also the nodal values within the phantom nodes need to be updated to time $t + \Delta t$ and this is accomplished using a linear interpolation from the current shock-fitting mesh to the grid-points of the background one. Once the phantom nodes have been updated, the shock-fitting mesh used in the current time interval can be deleted.

At this stage the numerical solution has correctly been updated at time $t + \Delta t$ within all grid points of the background tessellation and within all pairs of grid-points that belong to the fitted discontinuities, taking into account the shock front displacement. The next time step can be computed by re-starting from the first step 2.2 of the algorithm.

3 COOLFluiD

COOLFluiD [18–20, 23] is a collaborative simulation platform for high-performance computing, where each model or algorithm is encapsulated into independent *plug-in*'s that can be loaded on demand by user-defined applications. COOLFluiD main features include parallel solvers (e.g., Finite Volume, Residual Distribution, various higher-order methods) for hypersonic flows [8, 15, 39], all-speed plasmas [30, 38, 50], magnetohydrodynamics [2, 24, 27, 49], and radiation transport (based on Monte Carlo algorithms [44]).

3.1 Residual Distribution Solver

A prototype system of Partial Differential Equations (PDEs) including advection-diffusion-reaction terms, which represents the most general case for our target applications, can be expressed in differential form as:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}^c = \nabla \cdot \mathbf{F}^d + \mathbf{S}, \quad (14)$$

where \mathbf{U} are the conservative variables, \mathbf{F}^c are the convective fluxes, \mathbf{F}^d are the diffusive fluxes, and \mathbf{S} is the source term. The presence of any of those terms and their definitions depend on the actual physical model to be solved. The COOLFluiD *Residual Distribution* (RD) solver has been used to solve the PDEs in Eq.(14) for the present work. RD is a vertex-centered method which conveniently inherits compact stencils and continuous linear representation of the solution from Finite Element Methods, combined with upwinding and conservative properties typical of Finite Volume (FV) algorithms. In addition, its multi-dimensional character makes the resulting method very competitive for compressible flow simulations, generally more accurate and grid-independent than FV solvers on unstructured grids.

The RD algorithm discretizes the integrated version of Eq.(14) in each mesh vertex by (1) computing a *residual* (or *fluctuation*) ϕ^e in each element/cell e , (2) distributing residual fractions to the element vertexes, and (3) assembling all residual contributions at each vertex j from distance-1 neighboring elements. Those algorithmic steps are schematically depicted in Fig. 4, namely on the left for steps (1) and (2), on the right for step (3).

When including all terms and integrating in space, the semi-discretized form of the governing equations (14) for a vertex j reads:

$$\frac{\partial \mathbf{U}}{\partial t} \Omega_j + \phi_j^c = \phi_j^d + \phi_j^s, \quad (15)$$

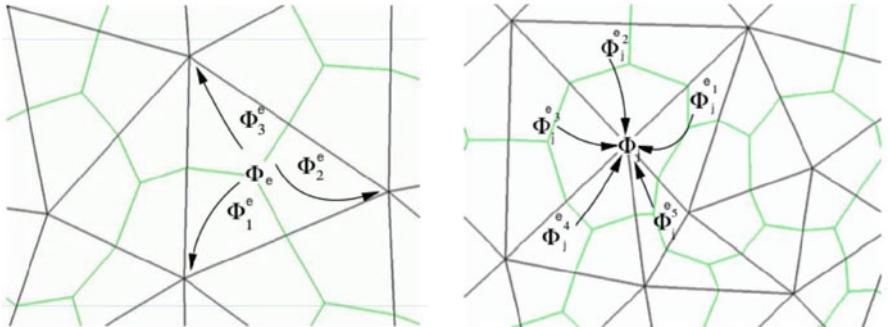


Fig. 4 Residual distribution concept: distribution of the residual for cell e to its vertexes i (left); assembly of all residual fractions from neighboring cells at each vertex j (right)

where Ω_j is the median dual cell (defined by green lines in Fig. 4), ϕ_j^c , ϕ_j^d , ϕ_j^s are fractions of the convective or diffusive fluxes and source terms which are assembled at j . In particular, convective fluxes are discretized using either the first-order N or the second-order B_x scheme [28] using Gnoffo's sensor [10]. The diffusive term is discretized with a Galerkin method. Finally, the source term, which is present in thermochemical nonequilibrium cases, is integrated with a one-point quadrature rule within a Petrov-Galerkin formulation and then distributed together with the convective flux [22]. Regarding time integration, either an explicit Forward Euler or implicit Backward Euler can be employed. In the latter case, the resulting linear system is solved using Generalized Minimum RESidual (GMRES) solvers in combination with a parallel Additive Schwarz preconditioner from the PETSc toolkit [3, 16].

A more detailed description of the COOLFluiD RD solver, which is used as a black-box by SF, lies outside the scope of the present work, but can be found in [17, 21, 22, 28, 29]. Therein, the interested reader can also find the corresponding equations for Euler and Navier-Stokes, including thermochemical nonequilibrium.

4 SF Code Design

The shock-fitting method presented so far, originally conceived within a procedural FORTRAN code, has been re-implemented and partially extended using modern object-oriented design techniques offered by the C++. To this end, the guiding principle has been to set up an algorithmic framework composed by independent dynamically-linked libraries, each encapsulating specific functionalities, and offering a core infrastructure potentially flexible to any future improvements or requirements. The architectural design of SF comprises two main parts, namely the *Framework* and the *Libraries* which are described in the following subsections.

4.1 Framework

The Framework defines the core abstract interfaces (APIs) to be implemented by the plugin libraries and some core functionalities, such as simulation configuration and control, mesh data structures, I/O. At the highest level, `ShockFittingAPI` defines C-style call-back functions allowing for interfacing SF with codes written in other languages (e.g., C, FORTRAN), `ShockFittingManager` takes care of the overall simulation configuration, while the `ShockFittingObject` is a facade that manages the creation, configuration, and interaction of the different algorithmic components within a simulation.

An overview diagram of the Framework is depicted in Fig. 5, where the columns (so-called *swimlanes*) represent the objects carrying out the general purposes. The small rectangles are the tasks assigned to the swimlane, or object, they belong to. These small rectangles come sequentially in succession by following the arrows: the process begins with the red oval-shaped *Start* and finishes with the red oval-shaped *End*. The user starts the simulation by calling the `TestStandardSF`, a simple list of commands needed to initiate the Framework components.

Globally each swimlane, and so each component, handles the following main commands: *create*, *configure*, *setup*, and *unsetup*. Through these commands each component creates and configures object parameters according to the user specifications inside the configuration file, allocates and deallocates component data, respectively. Another command, *process*, is used by some objects to make the following object operational.

By following the arrows from *Start* and passing through the several *setup* and *configure* rectangles, the *process StandardShockFitting* rectangle is reached in the third column. Here the shock-fitting algorithm of Sect. 2, fully encapsulated in the *StandardShockFitting* component (see Sect. 4.2.1 for details), which is a subclass of *ShockFittingObject*, is finally executed. Herein, the libraries are setup and then each object listed in the *LIBRARIES* rectangle accomplishes the task of the corresponding library (e.g., read, generate, remesh, convert). The set of operational libraries are cyclically called for a number of steps chosen by the user and sufficient to reach the convergence. After the iterative loop is terminated, all Framework objects are destroyed, the control returns to the user and the program ends.

4.2 Libraries

The Libraries provide implementation of the abstract APIs which are defined in the Framework. They are classified by topic: if one or more classes work to accomplish the same goal (e.g., re-meshing of the field, writing on output files, reading from input files), they are made to derive from a base class. The latter defines the library interface and assigns a name to it. Compared to the procedural language, it is as if the subroutines are first organized by accomplished tasks and then assigned to a library named as the achieving goal. The libraries can be expanded any time by adding new derived classes that must be consistent with the library they are going to belong to, in particular respecting its interface so that the dynamic interaction with the other SF components is preserved at the higher level.

A new library can be created (and registered) without being used. Moreover, the flexibility of SF allows the user to transparently choose which libraries must be executed through a configuration file, without needing to modify anything in the existing code (e.g., no `if - else` or file inclusions to be added anywhere). The set of libraries chosen on demand by the user determines the shock-fitting method to be executed.

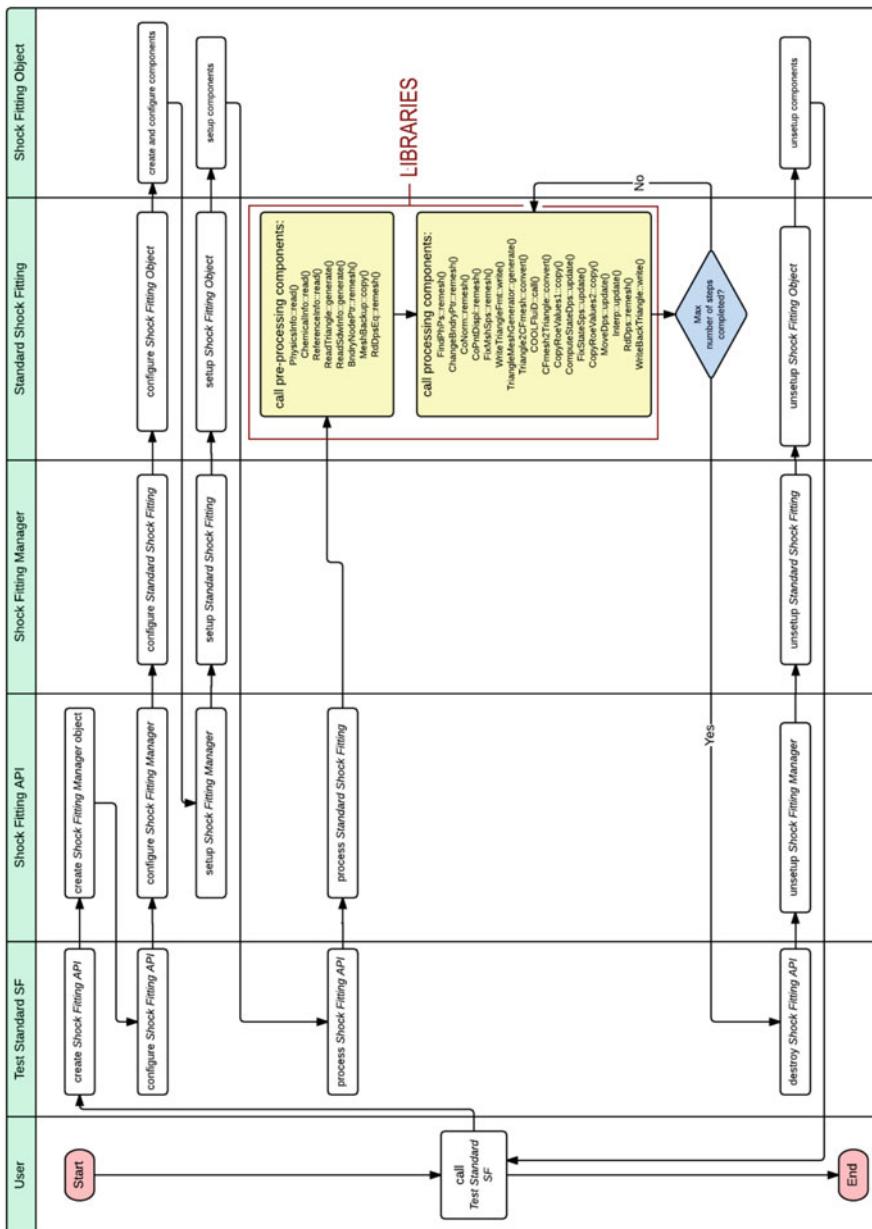


Fig. 5 Flow chart diagram showing the implementation of the SF solver

4.2.1 StandardShockFitting

The whole numerical method which has been described in Sect. 2 is encapsulated within StandardShockFitting, a subclass of ShockFittingObject (see Sect. 4.1). StandardShockFitting manages the interactions of the following entities (i.e., polymorphic objects), each one associated with a different library with the same name, grouping all its objects:

- MeshGeneratorSF defines the *generate()* function to build the mesh;
- RemeshingSF re-meshes the field through the *remesh()* function;
- ConverterSF adapts the data structure to a specified format using *convert()*;
- VariableTransformerSF converts variables via the *transform()* function;
- CFDsolverSF calls an arbitrary CFD solver via the *call()* function;
- CopymakerSF defines the *copy()* function to make copies of the mesh values;
- StateUpdaterSF corrects the CFD solution via the *update()* function;
- WritingMeshSF saves mesh data into output files via the *write()* function.

Each library includes all corresponding subclasses, each of which implements the main task (e.g., generate, remesh) in a customized way. By scanning the list above, one can identify the main steps of the shock-fitting algorithm presented in Sect. 2.

In order to examine more in depth the libraries concept, we focus our attention on the MeshGeneratorSF library, which is schematically depicted in Fig. 6. The mesh generator task is to collect all the information needed to build the mesh and generate it. It collects these information through its derived classes: ReadTriangle, ReSdwInfo, TriangleExe, and Tricall in our example.

The overall library is characterized by the *generate* method. Each member implements that method to fulfill its own goal: e.g., ReadTriangle generates the mesh field after reading data from the format required by TriangleMeshGenerator, ReSdwInfo inserts the discontinuity in the mesh field reading a file with the infor-

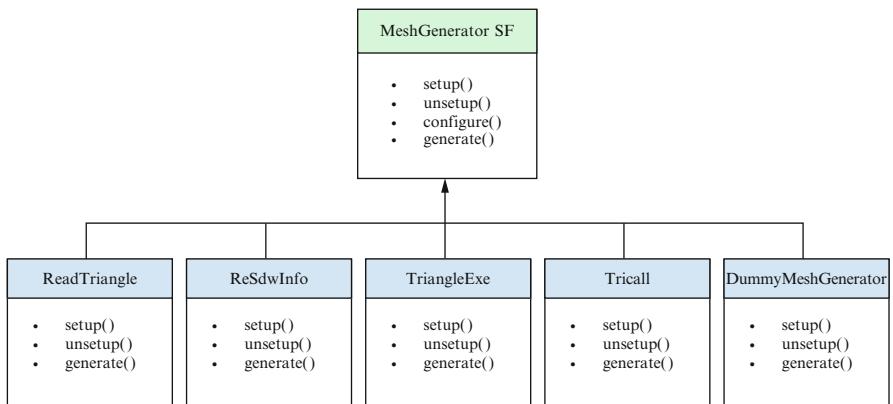


Fig. 6 MeshGeneratorSF library. The base object and its derived classes

mation concerning the shock and so on. In Fig. 6, also a `DummyMeshGenerator` member appears. Its purpose is to provide a default implementation, doing basically nothing. Following this well-defined design, new mesh generators (e.g., interfacing third party libraries or adhering to special formats) can easily be created and plugged into the SF framework. A flow chart shows implementation details of Standard-ShockFitting in Fig. 7. Herein, different swimlanes represent objects in charge of different operations, which are symbolized by small rectangles. Inside each rectangle, the action of the corresponding object is summarized. The operations, coming in succession within the diagram, trace the steps of the overall algorithm presented in Sect. 2, including some I/O operations and variables transformations which are not mentioned in the latter. In order to assist the reader to visually assign each object of Fig. 7 to the corresponding library it belongs to, a legend for the libraries colors is provided in Fig. 8.

4.3 The Code Configuration

All SF framework and library components are *self-configurable*, i.e. they can identify and set up by themselves their own parameters after parsing a configuration file. This concept was originally implemented in the `yagol` library [31], then further developed within COOLFluiD and `SConfig`, the library which is used by SF. The implementation of the self-configuration technique is described in [42].

Furthermore, SF components are also *self-registering*, meaning that an automatic object database (residing in static memory) is updated on-the-fly whenever a new component implementing a (abstract) framework API is made available via object providers. The self-registration [4] enables the creation of polymorphic objects by name and type via the abstract Factory pattern [9]. When setting up an SF simulation, Factories can be used to transparently create objects after querying the corresponding archive for that specific type of objects, therefore avoiding the need for using tedious and hardly maintainable `if-else` constructs (typical of procedural codes) for selecting which algorithm/model to instantiate. Implementation details of this powerful (but quite unused in scientific computing) technique, partially differing from the approach proposed in [4], are provided in [17, 20].

4.3.1 The Configuration File Format

The input file for the SF code, named `input.case`, fully reflects the flexibility of its design. Thanks to the above-mentioned self-configuration and self-registration techniques, it actually allows the end-user to control the instantiation of models and algorithms, more than just setting up their parameters. Each line of the configuration file has the form `.Key = Value`, in which `Key` represents an object or parameter while `Value` is the quantity assigned to key, e.g. an object name (i.e.

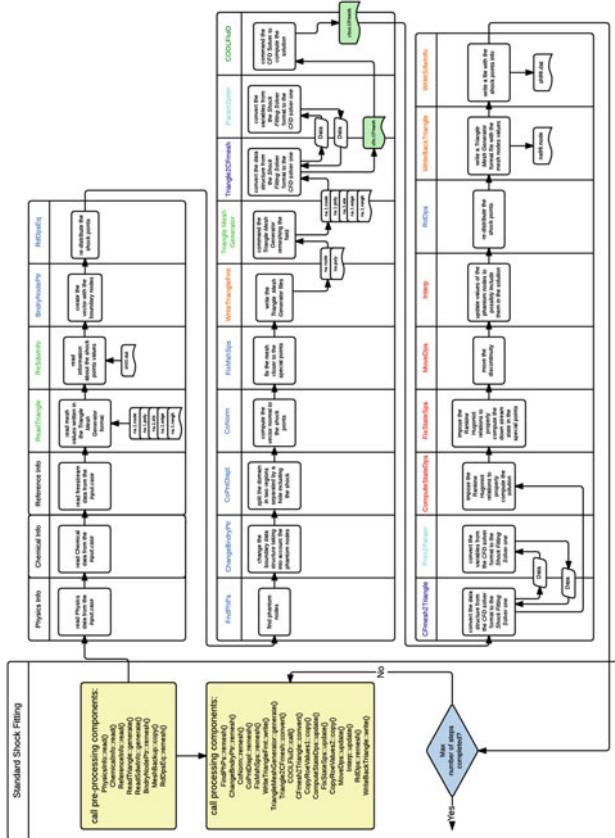


Fig. 7 Flow chart diagram showing the implementation of StandardShockFitting

Fig. 8 Table of the SF libraries colors which are used in Fig. 7

<i>CFD Solver SF</i>	
<i>Converter SF</i>	
<i>MeshGenerator SF</i>	
<i>Remeshing SF</i>	
<i>StateUpdater SF</i>	
<i>Variable Transformer SF</i>	
<i>WritingMesh SF</i>	

Listing 1 Example of the configuration settings for the MeshGeneratorSF.

```
. StandardShockFitting . MeshGeneratorList = ReadTriangle ReSdwInfo
. StandardShockFitting . MeshGeneratorList . ReadTriangle . InputFile = \
    na00 . node na00 . poly na00 . ele
. StandardShockFitting . ReSdwInfo . InputFile = sh00 . dat
```

its self-registration key), an alpha-numeric string, a numerical value, a boolean, an arbitrarily complex analytic function or an array of any of the previous. When objects/parameters to be configured belong to other objects, their corresponding key shows the object nesting explicitly, in a hierarchical way. In this case, the object/parameter to be configured is the last one appearing in the key on the left of the “=” operator and the value on the right will be assigned to it.

As a clarifying example, let us consider the configuration of the MeshGeneratorSF library of Fig. 6, as shown in Code Listing 1. Herein, ReadTriangle and ReSdwInfo are the names of the two MeshGeneratorSF objects to be instantiated via Factories, as set in the first line, while their corresponding input files are specified as nested parameters in the following lines.

4.4 Coupling to the CFD Solver

The coupling between SF and a generic CFD solver is accomplished via system calls (i.e., to setup and launch flow simulations) and I/O which are managed by concrete implementations of the CFDSolverSF wrapper (see Sect. 4.2.1). This allows each CFD solver to be treated as a black-box, without requiring any code modification. When coupling SF to COOLFluiD, both codes can read/write two file formats: CFmesh (i.e., COOLFluiD native format) or TECPLOT, both point-based, unstructured and ASCII. Despite the current coupling strategy, SF is designed to support call-back functions and, in the future, data will be transferred more efficiently directly through memory, as long as CFD solvers provide a suitable interface for this.

5 Numerical Results

Three representative cylinder cases are chosen to demonstrate the current capabilities of the SF code in combination with the COOLFluiD RD solver: (1) inviscid perfect gas at $M_\infty = 20$, (2) viscous perfect gas at $M_\infty = 17$, and (3) inviscid nitrogen flow in thermochemical nonequilibrium at $M_\infty = 6.13$. COOLFluiD was run using explicit time stepping for case (1) and implicit time stepping for (2) and (3).

5.1 Case 1: Perfect Gas Inviscid Flow Over Cylinder ($M_\infty = 20$)

The corresponding free-stream conditions for this case are listed in Table 1. The computational domain consists of an elliptic sector identified by a horizontal dimension of 2.5 m and a vertical dimension of 4.25 m. Slip wall, supersonic inlet and outlet conditions are imposed. No condition is imposed on the post-shock internal boundary resulting from the fitting. The shock-capturing (SC) solution is computed on a mesh of 9,097 triangles and 4,723 nodes which is shown in Fig. 9a. The same mesh is used as the initial background mesh for the SF code. The first *shocked* mesh resulting from remeshing is characterized by a slightly larger number of triangles (9,186) and nodes (4,851), as depicted in Fig. 9b. The first-order-

Table 1 Free stream conditions for Case 1

M_∞	$p_\infty [Pa]$	$T_\infty [K]$
20	10	273

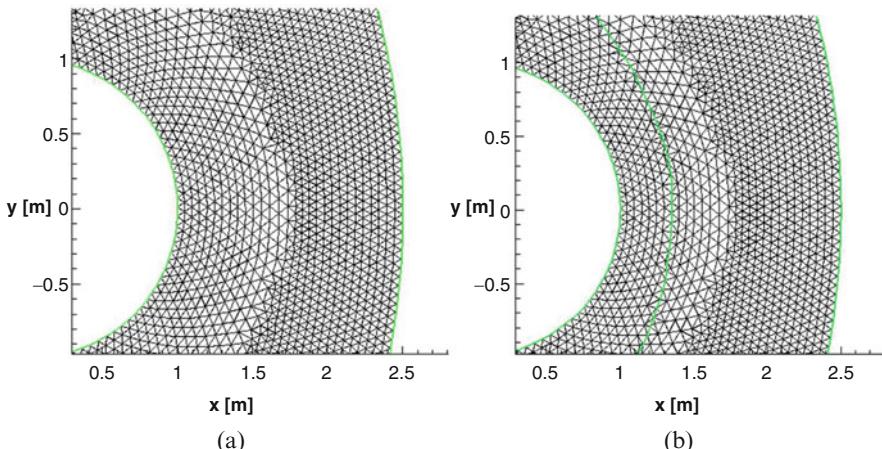


Fig. 9 Computational meshes for Case 1. (a) Initial mesh. (b). Shocked mesh after fitting

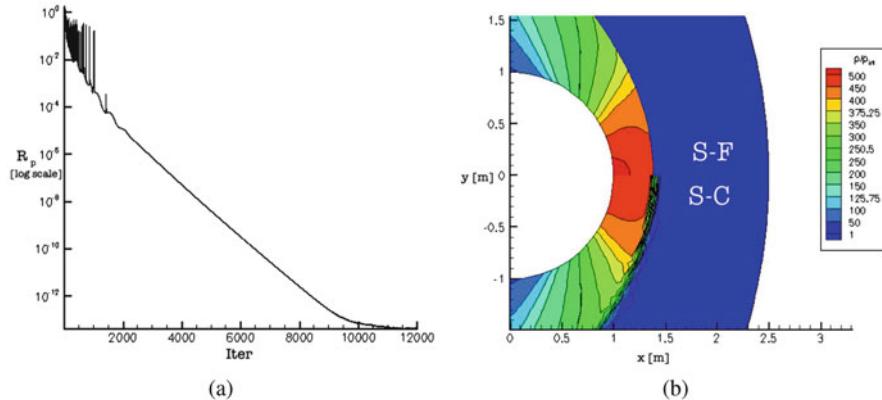


Fig. 10 Case 1: convergence and shock-capturing versus shock-fitting adimensional pressure field. (a) Pressure residual. (b) Zoomed view of the pressure field

accurate N scheme is used for both the SC and the SF solution. In Fig. 10a the trend of the pressure residual versus the number of steps is illustrated, showing a typical convergence history of a coupled SF/COOLFluiD simulation. The residual is computed on the grid-points of the background mesh through an $L2$ norm. During the first time steps, the shock moves and the connectivity of the overall mesh is modified, inducing large residual peaks. Once the shock position stabilizes, the mesh connectivity freezes and the residual smoothly decreases until the round-off error value. The same does not happen for the shock points, whose neighboring cells are always modified even for minimal changes in the shock position. For that reason the shock points are not included in the calculation of the residual. When comparing SC and SF solutions in Fig. 10b, the latter is characterized by a zero-thickness shock, while the former shows a smeared shock which is captured in 2–3 cells (this is also aggravated by the usage of a first-order scheme). The SF solution is compared with the one calculated by Lyubimov and Rusanov in the 70s. They computed mixed flows past non-slender bodies by a finite-difference method and listed the numerical solutions in tables [26]. Our results are compared against theirs, in terms of adimensional body surface pressure values in Fig. 11a and pressure isolines in the full domain in Fig. 11b, showing a good agreement.

5.2 Case 2: Perfect Gas Viscous Flow over Cylinder ($M_\infty = 17$)

This case corresponds to a benchmark proposed by Peter Gnoffo in [11], with conditions listed in Table 2. It belongs to a collection of tests made by the HEFSS NASA team to verify the unstructured FUN3D solver against the legacy structured

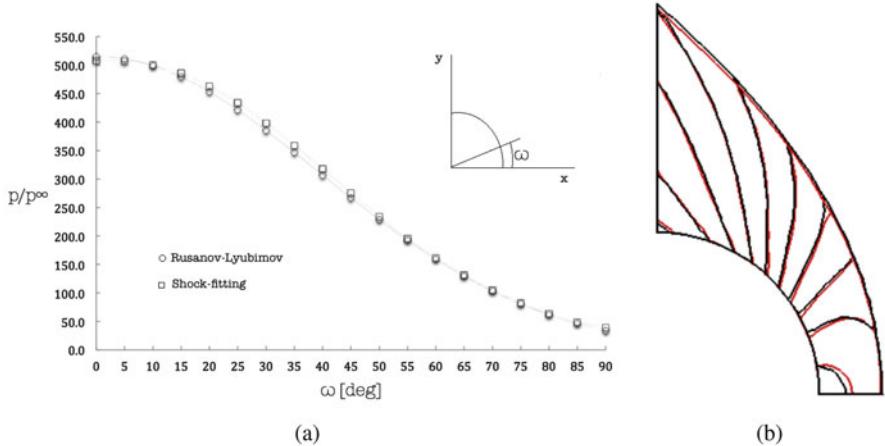


Fig. 11 Case 1: comparison of the adimensional pressure field on the surface and inside the domain (— [26], — SF). (a) Body surface pressure. (b) Isoline pressure field

Table 2 Free stream conditions for Case 2 from [11]

M_{∞}	$p_{\infty} [\text{Pa}]$	$T_{\infty} [\text{K}]$	$T_w [\text{K}]$
17	57.65	200	500

LAURA code. The computational domain around the cylinder with radius of 1 m is an elliptic sector identified by a horizontal dimension of 1.59 m and a vertical dimension of 2.62 m. No-slip isothermal wall at fixed temperature, supersonic inlet and outlet are the prescribed boundary conditions. The mesh includes 13,640 triangles, 6,993 nodes and was obtained by splitting an initial structured mesh. The initial mesh and the resulting *shocked* mesh (characterized by 13,826 triangles and 7,181 nodes) are shown in Fig. 12a and b, respectively.

The CFD solution is computed with a second-order B_x scheme [28] using Gnoffo's shock sensor [10] as blending coefficient. The convergence history in terms of pressure residual is shown in Fig. 13a, while the adimensional temperature field for the SC and SF solutions can be seen in Fig. 13b, demonstrating the undeniably higher accuracy of the SF solution. The SC and SF solutions have also been compared against the reference one from the NASA LAURA solver. In Fig. 14 the heat flux and the pressure coefficient are shown. An excellent agreement is found between the heating values of SF and LAURA outside of the stagnation region, while the SC solution shows a shifted trend. The stagnation heat flux is overpredicted by both SC and SF if compared to LAURA results, but SF does not show the cusp given by the SC code. The pressure coefficient perfectly matches LAURA solution.

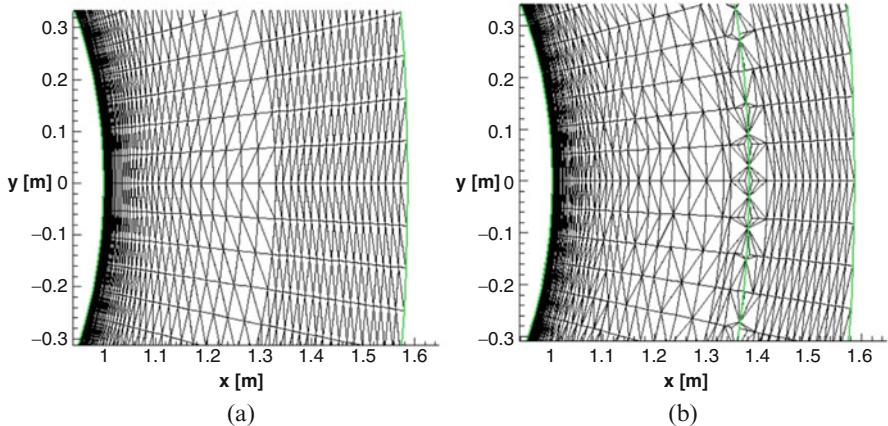


Fig. 12 Computational meshes for Case 2. (a) Initial mesh. (b) Shocked mesh after fitting

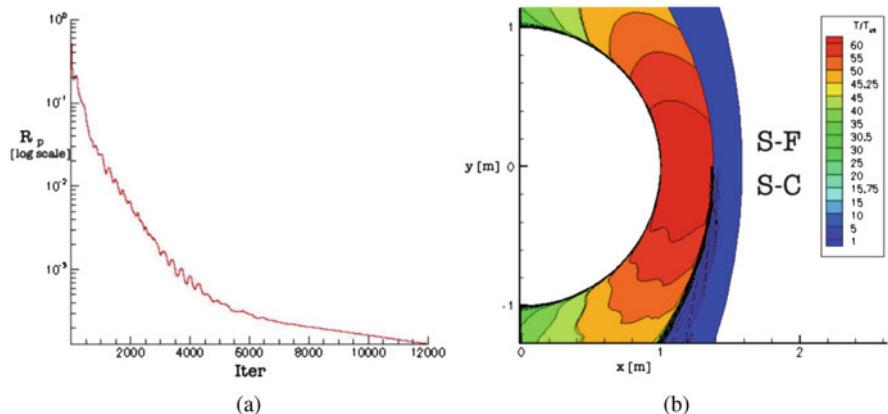


Fig. 13 Case 2: convergence and shock-capturing versus shock-fitting adimensional pressure field. **(a)** Pressure residual. **(b)** Zoomed view of the temperature field

5.3 Case 3: Inviscid Flow in Thermochemical Nonequilibrium Over Cylinder ($M_\infty = 6.13$)

This last case is based upon conditions (see Table 3) corresponding to an experiment in a nonequilibrium dissociating nitrogen flow carried out by Hornung [12]. The flow is here assumed to be inviscid and the cylinder has a radius of 0.0127 m. Park's thermochemical model which is used to describe the nitrogen dissociation is described in [36, 40] and full details about the corresponding COOLFluiD RD solver can be found in [22]. The computational domain is an elliptic sector identified by a horizontal dimension of 0.025 m and a vertical dimension of 0.05 m. Slip wall, supersonic inlet and outlet are imposed as boundary conditions. The SC solution

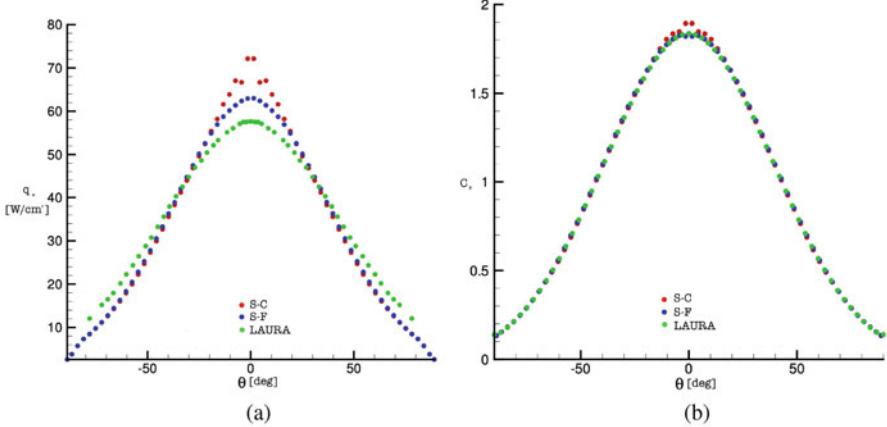


Fig. 14 Case 2: comparison of heat and pressure coefficient distributions for SC, SF, and NASA LAURA solvers. **(a)** Heating distribution. **(b)** Pressure coefficient

Table 3 Free stream conditions for Case 3 from [12]

M_∞	$p_\infty [\text{Pa}]$	$T_\infty [\text{K}]$	$T_{v\infty} [\text{K}]$	$u_\infty [\text{m/s}]$	$\alpha_{N\infty}$
6.13	2908	1833	1833	5594	0.07

is computed on a mesh of 29,282 triangles and 14,884 nodes, while the *shocked* mesh includes 29,422 triangles and 15,065 nodes, as shown in Fig. 15a and b, respectively. The solution provided by the SC code is computed through a first-order-accurate N_c scheme, while a second-order-accurate B_{xc} scheme [22, 28] is used for the fitted solution.¹ The SF solution is shown in terms of mass fractions of molecular nitrogen ($\alpha_{N_2} = \rho_{N_2}/\rho$) in Fig. 16a. The nitrogen dissociation affects the overall shock layer. Near the stagnation point, as long as the flow characteristic time tends to ∞ , the equilibrium should be locally reached. Nevertheless, near the shock region, the flow is far away from the equilibrium conditions and the molecular nitrogen concentration is carried away by the streamlines that follow the body. Along the streamlines that circumvent the stagnation point and go toward the expansion region, the flow is characterized by a decrease in temperature and pressure values. The dissociation processes are, therefore, frozen. This is clear in Fig. 16a, where, little away from the circumvention of the stagnation point, the flow moves forward without additional changes in the chemistry concentrations.

The SF solution is also compared directly against the experimental measurements from [12] in Fig. 16b, where the computed fringe pattern is superimposed onto Hornung's. The fringe shift number has been calculated through the expression provided in [12]:

¹The subscript c denotes the Contour Residual Distribution, aka *CRD*, variant of schemes.

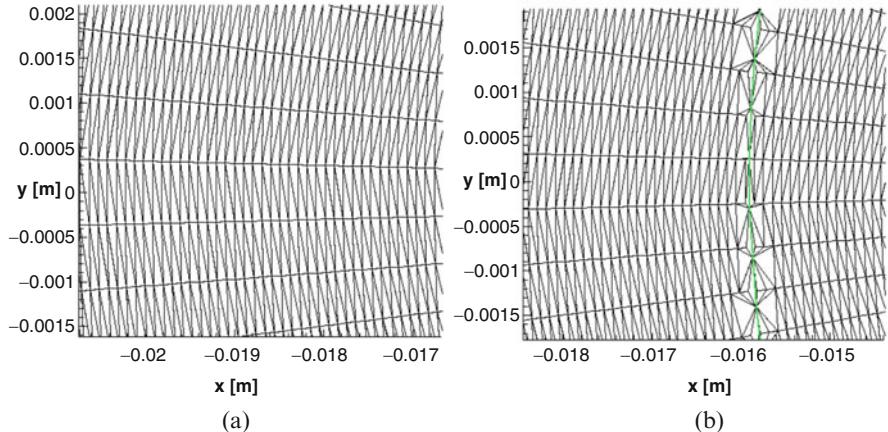


Fig. 15 Computational meshes for Case 3. (a) Initial mesh. (b) Shocked mesh after fitting

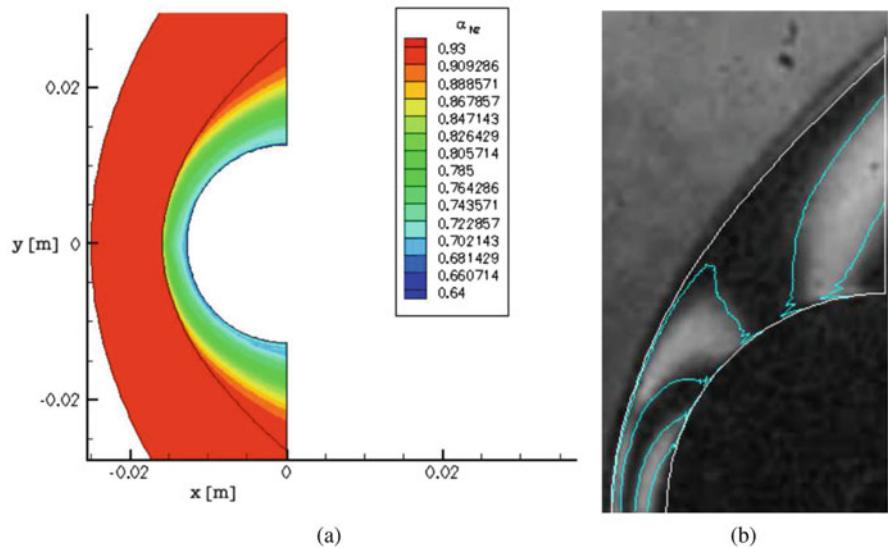


Fig. 16 Case 3: N_2 mass fractions and fringe patterns. (a) Contours of N_2 mass fractions. (b) Experimental [12] versus computed interferograms

$$\Delta\rho = \frac{4 \cdot 16 F \lambda}{L(1 + 0.28 \alpha)} \text{ g/cm}^3 \quad (16)$$

with $\Delta\rho$ being the density change set equal to $\rho - \rho_\infty$, $\alpha = \rho_N/\rho$ is the nitrogen dissociation fraction, $L = 0.1524$ is the geometrical path, and λ is the wavelength equal to 4300A. Figure 16b shows a slight mismatch in the shock layer thickness.

This can be explained by the usage of the inviscid model which, due to the absence of the boundary layer, tends to underpredict the shock standoff distance.

6 Conclusions

This chapter has presented an overview of the capabilities of SF, the first open source shock-fitting framework for unstructured grids. The code is designed for maximum flexibility, therefore relying upon modern object-oriented technology. Each component, in particular each algorithmic step, is encapsulated into a dynamic plugin which implements a well-defined interface. A concise description of the whole shock-fitting method which has been implemented has been provided. A selection of 2D numerical results have demonstrated that SF coupled to a CFD solver (COOLFluiD RD in this case) can provide improved solutions if compared to state-of-the-art shock-capturing solvers for hypersonic flows around blunt bodies, in both inviscid and viscous cases. Results with flows in thermochemical nonequilibrium are also encouraging. Ongoing work is focusing on extending the SF code in order to handle complex 3D geometries. To this end, the coupling to advanced open source tools such as CGAL [47] for surface reconstruction and Tetgen [46] for volume remeshing is needed.

References

1. Agin, G.J.: Fitting ellipses and general second-order curves. Technical report 8–9, The Robotics Institute, Carnegie-Mellon University, Pittsburgh (1981)
2. Alvarez Laguna, A., Lani, A., Deconinck, H., Mansour, N.N., Poedts, S.: A fully-implicit finite volume method for multi-fluid reactive and collisional magnetized plasmas on unstructured meshes. *J. Comput. Phys.* **318**(1), 252–276 (2016)
3. Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F.: Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries, chap. 8, pp. 163–202. Birkhäuser, Basel (1997)
4. Beveridge, J.: Self-registering objects in C++. *Dr. Dobb's J.* **288**, 38–41 (1998)
5. Blazek, J.: Computational Fluid Dynamics: Principles and Applications, 2nd edn. Elsevier Science, Amsterdam (2005)
6. Bonfiglioli, A., Grottadurea, M., Paciorri, R., Sabetta, F.: An unstructured, three-dimensional, shock-fitting solver for hypersonic flows. *Comput. Fluids* **73**, 162–174 (2013). <https://doi.org/10.1016/j.compfluid.2012.12.022>
7. Bonfiglioli, A., Paciorri, R., Campoli, L.: Unsteady shock-fitting for unstructured grids. *Int. J. Numer. Methods Fluids* **81**(4), 245–261 (2016). <https://doi.org/10.1002/fld.4183>. Fld.4183
8. Degrez, G., Lani, A., Panesi, M., Chazot, O., Deconinck, H.: Modelling of high-enthalpy, high-Mach number flows. *J. Phys. D Appl. Phys.* **42**(19), 194004 (2009). <http://stacks.iop.org/0022-3727/42/i=19/a=194004>
9. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series. Addison-Wesley, Reading (1994)
10. Gnoffo, P.A.: Updates to multi-dimensional flux reconstruction for hypersonic simulations on tetrahedral grids. 48th AIAA Aerospace Sciences Meeting, Orlando, FL (2010)

11. Gnoffo, P.A., White, J.A.: Computational aerothermodynamic simulation issues on unstructured grids. In: 37th AIAA ThermoPhysics Conference, Portland, OR (2004)
12. Hornung, H.G.: Non-equilibrium dissociating nitrogen flow over spheres and circular cylinders. *J Fluid Mech.* **53**, Part I, 149–176 (1972)
13. Ivanov, M.S., Bonfiglioli, A., Paciorri, R., Sabetta, F.: Computation of weak steady shock reflections by means of an unstructured shock-fitting solver. *Shock Waves* **20**(4), 271–284 (2010). <https://doi.org/10.1007/s00193-010-0266-y>
14. Kanamori, M., Suzuki, K.: Shock wave detection based on the theory of characteristics for CFD results. AIAA-2011-3681 (2011)
15. Knight, D., Longo, J., Drikakis, D., Gaitonde, D., et al.: Assessment of CFD capability for prediction of hypersonic shock interactions. *Prog. Aerosp. Sci.* **48–49**, 8–26 (2012)
16. Laboratory, A.N.: PETSc: portable, extensible toolkit for scientific computation (2007). [Http://www-unix.mcs.anl.gov/petsc](http://www-unix.mcs.anl.gov/petsc)
17. Lani, A.: An object oriented and high performance platform for aerothermodynamics simulation. Ph.D. thesis, Université Libre de Bruxelles (2008)
18. Lani, A., et al.: COOLFluiD Wiki page. <https://github.com/andrealani/COOLFluiD/wiki> (2016)
19. Lani, A., Quintino, T., Kimpe, D., Deconinck, H., Vandewalle, S., Poedts, S.: The COOLFluiD framework: design solutions for high-performance object oriented scientific computing software. In: Sunderan, V.S., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) Computational Science ICCS 2005. Lecture Notes in Computer Science 3514, vol. 1, pp. 281–286. Emory University, Springer, Atlanta (2005)
20. Lani, A., Quintino, T., Kimpe, D., Deconinck, H., Vandewalle, S., Poedts, S.: Reusable object-oriented solutions for numerical simulation of PDEs in a high performance environment. *Sci. Program. (Special Edition on POOSC 2005)* **14**(2), 111–139 (2006)
21. Lani, A., Mena, J.G., Deconick, H.: A residual distribution method for symmetrized systems in thermochemical nonequilibrium. In: 20th AIAA CFD Conference Honolulu, HI. AIAA-2011-3546 (2011)
22. Lani, A., Panesi, M., Deconinck, H.: Conservative residual distribution method for viscous double cone flows in thermochemical nonequilibrium. *Commun. Comput. Phys.* **13**, 479–501 (2013)
23. Lani, A., Villedieu, N., Bensassi, K., Kapa, L., Vymazal, M., Yalim, M.S., Panesi, M.: COOLFluiD: an open computational platform for multi-physics simulation and research. In: 21th AIAA CFD Conference, San Diego, CA. AIAA 2013-2589 (2013)
24. Lani, A., Yalim, M.S., Poedts, S.: A GPU-enabled finite volume solver for global magnetospheric simulations on unstructured grids. *Comput. Phys. Commun.* **185**(10), 2538–2557 (2014)
25. Lovely, D., Haimes, R.: Shock detection from computational fluid dynamics results. AIAA-99-3285 (1999)
26. Lyubimov, A.N., Rusanov, V.V.: Gas Flows Past Blunt Bodies, Volume II: Calculation Method and Flow Analysis. Nauka Press, Moscow (1970). NASA Technical Translation, F-714
27. Maneva, Y., Laguna, A.A., Lani, A., Poedts, S.: Multifluid modeling of magnetosonic wave propagation in the solar chromosphere - effects of impact ionization and radiative recombination. *Astrophys. J.* (2017). <https://arxiv.org/abs/1611.08439>
28. Mena, J.G., Pepe, R., Lani, A., Deconinck, H.: Assessment of heat flux prediction capabilities of residual distribution method: application to atmospheric entry problems. *Commun. Comput. Phys.* **17**(3), 682–702 (2015)
29. Mena, J.G., Lani, A., Deconinck, H.: An energy-dissipative remedy against carbuncle: application to hypersonic flows around blunt bodies. *Comput. Fluids* **133**, 43–54 (2016)
30. Munafò, A., Lani, A., Bultel, A., Panesi, M.: Modeling of non-equilibrium phenomena in expanding flows by means of a collisional-radiative model. *Phys. Plasma* **20**(7), 073501 (2013). <http://dx.doi.org/10.1063/1.4810787>
31. Pace, J.: Another getopt library. <http://yagol.sourceforge.net> (2003)
32. Paciorri, R., Bonfiglioli, A.: A shock-fitting technique for 2d unstructured grids. *Comput. Fluids* **38**(3), 715–726 (2009). <https://doi.org/10.1016/j.compfluid.2008.07.007>

33. Paciorri, R., Bonfiglioli, A.: Detection of shock waves from shock-capturing solutions. IMACS/ISGG Workshop, IAC-CNR, Roma, Italy (2011)
34. Paciorri, R., Bonfiglioli, A.: Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique. *J. Comput. Phys.* **230**(8), 3155–3177 (2011). <https://doi.org/10.1016/j.jcp.2011.01.018>
35. Paciorri, R., Bonfiglioli, A.: Recognition of shock-wave patterns from shock-capturing solutions. In: Di Giambardino, P., et al. (eds.) Computational Modeling of Objects Represented in Images. Fundamentals, Methods and Applications III. CRC Press, Taylor & Francis Group, London (2012)
36. Paciorri, R., Sabetta, F., Favini, B.: On the role of vibrational excitation in hypersonics flow computations. *Meccanica* **33**, 331–347 (1998)
37. Pagendarm, H., Seitz, B.: An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves. In: Palamidese, P. (ed.) Scientific Visualization: Advanced Software Techniques. Prentice Hall India, Delhi (1993)
38. Panesi, M., Lani, A.: Collisional radiative coarse-grain model for ionization in air. *Phys. Fluids* **25**, 057101 (2013)
39. Panesi, M., Lani, A., Magin, T., Pinna, F., Chazot, O., Deconinck, H.: Numerical investigation of the non equilibrium shock-layer around the expert vehicle. In: 38th AIAA Plasmadynamics and Lasers Conference, Miami, FL. AIAA Paper 2007-4317 (2007)
40. Park, C.: Assessment of a two-temperature kinetic model for dissociating and weakly ionizing nitrogen. *J. Thermophys.* **2**(1), 8–16 (1988)
41. Pepe, R., Bonfiglioli, A., Paciorri, R., Lani, R., Garicano Mena, J., Olliver-Gooch, C.F.: Towards a modular approach for unstructured shock-fitting. WCCM XI, ECCM V, ECFD VI (2014)
42. Quintino, T.: A component environment for high-performance scientific computing. Design and implementation. Ph.D. thesis, Katholieke Universiteit Leuven (2008)
43. Salas, M.: A Shock-Fitting Primer, 1st edn. CRC Applied Mathematics & Nonlinear Science. Chapman & Hall, Boca Raton (2009)
44. Santos, P.D., Lani, A.: An object-oriented implementation of a parallel Monte Carlo code for radiation transport. *Comput. Phys. Commun.* **202**, 233–261 (2016)
45. Shewchuk, J.R.: Triangle: engineering a 2D quality mesh generator and delaunay triangulator. In: Lin, M.C., Manocha, D. (eds.) Applied Computational Geometry: Towards Geometric Engineering. Lecture Notes in Computer Science, vol. 1148, pp. 203–222. Springer, Berlin (1996). From the First ACM Workshop on Applied Computational Geometry
46. Si, H.: TETGEN: a quality tetrahedral mesh generator and a 3d delaunay triangulator. <http://wias-berlin.de/software/tetgen/> (2016)
47. The Computational Geometry Algorithms Library. <http://www.cgal.org> (2016)
48. Wu, Z., Xu, Y., Wang, W., Hu, R.: Review of shock wave detection method in CFD post-processing. *Chin. J. Aeronaut.* **26**, 501–513 (2013)
49. Yalim, M.S., Abeele, D.V., Lani, A., Quintino, T., Deconinck, H.: A finite volume implicit time integration method for solving the equations of ideal magnetohydrodynamics for the hyperbolic divergence cleaning approach. *J. Comput. Phys.* **230**(15), 6136–6154 (2011)
50. Zhang, W., Lani, A., Panesi, M.: Analysis of non-equilibrium phenomena in inductively coupled plasma generators. *Phys. Plasma* **23**(7), 073512 (2016)

Shock-Fitting and Predictor-Corrector Explicit ALE Residual Distribution

L. Campoli, P. Quemar, A. Bonfiglioli, and M. Ricchiuto

1 Introduction and Main Results

This work is part of a long-term effort to develop a modern unstructured shock-fitting algorithm. The authors have so far shown the capabilities of their approach for steady flows in two [12, 13] and three [6] spatial dimensions, as well as some potential for time dependent moving shocks [7]. In all the mentioned works fitted shocks are treated as interior boundaries of zero thickness that are free to move throughout a triangular/tetrahedral mesh. This underlying unstructured triangulation/tetrahedrization covers the entire computational domain, and locally adapts to exactly fit the shock motion. The latter is ruled by the Rankine–Hugoniot jump relations which are in turn deduced from an approximate flow solution obtained by discretizing the compressible flow equations on either side of the discontinuity. To this end, in this work we will focus on the numerical solution of the perfect gas Euler equations. This contribution is a first step to evaluate the unsteady

L. Campoli

Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza University of Rome, Rome, Italy
e-mail: lorenzo.campoli@uniroma1.it

P. Quemar

LAGA, Université Paris 13 et LNHE, EDF R&D, Paris, France
e-mail: quemar@math.univ-paris13.fr

A. Bonfiglioli

Scuola di Ingegneria, Università degli Studi della Basilicata, Potenza, Italy
e-mail: aldo.bonfiglioli@unibas.it

M. Ricchiuto (✉)

Team CARDAMOM, Inria Bordeaux Sud-Ouest, Talence, France
e-mail: mario.ricchiuto@inria.fr

fitting method when schemes with different properties are used. We evaluate the results obtained with explicit residual distribution schemes in Arbitrary Lagrangian Eulerian (ALE) form, developed in [5, 16].

The paper investigates quantitatively and qualitatively the influence on the fitted solutions of the accuracy order (first or second here) of the method, as well as on the stabilization mechanism used. The main result is that accuracy seems to be the most influent factor.

2 Numerical Methods

2.1 Shock-Fitting Algorithm

The unstructured shock-fitting algorithm consists of two key ingredients: (1) a local re-meshing technique that constructs a time-dependent mesh in which the fitted discontinuities are internal boundaries of zero thickness and (2) an algorithm for solving the Rankine–Hugoniot jump relations that provide the Lagrangian velocity of the discontinuity and an updated set of dependent variables within the downstream side of the fitted shock.

More precisely, in two space dimensions, the fitted shock fronts are made of polygonal curves, i.e., a connected series of line segments (which we call the shock edges) that join the shock points. These shocks are free to move throughout a background triangular mesh that covers the entire computational domain (see Fig. 1a). Starting from a background grid, at a given time level n local mesh modifications are performed by re-generating a constrained Delaunay triangulation in the neighbourhood of the shock. This allows to ensure that the edges that make up the shock front are also part of the mesh that covers the entire computational domain (see Fig. 1b). The shock speed σ_{sh} is computed from the Rankine–Hugoniot jump relations within each of the shock points. Using nodal values of shock speed and shock normal \mathbf{n} , the fitted shock front is moved in a Lagrangian manner, as shown

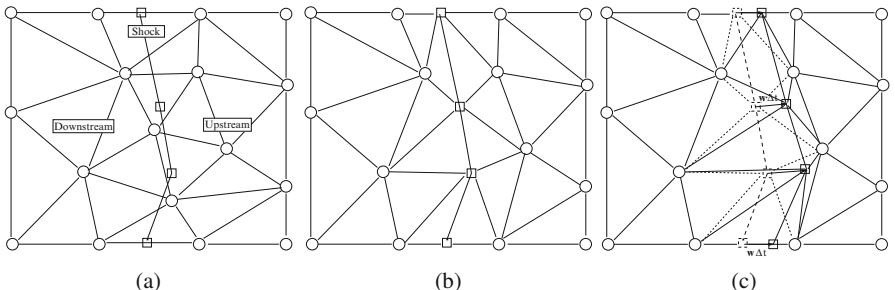


Fig. 1 Shock-fitting: schematic illustration of some of the algorithmic ingredients. (a) Background mesh. (b) “Shock-fitted” mesh. (c) Shock-motion + deformation

in Fig. 1c. A second-order-accurate temporal integration of the shock trajectory is obtained using a predictor-corrector scheme [7].

In the smooth regions on either side of the shock, a vertex-centred ALE discretization, described in Sect. 2.2, allows to obtain an approximation of the flow variables on the mesh. The ALE formulation allows to embed naturally in the schemes the presence of the shock front as a moving boundary. This is schematically shown in Fig. 1c where dashed lines are used to show the edges of the triangular cells when the shock front is at time level n and solid lines to show the same sides when the shock has reached time level $n + 1$.

The interested reader can refer to [7] for additional details on the unsteady fitting procedure.

2.2 Predictor-Corrector ALE Residual Distribution

We consider discretization methods for the Euler equations recast in the following ALE compact form:

$$\partial_t(J\mathbf{w}) + J\nabla \cdot (\mathbf{f}(\mathbf{w}) - \sigma\mathbf{w}) = 0 \quad (1)$$

with \mathbf{w} the array of conserved quantities (mass, momentum and energy), \mathbf{f} the conservative fluxes, J the determinant of the Jacobian of the transformation between the reference and actual frame and σ the local deformation velocity. To discretize (1), we consider the two-step explicit Residual Distribution (RD) method developed in [4, 15, 16], which can be written in the compact discrete form

$$|C_i^{n+1}| \mathbf{w}_i^{n+1} = |C_i^{n+1}| \mathbf{w}_i^* - \Delta t \sum_{K \in \mathcal{D}_i} \Phi_i^K(\mathbf{w}_h^n, \mathbf{w}_h^*) \quad (2)$$

where the fluctuations Φ_i^K define a splitting of the average element residual Φ^K defined by

$$\sum_{j \in K} \Phi_j^K = \Phi^K = \frac{1}{\Delta t} \left(\int_{K^{n+1}} \mathbf{w}_h^* - \int_{K^n} \mathbf{w}_h^n \right) + \frac{1}{2} \Phi^K(\mathbf{w}_h^n) + \frac{1}{2} \Phi^K(\mathbf{w}_h^*) \quad (3)$$

with

$$\sum_{j \in K} \Phi_j^K = \Phi^K = \int_{\partial K^{n+1/2}} (\mathbf{f}(\mathbf{w}_h) - \sigma_h \mathbf{w}_h) \cdot \mathbf{n} ds \quad (4)$$

The vector σ_h represents the mesh velocity with nodal values $\sigma_i = (\mathbf{x}_i^{n+1} - \mathbf{x}_i^n)/\Delta t$. The values \mathbf{w}_i^* are obtained from a first order predictor which is computed as

$$|C_i^{n+1}| \mathbf{w}_i^* = |C_i^{n+1}| \mathbf{w}_i^n - \Delta t \sum_{K \in \mathcal{D}_i} \tilde{\Phi}_i^K(\mathbf{w}_h^n) \quad (5)$$

where now the fluctuations $\tilde{\Phi}_i^K$ are a splitting of the following *geometrically non-conservative* average steady cell residual [4]

$$\sum_{j \in K} \tilde{\Phi}_j^K = \tilde{\Phi}^K = \int_{\partial K^{n+1/2}} \mathbf{f}(\mathbf{w}_h) \cdot \mathbf{n} \, ds - \int_{K^{n+1/2}} \boldsymbol{\sigma}_h \cdot \nabla \mathbf{w}_h \, dx \quad (6)$$

Almost all the geometrical quantities in the scheme are evaluated on the half-time $n + 1/2$ averaged configuration $\mathcal{T}_h^{n+1/2}$ which ensures

$$|K^{n+1}| - |K^n| = \Delta t \int_{\partial K^{n+1/2}} \boldsymbol{\sigma}_h \cdot \mathbf{n} \, ds = 0, \quad (7)$$

thus the satisfaction of a Discrete Geometric Conservation Law (DGCL).

The properties of the method introduced above are determined by the definition of the split residuals Φ_i^K , Φ_i^K and $\tilde{\Phi}_i^K$, which, following [4, 15, 16] have the form:

$$\Phi_i^K = \sum_{j \in K} \frac{m_{ij}^{K^{n+1}} \mathbf{w}_j^* - m_{ij}^{K^n} \mathbf{w}_j^n}{\Delta t} + \frac{1}{2} \Phi_i^K(\mathbf{w}_h^n) + \frac{1}{2} \Phi_i^K(\mathbf{w}_h^*) \text{ with } \Phi_i^K = \boldsymbol{\beta}_i \Phi^K(\mathbf{w}_h^n) \quad (8)$$

and similarly for $\tilde{\Phi}_i^K$. The $\boldsymbol{\beta}_j$ are distribution matrices, uniformly bounded w.r.t the cell residuals (6), (3), while the m_{ij}^K 's are mass matrix entries consistent with the definition of the spatial distribution [16]. Independently of the specific construction of these quantities, the above definitions give a scheme which is formally second order accurate in space and time, fully conservative, and verifying the DGCL. The interested reader is referred to [3, 4, 15, 16] for more details. In practice, we have used two main classes of methods: multidimensional upwind (MU) schemes, and centred schemes with some form of stabilization. In particular, the computations shown in the paper will compare results using the first order linear and monotone N scheme and the second order linear LDA scheme. These are MU distribution methods. A nonlinear method obtained by blending these two, and referred to as the LDAN scheme, is also tested. Two non-upwind methods are also considered, in particular, the explicit predictor-corrector formulation of the second order linear Streamline-Upwind (SU) method proposed in [16] (see also [9] and the references therein), and the nonlinear blended central (Bc) discretization obtained when blending the SU method with a limited Lax-Friedrich's distribution. The interested reader may consult the above-mentioned references for more details.

3 Numerical Results

3.1 Shock-Expansion Interaction

We start by investigating the behaviour of the schemes on the computation of an expansion fan interacting with a moving shock. The problem involves a rectangular cavity initially filled with an inviscid perfect gas at rest, as shown in the sketch of Fig. 2a. The left end of the domain acts as a piston which is impulsively set into motion, generating a planar shock wave moving towards the closed (right) end of the domain. Given the density ratio across the shock wave, all other kinematic and thermodynamic quantities follow from the Rankine–Hugoniot (R–H) jump relations and are summarized in Table 1. The upstream and downstream dimensionless quantities have been labelled with the subscripts 1 and 2 in Table 1. The Mach number in a reference frame attached to the shock is denoted by M_r , and its upstream value coincides with that of the dimensionless shock speed W_s , since a_1 has been chosen as the reference velocity.

After the impulsive startup, the piston starts slowing down at $t = t_0 + t_s$; its velocity and position are prescribed analytically by:

$$u_p(t) = \begin{cases} u_2 & t_0 < t \leq 0 \\ u_2 \exp^{-t} & t \geq 0 \end{cases}, \quad x_p(t) = \begin{cases} u_2 t & t_0 < t \leq 0 \\ u_2 (1 - \exp^{-t}) & t \geq 0 \end{cases}, \quad (9)$$

where $t_0 = -t_s$ and:

$$t_s = \frac{x_{sh}^0}{W_s - u_2} \simeq 0.3755$$

The flow evolution is sketched in the $x - t$ plane of Fig. 2c. We start the simulations at $t = 0$ with the shock located $x_{sh}^0 = 0.3$ unit lengths ahead of the piston and we initialize the solution using the initial conditions listed in Table 1.

The tail of the expansion wave (marked by arrows in Fig. 2c) is the straight, characteristic line of slope $u_2 + a_2$ that is shed from the origin of the space-time plane and divides the expansion wave from the uniform, shock-downstream region, labelled 2 in Fig. 2. Up to time:

$$\hat{t} = \frac{x_{sh}^0}{(u_2 + a_2) - W} = \frac{x_{sh}^0}{(1 - M_{r,2}) a_2} \simeq 0.8339$$

when the tail of the expansion wave meets the shock trajectory, the expansion region is a simple wave, so that an analytical solution can be computed as described, for instance, in [20, §4.1.5]. At later times, the expansion wave is reflected off the moving shock as an expansion wave of the opposite family, the shock starts reducing its speed and the region of the $x - t$ plane that is between the moving piston and the moving shock ceases to be a simple wave.

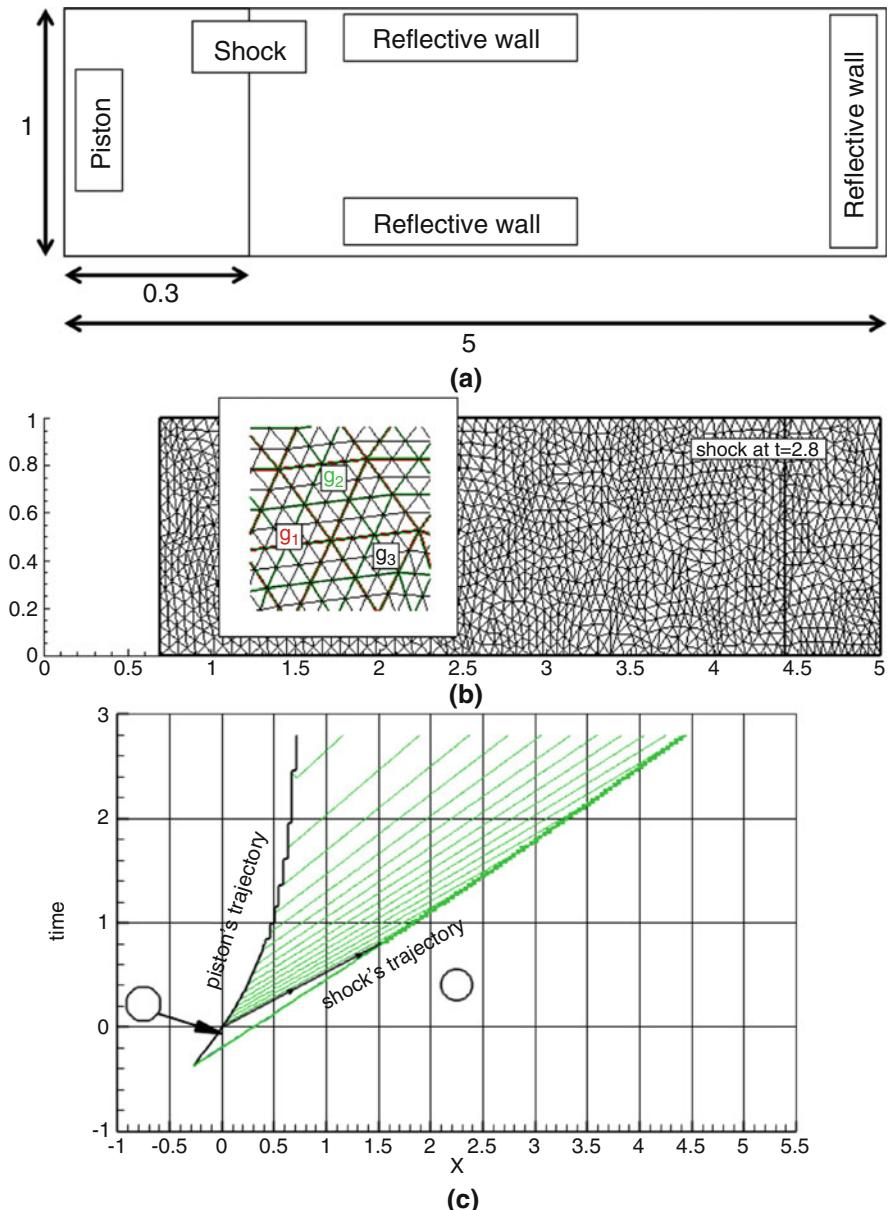


Fig. 2 Shock-expansion interaction. **(a)** Computational set-up at $t = 0$. **(b)** Nested meshes and shock position at $t = 2.8$. **(c)** Pressure iso-contours in the space-time plane

Table 1 Shock-expansion interaction: initial conditions

	ρ	u	v	p	M_r	a
Upstream (1)	1.4	0.0	0.0	1.0	1.5336	1.0
Downstream (2)	2.6872	0.7346	0.0	2.5772	0.6895	1.1588

Simulations have been advanced until the final time $t_2 = 2.8$ when the shock is located at $x_{sh}^{t_2} \simeq 4.43$ and has almost reached the closed (right) end of the cylinder. We present results obtained on a background grid of 639 nodes, 12448 triangles, with a shock front discretized by 41 points (cfr. right picture of Fig. 2a). During the time evolution, the streamwise size of the computational domain is reduced because of the piston's motion. All nodes of the background mesh move as the time elapses, but the topology of the mesh does not change. At any time instant, the location of the nodes of the background mesh is prescribed using the following equation:

$$x_i(t) = x_i(0) + x_p(t) \left(1 - \frac{x_i(0)}{L} \right)$$

where $L = 5$ is the streamwise length of the computational domain at $t = 0$.

Figure 3 provides the pressure field at three subsequent time instants for all the five different spatial discretization schemes and the two shock-modelling schemes. Apart from the excessive shock-thickness of the shock-capturing solutions, it is clear that shock-fitting provides a much cleaner representation of the shock-downstream region. Observe, in particular, that the crossflow non-uniformities are considerably less pronounced in the shock-fitting calculations.

Figure 4 shows the streamwise pressure profiles at two subsequent time instants for all the five spatial discretization schemes. Note that the figures report all the data points in the mesh. The superiority of the shock-fitting calculations is evident, particularly in the reduction of cross-shock spurious variations, despite the fact that both the fitted and captured solutions are represented on unstructured triangulations with almost identical topologies. We do observe an impact of the numerical discretization chosen. The upwind or centred nature of the method does not seem to affect the results as much as the resolution of the scheme. The N scheme clearly provides the “wrong boundary conditions” for the shock, which seems slightly misplaced w.r.t. the analytical solution, as one can see in the right frame of Fig. 4a. Shock amplitude and position are instead perfectly matched by the high order schemes, with perhaps a barely visible off-set present only in the LDAN result, shown in Fig. 4c. These differences are clearly much more important in the captured solutions: the linear, first-order-accurate N scheme smears the shock-profile over a considerable streamwise extent; as expected, the linear, second-order-accurate, but non-monotone LDA and SU schemes give rise to overshoots around the discontinuity; the non-linear, second-order-accurate LDAN scheme is monotonicity preserving, but the otherwise similar Bc scheme also overshoots the shock profile.

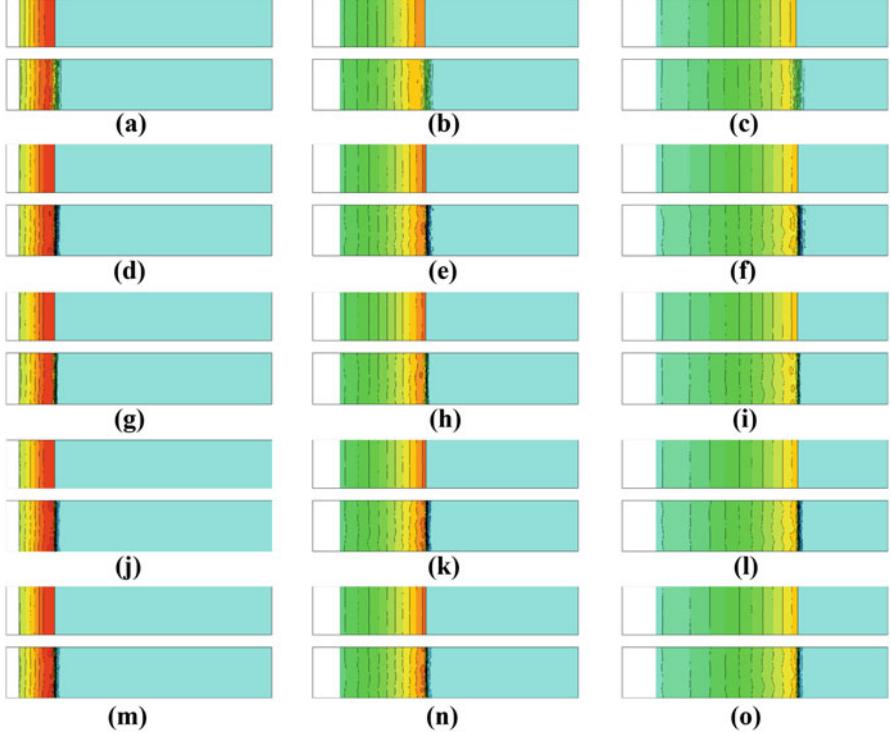


Fig. 3 Shock-expansion interaction: pressure contours at times $t = 0.4$ (*left*), $t = 1.2$ (*centre*) and $t = 2.0$ (*right*): 20 equally spaced contour levels between 0.5 and 2.6, fitted solutions on top and captured solutions on the bottom of each frame. **(a)** $t = 0.4$ N scheme. **(b)** $t = 1.2$ N scheme. **(c)** $t = 2.0$ N scheme. **(d)** $t = 0.4$ LDA scheme. **(e)** $t = 1.2$ LDA scheme. **(f)** $t = 2.0$ LDA scheme. **(g)** $t = 0.4$ LDAN scheme. **(h)** $t = 1.2$ LDAN scheme. **(i)** $t = 2.0$ LDAN scheme. **(j)** $t = 0.4$ SU scheme. **(k)** $t = 1.2$ SU scheme. **(l)** $t = 2.0$ SU scheme. **(m)** $t = 0.4$ Bc scheme. **(n)** $t = 1.2$ Bc scheme. **(o)** $t = 2.0$ Bc scheme

3.2 Shock–Vortex Interaction

The interaction between a shock and a vortex has been frequently reported in the literature as a tool for understanding the mechanisms of noise generation due to the interaction between a shock-wave and a turbulent flow [8, 10, 11, 14]. This problem has also been frequently used as a code verification benchmark [1, 18, 23]. Shock-fitting calculations of shock–vortex interactions have been previously reported in [2, 14, 21].

A uniform, supersonic stream, characterized by a shock-upstream Mach number $M_s = |\mathbf{u}_\infty|/a_\infty$, carries a vortex, from the left to the right of Fig. 5, towards a stationary normal shock. The computational domain, which is sketched in Fig. 5 along with the boundary conditions applied along its boundaries, is the rectangle $[0, 2L] \times [0, L]$. At the initial time, $t = 0$, the vortex is centred in $(x_v/L, y_v/L) = (0.5, 0.5)$ and the shock is located $0.2L$ downstream of the vortex centre. The

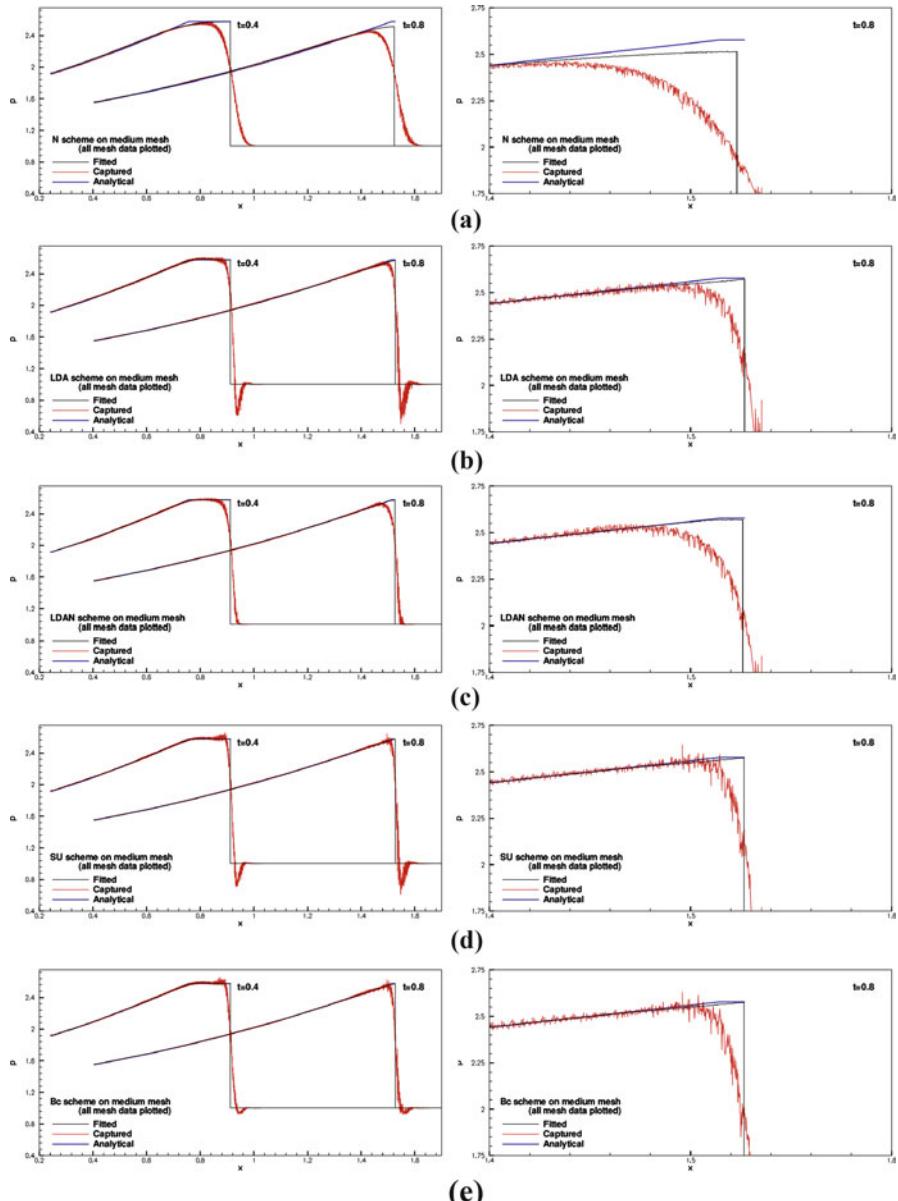


Fig. 4 Shock-expansion interaction: streamwise pressure profiles at two time instants for all the five spatial discretization schemes. Right column: zoom of the $t = 0.8$ peak. (a) N scheme. (b) LDA scheme. (c) LDAN scheme. (d) SU scheme. (e) Bc scheme

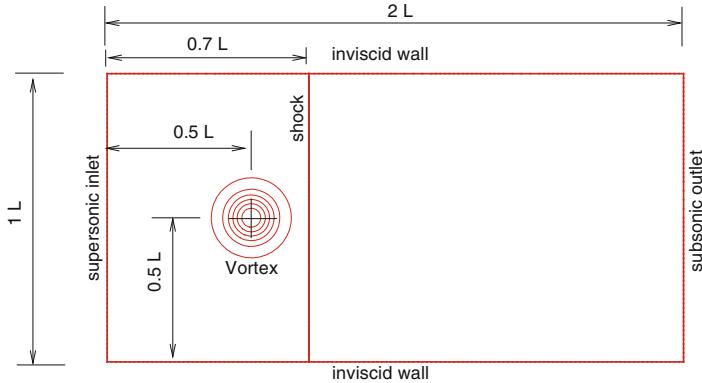


Fig. 5 Shock–vortex interaction: computational domain and boundary conditions

vortical structure we have used, which is an exact solution of the steady Euler equations, consists in a clockwise vortex characterized by a purely tangential velocity component. More precisely, using a polar coordinate system with the origin at the centre of the vortex and moving at constant speed $|\mathbf{u}_\infty|$, the perturbation velocity field $\tilde{\mathbf{u}}$ reads:

$$\tilde{u}_\theta = -\epsilon |\mathbf{u}_\infty| \tau e^{\alpha(1-\tau^2)} \quad (10a)$$

$$\tilde{u}_r = 0 \quad (10b)$$

In Eq. (10) $\tau = r/r_c$ is the non-dimensional radial distance from the pole of the moving reference frame and ϵ , α and r_c are user-defined parameters that control the shape and magnitude of the perturbation. This particular solution of the Euler equations features a divergence-free velocity field, which implies that density is constant along the streamlines and a non zero vorticity field. Since the free-stream flow is isentropic, the thermodynamic variables can be easily obtained from the linear momentum equation, and Crocco's form of the steady momentum equation implies that there must be a gradient of the perturbation total enthalpy in the radial direction, so that total enthalpy also changes across the streamlines.

A vortex Mach number, M_v , can be defined as the ratio between the maximum velocity perturbation and the sound speed, a_∞ , of the shock-upstream flow: $M_v = \max(\tilde{u}_\theta)/a_\infty$. For the particular vortical structure used here, the vortex Mach number becomes:

$$M_v = \epsilon M_s \frac{e^{(\alpha-\frac{1}{2})}}{\sqrt{2\alpha}} \quad (11)$$

Making the same choice of constants in Eqs. (10) and (11) also used in [1, 18, 23]: $\alpha = 0.204$ and $r_c/L = 0.05$, the vortex radius is about $0.35 L$, the maximum

velocity perturbation occurs at about $0.11 L$ from the vortex centre and the strength ϵ of the velocity perturbation follows from Eq. (11), once M_s and M_v have been set.

The topological pattern that arises once the vortex impinges on the shock depends upon both the shock and vortex strengths. We shall hereafter refer to the taxonomy adopted by Grasso and Pirozzoli [8], who define *weak* shock–vortex interactions as those that do not exhibit any shock reflection and *strong* shock–vortex interactions as those that feature secondary shocks; strong interactions can be further classified depending on whether a regular reflection or a Mach reflection occurs. The same authors identify in the (M_s, M_v) pair the two independent variables that govern the interaction: when the vortex Mach number is sufficiently low, weak interactions always occur; however, at a given shock strength, an increase in the vortex Mach number above the $M_v \approx 0.1 \div 0.2$ threshold causes the shock to fold thus producing reflected and diffracted shocks that yield either a regular or a Mach reflection, depending upon the value of M_v .

In the present work, we have set the shock and vortex Mach numbers respectively equal to $M_s = 2.0$ and $M_v = 0.2$, so that a *weak* interaction occurs. The reason for doing so is that the present shock-fitting algorithm is currently unable to automatically identify and follow the appearance of those secondary shocks that arise when a strong interaction takes place. Nonetheless, as demonstrated in [7], strong interactions can be simulated with the available algorithm, thanks to a hybrid mode of operation whereby the normal shock is fitted and the secondary shocks are captured. This hybrid modelling is however not suited for the grid-convergence analysis which will be performed hereafter, since it is impossible to separately identify the effects of the two different shock-modelling practices. Therefore, only weak interactions will be considered here and simulations will be performed using each of the five schemes, both in shock-fitting and shock-capturing mode.

When simulations are run in shock-fitting mode, the linear LDA and SU schemes, that are not monotonicity preserving, can be used without giving rise to oscillations because they are used in smooth regions of the flowfield. The use of non-linear schemes is unnecessary in this case, but non-linear schemes have also been tested in conjunction with shock-fitting to allow a comparison with the shock-capturing calculation using the same scheme.

When simulations are run in shock-capturing mode, it should be expected and it is indeed observed that the linear schemes give rise to wiggles around the captured shock wave. Nonetheless, shock-capturing calculations with linear schemes have also been performed to allow a comparison with the shock-fitting calculation using the same scheme.

A qualitative view of the solutions obtained with the two approaches is given in Fig. 6. The pictures show the total enthalpy contours in the solutions obtained with the LDA and LDAN schemes with shock capturing and fitting on the finest mesh level of Table 2. Note that the fitted results are virtually indistinguishable, and only those of the LDA are reported. As for the shock-expansion interaction, the qualitative difference is already quite striking from the comparison of the contours. In particular, besides the oscillations related to the approximation of the shock, we can see clearly that the contours downstream of the discontinuity are much

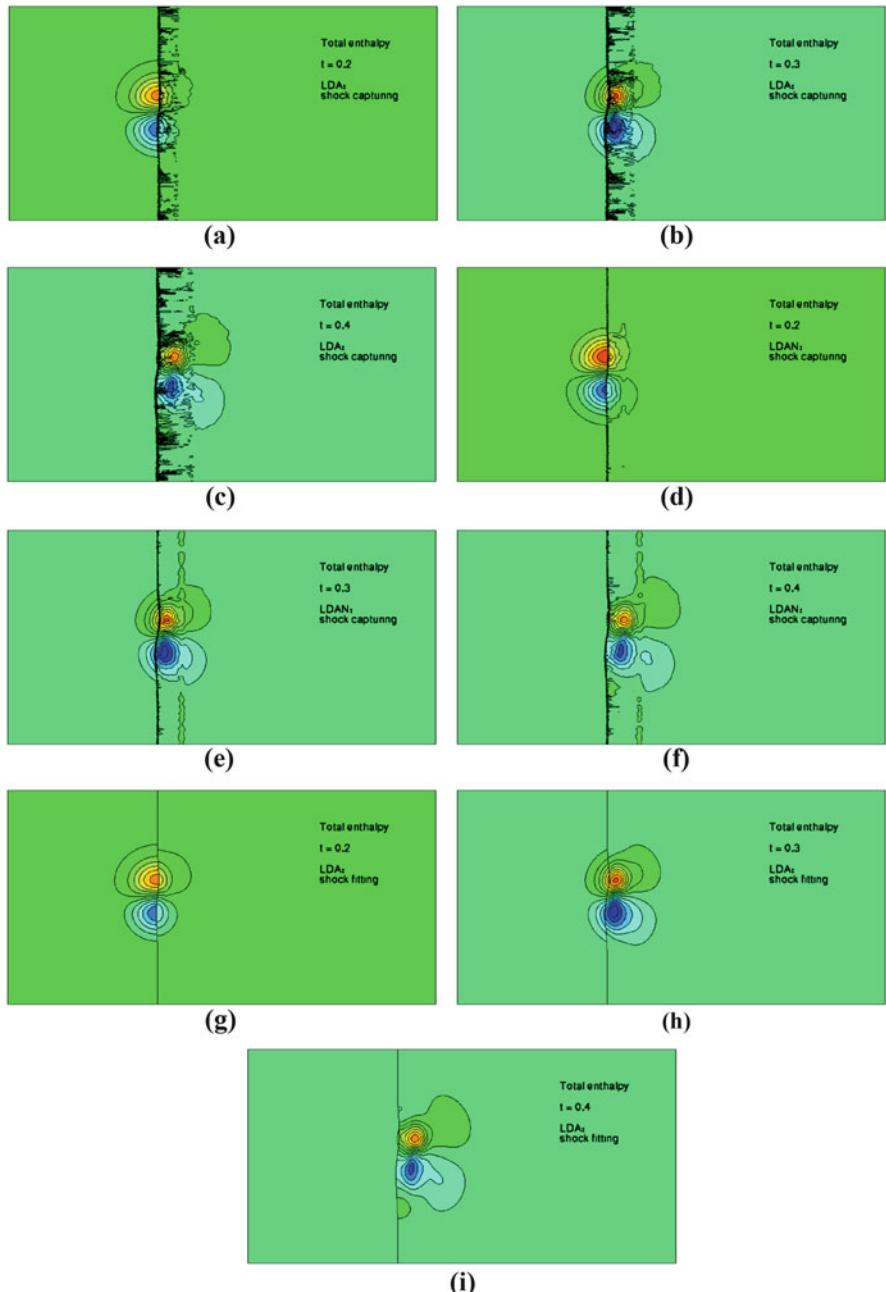


Fig. 6 Shock-vortex interaction: total enthalpy contours at times $t = 0.2$ (left), $t = 0.3$ (centre) and $t = 0.4$ (right): 15 equally spaced contour levels obtained with the shock-capturing versions of the LDA (top), and LDAN (middle), and with shock-fitting and LDA (bottom). (a) $t = 0.2$ LDA (capturing). (b) $t = 0.3$ LDA (capturing). (c) $t = 0.4$ LDA (capturing). (d) $t = 0.2$ LDAN (capturing). (e) $t = 0.3$ LDAN (capturing). (f) $t = 0.4$ LDAN (capturing). (g) $t = 0.2$ LDA (fitting). (h) $t = 0.3$ LDA (fitting). (i) $t = 0.4$ LDA (fitting)

Table 2 Shock-vortex interaction: grid features

Grid	N. of triangles	N. of gridpoints	Δt_k	h_k
g_1	6794	4181	0.00160	0.03000
g_2	27,128	14,437	0.00080	0.01500
g_3	108,438	55,277	0.00040	0.00750
g_4	433,778	218,269	0.00020	0.00375

less smooth in the captured solutions. This is more visible in the LDA results (Figs. 6d–f), however a similar behaviour is also present in the LDAN solutions. The fitted computations, on the other hand, show very nice and smooth contours.

In order to perform a quantitative comparison among the various combinations of spatial discretization schemes and shock-modelling practices, we have used the following global measure:

$$S_h(t) = \frac{1}{|\Omega|} \int_{\Omega} s_h(x, t) d\Omega \quad (12)$$

where $s = p\rho^{-\gamma}$ is entropy and we have computed the corresponding discretization error and order-of-convergence. To perform a grid convergence analysis, the computational domain has been discretized using a sequence of four nested Delaunay triangulations, hereafter labelled g_1 to g_4 . The coarsest g_1 mesh has been created using the Triangle code [22] by specifying a maximum area constraint, whereas the other three grid levels have been obtained by recursive subdivision of each triangle of the coarser level into four sub-triangles. Table 2 summarizes the characteristics of the various meshes that have been used both in the shock-capturing computation and as *background* triangulations in the shock-fitting computation. The dimensionless, characteristic spatial size h_k of grid k has been set equal to the uniform spacing along the boundaries of the computational domain; observe that both h_k and the dimensionless time-step length Δt_k are halved when moving from grid level k to the finer grid level $k + 1$.

Whenever an exact, or manufactured solution to the set of governing PDEs is available, the discretization error can be evaluated straightforwardly. However, when the exact solution is not known, the discretization error of a numerical solution and the convergence properties of a numerical scheme can still be estimated using a suite of techniques collectively known as Richardson extrapolation (RE). There exist different variations of the RE technique depending on the number of grid levels involved and on the expression of the leading error terms that is a priori assumed in the analysis; hereafter, we refer to the generalized RE technique of fixed order, following the nomenclature used in [19]. The term generalized refers to the fact that the RE technique is applied to n th-order-accurate schemes and grids refined by an arbitrary factor r . Note that the use of nested triangulations fixes the grid refinement ratio between each couple of meshes in the sequence to a value $r = 2$. It is also assumed that the order of convergence n is known (hence, the term fixed) and is typically set equal to the design order of the scheme. The method used here relies on an error expansion of the type

$$\epsilon_h(x) = u_h(x) - u_0(x) = \sum_{p=n}^{\infty} g_p(x) h^p = g_n(x) h^n + HOT \quad (13)$$

Neglecting the *HOT*, writing Eq. (13) for two grid levels h and rh , and solving for the unknown exact solution, one obtains:

$$\tilde{u}_0 = u_h - \frac{u_{rh} - u_h}{(r^n - 1)} \quad (14)$$

In Eq. (14) we have used a tilde to underline the fact that \tilde{u}_0 is only an approximation of the corresponding quantity that appears in Eq. (13), because the higher-order contributions have been neglected.

Since an exact solution is not known, for the present test-case, Eq. (14) has been used as follows:

$$\tilde{S}_0 = S_{h_4} - \frac{S_{h_3} - S_{h_4}}{3}$$

to provide an approximation of the *exact* entropy integral of Eq. (12). This is a rather common practice that, according to Roache [17], has been in use in the CFD arena since the early 1990s, even though the procedure is somewhat questionable, because the order is a-priori given in order to perform the RE.

The results obtained with this procedure are summarized in the convergence plots of Fig. 7. The plots show that all the captured solutions converge *at best* with a first order rate. When removing the error due to the shock-capturing, all the schemes behave as if the solution were smooth. In particular, all the schemes provide convergence rates very close to those observed in [16]: 2 for the liner LDA and SU schemes, very close to 2 for the Bc scheme, a rate of roughly 1.5 for the LDAN scheme, and an order 1 for the N scheme.

4 Conclusion and Outlook

We have provided a preliminary study of the behaviour of explicit ALE residual distribution when coupled with shock-fitting. The results show that the fitted solutions are quite independent on the stabilization mechanism used. Multidimensional upwind schemes, and centred schemes with some form of artificial dissipation yield very similar results. What makes most of the difference is the accuracy of the discretization. Even with fitting, first order schemes fail to provide the correct boundary conditions for the discontinuity, producing a visible error in shock position and strength. High order discretizations, both linear and nonlinear, are thus required. A comparison with captured solutions shows two main results: fitting allows to recover second order of accuracy where capturing leads to first order; the fitted results are much less sensitive to the use of monotonicity preserving methods, as well as to mesh irregularities due to the absence of mesh dependent errors generated

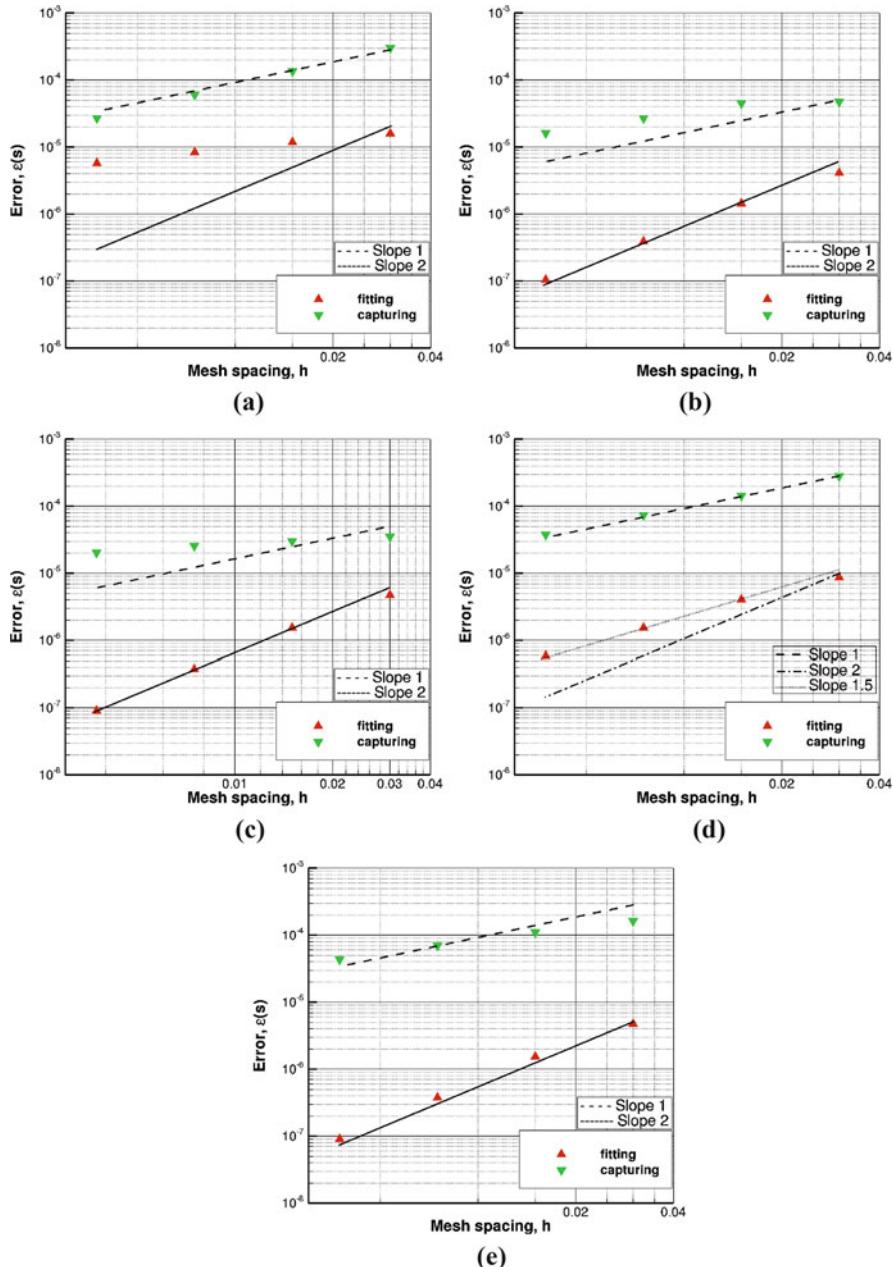


Fig. 7 Convergence histories obtained by means of Richardson Extrapolation for the shock-vortex interaction problem at time $t = 0.4$. (a) $t = 0.4$ N scheme. (b) $t = 0.4$ LDA scheme. (c) $t = 0.4$ SU scheme. (d) $t = 0.4$ LDAN scheme. (e) $t = 0.4$ Bc scheme

in the captured region. In addition anomalies related to the mesh dependent nature of the capturing error are removed so that the methods behave as if the flow were smooth. This has been proven on two test cases.

Natural extensions of this work will be to consider more complex cases, and more interesting discretization strategies. The first aspect may involve stronger shock–vortex interactions, with the appearance of secondary (lambda) shocks. Concerning the discretization, the ALE framework used here would allow quite naturally the use of a hybrid fitting/adaptive approach, with an r -adaptation strategy in the non-fitted domains, as proposed, e.g., in [3]. Lastly, a very interesting test would be to repeat the studies of this paper when the fitting is performed using non-conservative formulations in the CFD part. This may reveal very useful when considering flows also involving strong contact discontinuities, and chemistry.

Acknowledgements The authors are indebted to R. Paciorri for the many constructive discussions during the development of this work.

References

1. Abgrall, R., Mezine, M.: Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. *J. Comput. Phys.* **188**(1), 16–55 (2003). [https://doi.org/10.1016/S0021-9991\(03\)00084-6](https://doi.org/10.1016/S0021-9991(03)00084-6)
2. Arina, R., Marsilio, R.: A shock-fitting spectral method for the study of shock-vortex interactions. In: 28th Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics (AIAA) (1997). <https://doi.org/10.2514/6.1997-1839>
3. Arpaia, L., Ricchiuto, M.: r-adaptation for Shallow Water flows: conservation, well balancedness, efficiency. Research Report RR-8956, Inria Bordeaux Sud-Ouest (2016)
4. Arpaia, L., Ricchiuto, M., Abgrall, R.: An ale formulation for explicit runge-kutta residual distribution. *J. Sci. Comput.* **190**(34), 1467–1482 (2014)
5. Arpaia, L., Ricchiuto, M., Abgrall, R.: An ALE formulation for explicit Runge-Kutta residual distribution. *J. Sci. Comput.* 1–46 (2014). <https://doi.org/10.1007/s10915-014-9910-5>
6. Bonfiglioli, A., Grottadurea, M., Paciorri, R., Sabetta, F.: An unstructured, three-dimensional, shock-fitting solver for hypersonic flows. *Comput. Fluids* **73**, 162–174 (2013)
7. Bonfiglioli, A., Paciorri, R., Campoli, L.: Unsteady shock-fitting for unstructured grids. *Int. J. Numer. Methods Fluids* **81**(4), 245–261 (2016). <https://doi.org/10.1002/fld.4183>. Fld.4183
8. Grasso, F., Pirozzoli, S.: Shock-wave-vortex interactions: shock and vortex deformations, and sound production. *Theor. Comput. Fluid Dyn.* **13**(6), 421–456 (2000). <https://doi.org/10.1007/s001620050121>
9. Hughes, T., Scovazzi, G., Tezduyar, T.: Stabilized methods for compressible flows. *J. Sci. Comput.* **43**, 343–368 (2010)
10. Inoue, O., Hattori, Y.: Sound generation by shock-vortex interactions. *J. Fluid Mech.* **380**, 81–116 (1999). <https://doi.org/10.1017/S0022112098003565>
11. Meadows, K.R., Kumar, A., Hussaini, M.: Computational study on the interaction between a vortex and a shock wave. *AIAA J.* **29**(2), 174–179 (1991). <https://doi.org/10.2514/3.59916>
12. Paciorri, R., Bonfiglioli, A.: A shock-fitting technique for 2D unstructured grids. *Comput. Fluids* **38**(3), 715–726 (2009)
13. Paciorri, R., Bonfiglioli, A.: Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique. *J. Comput. Phys.* **230**(8), 3155–3177 (2011). <https://doi.org/10.1016/j.jcp.2011.01.018>

14. Pao, S., Salas, M.: A numerical study of two-dimensional shock vortex interaction. In: Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics (1981). <https://doi.org/10.2514/6.1981-1205>
15. Ricchiuto, M.: An explicit residual based approach for shallow water flows. *J. Comput. Phys.* **80**, 306–344 (2015)
16. Ricchiuto, M., Abgrall, R.: Explicit Runge-Kutta residual distribution schemes for time dependent problems: second order case. *J. Comput. Phys.* **229**(16), 5653–5691 (2010). <https://doi.org/10.1016/j.jcp.2010.04.002>
17. Roache, P.: Quantification of uncertainty in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **29**, 123–160 (1997). <https://doi.org/10.1146/annurev.fluid.29.1.123>
18. Rossiello, G., De Palma, P., Pascazio, G., Napolitano, M.: Second-order-accurate explicit fluctuation splitting schemes for unsteady problems. *Comput. Fluids* **38**(7), 1384–1393 (2009). <https://doi.org/10.1016/j.compfluid.2008.01.021>. Special Issue Dedicated to Professor Alain Lerat on the Occasion of his 60th Birthday
19. Roy, C.J., McWherter-Payne, M.A., Oberkampf, W.L.: Verification and validation for laminar hypersonic flowfields Part 1: verification. *AIAA J.* **41**(10), 1934–1943 (2003)
20. Salas, M.: A Shock-Fitting Primer, 1st edn. CRC Applied Mathematics & Nonlinear Science. Chapman & Hall, London (2009)
21. Salas, M.D., Zang, T.A., Hussaini, M.Y.: Shock-Fitted Euler Solutions to Shock-Vortex Interactions, pp. 461–467. Springer, Berlin (1982). https://doi.org/10.1007/3-540-11948-5_59
22. Shewchuk, J.R.: Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In: Lin, M.C., Manocha, D. (eds.) *Applied Computational Geometry: Towards Geometric Engineering*. Lecture Notes in Computer Science, vol. 1148, pp. 203–222. Springer, Berlin (1996). From the First ACM Workshop on Applied Computational Geometry
23. Shu, C.W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, pp. 325–432. Springer, Berlin (1998)

A Shock-Fitting Technique for ALE Finite Volume Methods on Unstructured Dynamic Meshes

Jun Liu and Dongyang Zou

1 Introduction

Shock waves occur when the gas is sharply compressed. In all flow fields in which they occur, shock waves play an important role that affects the overall flow behavior. Computing shock waves constitute a difficult problem.

In 1950, von Neumann pioneered the concept of adding sufficient artificial viscosity to the inviscid flow equations to capture the shock [33]. In 1954, Lax proposed the famous “weak solution theory” which solves the governing equations in integral form rather than in differential form [13]. Based on the weak solution theory, various shock-capturing schemes were devised and widely used for solving two-dimensional steady flows [14, 21]. However, owing to the presence of spurious numerical oscillations in the vicinity of the shock waves, applications of high-order shock-capturing schemes were greatly limited.

Subsequently, studies of the underlying mechanisms that produce these oscillations have attracted the attention of many scientists [9, 16, 37]. These studies proved to be successful and led to algorithms known as oscillation-free schemes. Many capturing algorithms were developed and used, including total variation diminishing (TVD) schemes [8, 34–36], essentially non-oscillatory (ENO) schemes [10], weighted essentially non-oscillatory (WENO) schemes [17], and non-oscillatory and non-free-parameter dissipation difference (NND) schemes [38].

Despite several creative mathematical solutions proposed to improve the basic shock-capturing technique, it still failed at what was its initial goal. Owing to the inability of high-order capturing schemes to pass information through a discontinuity correctly [5], limiter functions are needed in these shock-capturing schemes.

J. Liu • D. Zou (✉)

Dalian University of Technology, Dalian 116024, P.R. China

e-mail: liujun65@dlut.edu.cn; zoudy89@gmail.com

A limiter function takes effect in the region with a steep gradient so that a discontinuity is smeared over a few grid points in a shock-capturing algorithm. Therefore, a captured shock relies heavily on the computational mesh. Mesh refinement is a direct way of increasing the solution quality without considering the computational cost. However, mesh refinement does not reduce the oscillations, rather it decreases their wave length [15]. In other words, improving solution quality by refining the mesh has not changed the dependency on the use of limiter functions. The situation is even worse if the computational mesh is unstructured. This is because there is always at least one face that is not aligned with the shock on an unstructured mesh. The damage brought about by the loss of orthogonality increases the amplitude of these spurious oscillations. Some studies [6, 28] on a truly multidimensional or rotational upwind scheme have been carried out to eliminate these spurious oscillations. Owing to the limitations of the current technology, the studies on multidimensional schemes have not entirely solved the problems of oscillations. As Arora and Roe [2] pointed out, it is difficult to ensure a captured shock to be both narrow and oscillation-free.

As an alternative, the shock-fitting technique is reconsidered to eliminate the disadvantages in dealing with shock waves using the shock-capturing technique. The shock-fitting technique has been a reliable tool for simulating a shock wave since the dawn of computational fluid dynamics (CFD). In the 1960s, Abbott and Moretti [1] developed a shock-fitting technique together with the Lax–Wendroff scheme, which produced a set of solutions on supersonic blunt body flows requiring only 6 min on a CDC 6600 computer. This work greatly advanced the development of computational gas dynamics at the time. In 1967, Richtmyer [27] proposed what was known as the floating shock-fitting method, which allowed the fitted shock wave to float on the background mesh. The floating shock-fitting technique eliminated the problems associated with partitioning the flow field using boundary-shock implementation. Salas [29] applied the floating shock-fitting technique to simulate flows with internal discontinuities, including shock waves and contact surfaces, which greatly promoted the development of the shock-fitting technique. The floating shock-fitting technique was implemented and combined with a second-order accurate upwind scheme by Hartwich [11], which made it possible to construct a general, neoconservatives Euler solver. With additional studies, increasingly complicated flows were solved using the shock-fitting technique [31]. For example, Nasuti and Onofri used Moretti's shock-fitting technique to compute flows, including triple points and shock interactions[22]. In addition, high-order accurate algorithms were applied using the shock-fitting technique. In 1999, Kopriva [12] combined the shock-fitting technique with the spectral method and applied it for the computation of supersonic flows. Indeed, the idea of combining the spectral approximation performed only on smooth portions of the flow with a precise description of a shock discontinuity provided by the shock-fitting technique has gained significance in the construction of new computational algorithms. More recently, Zhong et al. [25, 26] coupled the high-order finite difference (FD) method with the shock-fitting method and applied it to hypersonic flows.

Although considerable progress has been achieved with the shock-fitting technique, it has always been unpopular owing to difficult implementation. In terms of coding simplicity, special treatment for discontinuities is not necessary in the shock-capturing. By contrast, in shock-fitting, there are often special points requiring ad hoc treatments, in addition to the topological difficulties that accompany the presence of multiple shocks. The rapid growth of computational resources has given strong support to CFD. Thus, when computational resources are not limited, most people would prefer a simple algorithm rather than an efficient one.

Researchers practicing shock-fitting proved to be resourceful in search for a viable solution. A shock-fitting technique implemented on unstructured grids has been presented by Paciorri and Bonfiglioli [23, 24] and by Bonfiglioli et al. [4]. Compared with the traditional boundary shock-fitting technique on structured grids, the unstructured version of the technique does not suffer from the strong topological limitations that plague boundary shock-fitting implementations on structured grids. In addition, compared with the floating shock-fitting technique, the coupling between shock-fitting algorithms and existing gas dynamics solvers is much simpler for unstructured than it is for structured grids. Salas [30] supported the work based on unstructured grids because of its great potential for generalizing the shock-fitting technique in the future.

In the present work, we improved the shock-fitting method based on unstructured dynamic-mesh that is presented in the reference [39]. First, grid nodes are defined as either shock points or common points according to a shock detection technique. Two or more sets of states are defined in a shock point, and it is permitted to move. The motion and flow states of a shock node are determined by the R-H jump relations. Along with the motion of the shock nodes, an unstructured dynamic meshes algorithm is used to update the position of the common nodes to ensure the high quality of the computational mesh. Then, the nature of the cell faces can also be determined based on the definition of cell nodes. If all the nodes of a cell face are shock points, it is defined as a shock face. The fluxes across a shock face are set equal to the upstream fluxes. In the computational process, the nature of the nodes is permitted to change. Thus, it is easier to apply this method in complex problems, even with topological change.

2 Computational Algorithms

In this work, we have combined the shock-fitting algorithm with an existing shock-capturing code [19], which includes an unstructured dynamic-mesh algorithm. The computational code is named the Mixed Capturing and Fitting Solver (MCFS). In the MCFS code, the present shock-fitting algorithm is used to determine the motion of the shock points and compute the boundary fluxes across each shock boundary face.

We will consider a flow field in which several shock waves occur. Using MCFS to simulate the flow field, three types of solutions can take place. One of these is the captured solution. In this solution, no shock point is defined in the flow field. By contrast, a fully-fitted solution is obtained when all shock waves in the flow field

are fitted as the shock points. If some, but not all shock waves are fitted, a hybrid solution is obtained.

This section provides a detailed description of the proposed fitting technique along with a brief introduction to the shock-capturing algorithm and unstructured dynamic-mesh technique.

2.1 Shock-Capturing Algorithm

During the past decade, we have been studying the unstructured dynamic-mesh technique, and we developed an unstructured code for cell-centered finite volume methods, which was successfully applied to solve many problems in science and engineering. A detailed description of the solver's basic features is given in [19], whereas a brief discussion of its main features is provided in this section.

The flow is assumed to be governed by the time-dependent Euler equations, which are written in differential form using an arbitrary Lagrangian–Eulerian (ALE) formalism as:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{0}. \quad (1)$$

The conservative variables and convective fluxes are defined as

$$\begin{aligned} \mathbf{Q} &= \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho(u - x_t) \\ \rho u(u - x_t) + p \\ \rho v(u - x_t) \\ \rho w(u - x_t) \\ \rho u H - \rho E x_t \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho(v - y_t) \\ \rho u(v - y_t) \\ \rho v(v - y_t) + p \\ \rho w(v - y_t) \\ \rho v H - \rho E y_t \end{pmatrix}, \\ \mathbf{H} &= \begin{pmatrix} \rho(w - z_t) \\ \rho u(w - z_t) \\ \rho v(w - z_t) \\ \rho w(w - z_t) + p \\ \rho w H - \rho E z_t \end{pmatrix}, \end{aligned} \quad (2)$$

where ρ is density; u , v , and w are the velocity components in x , y , and z directions respectively, and p is the static pressure; x_t , y_t , and z_t are the grid velocity components in x , y , and z directions respectively. The specific total energy E and enthalpy H are given by

$$\begin{aligned} E &= \frac{1}{\gamma - 1} \frac{p}{\rho} + \frac{1}{2} (u^2 + v^2 + w^2), \\ H &= \frac{\gamma}{\gamma - 1} \frac{p}{\rho} + \frac{1}{2} (u^2 + v^2 + w^2) \end{aligned} \quad (3)$$

where γ denotes the specific heat ratio.

The Eq. (1) are discretized using a cell-centered finite-volume procedure, which employs unstructured triangular and quadrilateral meshes. The cell residual \mathbf{Res}_i is defined as the numerical approximation of the integral over the cell of the spatial operator divided by the cell volume v_i , that is:

$$\mathbf{Res}_i = - \sum_{k=1}^n \phi_{ik}(\mathbf{n}_{ik}) \Delta s_{ik} / v_i, \quad (4)$$

where k spans the n control volumes that are neighbors of i -th volume, Δs_{ik} is the area of k -th cell face, \mathbf{n}_{ik} is the unit vector of the outward normal to the face, and $\phi_{ik}(\mathbf{n}_{ik})$ is the numerical flux, which is a numerical approximation to $\mathbf{F}_c \cdot \mathbf{n}_{ik}$. Although, flux-vector-splitting and flux-difference-splitting methods are both available in the solver, only van Leer's scheme has been used to compute the numerical results to be presented in Sect. 3. Equation (1) can be reduced to:

$$\frac{\partial \bar{\mathbf{Q}}_i}{\partial t} = \mathbf{Res}_i. \quad (5)$$

In the work, Eq. (5) has been integrated in time using a four-stage Runge–Kutta method.

The numerical flux $\phi_{ik}(\mathbf{n}_{ik})$ is evaluated by using the left and right states of the interface. If the cell-averaged values are taken as the left and right states, the numerical method will be a first-order accurate in space. To achieve second-order accuracy, the Gauss–Green method is applied to calculate the cell gradient $\nabla \bar{\mathbf{Q}}_i$. Therefore, the left and right states of face k are:

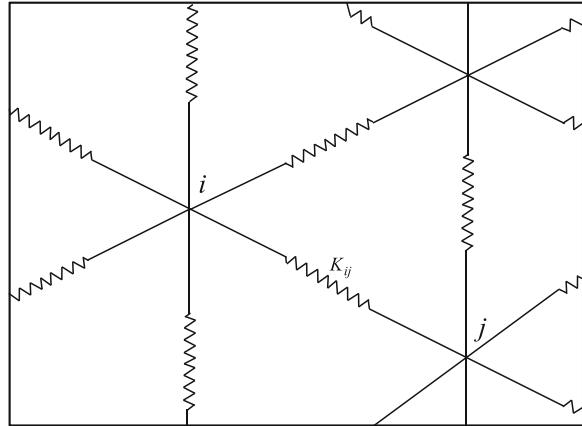
$$\begin{aligned} \bar{\mathbf{Q}}_k^L &= \bar{\mathbf{Q}}_i + \psi_i \nabla \bar{\mathbf{Q}}_i \cdot (\mathbf{O}_k - \mathbf{O}_i) \\ \bar{\mathbf{Q}}_k^R &= \bar{\mathbf{Q}}_j + \psi_j \nabla \bar{\mathbf{Q}}_j \cdot (\mathbf{O}_k - \mathbf{O}_j), \end{aligned} \quad (6)$$

where k refers to the number of the interface between the i -th and j -th control volumes; \mathbf{O}_i \mathbf{O}_j are the position vectors of the centroid of the cells; and \mathbf{O}_k is the position vector of the centroid of face k . The slope limiter of Venkatakrishnan [32] has been used for the calculation of ψ in second-order accurate flow simulations.

2.2 Unstructured Dynamic-Mesh Technique

A mesh deformation strategy based on both linear and torsional spring analogies is used to update the mesh while maintaining the quality of the grid. According to this mesh deformation strategy, the collection of the edges of the mesh is simulated by a network of linear springs, as proposed by Batina [3]. As shown in Fig. 1, the springs

Fig. 1 Sketch of springs analogies



are taken as linear, and Hooke's law determines the force at every node i exerted by the nodes j , which are connected to node i :

$$\mathbf{R}_i = \sum_{j=1}^{N_i} K_{ij}(\mathbf{x}_j - \mathbf{x}_i). \quad (7)$$

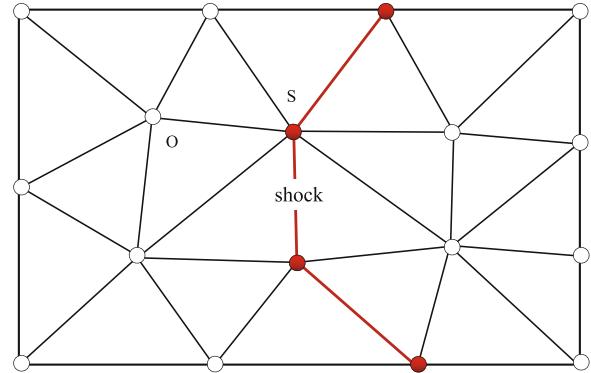
In Eq. 7, N_i is the number of neighbors of the node i and K_{ij} is the stiffness of the spring connecting nodes i and j . Note that if nodes i and j do not connect to each other, $K_{ij} = 0$. When the initial computational grids are generated, the forces at every node are computed, and their magnitude at every node i does not change during the deformation process. When the boundary nodes move, the new positions of the interior cell nodes can be determined by solving the following linear system:

$$\begin{bmatrix} -\sum_{j=1}^{N_1} K_{1j} & & & \\ & -\sum_{j=1}^{N_2} K_{2j} & K_{ik} & \\ & & K_{ik} & \ddots \\ & & & -\sum_{j=1}^{N_m} K_{mj} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_m \end{bmatrix}. \quad (8)$$

After regrouping, the iterative equation to be solved yields:

$$\mathbf{x}_i^{k+1} = \frac{\sum_{j=1}^{N_i} K_{ij} \mathbf{x}_j^k - \mathbf{R}_i}{\sum_{j=1}^{N_i} K_{ij}}. \quad (9)$$

Fig. 2 Definition of properties of grid nodes



The Eq. 9 is solved for every internal node in the mesh, whereas the positions of the boundary nodes are fixed during the calculation. More details are given in [7].

2.3 Shock-Fitting Algorithm

To illustrate the algorithmic features of the proposed methodology, we consider a two-dimensional domain and a shock wave crossing this domain. In solving such a problem, computational processes are introduced step by step.

Step 1: Definition of the properties of the grid nodes.

Unstructured triangular grids are used to discretize the computational domain. After discretizing, the grid nodes are defined as either shock points (labeled as ‘S’) or common points (labeled as ‘O’) as shown in Fig. 2. Two or more sets of states are assigned to each shock node, but only one set to common nodes.

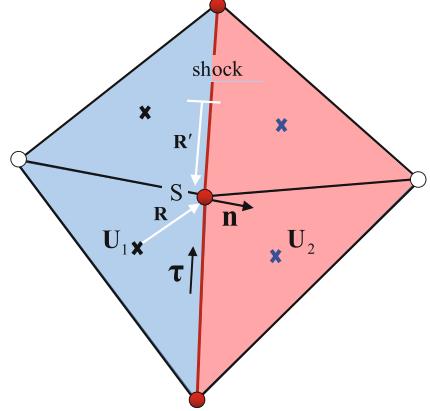
A shock detection technique is needed to determine the position of the shock wave(s) and define the nature of the grid nodes. In this work, we detected the positions of the shock waves using Mach number iso-contours.

Step 2: Determination of the states on shock nodes.

In this work, a cell-centered finite volume method is chosen as the shock-capturing solver. Therefore, the states of the grid nodes are obtained by averaging the flow parameters at the center of the cells. For a common node, the flow states can be computed using the states of cell-averaged values of the neighboring cells. However, there are two sets of states on a shock node.

We first identify the upstream region and downstream region according to the entropy magnitudes on both sides of a shock face. Then, the upstream states and downstream states can be averaged by using the flow variables at the center of cells, which belong to the upstream region \mathbf{U}_1 and downstream region \mathbf{U}_2 respectively. The symbol \mathbf{V} represents the flow states in a grid node and the subscripts u and d are used to mark the states belonging to the upstream and downstream regions:

Fig. 3 Determination of states on shock nodes



$$\mathbf{V}_u = \frac{\sum_{k=1}^{N_1} \alpha_k \mathbf{U}_{1k}}{\sum_{k=1}^N \alpha_k}, \quad \mathbf{V}_d = \frac{\sum_{k=1}^{N_2} \alpha_k \mathbf{U}_{2k}}{\sum_{k=2}^N \alpha_k}. \quad (10)$$

where N_1 and N_2 are the number of upstream cells and downstream cells neighboring node S , α_k is the weight coefficient and $\alpha_k = \frac{\chi}{|\mathbf{R}|}$; As shown in Fig. 3, \mathbf{R} is a vector joining the k -th cell center to the shock point S . The parameter χ is used to determine whether node S belongs to the domain of dependence of the k -th cell, and we have

$$\chi = \begin{cases} 0, & \text{if } M_{\tau k} \leq -1 \\ 1, & \text{if } M_{\tau k} > -1 \end{cases}. \quad (11)$$

where $M_{\tau k} = \frac{\mathbf{u}_{2k} \cdot \boldsymbol{\tau}}{a_{2k}}$ is the tangential Mach number of the k -th downstream cell along the vector $\boldsymbol{\tau}$; and $a_{2k} = \sqrt{\gamma \frac{p_{2k}}{\rho_{2k}}}$ represents the downstream sound speed.

In addition to the upstream and downstream states, we define the normal unit vector \mathbf{n} within each shock node using the unit vectors normal to the neighboring shock faces:

$$\mathbf{n} = \frac{\sum_{k=1}^N \alpha'_k \mathbf{n}'_k}{\sum_{k=1}^N \alpha'_k} \quad (12)$$

In Eq. (12) N is the total number of shock faces linked to the shock point S ; \mathbf{n}_k is the unit vector normal to the k -th shock face, and pointing from the upstream to the downstream region of the shock face, α'_k is a weight coefficient and $\alpha'_k = \frac{\chi}{|\mathbf{R}'|}$. \mathbf{R}' is a vector from the center of the k -th shock face to the shock point S ; the parameter χ is defined according to Eq. (11).

According to the characteristics of supersonic flows, the downstream states \mathbf{V}_d in Eq. (10) computed by the downstream states are also incorrect.

For a shock node, the values on both sides are related through the R-H relations, which are used to recompute the downstream values \mathbf{V}_d . To distinguish the recomputed downstream states from previous ones, we write the corrected values as \mathbf{V}'_d . There are four unknown quantities in the downstream region that need to be determined, i.e., ρ'_d , u'_d ($u'_d = \mathbf{u}'_d \cdot \mathbf{n}$), p'_d , and w ($w = \mathbf{W} \cdot \mathbf{n}$). As the R-H relations written in the shock-normal direction \mathbf{n} provide only three equations, another one is needed. Fortunately, we know that the downstream field information propagates along a generator of the characteristic conoid, reaching the shock. The influence of the downstream flow on the upstream shock is carried along this conoid. Therefore, a Riemann variable J propagating upstream can also be used to determine the boundary conditions.

The upstream normal component of the Mach number, relative to the shock front at $t + \Delta t$ is expressed as:

$$M_{u,\text{rel}} = \frac{\mathbf{u}_u \cdot \mathbf{n} - w}{a_u}, \quad (13)$$

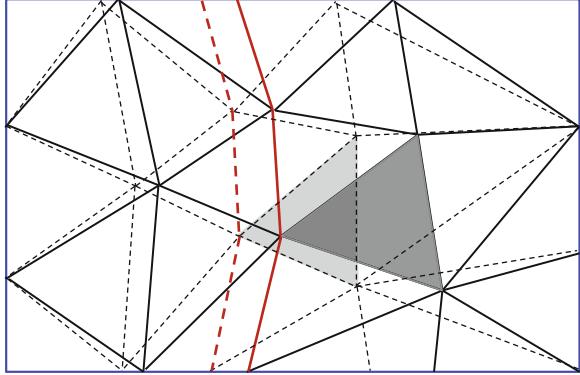
where a_u represents the acoustic speed of the upstream region. In terms of $M_{u,\text{rel}}$ and the boundary values in the downstream region, we have:

$$\begin{aligned} p'_d &= \frac{p_u (2\gamma M_{u,\text{rel}}^2 - (\gamma - 1))}{\gamma + 1} \\ \rho'_d &= \frac{\rho_u (\gamma + 1) M_{u,\text{rel}}^2}{(\gamma - 1) M_{u,\text{rel}}^2 + 2} \\ \frac{\rho'_d (u'_d - \mathbf{u}_u \cdot \mathbf{n} + a_u M_{u,\text{rel}})}{\gamma p'_d} &= \frac{(\gamma - 1) M_{u,\text{rel}}^2 + 2}{2\gamma M_{u,\text{rel}}^2 - (\gamma - 1)} \\ J &= \frac{2\sqrt{\gamma p'_d / \rho'_d}}{\gamma - 1} - u'_d = \frac{2\sqrt{\gamma p_d / \rho_d}}{\gamma - 1} - \mathbf{u}_d \cdot \mathbf{n}. \end{aligned} \quad (14)$$

Equation (14) is a set of four equations in the four unknowns. These four equations can be combined into a single equation for $M_{u,\text{rel}}$ as follows:

$$\begin{aligned} \frac{1}{\gamma - 1} \sqrt{(2\gamma M_{u,\text{rel}}^2 - (\gamma - 1)) ((\gamma - 1) M_{u,\text{rel}}^2 + 2) / M_{u,\text{rel}}^2 + (M_{u,\text{rel}}^2 - 1) / M_{u,\text{rel}}} \\ = \frac{\gamma + 1}{2a_u} (J + \mathbf{V}_u \cdot \mathbf{n}). \end{aligned} \quad (15)$$

Fig. 4 Sketch of the present shock-fitting technique; *solid line*: meshes at $T = t$; *dashed line*: meshes at $T = t + \Delta t$



The left-hand side of Eq. (15) varies almost linearly with $M_{u,\text{rel}}$. Thus, Eq. (15) can be solved readily by Newton iterations using the value of $M_{u,\text{rel}}$ at t as a first guess. The four unknown parameters follow from Eqs. (13) and (14) once $M_{u,\text{rel}}$ is known.

For a 2D or 3D problem, the tangential velocity can readily be determined according to:

$$\mathbf{u}'_d \cdot \boldsymbol{\tau} = \mathbf{u}_u \cdot \boldsymbol{\tau}. \quad (16)$$

Here, $\boldsymbol{\tau}$ indicates the tangential direction to the boundary. Thus, the boundary velocity vector in the downstream region is rewritten as $\mathbf{u}_d = u'_d \mathbf{n} + (\mathbf{u}_u \cdot \boldsymbol{\tau}) \boldsymbol{\tau}$.

Step 3: Mesh deformation and re-meshing..

A key feature of this shock-fitting technique is that the motion and deformation of shock waves is described by a grid movement. The calculation in Step 2 provides not only the states of the shock nodes, but also their velocities. The shock fronts evolve during the calculations and their shapes and positions vary in time. Furthermore, the new positions of the internal grid points and the grid velocity can be calculated according to a mesh deformation strategy.

The internal nodes of the computational mesh move along with the fitted discontinuities during the entire computational process. As sketched in Fig. 4, the shockwave marked by the solid red line at $T = t$ moves to a new position marked by the dashed red line at $T = t + \Delta t$. With the motion of the fitted shock wave, internal nodes of the computational mesh move to new positions. Then, a new mesh with the same topological structure is obtained.

Combined with the mesh deformation strategy, a local re-meshing strategy is also implemented to solve moving boundary problems with large displacements. The grid quality is defined as: $q = \frac{2.R_1}{R_2}$ for triangles and as $q = \frac{3.R_1}{R_2}$ for tetrahedrons, where R_1 and R_2 represent the radii of in-circle (sphere for tetrahedrons) and circum-circle (sphere for tetrahedrons). The grid quality q varies from 0 to 1. We can evaluate the grid quality by using the defined grid quality factor. If the grid quality factor is less than a preset value, i.e., 0.01, the respective cells are of bad quality.

These skew cells and their neighboring cells are removed and the resulting hole is then re-meshed using new cells of better quality. In the process of re-meshing, a method for transferring flow information among meshes is utilized. It has been proven that no errors are introduced in either time or space [18].

Step 4: Solution update.

After the previous three steps, the solution can be determined at time level $t + \Delta t$ using the aforementioned shock-capturing algorithm. For finite volume methods, the process of determining the fluxes across each cell face is the most important. In the present shock-fitting technique, the fluxes across a common cell face can be determined using various flux schemes, such as van Leer (FVS), Roe (FDS), etc. By contrast, if the cell face is a shock face, the fluxes are easier to determine. In the present shock-fitting technique, the fluxes across a shock face can be computed

$$\Phi = [\mathbf{F}, \mathbf{G}, \mathbf{H}]_u^T \cdot \mathbf{n} = \begin{bmatrix} a_1 \\ a_1 \mathbf{V}_x + p n_x \\ a_1 \mathbf{V}_y + p n_y \\ a_1 \mathbf{V}_z + p n_z \\ a_1 E + p a_2 \end{bmatrix}_u, \quad (17)$$

where $a_1 = \rho(\mathbf{V} - w \cdot \mathbf{n}) \cdot \mathbf{n}$, $a_2 = \mathbf{V} \cdot \mathbf{n}$, $E = \frac{1}{2} \mathbf{V} \cdot \mathbf{V} + \frac{p}{\rho(\gamma-1)}$. w denotes the velocity of the shock faces, which can be computed by averaging the velocities of the shock nodes in *Step (2)*.

3 Applications

In this section, three cases are presented to assess the performance of the present shock-fitting technique.

3.1 Blunt Body Problem

First, we consider the high-speed flow past a circular cylinder at the free-stream Mach number, $M = 20$. The computational domain surrounds a semi-circular cylinder with a radius $R = 1$, as shown in Fig. 5. This problem has been studied by various approaches as a simple case of a shock in a non-uniform flow [20, 23]. Therefore, there are many results that can be used to evaluate the quality of the present shock-fitting results.

The POINTWISE mesh generator has been used to create a computational grid by specifying the distribution of the boundary nodes. These are evenly spaced and the distance between two adjacent boundary nodes of the computational grid equals $0.1R$. The computational mesh is made of 1,460 isotropic cells.

Fig. 5 Blunt body: computational domain

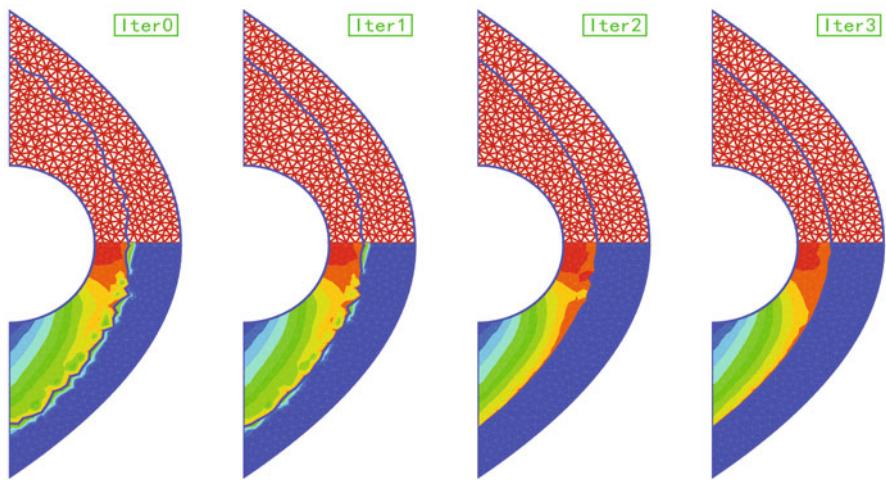
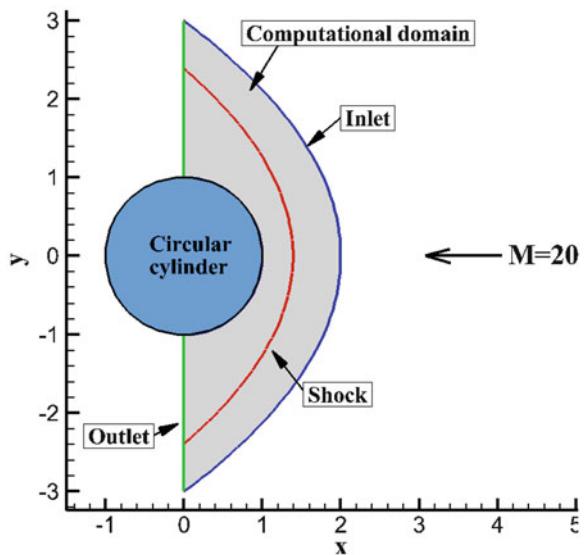


Fig. 6 Blunt body: process of shock convergenc

The captured solution is used to initialize the flow field and to determine an initial shock position as shown in Fig. 6(Iter0). Successively, the flow field and shock position are integrated in time until a steady state is reached. The shock front becomes smooth and the upstream state becomes uniform. Figure 7 shows the Mach number iso-contours computed using shock-fitting.

A quantitative analysis of the shock-fitting solutions was carried out by comparing the estimates of the shock position and pressure distribution computed by shock-fitting with the reference solution computed by Lyubimov and Rusanov [20].

Fig. 7 Blunt body: Mach number iso-contours

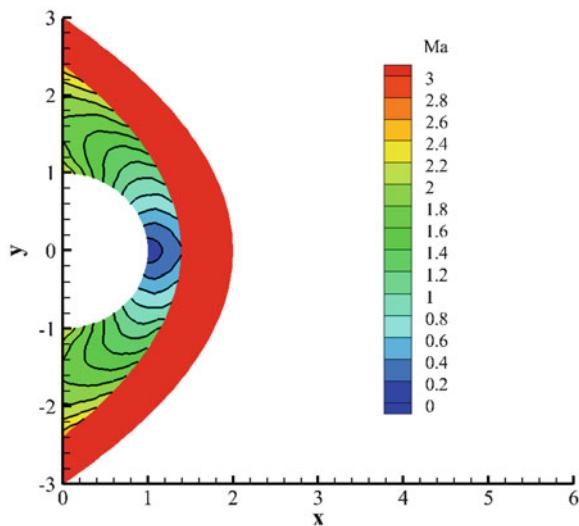
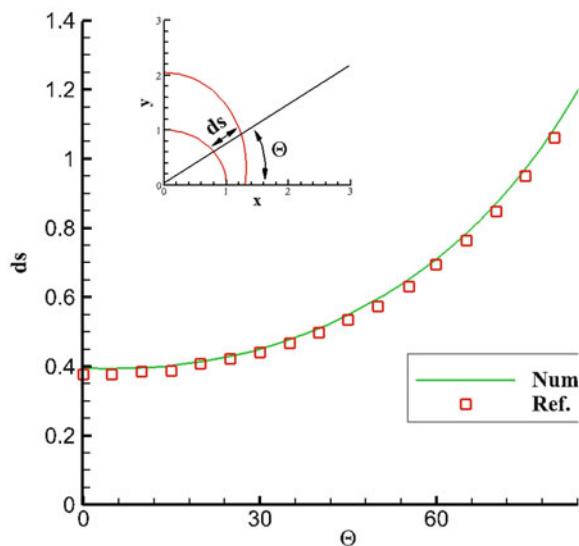


Fig. 8 Blunt body:
comparison of shock-wall
distance between the
computed solutions and
reference solution



In Figs. 8 and 9, the distance (ds) between the shock and the cylinder wall and the normalized pressure at the wall is plotted against the azimuthal angle (Θ). This comparison shows that the solutions computed by the proposed shock-fitting methodology are in good agreement with the reference solution.

Fig. 9 Blunt body: comparison of normalized pressure at wall between the computed solutions and reference solution

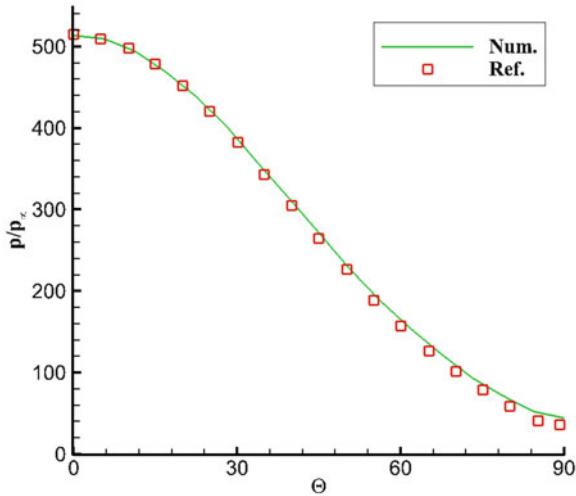
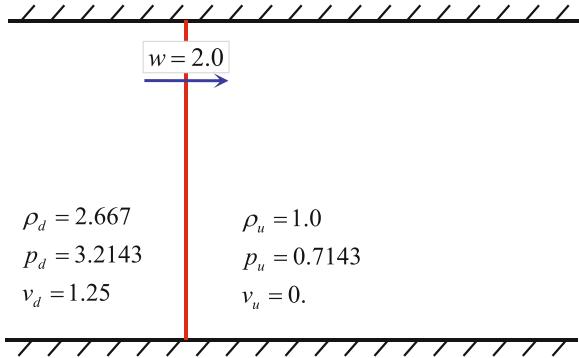


Fig. 10 Constant section duct: test case sketch



3.2 Shock Moving Through a Constant Section Duct

We consider a normal shock wave moving through a constant section duct. The shock front velocity is 2.0 in the x direction. Although it is very simple, this case can check the computational accuracy of the present algorithm. Figure 10 shows a sketch of the test case and the nondimensional upstream and downstream states.

The computational domain is a $[1.0 \times 0.5]$ rectangle and it is discretized using a triangular unstructured grid made of 2864 cells and 1535 nodes shown in Fig. 11. The shock wave is initially located at $x = 0.25$.

Based on the known location of the shock wave, we label all the nodes of $x = 0.25$ as shock nodes. In this case, there are 26 shock nodes and 25 shock faces. The fluxes across these labeled shock faces are computed using the algorithm presented in Sect. 2.3 and the fluxes across other faces are obtained by the van Leer scheme. In the process of computation, these shock nodes move versus time. Thus, the computational grid also appears distorted.

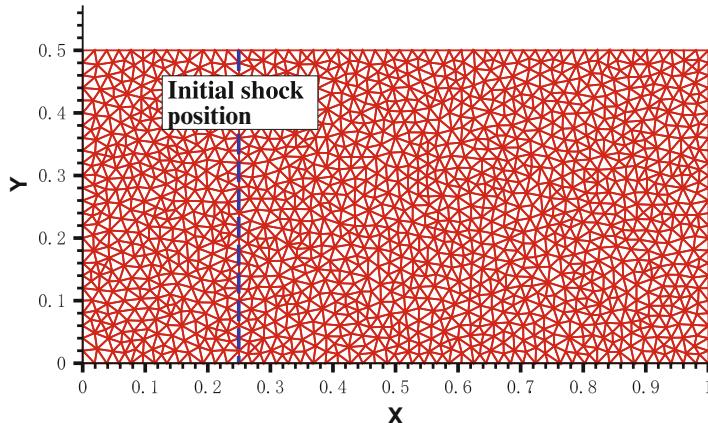


Fig. 11 Constant section duct: computational mesh

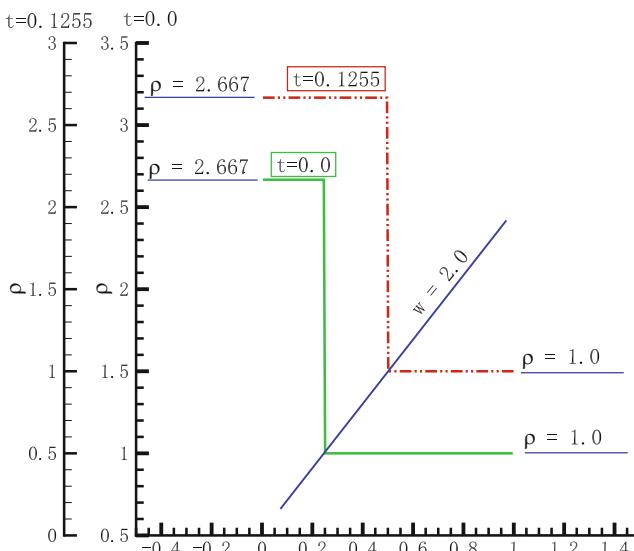
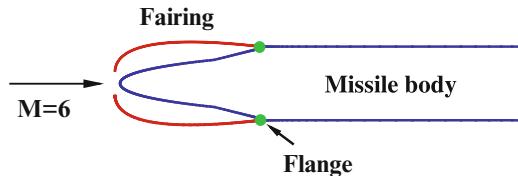


Fig. 12 Constant section duct: density distribution

Figure 12 shows the density distribution along $y = 0.2$. In this figure, the green and red lines represent the density distribution at $t = 0$ and $t = 0.1255$ respectively. The results show: the density jumps across the shock wave and the computed shock wave has no thickness, which conforms to the character of the shock wave described by the Euler equation. In addition, no spurious numerical oscillations occur. The shock wave in the flow field moves from the initial location at $x = 0.25$ to the location at $x = 0.5$ at $t = 0.1225$. The computational results show us the validity and satisfactory performance of the present algorithm.

Fig. 13 Fairing separation: sketch of the fairing separation model



3.3 Fairing Separation

As shown in Fig. 13, we consider a missile in a supersonic $M = 6$ stream. The missile body carries two pieces of fairings, which can rotate around flanges. At the beginning, the pitch angles are 5° and -5° respectively. The free stream flows inside the region surrounded by fairings and the missile body and increases the pressure inside this region. Because the pressure inside the fairings is higher than outside them, the fairings rotate around the flanges. Both the shock-capturing and shock-fitting techniques are used to simulate this process.

In the shock-fitting simulation, the upstream state in the shock points is set equal to the free stream states, whereas the initial downstream state is initialized by the steady solution computed by the shock fitting. Figure 14 displays the comparison between the shock-fitting and shock-capturing solutions obtained by the same code. It is clear that the captured shock waves have a thickness of a few cells, which is greater than the physical thickness.

4 Conclusion

In this work, a shock-fitting technique for cell-centered finite volume methods on unstructured dynamic meshes was developed. By defining the shock points, the values for the shock points can be computed correctly by solving R-H jump relations. Following this procedure, the fluxes across a shock face are set to equal the upstream fluxes. At the same time, the slope of the cell face was adjusted to align with the real shock surface using the unstructured dynamic meshes technique. Some simple 2D test cases are given and the numerical results show that the present shock-fitting technique retains the good feature of traditional shock-fitting methods. In the future, more complex cases, such as 3D problems with complex configuration or flow structure, unsteady problems with topological change, etc., will be considered.

Acknowledgements This work is partially supported by the National Natural Sciences Foundation of China under the grants of 91541117.

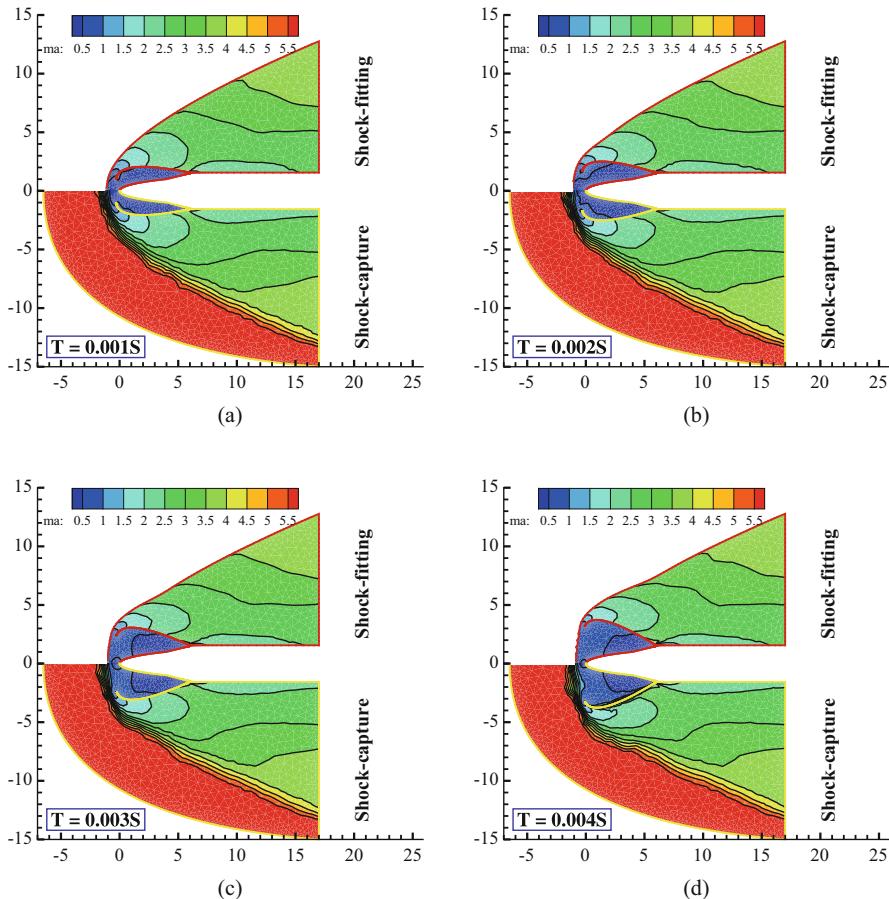


Fig. 14 Fairing separation: Mach contours change versus time during fairing separation. **(a)** T=1 ms. **(b)** T=2 ms. **(c)** T=3 ms. **(d)** T=4 ms

References

- Abbett, M., Moretti, G.: A time-dependent computational method for blunt body flows. *AIAA J.* **4**(12), 2136–2141 (1966). <https://doi.org/10.2514/3.3867>
- Arora, M., Roe, P.L.: On postshock oscillations due to shock capturing schemes in unsteady flows. *J. Comput. Phys.* **130**(1), 25–40 (1997). <https://doi.org/10.1006/jcph.1996.5534>
- Batina, J.T.: Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA J.* **28**(8), 1381–1388 (1990). <https://doi.org/10.2514/3.25229>
- Bonfiglioli, A., Grottadurea, M., Paciorri, R., Sabetta, F.: An unstructured, three-dimensional, shock-fitting solver for hypersonic flows. *Comput. Fluids* **73**, 162–174 (2013). <https://doi.org/10.1016/j.compfluid.2012.12.022>
- Carpenter, M.H., Casper, J.H.: Accuracy of shock capturing in two spatial dimensions. *AIAA J.* **37**, 1072–1079 (1999). <https://doi.org/10.2514/2.835>

6. Dadone, A., Grossman, B.: A multi-dimensional upwind scheme for the Euler equations. In: 13th International Conference on Numerical Methods in Fluid Dynamics: Proceedings of the Conference Held at the Consiglio Nazionale delle Ricerche, vol. 414, pp. 95–99 (1993)
7. Guo, Z.: Numerical simulation technique research for unsteady multi-body flow field involving moving boundaries. National University of Defense Technology, Changsha (2002)
8. Harten, A.: High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **135**, 260–287 (1997)
9. Harten, A., Hyman, J.M., Lax, P.D.: On finite-difference approximations and entropy conditions for shocks. *Commun. Pure Appl. Math.* **29**(3), 297–322 (1976)
10. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order essentially non-oscillatory schemes. *J. Comput. Phys.* **131**(1), 3–47 (1987). <https://doi.org/10.1006/jcph.1996.5632>
11. Hartwich, P.M.: Fresh look at floating shock fitting. *AIAA J.* **29**(7), 1084–1091 (1990). <https://doi.org/10.2514/3.10707>
12. Kopriva, D.A.: Shock-fitted multidomain solution of supersonic flows. *Comput. Methods Appl. Mech. Eng.* **175**, 383–394 (1999). [https://doi.org/10.1016/S0045-7825\(98\)00362-4](https://doi.org/10.1016/S0045-7825(98)00362-4)
13. Lax, P.D.: Weak solutions of nonlinear hypersonic equations and their numerical computation. *Commun. Pure Appl. Math.* **7**, 159–193 (1954)
14. Lax, P.D., Wendroff, B.: Systems of conservation laws. *Commun. Pure Appl. Math.* **13**, 217–237 (1960)
15. Lee, T.K., Zhong, X.: Spurious numerical oscillations in simulation of supersonic flows using shock-capturing schemes. *AIAA J.* **37**(3), 313–319 (1999). <https://doi.org/10.2514/2.732>
16. Li, S.: Theory of Dissipative Conservation Schemes. High Education Press, Beijing (1997)
17. Liu, X., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **115**(1), 200–212 (1994). <https://doi.org/10.1006/jcph.1994.1187>
18. Liu, J., Bai, X., Wang, W.: High accurate method for transferring flow information among meshes and its validation. *Acta Aerodyn. Sin.* **28**(01), 1–6 (2010)
19. Liu, J., Bai, X., Guo, Z.: The computational method of unstructured moving grids. National University of Defence Technology Press, Changsha (2009)
20. Lyubinov, A., Rusanov, V.: Gas Flow Past Blunt Bodies. Part II: Tables of Gasdynamic Functions. Translation of Nauka Press, Moscow (1970)
21. MacCormack, R.W.: The effect of viscosity in hypervelocity impact cratering. *J. Spacecr. Rocket.* **40**(5), 757–763 (2003)
22. Nasuti, F., Onofri, M.: Analysis of unsteady supersonic viscous flows by a shock-fitting technique. *AIAA J.* **34**(7), 1428–1434 (1996). <https://doi.org/10.2514/3.13249>
23. Paciorri, R., Bonfiglioli, A.: A shock-fitting technique for 2d unstructured grids. *Comput. Fluids* **38**(3), 715–726 (2009). <https://doi.org/10.1016/j.compfluid.2008.07.007>
24. Paciorri, R., Bonfiglioli, A.: Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique. *J. Comput. Phys.* **230**(8), 3155–3177 (2011). <https://doi.org/10.1016/j.jcp.2011.01.018>
25. Prakash, A., Parsons, N., Wang, X., Zhang, X.: High-order shock-fitting methods for direct numerical simulation of hypersonic flow with chemical and thermal nonequilibrium. *J. Comput. Phys.* **230**(23), 8474–8507 (2011). <https://doi.org/10.1016/j.jcp.2011.08.001>
26. Rawat, P.S., Zhong, X.: On high-order shock-fitting and front-tracking schemes for numerical simulation of shock-disturbance interactions. *J. Comput. Phys.* **229**(19), 6744–6780 (2010). <https://doi.org/10.1016/j.jcp.2010.05.021>
27. Richtmyer, R.D.: Difference Methods for Initial-Value Problems, 2nd edn. Interscience Tracts in Pure and Applied Mathematics (1967). <https://doi.org/10.1063/1.3060778>
28. Rumsey, C.L., van Leer, B., Roe, P.L.: A grid-independent approximate Riemann solver with applications to the Euler and Navier-Stokes equations. In: 29th Aerospace Sciences Meeting, pp. 1–12 (1991)
29. Salas, M.D.: Shock fitting method for complicated two-dimensional supersonic flows. *AIAA J.* **14**(5), 583–588 (2009). <https://doi.org/10.2514/3.61399>
30. Salas, M.D.: A Shock-Fitting Primer. CRC Press, Boca Raton (2010)

31. Trepanier, J.Y., Paraschivoiu, M., Reggio, M., Camarero, R.: A conservative shock fitting method on unstructured grids. *J. Comput. Phys.* **126**(2), 421–433 (1996). <https://doi.org/10.1006/jcph.1996.0147>
32. Venkatakrishnan, V.: On the accuracy of limiters and convergence to steady state solutions. In: 31st Aerospace Sciences Meeting, pp. 1–10 (1993). <https://doi.org/10.2514/6.1993-880>
33. Von Neumann, J., Richtmyer, R.D.: A method for the numerical calculation of hydrodynamic shocks. *J. Appl. Phys.* **21**, 232–237 (1950). <https://doi.org/10.1063/1.1699639>
34. Yee, H.C., Harten, A.: Implicit TVD schemes for hyperbolic conservation laws in curvilinear coordinates. *AIAA J.* **25**(2), 266–274 (1987). <https://doi.org/10.2514/3.9617>
35. Yee, H.C., Sweby, P.K.: Aspects of numerical uncertainties in time marching to steady-state numerical solutions. *AIAA J.* **36**(5), 712–724 (1998). <https://doi.org/10.2514/2.459>
36. Yee, H.C., Klopfer, G.H., Montagni, J.L.: High-resolution shock-capturing schemes for inviscid and viscous hypersonic flows. *J. Comput. Phys.* **88**(1), 31–61 (1990)
37. Zhang, H.: The exploration of the spatial oscillations in finite difference solutions for Navier-Stokes shocks. *Acta Aerodyn. Sin.* **1**, 12–19 (1984)
38. Zhang, H.: Non-oscillatory and non-free-parameter dissipation difference scheme. *Acta Aerodyn. Sin.* **2**, 143–165 (1988)
39. Zou, D., Xu, C., Dong, H., Liu, J.: A shock-fitting technique for cell-centered finite volume methods on unstructured dynamic meshes. *J. Comput. Phys.* **345**, 866–882 (2017)

Inserting a Shock Surface into an Existing Unstructured Mesh

Daniel Zaide and Carl Ollivier-Gooch

1 Introduction

To succeed, unstructured mesh shock fitting schemes require a meshing approach that will add a shock front to the mesh while leaving the remainder of the mesh essentially unchanged. Minimizing changes to the mesh reduces the time and effort required to interpolate the solution from one mesh to another, which in turn reduces the impact of the interpolation on solution accuracy.

As a result, regenerating a mesh with the domain boundary modified to include the shock is a poor alternative from the point of view both of the requirements of the shock fitting algorithm and of the time required for meshing. We propose instead to directly insert a surface into the mesh used in the simulation, with only local modifications.

Most Delaunay refinement mesh generation schemes [1, 3, 5, 6, 8, 13, 15, for example] begin by inserting the points defining the domain boundary, or some sampling of that boundary, into a background mesh that bounds the domain. Next, the faces of the boundary are recovered in some way to produce an initial, very coarse and very poor quality mesh of the domain. While one can argue that this is in some sense insertion of a surface into an existing mesh, this is not adequate for our purposes.

D. Zaide

Advanced Numerical Simulation Laboratory, The University of British Columbia,
Vancouver, BC, Canada

Synopsys, Inc., Mountain View, CA, USA
e-mail: dan.zaide@gmail.com

C. Ollivier-Gooch (✉)

Advanced Numerical Simulation Laboratory, The University of British Columbia,
Vancouver, BC, Canada
e-mail: cfo@mech.ubc.ca

There are two primary deficiencies in simply applying such a surface recovery technique for adding a shock surface¹ to an existing computational mesh. First, the existing mesh has a length scale associated with it, and the discretization of the surface we add must be compatible with that length scale. Second, unless care is taken, adding an arbitrary surface to an existing unstructured mesh can produce arbitrarily short edges and arbitrarily bad cells, which we wish to avoid. To the best of our knowledge, ours is the only scheme designed to insert a surface in an existing unstructured mesh and produce results of guaranteed mesh quality.

We begin with two inputs, an initial unstructured mesh and a discrete surface.

In 2D, all initial meshes shown here are produced using Shewchuk's improvement on Ruppert's 2D Delaunay refinement algorithm, with extensions for varying length scale and curved boundaries [10, 13, 15], guaranteeing a minimum angle of 25.66° . The curve is defined by a set of Bezier spline control points, which we interpolate using a standard parametric cubic spline interpolation. The curve is then sampled on the mesh to determine length scales at various points on the curve, creating an initial distribution of vertices to insert in the mesh. We then move these vertices along the curve using the principle of equidistribution, resulting in a vertex distribution with spacing comparable to the underlying mesh.

In 3D, our initial meshes are again produced by Delaunay refinement (Shewchuk's 3D scheme [14]). In 3D, the surface to be inserted is given by a triangulation. Again, we produce a spline interpolation of the surface using the Common Geometry Module [16]. We discretize the curves bounding the surface just as in 2D, then sample the surface [7] to produce a new surface triangulation that has good geometric quality and a length scale comparable to the volume mesh.

With the new vertex locations determined, vertices in the original mesh that are near the new surface are removed by edge contraction [9], creating an open region in the mesh to insert these new points. The surface vertices are then safe to insert, and the mesh is reconnected appropriately. Recovery of the new surface is easy by design: the vertices removed from the initial mesh ensure that entities on the surface are not encroached. Local refinement is then used to clean up the mesh. A final check is performed to ensure an appropriate mesh quality is retained. The overall algorithm for surface insertion is outlined in Fig. 1 and described in detail below.

2 Approach

2.1 Curve Discretization

The length scale utilized on the curve is first defined from a vertex length scale function $LS(v)$. While there are many choices for vertex length scale on an existing

¹We will use “surface” to mean a $d - 1$ -dimensional curved entity embedded in a d -dimensional space. That is, when it is not ambiguous to do so, we will use surface to refer to a two-dimensional surface that we wish to insert into a volume mesh and to a one-dimensional curve that we wish to insert into a two-dimensional mesh.

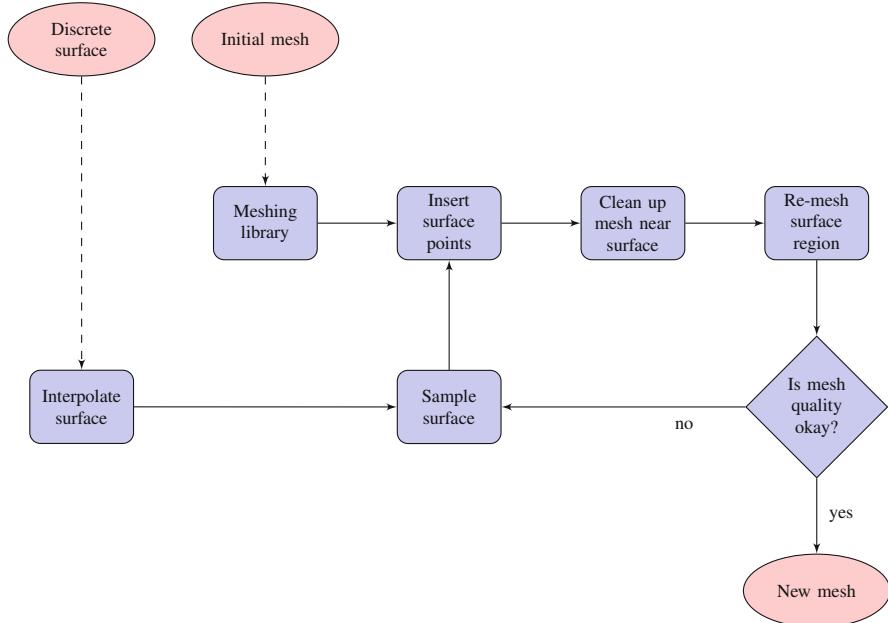


Fig. 1 Algorithm flowchart

mesh, in this work we use the radius of a sphere around the vertex based on neighboring vertices, calculated as

$$LS(v) = \left(d \sum_{i=1}^N V_i / \sum_{i=1}^N \theta_i \right)^{\frac{1}{d}} \quad (1)$$

where d is the spatial dimension of the problem, N is the number of neighboring cells, with volume V_i for neighbor i , which has a solid angle of θ_i at vertex v . With this definition, the length scale at any point can be determined by barycentric interpolation within a cell. Consider a location (x, y, z) in a simplex. The length scale is then

$$LS(x(t), y(t), z(t)) = \sum_{i=1}^{d+1} \lambda_i LS(v_i) \quad (2)$$

for barycentric coordinates λ_i corresponding to (x, y, z) . This leads to a continuous piece-wise linear description of the length scale on the boundary curve, which we will use in the discretization of our curve and subsequent surface sampling. Lastly, to ensure the curve is sufficiently resolved, we use the smaller of the length scale and the radius of curvature such that the length scale is

$$LS(t) = \min (LS(x(t), y(t), z(t)), \rho(t)) \quad (3)$$

where t is a parametric coordinate along the curve and $\rho(t)$ is the radius of curvature for the curve.

We begin our curve discretization by defining the parametric cubic B-spline representation of the curve by parameter t and coordinates $(x(t), y(t))$, with arc length $\ell(t)$ such that $\ell(0) = 0$. We use the length scale defined above.

We define the computational domain by ξ running from 0 to N , and the length of the curve in this space as $\ell = \ell(\xi)$. The governing equation for equidistribution of length scale along a curve is the steady-state solution to the moving mesh PDE (MMPDE), written in one dimension as

$$\frac{d}{d\xi} \left(\frac{1}{LS(\ell)} \frac{d\ell}{d\xi} \right) = 0. \quad (4)$$

As we only interested in getting a “good” spacing and not in solving this PDE exactly, we solve Eq. 4 approximately using Gauss-Seidel iteration from an initial spacing of N points, defining discrete arc-lengths $i = 1, \dots, N$ and updating as

$$\ell_i^{n+1} = \frac{LS\left(\ell_{i-\frac{1}{2}}^{n+1}\right) \ell_{i+1}^n + LS\left(\ell_{i+\frac{1}{2}}^n\right) \ell_{i-1}^{n+1}}{LS\left(\ell_{i-\frac{1}{2}}^{n+1}\right) + LS\left(\ell_{i+\frac{1}{2}}^n\right)} \quad (5)$$

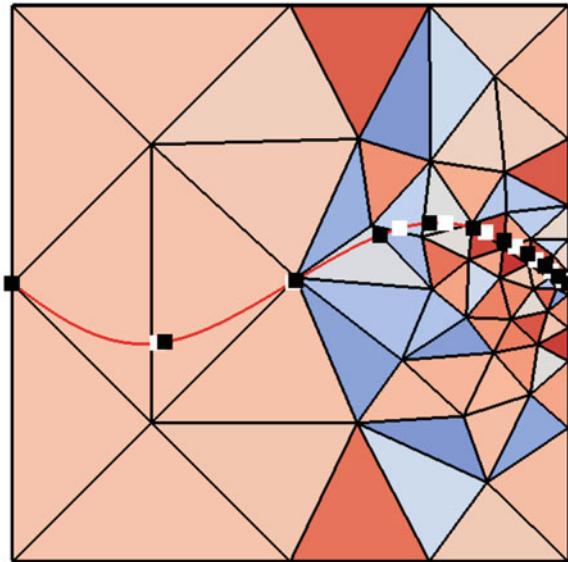
until the L_2 norm of the change in point positions is small.

To determine the initial spacing, a point is placed at the beginning of the curve ($\xi = 0$) and we work our way along the curve, placing points based on the length scale of the previous point, until we get close to the end. Towards the end of the curve, points are placed until one falls beyond the end of the curve; this last point is placed at the end of the curve instead. This leads to a reasonably good initial spacing on the curve, much better than a uniform or geometric spacing would provide, giving us the white points shown in Fig. 2. This also determines N , the number of points. Next, we solve the MMPDE, producing the black points in Fig. 2, which will be inserted into the mesh; at this stage, we do not add or remove points as the MMPDE is solved, though recent research has explored and developed this idea [12].

2.2 Surface Discretization

With the boundary curve spacing determined, we use the sampling algorithm of Gosselin and Ollivier-Gooch [6] to triangulate the surface based on an initial triangulation from boundary curve vertices. This algorithm samples each smooth surface, first by sampling boundary curves, and then by refining the restricted Delaunay triangulation, targeting topological violations and poor quality triangles.

Fig. 2 Initial (white) and final (black) points to insert



We build off of this, using our equidistributed boundary curve discretization as the boundary curve sampling. We do not solve the moving mesh equation on the surface due to the computational expense. To obtain a surface sampling based on the length scale of the initial mesh, we refine the surface triangulation until

$$\sqrt{2}r_{\text{circum}} < \frac{1}{3} \sum_{i=1}^3 LS(\mathbf{x}(v_i)) \quad (6)$$

is satisfied for each surface triangle, where the length scale is computed using barycentric interpolation of the vertex coordinates on the initial mesh. We then continue sampling to meet minimum mesh quality bounds for the surface triangulation; there are theoretical guarantees that we can meet the same quality bounds for the surface mesh that we cited above for 2D meshes. See [6] for details.

For shock-fitting applications, it is important to note that this surface sampling scheme can easily handle non-manifold surfaces. Non-manifold curves, like triple lines, can be discretized like any other curve bounding a surface. Gosselin's sampling scheme by design samples each smooth surface part of a piecewise smooth surface separately, making a non-manifold collection of surfaces no more difficult to sample properly than a manifold collection of surfaces.

Figure 3 illustrates the sampling process for a simple bilinear surface in 3D. The original surface (Fig. 3a) is given as a triangulation, whose edges are clearly visible thanks to differences in shading induced by changes in the surface normal. We sample the boundary curves (Fig. 3b) and then the surface itself (Fig. 3c).

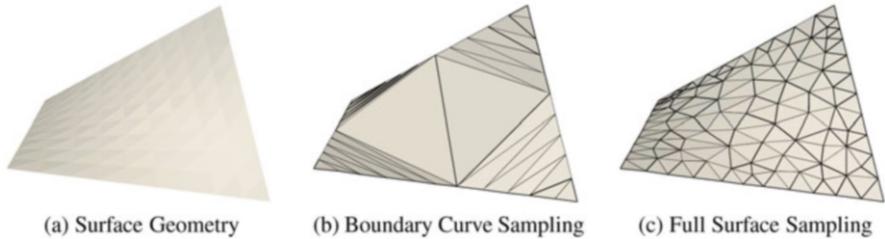


Fig. 3 Representative sampling process. (a) The initial surface. (b) Equidistribution of points along the boundary curve. (c) The surface is refined to match the length scale of the initial mesh

2.3 Creating a Cavity in the Mesh

Once the locations of the new vertices on the surface have been determined, the mesh needs to be prepared for their insertion. We have two goals here, both consistent with ensuring that we can produce quality isotropic meshes at the end of the process.

We need to prevent the formation of short edges in the mesh, so we must ensure the new vertices will be inserted sufficiently far from existing vertices. To do this, all vertices in the original mesh that are within one half of a length scale from any vertex in the surface are flagged for removal.

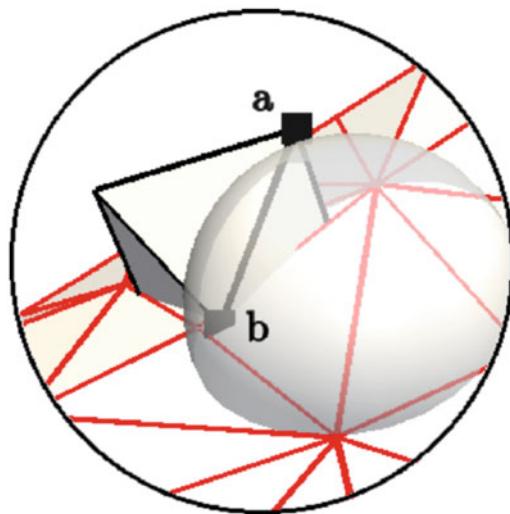
We also need to ensure that we create no small dihedral angles involving points in the original mesh and faces on the new surface. To accomplish this, we flag for removal all vertices that encroach on any of the surface faces.² This has the additional advantage that all the surface faces are guaranteed to exist in a Delaunay mesh once their points are all inserted.

An example of this approach is shown in Fig. 4, where vertex a is flagged for removal because it is too close to a surface vertex and vertex b is flagged for removal because it encroaches on the surface face. We use a tree-based geometric search to identify these vertices, with time complexity of $N_f \log N_v$, where N_f is the number of surface faces and N_v is the number of vertices in the volume mesh.

To remove vertices from the mesh, we use the edge contraction algorithm of Ollivier-Gooch [9]. This algorithm removes vertices by moving a vertex on to one of its incident vertices, collapsing an edge. To determine which edge to collapse, all incident vertices are checked, and the incident vertex that preserves boundary topology, mesh validity (no inverted cells), and has the maximum quality for the resulting mesh fragment is chosen. In this work, the quality criterion used during edge contraction maximizes the minimum sine of angles (dihedral angles in 3D).

²That is, we remove all points lying inside the equatorial ball of any surface face. The equatorial ball of a face is the (unique) smallest ball that passes through all of its vertices.

Fig. 4 Identifying vertices to remove. This representative figure shows a single tetrahedron of the volume mesh cut by the sampled surface. The vertices highlighted in black are flagged for removal: vertex **a** is flagged as it is too close to a vertex in the surface mesh and vertex **b** is flagged for encroaching on a surface face



In three dimensions, there can be vertices that cannot be removed with edge contraction, as there are no incident edges to collapse that would result in a valid mesh. This is especially true on the boundary, because surface vertices cannot be moved onto interior vertices, limiting the choice of candidate incident vertices. In our approach, these vertices are left in the mesh, and are removed during surface insertion, where the changes in mesh topology due to insertion generally allow for their removal. In practice, this results in the removal of all flagged vertices after surface insertion, and a locally Delaunay mesh with well-spaced vertices near the incoming surface.

Figure 5 shows the results of vertex removal. Because point removal is a strictly local operation, the mesh changes only near the surface (within about one length scale of surface being inserted), leaving the rest of the mesh untouched.

2.4 Inserting the New Surface

With the vertices removed and a region for the surface cleared out, the new vertices on the sampled surface can be inserted into the mesh, and the surface can be recovered. The vertices can be inserted by using the Watson algorithm [17] to produce a Delaunay mesh directly. If all vertices flagged for removal were successfully removed, then no faces or edges in the new surface can be encroached, and the surface is automatically recovered in the Delaunay mesh. However, it is not always possible to remove all vertices tagged for removal from the volume mesh by edge contraction, which nullifies our guarantee. Because of this, to improve robustness in 3D, we choose instead to insert vertices by splitting their containing cell into four new cells and swapping to recover each face. After the three vertices

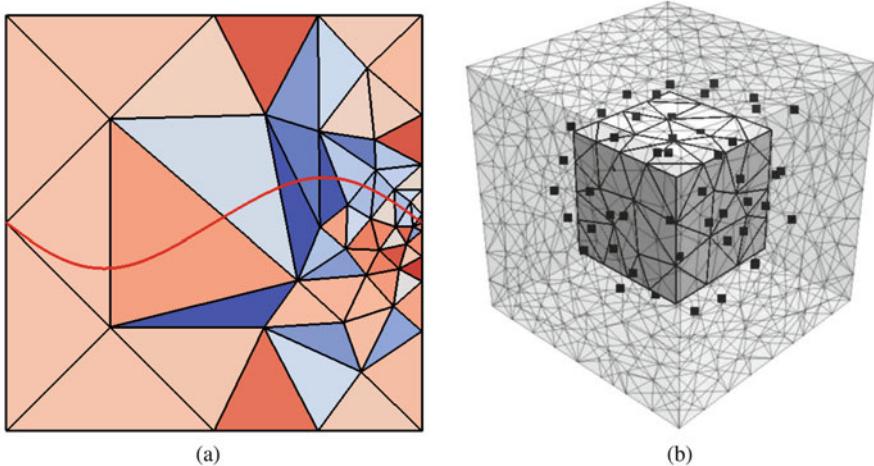


Fig. 5 Removing points near a surface to be inserted into a mesh. **(a)** In two dimensions. **(b)** Points to be removed in three dimensions

for a surface triangle (or two vertices for an edge, in 2D) are inserted, we swap faces and edges to recover the surface face, which is then locked as an internal boundary face. As we have removed most or all of the vertices that would have encroached on the newly inserted faces, they are easily recovered, and are often present in the mesh simply from insertion, even though the mesh is not locally Delaunay. After surface recovery, we make a second attempt to remove vertices whose removal failed in the first pass, usually removing nearly all of the remainder. The results of this insertion, prior to any mesh cleanup, are shown in Fig. 6.

2.5 Mesh Cleanup

After inserting all of the vertices on a surface and recovering, the surface has now been completely inserted into the initial mesh. This surface insertion process is designed for robust surface recovery rather than mesh quality, so there are often poor quality cells near the newly inserted surface. These cells are almost always slivers, because our length scale based surface sampling results in well-spaced vertices; in particular, there can be no edges that are shorter than half the local length scale.

The next step is mesh cleanup, where these poor quality cells are flagged and removed from the mesh and the final mesh is obtained. Of the wealth of available techniques for removing badly shaped cells and improving the overall mesh quality, we make use of three: refinement, swapping, and smoothing. As only cells near the surface have been modified, all of these techniques are applied locally, on cells with at least one vertex affected by the insertion procedure.

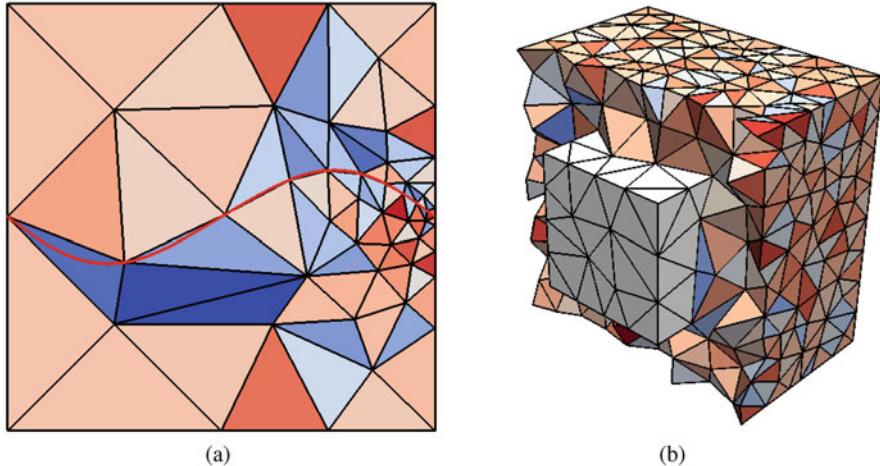


Fig. 6 Inserting points into the original mesh and recovering the new surface. Elements are colored by smallest (dihedral) angle. **(a)** In two dimensions. **(b)** In three dimensions

2.5.1 Two Dimensions

We use the refinement techniques described in [1, 11]. These involve inserting vertices at the circumcenters of badly shaped cells until the desired quality is met. As only cells near the curve have been modified, refinement is done locally. We also intentionally avoid boundary edge splitting to preserve our length scale based spacing, with the exception of small angle protection, done prior to refinement by inserting onto boundary edges using concentric circles, as described by Miller et al. [8]. While this adds additional vertices on the sampled surface that do not correspond to the original length scale, they guarantee the termination of the refinement algorithm around small angles in the inserted geometry. Following refinement, one pass of the smoothing of Freitag and Ollivier-Gooch [4] is done on all vertices adjacent to the curve. Smoothing further improves the mesh around the newly inserted surface, resulting in a final mesh that not only contains the inserted curve, but has maintained a reasonable quality while only affecting a small number of vertices. An example is shown in Fig. 7a.

2.5.2 Three Dimensions

For refinement, we use standard Delaunay refinement with circumcenter insertion, as described in [7]. Refinement is done only on the poor quality cells identified near the surface. We also intentionally avoid boundary edge splitting at this stage to preserve our length scale based spacing. Following the refinement procedure, the vertex smoothing and face/edge swapping of Freitag and Ollivier-Gooch [4] is

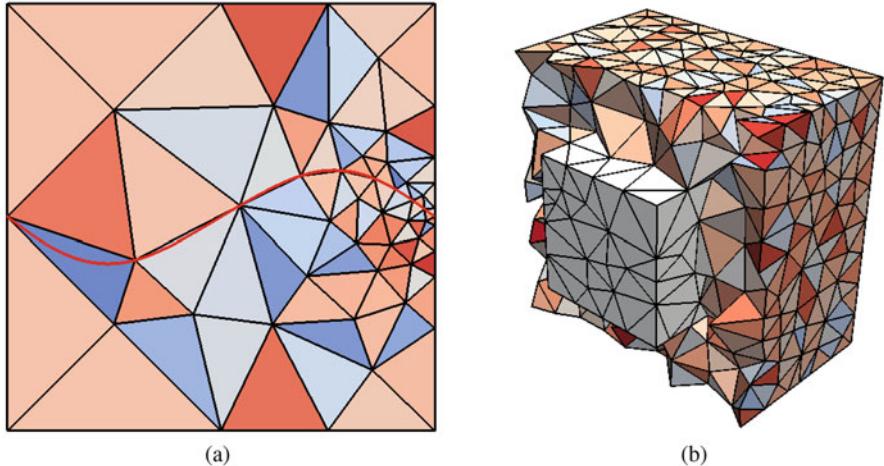


Fig. 7 Example meshes after cleanup. Elements are colored by smallest (dihedral) angle on the same scale as before. **(a)** In two dimensions. **(b)** In three dimensions

performed on cells affected by the surface insertion. In this work, two passes of swapping are done, followed by two passes of smoothing. We have demonstrated elsewhere that this results in a reasonable quality mesh near the surface, with the majority of initial mesh left unchanged. While other combinations of refinement, swapping, and smoothing could be used, in practice this is sufficient for our needs, as shown in Fig. 7b.

3 Examples

3.1 Two Dimensions

Here we present two additional examples of curve insertion into existing meshes, as well as reprising the example shown step by step above. Figure 8 shows these three examples. On the left, we show the before-and-after pictures for inserting a cubic spline into a uniform mesh in a square. In the center is the example shown above, with the same cubic spline inserted into a non-uniform mesh. Finally, on the right, we show the result of inserting a maple leaf outline into a mesh in a circle; because of small radius of curvature in the inserted curve, there are inevitably some short edges—and hence small triangles—created in this mesh, to simultaneously satisfy geometric and mesh quality requirements. In each case, the final mesh shows similar mesh quality to the initial mesh with no major changes in mesh resolution. We also provide quantitative information about these three cases in Table 1. The table shows

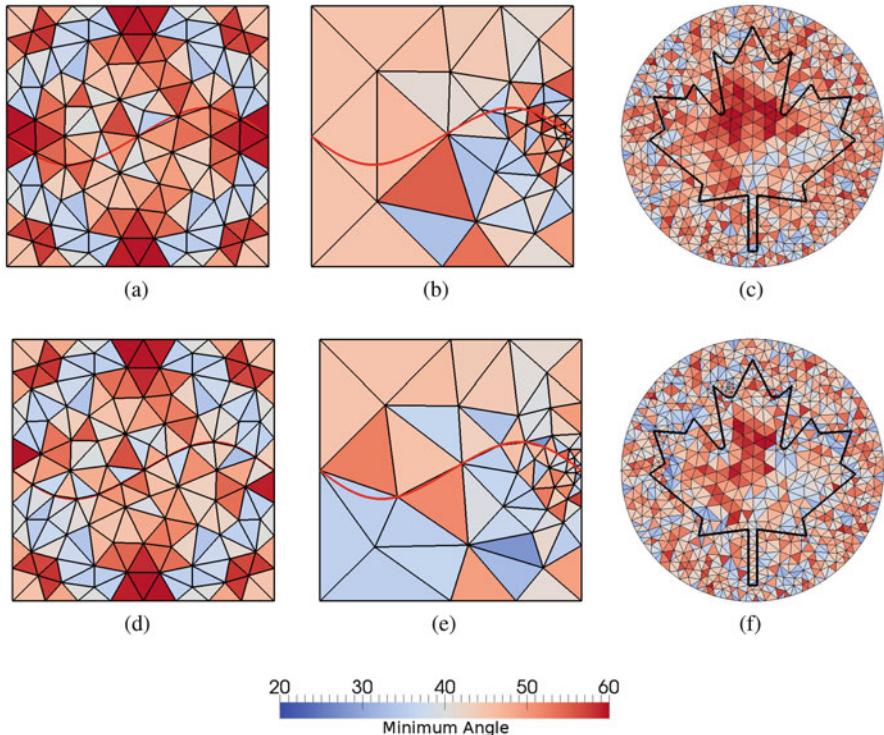


Fig. 8 Two-dimensional examples. These are the coarse meshes of Table 1. (a) Initial mesh and curve, spline with a uniform background mesh. (b) Initial mesh and curve, spline with a non-uniform background mesh. (c) Initial mesh and curve, maple leaf with a uniform background mesh. (d) Final mesh and curve, spline with a uniform background mesh. (e) Final mesh and curve, spline with a non-uniform background mesh. (f) Final mesh and curve, maple leaf with a uniform background mesh

initial and final mesh size and quality data; final mesh quality meets the guarantees of the post-processing refinement scheme, while mesh size is relatively unchanged. Note also that nearly all vertices are unaffected by curve insertion; even for the maple leaf case, which has a much longer curve relative to local mesh length scale, only a few percent of vertices are affected by adding the curve to a fine mesh. The middle columns of the table show the number of vertices on the curve, removed near the curve, added during refinement to meet mesh quality requirements, and moved by mesh smoothing.

Table 1 Summary of 2D results

Example	Initial			Number of vertices ..			Final		
	Verts	Min∠	Removed	On curve	Refine	Smoothed	Verts	Min∠	Verts unchanged
Spline, uniform mesh	C	104	34.02°	7	11	0	2	108	33.31°
	M	8315	30.24°	71	112	0	42	8356	30.01°
	F	40,315	30.24°	157	253	0	110	40,411	30.04°
Spline, non-uniform mesh	C	49	30.40°	7	8	0	7	50	30.18°
	M	8254	30.17°	74	107	0	51	8287	29.04°
	F	33,952	30.06°	214	316	0	117	34,054	30.06°
Maple leaf	C	614	30.01°	66	125	24	80	697	25.84°
	M	14,601	30.07°	356	560	38	296	14,843	26.63°
	F	57,077	30.30°	719	1105	36	640	57,499	26.78°

C = coarse, M = medium, F = fine

3.2 Three Dimensions

In three dimensions, we present two additional cases, a twisted torus and a mechanical part, whose geometries are shown in Fig. 9. Each of these geometries is inserted into an existing mesh in a cube. Figure 10a, b shows the surfaces, sampled to match the length scale of the original mesh, while Fig. 10c, d shows the final coarse meshes, with tetrahedra colored by their smallest dihedral angle. Note that the mechanical part imprints the bottom of the domain into which it is inserted. To demonstrate that the resulting meshes are of good quality, regardless of mesh resolution, we show histograms of dihedral angles for three mesh densities for both of these cases (Fig. 11 for the twisted torus geometry and Fig. 12 for the mechanical part geometry). Finally, as for the 2D examples, we provide a quantitative summary for these 3D examples in Table 2.

3.3 A Two-Dimensional Shock Fitting Example

We turn now to a simple shock fitting example. The physical problem is a supersonic flow of Mach 1.85 around a cylinder. Our example focuses on the mesh modification required to represent the shock, without regard here to the flow physics. Information about the shock is given in the form of a polyline identifying points on the shock.³ Figure 13a shows the initial (very coarse) mesh and the spline interpolation of the shock points. As before, we sample the curve (Fig. 13b). Next, we identify points that are too close to the curve or to the sample points to be consistent with good mesh quality (Fig. 13c), and remove these points (Fig. 13d). Then we insert the sample points into the mesh (Fig. 13e); note that more points are added on the shock near

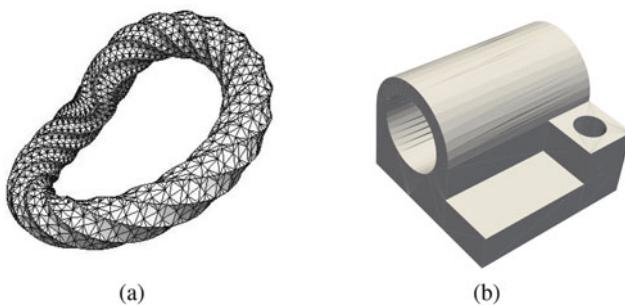


Fig. 9 Surface representations for 3D test cases. (a) Twisted torus geometry. (b) Mechanical part geometry

³Our thanks to Renato Paciorri and Aldo Bonfiglioli for providing the shock data for this test case.

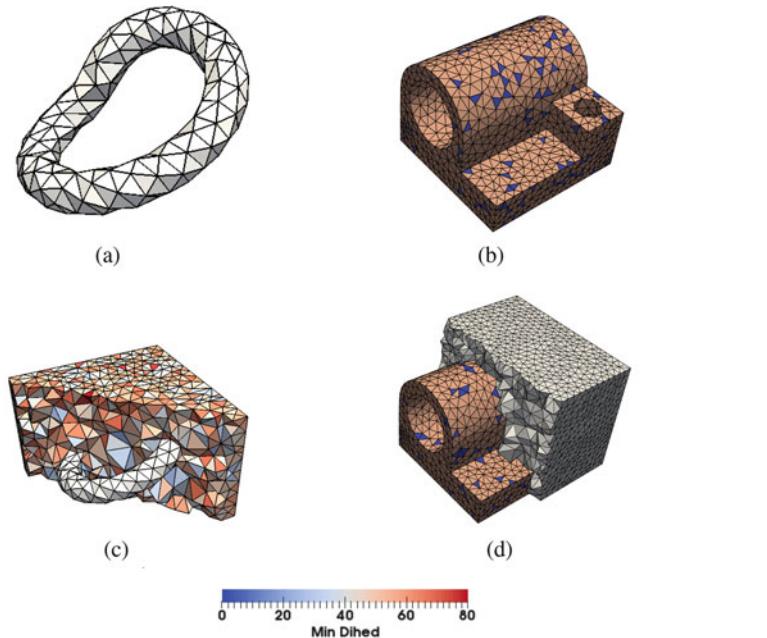


Fig. 10 Inserting the objects of Fig. 9 into meshes in a cube. These are the coarse meshes in Table 2. (a) Coarse sampling of the twisted torus. (b) Coarse sampling of the mechanical part. (c) Cutaway of final torus mesh. (d) Cutaway of final mechanical part mesh

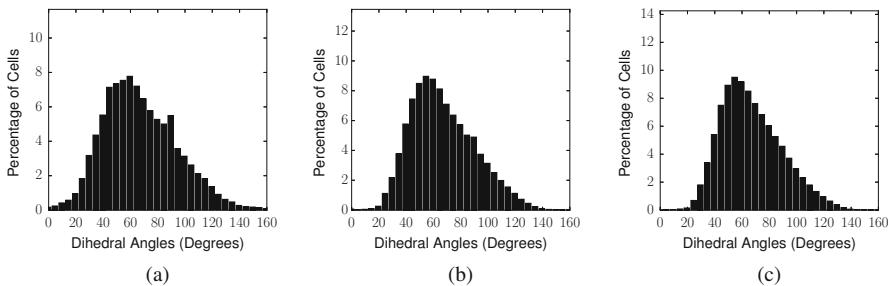


Fig. 11 Distribution of dihedral angles for the twisted torus meshes. (a) Coarse mesh. (b) Medium mesh. (c) Fine mesh

the boundary because of small angle considerations. Finally, we improve the mesh quality, only touching points that are first or second neighbors of points inserted on the shock (Fig. 13f). As with the previous examples, the output mesh quality exceeds the mesh quality guarantees for Delaunay refinement mesh generation schemes.

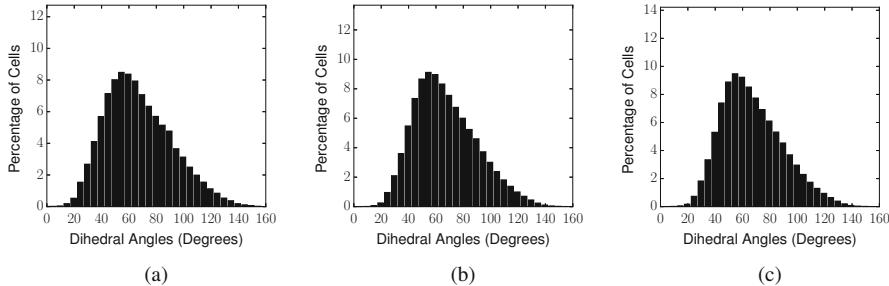


Fig. 12 Distribution of dihedral angles for the mechanical part meshes. (a) Coarse mesh. (b) Medium mesh. (c) Fine mesh

Table 2 Summary of 3D results

Example		Initial	Surface Recovery		Final	
		Verts	Verts removed	Verts added	Verts	% Unch.
Twisted torus	C	6616	517	973	7125	75%
	M	43,525	2150	2993	44,402	87%
	F	317,218	9994	9164	316,340	93%
Mechanical part	C	6854	853	1545	7333	66%
	M	45,775	3519	5468	47,201	78%
	F	196,637	9269	16,124	202,433	96%

C = coarse, M = medium, F = fine

3.4 A Three-Dimensional Shock Fitting Example

We conclude with a three-dimensional shock fitting example. The physical problem is a wave rider re-entry capsule at Mach 25 and 45° angle of attack. In this case, the shock surface was specified as an STL triangulation.⁴ In this case, the input surface triangulation (shown on the left in Fig. 14) has a very small number of especially poor triangles, with aspect ratio exceeding 5000 and minimum angle of a few tenths of a degree. The re-sampled surface (on the right in the figure) has a minimum angle in excess of 30°. Figure 15 shows two cutaway views of the volume mesh with the surface inserted, from different angles. Note particularly the matching of surface mesh and volume mesh edge lengths at the inserted surface, which is visually comparable to that at the boundaries of the original mesh.

This mesh has been successfully read in by the hypersonic shock fitting code of Bonfiglioli et al. [2] and advanced in time. Code integration is not yet complete at the time of this writing, but our near-term plan is to update the mesh to reflect movement of the shock surface by adding and removing points near the shock

⁴Again, we thank Renato Paciorri and Aldo Bonfiglioli for providing this data from their shock fitting code.

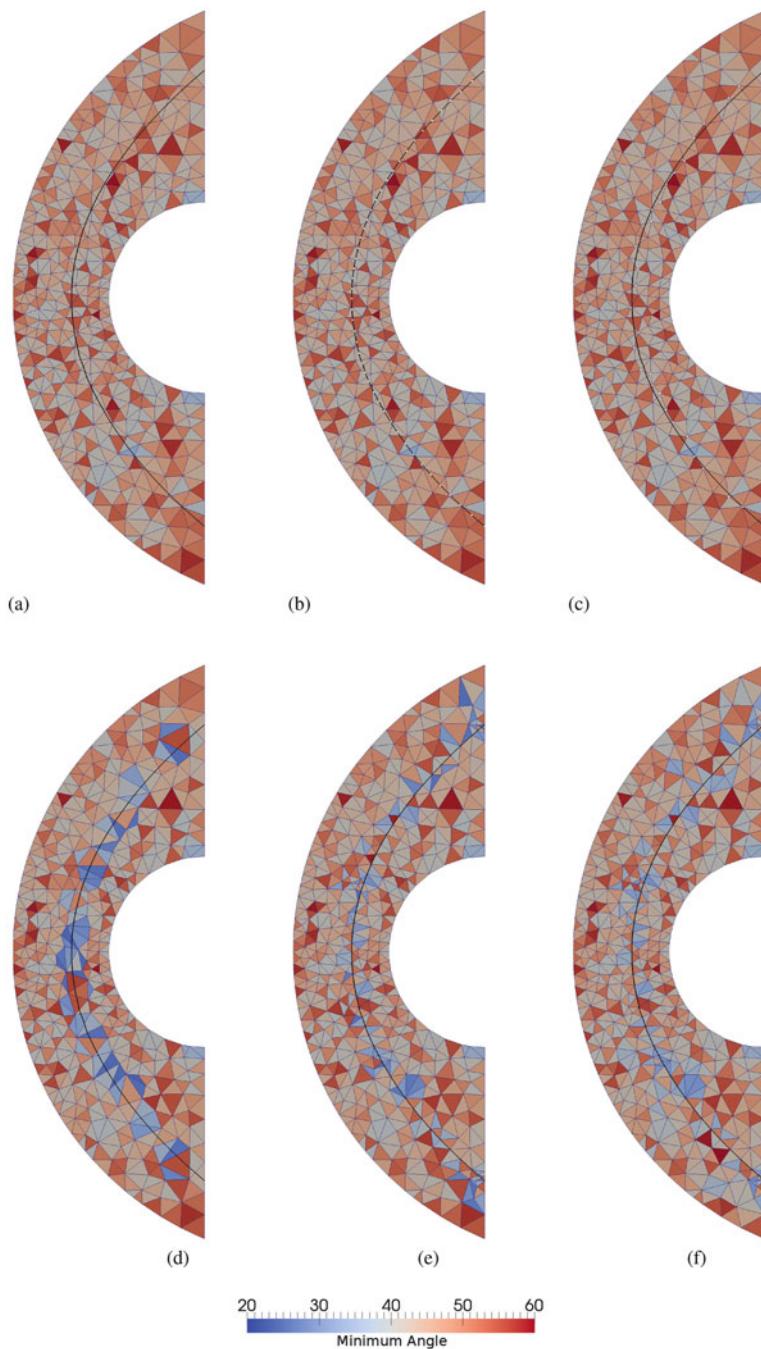


Fig. 13 Proof-of-concept shock example in 2D. **(a)** Initial mesh, with shock curve to insert. **(b)** Initial mesh, with sample points. **(c)** Initial mesh, with points to remove. **(d)** Mesh after point removal. **(e)** Mesh after insertion of sample points. **(f)** Final mesh

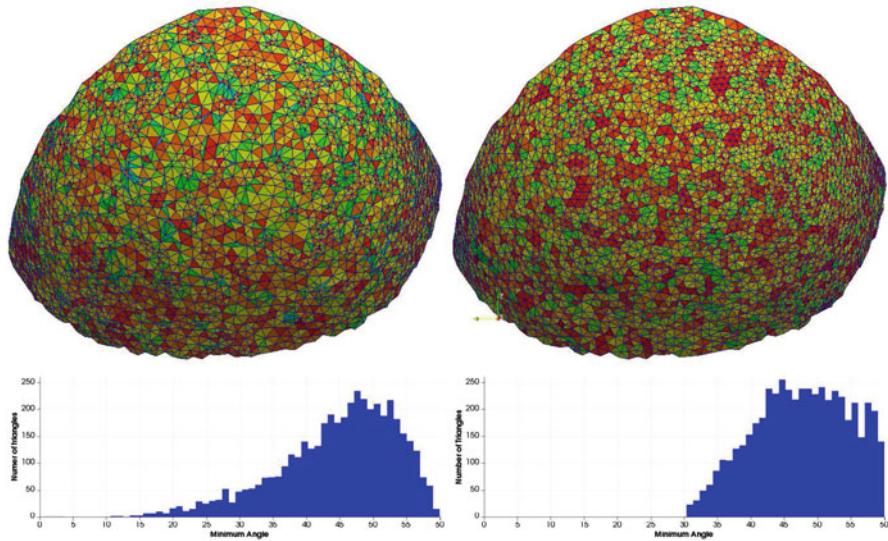


Fig. 14 Comparison of the initial (left) and re-sampled (right) surface meshes for the 3D waverider example, as seen from upstream. Minimum angle of surface triangles is shown in the histograms below

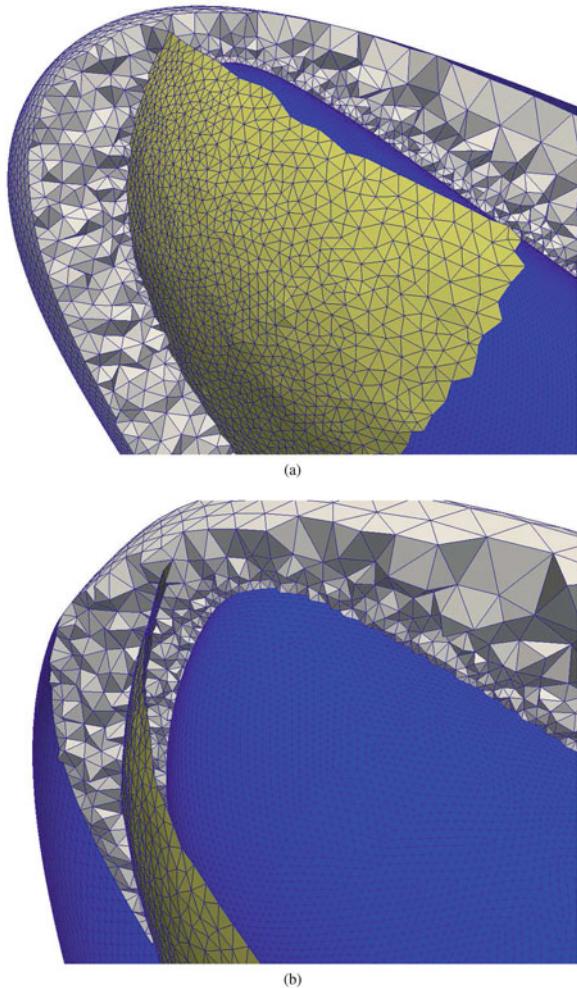
surface to reflect their proximity to the surface (distance relative to the local length scale in the original volume mesh). This approach will minimize meshing work⁵ and also the disruptions in convergence that would occur with more intrusive mesh updates.

4 Conclusions and Future Work

In this chapter, we have described an approach for inserting a surface into an existing mesh. The surface is given discretely on input; we begin by producing a spline of the surface so that we can re-discretize it with a length scale determined by the characteristic size of the mesh locally. We sample the surface; in three dimensions, we sample bounding curves first, then the surface itself. Next, we identify vertices in the existing mesh that encroach on faces of the sampled surface or are too close to sample vertices; these vertices are removed by edge contraction. We insert the surface sample points and recover the surface; because none of the surface faces are encroached (barring occasional vertex removal failures in 3D), recovery is easy.

⁵Especially compared with the current approach in Bonfiglioli et al's work flow, which is to regularly regenerate the mesh with the shock surface moved and points too near the surface ignored.

Fig. 15 Two cutaway views of the waverider mesh with the shock surface inserted into it. The *yellow surface* in the middle of each figure is the shock; the *blue surfaces* are the body and farfield boundary (only backward-facing triangles visible), while the *gray triangles* are a crinkle cut of the volume mesh along the body plane of symmetry. **(a)** View from upstream of the shock surface. **(b)** View between the shock surface and the body



Finally, we perform some local cleanup to the mesh near the surface to improve mesh quality.

Our approach results in proper length scale on the surface, as determined by the length scale of the background mesh, and in high quality output meshes. Equally importantly, we change only a small fraction of the mesh: only a few cell layers on each side of the inserted surface are ever changed. This both improves execution time for surface insertion and reduces the number of mesh points where solution interpolation is required.

We are currently working to integrate our scheme with the unstructured shock fitting scheme of Bonfiglioli et al. [2].

Acknowledgements This work was supported financially by Intel Corporation, with technical monitors Peter Fleischmann, Stephen Cea, and Anil Sehgal, and by Università di Roma “La Sapienza” while the second author was a visiting professor there.

References

1. Boivin, C., Ollivier-Gooch, C.F.: Guaranteed-quality triangular mesh generation for domains with curved boundaries. *Int. J. Numer. Methods Eng.* **55**(10), 1185–1213 (2002)
2. Bonfiglioli, A., Grottadurea, M., Paciorri, R., Sabetta, F.: An unstructured, three-dimensional, shock-fitting solver for hypersonic flows. *Comput. Fluids* **73**, 162–174 (2013)
3. Cheng, S.W., Dey, T.K., Ramos, E.A., Ray, T.: Sampling and meshing a surface with guaranteed topology and geometry. *SIAM J. Comput.* **37**(4), 1199–1227 (2007)
4. Freitag, L.A., Ollivier-Gooch, C.F.: Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Methods Eng.* **40**(21), 3979–4002 (1997)
5. George, P.L., Borouchaki, H.: Delaunay Triangulation and Meshing: Application to Finite Elements. Hermès, Paris (1998)
6. Gosselin, S., Ollivier-Gooch, C.F.: Constructing constrained Delaunay tetrahedralizations of volumes bounded by piecewise smooth surfaces. *Int. J. Comput. Geom. Appl.* **21**(5), 571–594 (2011)
7. Gosselin, S., Ollivier-Gooch, C.F.: Meshing volumes with curved boundaries using constrained Delaunay tetrahedralization. *Int. J. Numer. Methods Eng.* **87**(8), 795–820 (2011)
8. Miller, G.L., Pav, S.E., Walkington, N.: When and why Ruppert’s algorithm works. In: Proceedings of the 12th International Meshing Roundtable (2003)
9. Ollivier-Gooch, C.F.: Coarsening unstructured meshes by edge contraction. *Int. J. Numer. Methods Eng.* **57**(3), 391–414 (2003)
10. Ollivier-Gooch, C.F.: GRUMMP version 0.7.0 User’s Guide. Technical report, Department of Mechanical Engineering, The University of British Columbia (2017)
11. Ollivier-Gooch, C.F., Boivin, C.: Guaranteed-quality simplicial mesh generation with cell size and grading control. *Eng. Comput.* **17**(3), 269–286 (2001)
12. Ong, B., Russell, R., Ruuth, S.: An h-r moving mesh method for one-dimensional time-dependent PDEs. In: Proceedings of the 13th International Meshing Roundtable, pp. 39–54. Springer, Berlin (2013)
13. Ruppert, J.: A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms* **18**(3), 548–585 (1995)
14. Shewchuk, J.R.: Delaunay refinement mesh generation. PhD thesis, School of Computer Science, Carnegie Mellon University (1997)
15. Shewchuk, J.R.: Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom. Theory Appl.* **22**(1–3), 21–74 (2002)
16. Tautges, T.J.: CGM: a geometry interface for mesh generation, analysis and other applications. *Eng. Comput.* **17**(3), 299–314 (2001)
17. Watson, D.F.: Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput. J.* **24**(2), 167–172 (1981)

Shock Wave Detection Based on the Theory of Characteristics

Masashi Kanamori and Kojiro Suzuki

1 Introduction

The shock-fitting method is a fascinating approach to solving a flowfield that contains shock waves because it successfully treats the shock waves by considering them as boundary conditions rather than some parts of the solution within the computational domain. This means that one can avoid treating the shock waves with complex techniques such as total variation diminishing (TVD) [23, 24] or an essentially non-oscillatory (ENO) [7, 17, 18] scheme. Application of the shock-fitting method is, however, limited since it is necessary to know the locations of the shock waves to determine the bounds of the computational region. The detection of shock waves from CFD results is therefore of a great importance to enable the shock fitting method to be extended to more complex flowfields.

Several shock detection methods have been investigated to date [2–6, 9, 11–13]. These techniques can be classified into three types: utilizing a spatial derivative of the primitive variables [2–4, 11, 12], fitting the numerical result with the analytical solution of the local Riemann problem [5, 6], and finding locations where the characteristics collide with each other [9].

The first approach is based on the assumption that a gradient of the primitive variable, such as pressure or density, is perpendicular to the shock front. Buning and Steger [2] mentioned for the first time that shock waves were obtained as

M. Kanamori (✉)

Institute of Aeronautical Technology, Japan Aerospace Exploration Agency, 7-44-1,
Jindaijihigashi-machi, Chofu, Tokyo, Japan
e-mail: kanamori.masashi@jaxa.jp

K. Suzuki

Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5 Kashiwanoha,
Kashiwa, Chiba, Japan
e-mail: kjsuzuki@k.u-tokyo.ac.jp

locations where the Mach number along a pressure gradient becomes unity. This method, however, detects not only shock waves but also other types of waves, and Liou et al. [11] introduced three filtering techniques to eliminate such spurious detections. Lovely and Haimes [4] successfully extended the approach to unsteady flows. Ma et al. [13] proposed another approach to detecting shock waves, in which shock waves are extracted as iso-surfaces corresponding to a zero second spatial derivative of the density along the flow direction.

In the second methodology, they considered the local Riemann problem for one-dimensional unsteady flows in each computational cell and detected shock waves by fitting the numerical result with the analytical solution of the problem [5, 6]. The distinctive feature of this approach is that in addition to shock waves, it can detect other types of waves such as contact discontinuities and expansion waves. Two-dimensional shock waves can also be detected by determining a proper direction in which the flowfield can be treated as one-dimensional flow [6].

The third approach is based on the most rigorous definition of shock waves in an inviscid flow from a theoretical point of view; i.e. a shock wave is identical to the collision of characteristics of the same family [26]. In the shock-fitting code implemented by Moretti [14], shock waves were obtained by finding the intersection of the characteristics in one-dimensional unsteady flow; i.e. considering the characteristic velocity which is equal to $u \pm a$, where u and a respectively denote the flow velocity and the speed of sound, a shock wave emerges at the inflection point of the characteristic velocity. Kanamori and Suzuki [9] proposed a method utilizing the characteristics for two-dimensional steady flow. They extracted the collision by considering the local linearization of the vector field of the characteristics. Their method was successfully extended to unsteady and three-dimensional flowfields by respectively transforming a coordinate moving with the shock wave [9] and extending the idea of characteristics to three-dimensional space [10]. An advantage of this method is that shock waves can be detected (1) based on a rigorous and simple definition and (2) utilizing no empirical values or arbitrariness.

This paper mainly introduces the third approach. Section 2 introduces the theory of characteristics for two-dimensional steady flow and the definition of a shock wave from the viewpoint of characteristics. Section 3 describes details of the shock wave detection algorithm along with applications. Sections 4 and 5 describe extensions of our method to two more complex domains: unsteady flowfields and three-dimensional flowfields, respectively. The paper ends with concluding remarks in Sect. 6.

2 The Theory of Characteristics

This section describes the theory of characteristics in two-dimensional steady, supersonic flow and discusses the relationship between characteristics and shock waves. Denoting by θ the argument of the flow velocity and w the Prandtl-Meyer function [1], transport equations for $\theta + w$ and $\theta - w$

$$\frac{\partial}{\partial C^\mp}(\theta \pm w) = 0. \quad (1)$$

can be rewritten as follows:

$$\frac{\sqrt{M^2 - 1} \cos \theta \pm \sin \theta}{M} \frac{\partial(\theta \pm w)}{\partial x} + \frac{\sqrt{M^2 - 1} \sin \theta \mp \cos \theta}{M} \frac{\partial(\theta \pm w)}{\partial y} = 0. \quad (2)$$

The coefficient of each derivative forms a component of the characteristic velocity for the Riemann invariants, i.e. $\theta \pm w$ are transported at speeds of $(\sqrt{M^2 - 1} \cos \theta \pm \sin \theta)/M$ in the x direction and $(\sqrt{M^2 - 1} \sin \theta \mp \cos \theta)/M$ in the y direction. One can define the coefficients as propagation velocity vectors for the Riemann invariants, denoted by $\mathbf{f}(\mathbf{x})$:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} (\sqrt{M^2 - 1} \cos \theta \pm \sin \theta) / M \\ (\sqrt{M^2 - 1} \sin \theta \mp \cos \theta) / M \end{bmatrix}. \quad (3)$$

Equation (3) can be used to obtain the characteristics from CFD results by solving the following streamline equations for the characteristics $\mathbf{f}(\mathbf{x})$,

$$\frac{d\mathbf{x}}{d\tau} = \mathbf{f}(\mathbf{x}) \quad \text{or} \quad \frac{d}{d\tau} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (\sqrt{M^2 - 1} \cos \theta \pm \sin \theta) / M \\ (\sqrt{M^2 - 1} \sin \theta \mp \cos \theta) / M \end{bmatrix}, \quad (4)$$

where τ is a pseudo-time parameter. The characteristic curves C^\pm can be obtained by integrating Eq. (4).

2.1 Shock Wave and Characteristics

By definition, each characteristic curve should not interact with other characteristics of the same family. At locations where characteristics of the same type collide with each other, shock waves occur. Figure 1 clearly indicates the relationship between shock waves and characteristics in the supersonic flowfield around a circular cylinder. C^- curves collide with each other on a certain curve which is identical to the bow shock wave. The characteristic curves in Fig. 1 were obtained by solving Eq. (4) using a fourth-order 4-step Runge-Kutta method.

The above argument indicates that shock waves can be extracted from a CFD result as a series of positions where characteristics of the same family collide with each other. The method of finding such locations is introduced in the next section.

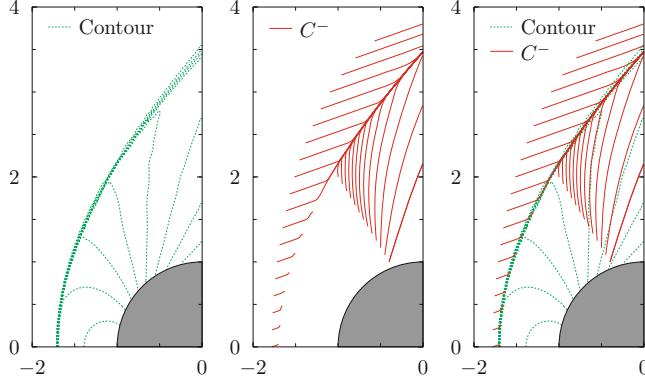


Fig. 1 Characteristics of supersonic flow around a circular cylinder (*Left*: Pressure contour, *Middle*: C^- , *Right*: superposition of contours and C^-)

3 Shock Wave Detection in Two-Dimensional Steady Flow

This section describes the shock wave detection algorithm and presents examples of its application.

3.1 Characteristic Equation as a Dynamical System

Calculating locations where two or more curves collide is generally computationally expensive. This section therefore extracts such locations as a critical line of Eq. (4) in each cell.

Consider the following two-dimensional linear ODE with respect to \mathbf{x} ,

$$\frac{d\mathbf{x}}{d\tau} = A\mathbf{x} + \mathbf{b}. \quad (5)$$

The solution of this equation is easily obtained and is expressed as follows: [8]

$$\mathbf{x} = \mathbf{x}_0 + C_1 \exp(\lambda_1 \tau) \mathbf{r}_1 + C_2 \exp(\lambda_2 \tau) \mathbf{r}_2, \quad (6)$$

where λ_i and \mathbf{r}_i are respectively the i -th eigenvalue and the corresponding eigenvector of the matrix A , and C_i are constants of integration. The shape of the solution Eq. (6) is classified into several patterns depending on the combination of eigenvalues as shown in Fig. 2. Only the cases of two distinct real eigenvalues are depicted in Fig. 2, since solution curves with complex or duplicated eigenvalues are of no interest here. It can easily be recognized what each pattern in Fig. 2 has in common: there are two straight line solutions, which are so-called *critical lines*, and

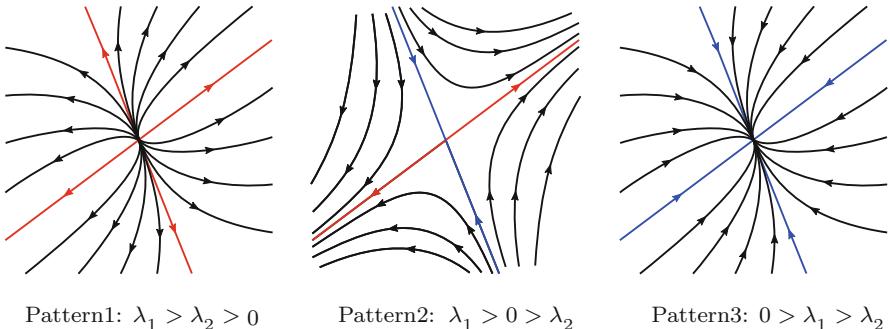


Fig. 2 Solution curves of Eq. (5) depending on its eigenvalues λ_i

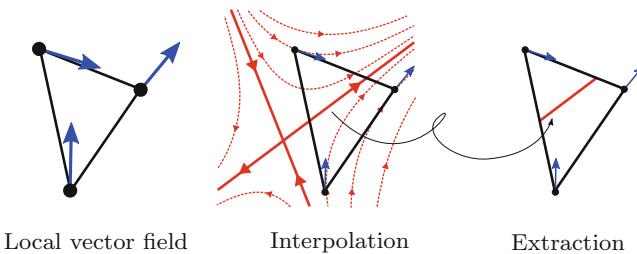


Fig. 3 Linearization of vector field in each cell

their directions depend on the signs of their eigenvalues. The direction vector of each critical line is identical to the eigenvector. The intersection of these two lines is often referred to as a *critical point* or a *fixed point*, denoted by \mathbf{x}_0 in Eq. (6). The important point here is that all the solution curves converge to a certain critical line when one or more eigenvalues are negative. Such a critical line is considered as a shock wave. In fact, it is easily recognized from the figure that all the solution curves converge to the critical line.

This indicates that the proposed method detects a shock wave as a convergent line of characteristics rather than as a series of points at which the collision of each characteristic occurs. Shock waves are found as a critical line in each computational cell, and it is necessary to linearize Eq. (4) for each cell to achieve this. Figure 3 helps the reader to understand this process. Note here that the vector at each vertex of the cell is identical to the right-hand side of Eq. (4). Using these vectors, local linearization is applied to obtain a linear ODE, Eq. (5). This process is essentially identical to applying an appropriate linear solution to the local vector field; i.e. a vector at each vertex becomes tangential to the solution curve passing through the vertex. The shock wave is finally obtained as a critical line of the locally linearized ODE. Repeating this process for each cell results in detection of the shock wave.

3.2 Algorithm for Two-Dimensional Steady Shock Wave Detection

The shock wave detection algorithm is summarized as follows:

1. Calculate the propagation velocity vectors for the Riemann invariants $\mathbf{f}(\mathbf{x})$ as described by Eq. (3) at each grid point.
2. Construct triangular cells from three neighboring grid points and calculate the right-hand side of Eq. (5), $\mathbf{A}\mathbf{x} + \mathbf{b}$, from the vectors $\mathbf{f}(\mathbf{x})$ at the three grid points.
3. Obtain the critical line for Eq. (5) and if it passes through the cell, consider the shock-crossing condition. That is, calculate the velocity component perpendicular to the critical line, denoted by V_n , at each grid point and check whether or not the Mach number $M_n = V_n/a$ satisfies the following relations:

$$(M_n)_L > 1 \text{ and } (M_n)_R < 1 \quad \text{or} \quad (M_n)_L < 1 \text{ and } (M_n)_R > 1, \quad (7)$$

where a is the speed of sound. The point L is determined as the furthermost point from the critical line to avoid the intermediate region of the numerical shock wave. Point R is set so that points L and R possess line symmetry with respect to the critical line. $(M_n)_R$ is calculated by extrapolating \mathbf{u}_R and a_R from the information at each vertex. The shock-crossing condition prevents the possibility of detecting slip lines.

A special treatment is, however, necessary when considering subsonic regions since no characteristic is defined in such regions. In fact, the propagation velocity $\mathbf{f}(\mathbf{x})$ cannot be calculated when $M < 1$. In a subsonic region, information can propagate throughout the entire region regardless of flow direction, but on the other hand information that induces a shock wave propagates in almost the opposite direction to the flow. This observation implies that characteristics that have almost the opposite direction to the flow velocity can be regarded as representative of the characteristics in the subsonic region. In fact, it is obvious that such shock waves can always be detected as long as the characteristics in the region have such a direction. The procedure for calculating characteristics in a subsonic region is as follows:

1. Calculate new velocities $\mathbf{V}'_1 = \frac{1}{2}\mathbf{V} + \mathbf{a}$ and $\mathbf{V}'_2 = \frac{1}{2}\mathbf{V} - \mathbf{a}$, where \mathbf{V} denotes the flow velocity, as illustrated in Fig. 4. Here, \mathbf{a} is defined as a vector that is orthogonal to \mathbf{V} and the length of which is equal to the speed of sound a .
2. Calculate $\mathbf{f}(\mathbf{x})$ defined in Eq. (3) from the velocities \mathbf{V}'_1 and \mathbf{V}'_2 . This yields four characteristics shown in Fig. 4. Select the two that are directed against the flow velocity \mathbf{V} , which are denoted by “ $\mathbf{f}(\mathbf{x})$ for C^\pm ” in Fig. 4.

Note that an advantage of this method is that no thresholdings are used, and thus no adjustments are needed to eliminate difficulties associated with the use of thresholdings, as was discussed in Sect. 1.

Fig. 4 Calculation of characteristics for the subsonic region

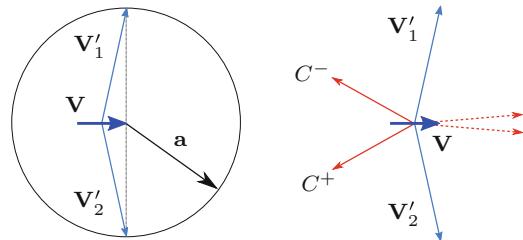
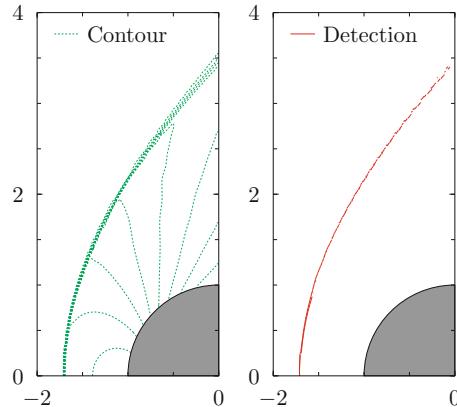


Fig. 5 Shock wave detection result for supersonic flow around a circular cylinder



3.3 Examples of Application

The shock detection method is now applied to two numerical results. All flowfields in this section were obtained by solving two-dimensional compressible Euler equations. The Simple High-resolution Upwind Scheme (SHUS) [16] with third-order MUSCL interpolation [20, 21] and the LU-SGS implicit scheme [25] were respectively utilized as the advection scheme and the time integration method.

The first example is supersonic flow around a circular cylinder. A strong bow shock wave exists in front of the cylinder, forming a subsonic region. The number of the grid points is 115×80 . Figure 5 shows the detection results, clearly indicating that a shock wave in this flowfield can be successfully detected using the proposed method even in the region where the shock wave causes the flowfield to decelerate from supersonic to subsonic speed.

Supersonic flow around a double wedge is considered as the second example, illustrated in Fig. 6. Two attached shock waves emanate at the edges and intersect. From the intersection, a slip line is generated along the flow. This flow is thus suitable for validating the present method because the method determines only shock waves while eliminating other discontinuities. The number of grid points is 249×200 . Note that no slip line is evident in the results of the shock wave detection method (the right-hand plot in Fig. 6), indicating that the detection method distinguishes slip lines from shock waves.

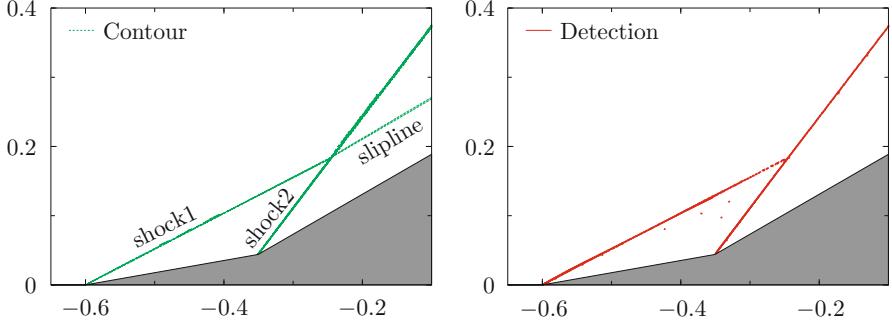


Fig. 6 Density contour plot for supersonic flow around a double wedge (*left*) and its shock wave detection result (*right*)

It is important to evaluate the sensitivity of the shock detection algorithm to the computational grid. Consider a Mach 2 supersonic flow around a 5° wedge with no angle of attack. This results in an oblique shock wave with a shock angle of $\beta = 34.3^\circ$ and a strength of $p_2/p_1 = 1.315$, which is relatively weak. The purpose of this problem is to evaluate the sensitivity of the result to the angle between the grid lines and the shock wave, denoted by α , over the range $0\text{--}20^\circ$. The effect of the cell aspect ratio, defined as w/h , is also examined by changing it from 0.5 to 1.5. Some steps were calculated using the CFD method introduced above starting with an exact solution as the initial condition. Figure 7 summarizes the detection result for various values of α and w/h . The exact shock wave and the computational grid lines are also depicted as a double line and dotted rectangles respectively. Note that every 15th grid line is shown in Fig. 7. Shock waves are successfully detected as continuous lines when the grid is perfectly aligned with the shock wave. On the other hand, a slight difference between the two, that is a non-zero value of α , leads to a disturbing consequence; the detected results form a group of many small pieces around the exact shock location. Such a result is inevitable unless CFD analysis is conducted with a well-aligned grid. In other words, one can recognize the locations where the grid is poorly aligned with respect to the shock wave from such a detection result. On the other hand, compared to the effect of α , aspect ratio has only a tiny effect on the detection result. This emphasizes the importance of aligning the grid lines with the shock wave.

4 Unsteady Shock Wave Detection

This section extends the shock wave detection method to an unsteady flowfield. The method introduced in the previous section cannot detect moving shock waves by itself and so some remedies are necessary. The key point here is the introduction of a shock-stationary coordinate which moves with the shock wave.

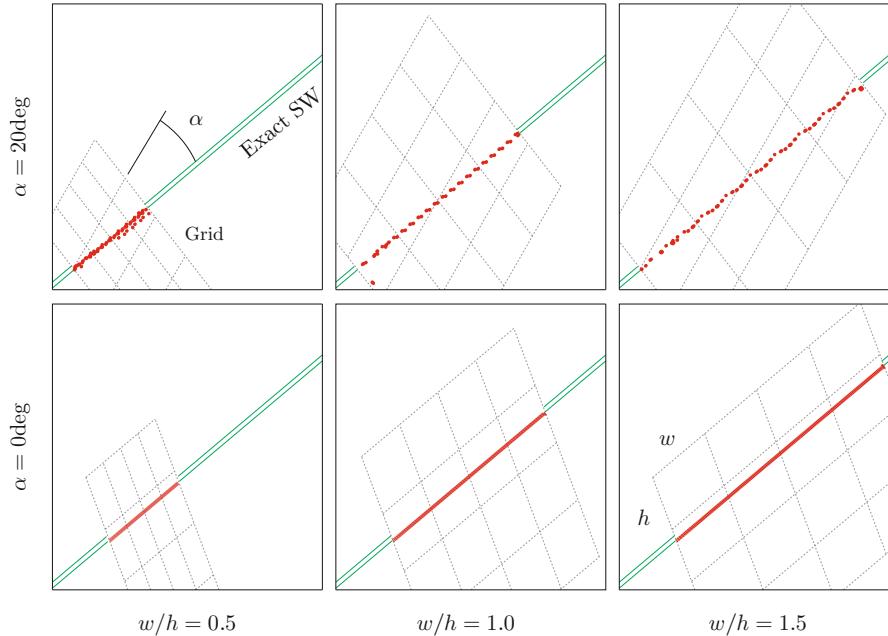


Fig. 7 Sensitivity of aspect ratio, w/h , and angle of grid line from exact shock wave, α , on shock wave detection result (every 15th grid line is shown as “Grid”)

4.1 Difficulties on Detection of Moving Shock Waves

Care must be taken when utilizing the shock wave detection method for an unsteady flowfield since both pre- and post-shock regions might be subsonic if the shock wave propagates at supersonic speed. For example, consider a moving shock wave with a shock Mach number of M_s traveling along a one-dimensional tube filled with a stationary gas. The Mach number in the region after the shock wave, M_1 , is obtained as follows: [1]

$$M_1 = (M_s^2 - 1) \left[\left(\gamma M_s^2 - \frac{\gamma - 1}{2} \right) \left(\frac{\gamma - 1}{2} M_s^2 + 1 \right) \right]^{-1/2}. \quad (8)$$

This equation indicates that both the regions before and after the shock wave become subsonic if $M_s < 2.07$ for a perfect gas ($\gamma = 1.4$). For such a case, the characteristics cannot be evaluated in the sense of a steady flowfield although a moving shock wave certainly exists. In fact, the Mach number in the region upstream of a shock wave must be supersonic if the shock wave is stationary. This contradiction can be reasonably resolved by observing the flowfield from a coordinate moving with the shock wave, and indeed Eq.(8) itself is often derived

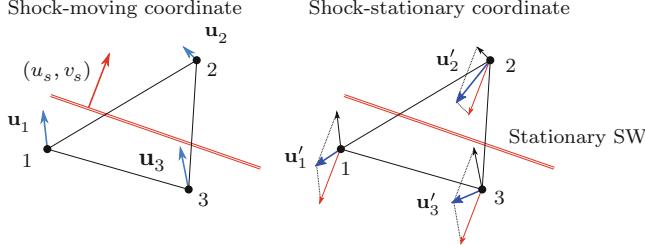


Fig. 8 Shock-moving (left) and shock-stationary (right) coordinates

using such a coordinate transformation. The authors define such a coordinate as a *shock-stationary coordinate* and consider moving shock waves as stationary ones.

The velocity of the moving shock wave must be determined in order to achieve the coordinate transformation. A common practice to evaluate the velocity is to consider the conservation of total enthalpy. Assuming steady inviscid flowfield within each computational cell, the total enthalpy evaluated with the flow velocity relative to the moving shock wave, denoted by $\mathbf{u} - \mathbf{u}_s = [u - u_s, v - v_s]$, should be unchanging; that is,

$$h_t = \frac{\gamma}{\gamma - 1} \frac{p}{\rho} + \frac{1}{2} \{(u - u_s)^2 + (v - v_s)^2\}, \quad (9)$$

should be the same everywhere. The authors therefore consider the conservation of h_t in each triangular cell as shown in Fig. 8. Considering that the values of total enthalpy at each vertex of the local triangular cell should be equal yields the following equation:

$$\frac{\gamma}{\gamma - 1} \frac{p_i}{\rho_i} + \frac{1}{2} \{(u_i - u_s)^2 + (v_i - v_s)^2\} = \frac{\gamma}{\gamma - 1} \frac{p_j}{\rho_j} + \frac{1}{2} \{(u_j - u_s)^2 + (v_j - v_s)^2\} \quad (i \neq j), \quad (10)$$

where subscripts i and j denote the grid points, or the vertices of the cell, and take the values 1, 2, or 3 as shown in Fig. 8. The equation can be rewritten in a linear form for u_s and v_s as

$$2(u_i - u_j)u_s + 2(v_i - v_j)v_s = u_j^2 - u_i^2 + v_j^2 - v_i^2 - \frac{2\gamma}{\gamma - 1} \left(\frac{p_i}{\rho_i} - \frac{p_j}{\rho_j} \right). \quad (11)$$

Equation (11) is the relation between the grid points i and j . Considering the combinations of two points from the three vertices of each cell, two independent equations are obtained resulting in a unique set of velocity components, u_s and v_s .

Difficulties arise when obtaining the moving shock velocity from Eq. (11). The reader should pay attention to the meaning of Eq. (11), which gives the velocity of a moving shock under the assumption of total enthalpy conservation; the existence

of moving shock waves is not considered in Eq. (11). This may result in a spurious shock detection. Such a misdetection often occurs in the freestream region, where the flowfield is almost uniform and values differ slightly from the freestream due to limitations such as the finite precision of the numerical representation. One can avoid evaluating such a region by introducing a threshold that distinguishes computational cells containing a moving shock wave from physically meaningless cells.

A better approach is expected by introducing the Rankine-Hugoniot condition [19], which states that for a one-dimensional flowfield, the variables must satisfy the relation

$$\mathbf{F}_R - \mathbf{F}_L = s(\mathbf{Q}_R - \mathbf{Q}_L), \quad (12)$$

where $\mathbf{Q} = [\rho, \rho u, \rho e]$ denotes the conserved variables, $\mathbf{F} = [\rho u, \rho u^2 + p, \rho(e + p)]$ denotes the flux, s denotes the speed of the shock wave motion, and the subscripts L and R correspond to the left- and right-hand sides of the moving shock wave, respectively. The Rankine-Hugoniot relation can be extended to two-dimensional flows. Consider the component perpendicular to the shock wave. The relation between the pre- and post-shock regions regarding conservation of mass can be obtained as follows:

$$s(\rho_R - \rho_L) = [\rho(\mathbf{u} \cdot \mathbf{n})]_R - [\rho(\mathbf{u} \cdot \mathbf{n})]_L, \quad (13)$$

where ρ , \mathbf{u} and \mathbf{n} respectively denote a density, a velocity vector, and a unit vector normal to the shock wave. The vector \mathbf{n} has already been obtained since the direction of the moving shock wave was derived when defining the shock-stationary coordinate. The moving velocity of the shock wave s can then be obtained from Eq. (13).

4.2 Algorithm for Unsteady Shock Wave Detection

The algorithm for detecting unsteady shock waves is summarized as follows:

1. Calculate the moving velocity of the shock wave (u_s, v_s) , which can be obtained by solving Eq. (11) using three neighboring grid points.
2. Calculate the propagation velocity vectors of the Riemann invariants $\mathbf{f}(\mathbf{x})$ in shock-stationary coordinates, $(u'_i, v'_i) = (u_i - u_s, v_i - v_s)$.
3. Construct triangular cells from three neighboring grid points and calculate the right-hand side of Eq. (5) $\mathbf{Ax} + \mathbf{b}$ from the vector $\mathbf{f}(\mathbf{x})$ at the three grid points.
4. Determine the critical line for the linear system $d\mathbf{x}/d\tau = \mathbf{Ax} + \mathbf{b}$.
5. Calculate the speed of the shock wave s from Eq. (13). (It should be noted that the direction of movement of the shock wave \mathbf{n} has already been obtained as a unit normal vector of the critical line.) The correct shock velocity $(u_s, v_s)_{RH}$ is then calculated as $s \times \mathbf{n}$.

6. Calculate the relative velocity (u'_i, v'_i) at each grid point using $(u_s, v_s)_{RH}$ and consider the shock-crossing condition, Eq.(7).
7. Determine the critical line as a shock wave if the shock-crossing condition is satisfied.

4.3 Examples of Application

The unsteady shock wave detection algorithm is now applied to a so-called double Mach reflection problem [22]. A planar shock wave moving at Mach 10 impinges into wall which is inclined at 30° . As a result, two series of Mach reflections occur along with a slip line as indicated in Fig. 9a. This problem is therefore appropriate for evaluating the applicability of the unsteady shock wave detection method. The number of grid points is 541×400 in directions along and normal to the bottom wall, respectively. To achieve an accurate temporal integration, a 3-step third-order TVD Runge-Kutta time integration scheme [7, 17] is utilized in this computation. The other computational methods are the same as in Sect. 3.3.

Figure 9b depicts the result using the unmodified shock wave detection method, i.e. the algorithm for steady flow. The figure clearly shows that neither weak shock nor strong shock waves can be detected. Figure 9c shows the result using the method modified with the shock-stationary coordinate concept, i.e. omitting steps 5–7 in Sect. 4.2. The result shows that almost all shock waves are detected along with many misdetections. Figure 9d shows the result using the method with the shock-stationary coordinate and the Rankine-Hugoniot condition. Almost all the errors observed in Fig. 9c are eliminated.

In conclusion, the combination of the shock-stationary coordinate and the Rankine-Hugoniot condition can improve the detection of moving shock waves without introducing thresholds.

5 Three-Dimensional Shock Wave Detection

This section extends the shock wave detection method described in Sect. 3 to three-dimensional flowfields. Unlike two-dimensional flowfields, a three-dimensional flowfield has an infinite number of characteristics and so the choice of an appropriate characteristic is necessary.

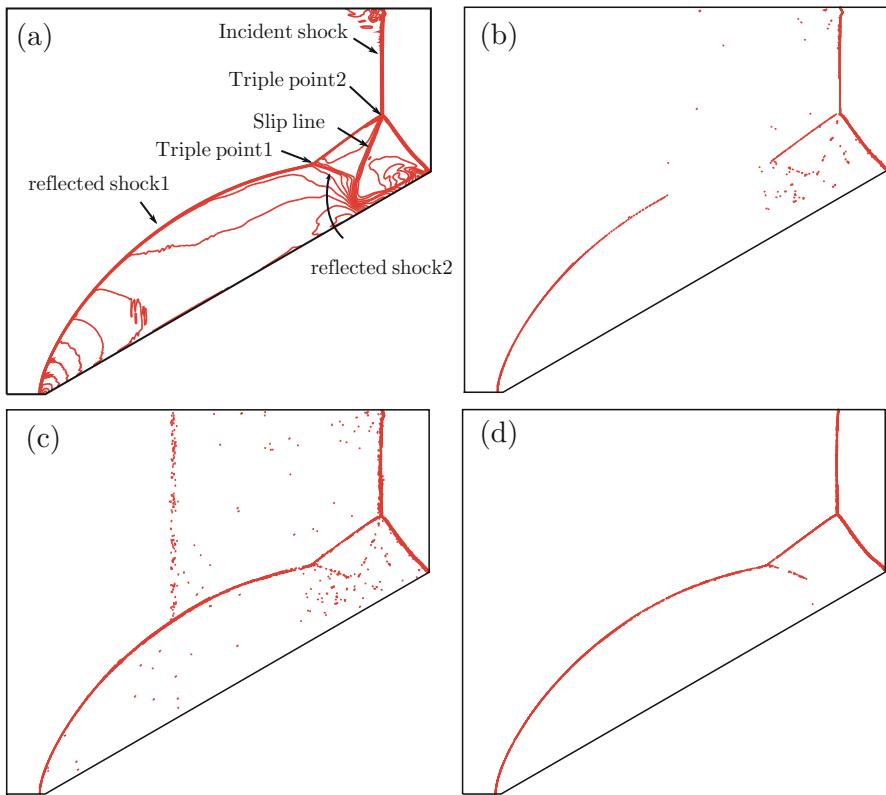
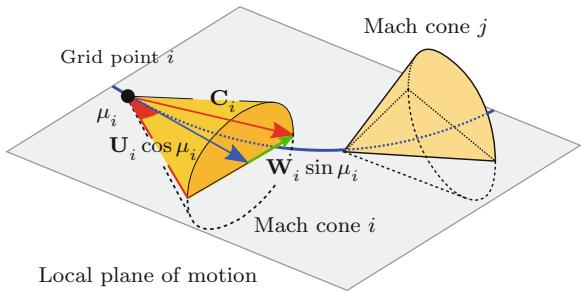


Fig. 9 Shock detection result for unsteady flow calculation using the shock wave detection method (a) density contour, (b) without modification, (c) with shock-stationary coordinate system, (d) with shock-stationary coordinate system and the Rankine-Hugoniot condition

5.1 Extension of Characteristics from Two- to Three-Dimensional Space

There are an infinite number of the characteristics in a three-dimensional inviscid supersonic flowfield. Each characteristic is equivalent to the generating line of the local Mach cone. Thus, one should consider the collision of these generating lines in three-dimensional space. The relations between each generating line are, however, little understood. In a two-dimensional flowfield, there are only two invariants $\theta \pm w$ corresponding to C^\mp , and only the collision of characteristics of the same family causes shock waves to be generated. In a three-dimensional flowfield, on the other hand, there exist an infinite number of characteristics and the correspondence between each generating line is unknown; that is, when the generating line of a local Mach cone collides with that of another local Mach cone, it cannot be determined whether or not a shock wave results. The authors therefore introduce

Fig. 10 Definition of characteristics in a three-dimensional flowfield



the idea of choosing the representative of the generating line which contributes to the generation of shock waves.

Consider a streamline and local Mach cones at two points i and j on the streamline as shown in Fig. 10. In general, if a shock wave is located between the two points i and j , the streamline will change direction suddenly and the two Mach cones should collide with each other. Here, the streamline above is assumed to be a planar curve in a certain plane in the neighborhood of the point of intersection between the streamline and the shock wave. This becomes exact in the limit of i and j to the point of intersection. The authors define such a plane as a local plane of motion. \mathbf{C}_i is one of the intersections between the Mach cone i and the local plane of motion, which by definition is the characteristics.

There are two intersections between the Mach cone and the local plane of motion, \mathbf{C}_i and the vector on the other side of the flow velocity vector. Considering the collision of two Mach cones, however, it is obvious that \mathbf{C}_i is the first section of the Mach cone i that collides with the Mach cone j . The authors therefore define the vector \mathbf{C}_i as a characteristic which induces the generation of a shock wave and consider shock detection based on the vector field. \mathbf{C}_i can be expressed as the following relation:

$$\mathbf{C}_i = \mathbf{U}_i \cos \mu_i + \mathbf{W}_i \sin \mu_i, \quad (14)$$

where \mathbf{U}_i and \mathbf{W}_i respectively denote unit vectors tangential to and normal to the streamline at the grid point i . \mathbf{U}_i is obtained by normalizing the flow velocity \mathbf{u}_i . \mathbf{W}_i is in the local plane of motion and normal to the vector \mathbf{U}_i , which can be obtained as a linear combination of \mathbf{U}_i and the acceleration vector denoted by \mathbf{a}_i . Any velocity vectors on the streamline are always in the local plane of motion by definition, i.e. the rate of change of these vectors, the acceleration vector \mathbf{a}_i , is also in the plane. This means that the local plane of motion is defined by two independent vectors: velocity vector \mathbf{u}_i and acceleration vector \mathbf{a}_i . \mathbf{a}_i can be obtained by differentiating \mathbf{u}_i with respect to τ , or a parameter along the streamline. Considering the linear interpolation of \mathbf{u}_i , namely $\mathbf{u}_i = \mathbf{P}\mathbf{x}_i + \mathbf{q}$, the differentiation can be calculated as follows:

$$\mathbf{a}_i = \frac{d}{d\tau} \mathbf{u}_i = \frac{d}{d\tau} (\mathbf{P}\mathbf{x}_i + \mathbf{q}) = \mathbf{P} \frac{d}{d\tau} \mathbf{x}_i = \mathbf{P} (\mathbf{P}\mathbf{x}_i + \mathbf{q}). \quad (15)$$

The vector \mathbf{W}_i is obtained using \mathbf{U}_i and \mathbf{a}_i as follows:

$$\mathbf{W}_i = \frac{\mathbf{w}_i}{|\mathbf{w}_i|}, \quad \mathbf{w}_i = \mathbf{a}_i - (\mathbf{U}_i \cdot \mathbf{a}_i) \mathbf{U}_i. \quad (16)$$

As described in Sect. 3.2, a special treatment is necessary in subsonic regions. For such cases, a representative of the characteristics is considered as follows:

$$(\mathbf{C}_i)_{\text{subsonic}} = \mathbf{U}_i \cos(\mu_i)_{\text{subsonic}} + \mathbf{W}_i \sin(\mu_i)_{\text{subsonic}}, \quad (\mu_i)_{\text{subsonic}} = 2 \sin^{-1} \frac{2}{\sqrt{M^2 + 4}}. \quad (17)$$

This definition is consistent with the two-dimensional shock detection introduced in Sect. 3.

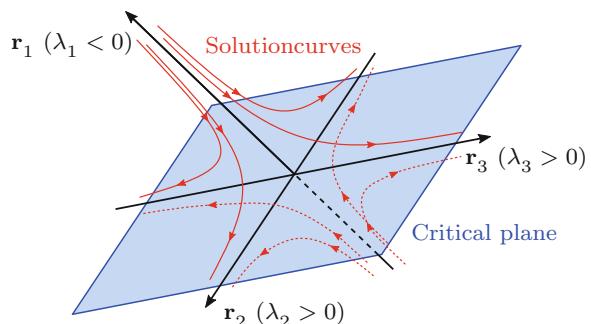
After calculating the vector \mathbf{C}_i using Eq. (14), shock wave detection is conducted; i.e. calculate \mathbf{A} and \mathbf{b} for the vector \mathbf{C}_i using four neighboring grid points and obtain the eigenvalues of the matrix \mathbf{A} . For negative eigenvalues, there exists a convergent plane of the solution curves, which is called a *critical plane*. Figure 11 shows typical solution curves for a three-dimensional linear equation with one negative and two positive eigenvalues. As shown in Fig. 11, all solution curves converge to the plane which contains the eigenvectors \mathbf{r}_2 and \mathbf{r}_3 . In other words, the critical plane corresponding to the negative eigenvalue can be obtained from the remaining two eigenvectors.

5.2 Algorithm for Three-Dimensional Shock Wave Detection

The algorithm for three-dimensional shock detection is summarized as follows:

1. Construct tetrahedral cells with four neighboring grid points and linearly interpolate the flow velocity \mathbf{u} , i.e. $\mathbf{u} = \mathbf{P}\mathbf{x} + \mathbf{q}$. In three-dimensional space, four

Fig. 11 Typical solution curves of Eq. (5) for the three-dimensional case



grid points are required to determine the linear interpolation uniquely. For a hexahedral cell, divide it into six tetrahedral cells. This method of dividing cells is not unique, but the authors have confirmed that the results are almost independent of the choice of division.

2. Calculate the characteristics which contribute to shock wave generation, denoted by \mathbf{C}_i , which are expressed by Eq. (14) at each grid point.
3. Construct a linear interpolation of the characteristics \mathbf{C}_i in the tetrahedral cell and obtain the critical plane for each negative eigenvalue. If the plane passes through the cell, define the surface as a shock wave.

5.3 Examples of Application

Two results of applying the three-dimensional shock wave detection algorithm are now presented.

The first example is a supersonic inviscid flow around a cone. The aim of this application is to validate that Eq. (14) is an appropriate representative of the characteristics in three-dimensional space. The freestream has an angle of attack of 10° to break the axisymmetry of the flowfield. In fact, the flowfield is virtually identical to a two-dimensional one if no angle of attack is considered. The half-cone angle is set to 15° . The number of grid points is $151 \times 91 \times 151$, about 2.1 million grid points. Figure 12 illustrates the detection results along with pressure contours projected onto some meridional planes. Compared with the pressure contours, the detection method successfully extracts shock waves even in a three-dimensional flowfield. This also indicates the appropriate definition of the characteristics, defined as Eq. (14). It should be noted that the detection result forms a continuous line since the grid line is almost perfectly aligned with the shock wave in this case, as was discussed in Sect. 3.3.

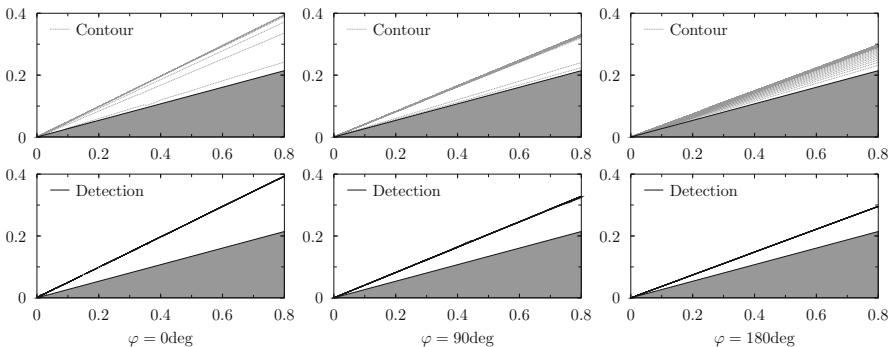


Fig. 12 Shock detection results (black, solid line) and pressure contour diagrams (gray, dotted line)

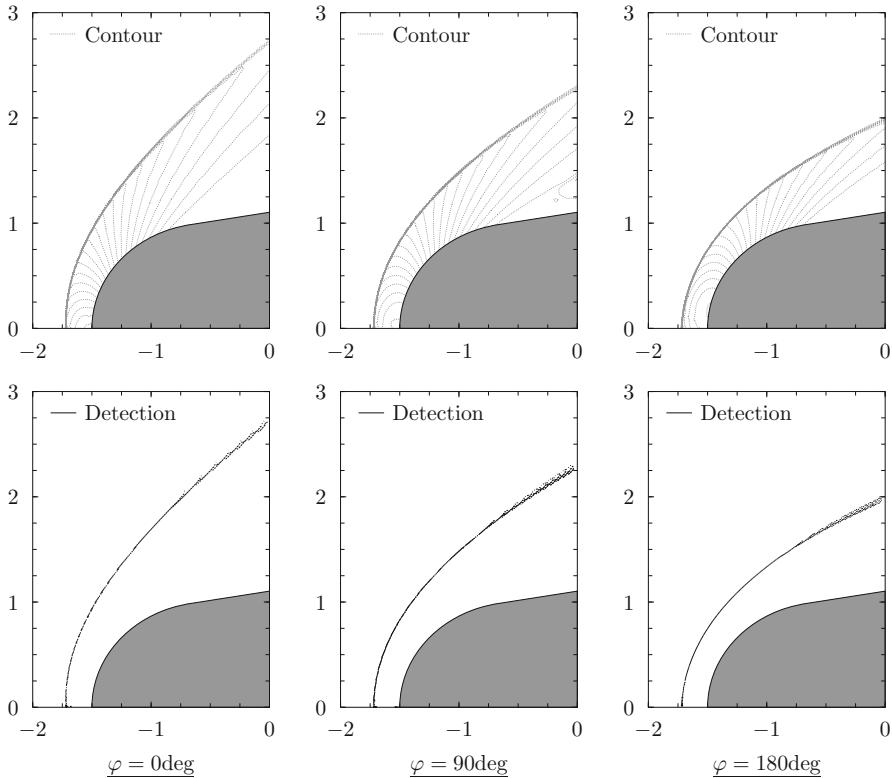


Fig. 13 Shock detection results (black, solid line) and pressure contour diagrams (gray, dotted line)

The second example is a supersonic inviscid flow around a sphere-cone with an angle of attack of 10° . In this flowfield, a strong bow shock wave emanates in front of the body, resulting in the formation of a subsonic region. This example is therefore appropriate for confirming the characteristics in a subsonic region defined by Eq. (17). In addition, a curved shock wave is suitable for discussing the effect of grid lines. The nose radius and the half-cone angle are set to 1 and 10° respectively. The number of grid points is $151 \times 91 \times 120$, about 1.8 million grid points. Figure 13 shows shock detection results and pressure contours on several meridional planes. The bow shock wave is detected successfully as well as the attached shock as in the previous example.

The effect of grid lines on the detection is shown in Fig. 14. Unlike the previous example, the detected shock waves are somewhat “dispersive,” or like a group of line segments, in the region on the top of the body (denoted as “Region A” in Fig. 14) while a perfectly continuous shock wave is obtained in region B. The difference between these two is the angle between the shock wave and grid lines. From the discussion in an earlier example, the present method can detect shock

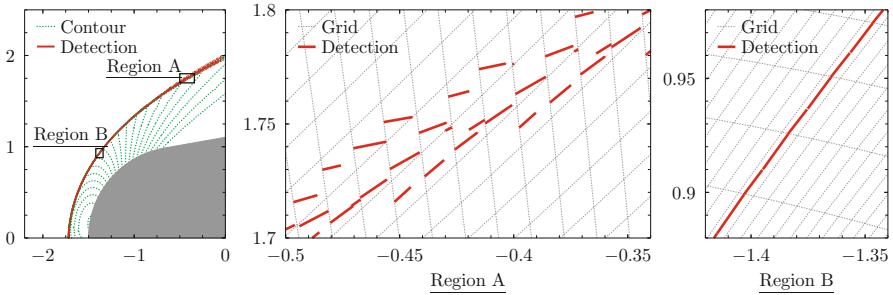


Fig. 14 Closeup view of shock detection result and pressure contours for a supersonic flow around a sphere-cone with computational grid

waves perfectly if the grid lines are parallel to the shock wave, and region B is just such a desirable case. In region A, on the other hand, shock waves cannot be detected correctly because the grid lines are not parallel to the shock. This situation can be summarized by saying that the more the grid lines become parallel to the shock wave, the more the detection results improve.

6 Concluding Remarks

This paper introduces a method for detecting shock waves in shock-capturing CFD results. The idea is very simple; a collision of characteristics of the same family causes the generation of a shock wave. The authors propose a method to extract such locations as the critical line of a linearized equation of the vector field of the characteristics. As a result, successful detection of shock waves is achieved.

This paper also shows extensions of the method to unsteady and three-dimensional flowfields. Moving shock waves can easily be detected by introducing a shock-stationary coordinate. In addition, the Rankine-Hugoniot condition improves the accuracy of detecting moving shock waves. The case of detecting three-dimensional shock waves has a problem in that a characteristic in three-dimensional space is the generating line of a local Mach cone and the relations between two distinct local Mach cones are little understood. The authors therefore show a method for choosing an appropriate representative from an infinite number of generating lines for generating shock waves in three-dimensional space. As a result, the shock detection method for a three-dimensional flowfield can be defined as a simple extension of the method for two-dimensional space.

The authors consider that the method helps to extend the shock-fitting methodology to more complex flowfields. Combining the shock detection method with the framework of unstructured computation is of great interest. Paciorri and Bonfiglioli showed some interesting results with their framework of shock-fitting, unstructured-mesh hybrid [15]. Recalling that the method described in this paper detects a shock

wave for each triangular or tetrahedral cell, it can be applied to an unstructured mesh. Such a combination is therefore expected to accelerate the applicability of the shock-fitting methodology.

The authors also consider that the method not only has remarkable potential as a visualization tool, but also reveals some interesting features, i.e. shock waves can be obtained as perfectly continuous curves or surfaces under the condition that the computational grid lines are well-aligned with the shock wave. Conversely, such a detection presents an undeniable evidence of well-aligned computational mesh and therefore that correct computation is guaranteed from the point of view of resolving shock waves. This also indicates that the present method is also useful as an evaluation tool to assess the quality of CFD solutions including shock waves.

References

1. Anderson, J.D.J.: Modern Compressible Flow with Historical Perspective. McGraw-Hill Publishing Company, New York (1990)
2. Buning, P.G., Steger, J.L.: Graphics and Flow Visualization in Computational Fluid Dynamics. AIAA Paper 85-1507, pp. 162–167 (1985)
3. Darmofal, D.: Hierarchical visualization of three-dimensional vortical flow calculation. Ph.D. thesis, Massachusetts Institute of Technology (1991)
4. Darmofal, D.: An image analysis based approach to shock identification in CFD. AIAA Journal 95-0117 (1995)
5. Glimm, J., Grove, J.W., Kang, Y., Lee, T., Li, X., Sharp, D.H., Yu, Y., Ye, K., Zhao, M.: Statistical Riemann problems and a composition law for errors in numerical solutions of shock physics problems. SIAM J. Sci. Comput. **26**, 666–697 (2004)
6. Glimm, J., Grove, J.W., Kang, Y., Lee, T., Li, X., Sharp, D.H., Yu, Y., Ye, K., Zhao, M.: Errors in numerical solutions of spherically symmetric shock physics problems. Contemp. Math. **371**, 163–179 (2005)
7. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes, III. J. Comput. Phys. **71**, 231–303 (1987)
8. Hirsch, M.W., Smale, S., Devaney, R.L.: Differential Equations, Dynamical Systems, and an Introduction of Chaos. Academic Press, Cambridge (2003)
9. Kanamori, M., Suzuki, K.: Shock wave detection in two-dimensional flow based on the theory of characteristics from CFD data. J. Comput. Phys. **230**(8), 3085–3092 (2011)
10. Kanamori, M., Suzuki, K.: Three-dimensional shock wave detection based on the theory of characteristics. AIAA J. **51**(9), 2126–2132 (2013)
11. Liou, S.P., Mehlig, S., Singh, A., Edwards, D., Davis, R.: An image analysis based approach to shock identification in CFD. AIAA Paper 95-0117 (1995)
12. Lovely, D., Haimes, R.: Shock detection from computational fluid dynamics results. AIAA Paper 99-3285 (1999)
13. Ma, K.L., Rosendale, J.V., Vermeer, W.: 3D shock wave visualization on unstructured grids. In: Crawfis, R., Hansen, C. (eds.) Proceedings of the 1996 Symposium on Volume visualization, pp. 87–104 (1999)
14. Moretti, G.: Computation of flows with shocks. Annu. Rev. Fluid Mech. **19**, 313–337 (1987)
15. Paciorri, R., Bonfiglioli, A.: Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique. J. Comput. Phys. **230**(8), 3155–3177 (2011)
16. Shima, E., Jounouchi, T.: Role of CFD in aeronautical engineering (no.14) - AUSM type upwind schemes. In: Ebihara, M. (ed.) Proceedings of the 14th NAL Symposium on Aircraft Computational Aerodynamics, pp. 7–24 (1999)

17. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**, 439–471 (1988)
18. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *J. Comput. Phys.* **83**, 32–78 (1989)
19. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction. Springer, Berlin (2009)
20. van Leer, B.: Toward the ultimate conservative difference scheme. 4, a new approach to numerical convection. *J. Comput. Phys.* **23**, 276–299 (1977)
21. van Leer, B.: Toward the ultimate conservative difference scheme. 5, a second-order sequel to godunov’s method. *J. Comput. Phys.* **32**, 101–136 (1979)
22. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid with strong shock. *J. Comput. Phys.* **54**, 115–173 (1984)
23. Yee, H.C.: Construction of explicit and implicit symmetric TVD schemes and their applications. *J. Comput. Phys.* **68**, 151–179 (1987)
24. Yee, H.C., Warming, R.F., Harten, A.: Implicit total variation diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.* **57**, 327–360 (1985)
25. Yoon, S., Kwak, D.: An implicit three-dimensional Navier-Stokes solver for compressible flow. *AIAA J.* **30**(11), 2635–2659 (1992)
26. Zeldovich, Y.B., Raizer, Y.P.: Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena. Dover Publications, Mineola (2002)

Part III
Tribute to Gino Moretti

Gino Moretti: A Man for All Seasons

Manuel D. Salas

[He] left [his] signature on us, the literature of the teacher who writes on minds. I have had many teachers who told me soon-forgotten facts but only [two] who created in me a new thing, a new attitude and a new hunger. I suppose that to a large extant I am the unsigned manuscript of that ... teacher. What deathless power lies in the hands of such a person. – John Steinbeck, *Like Captured Fireflies*, 1955

The play *A man for all seasons* is Robert Bolt's portrait of Sir Thomas More, Lord Chancellor of England, as the ultimate man of conscience, remaining true to his principles under all circumstances and at all times. Robert Whittington, a poet contemporary of Thomas More, wrote in 1520: "More is a man of an angel's wit and singular learning. I know not his fellow. For where is the man of that gentleness, lowliness and affability? And, as time requireth, a man of marvelous mirth and pastimes, and sometime of as sad gravity. A man for all seasons." Those same words also describe the Gino Moretti that I knew.

Here I write not so much about Moretti's technical work which has already been discussed at length by me and others, but about Moretti the man. If I am successful, I hope to convey to you his-sense-of-life. And by that I mean an appraisal of the man, of his nature and of his character.

When I graduated from High School in 1965 I knew two things: I wanted to be an Aeronautical Engineer and I wanted to study under Antonio Ferri, whom I thought was the premier high speed aerodynamicist at that time. Ferri was the Chair of the Department of Aeronautics at the Polytechnic Institute of Brooklyn. But to my surprise when I arrived at the Polytechnic, Ferri and a cadre of professors had hightailed to New York University. I was dismayed. I decided to finish my bachelors at the Polytechnic and then enroll for a Masters at NYU. It would not happen.

M.D. Salas (✉) (retired)
NASA Langley Research Center, Hampton, VA, USA
e-mail: manny.salas@gmail.com

Fig. 1 Gino Moretti at the University of Rome, La Sapienza, 2006



The next year, my sophomore year, I was supposed to take my first Fluid Dynamics class. I was looking forward to this class. The Aeronautics class was large, so the Polytechnic offered two parallel courses. One was given by a faculty member I was familiar with, the other by a new faculty member, Gino Moretti, of whom I knew nothing. I took his class and it changed everything.

I remember that on the first day when I entered the class the front row was empty and I took a chair there. A few minutes later Moretti came in, wearing a short sleeve shirt, just like in the photo, Fig. 1, and began lecturing and at that moment I knew that this was someone extraordinary. Moretti was a gifted story teller, he was trained in classical mechanics and applied mathematics by the Italian masters of his generation Carlo Ferrari (1903–1996), Francesco Tricomi (1897–1978), Alessandro Terracini (1889–1968), and Guido Fubini (1879–1943). He was a seasoned lecturer and had already published several textbooks in mathematics, physics, and mechanics. He had a love and an innate knowledge of fluid mechanics. His blackboard sketches and illustrations were small artworks. He had taught this course for several years at the University of Cordova and the National University of Cuyo in Bariloche, Argentina. The lectures just flowed with ease and clarity. All these elements combined into a masterwork class.

My classmates however complained about how difficult the homework and exams were. They said he was “old school”. I don’t think they knew exactly what they meant by that. I think that what they were trying to say was that he was a master-craftsman and fluid dynamics was his craft.

What I did not know back then was that on March of 1958, Moretti had traveled from Argentina to the US, up the East coast, looking for employment and by happenstance had arranged a meeting with Ferri at Ferri’s company, GASL.

To set this in context, remember that the previous year, 1957, the Russians had launched Sputnik 1, the space race was on and the US was not winning. The birth of Computational Fluid Dynamics was about to take place.

This is Moretti's description of that meeting, of course the dialogue is in Italian, therefore it is loud and I can envision a lot of hand waving:

Moretti writes: "Ferri had just arrived from [a trip to] Italy ... I was somewhat intimidated. Ferri's directness and irreverence impressed me. Carlo Ferrari, my mentor and universally considered as one of the most respected specialists in Fluid Mechanics, according to Ferri was *un salame*, [a sausage, which Moretti translates as] a silly goose, only capable of writing papers, papers that were all wrong.¹ Looking at me with extreme seriousness, Ferri went down to business:

Ferri: I could hire you and you could even be happy here. But your wife won't; believe me; I know how things go, therefore... I can't hire you."

Wow! Ferri had just hit Moretti's most sensitive spot, his wife. That triggered his first fight with Ferri.

"Moretti: If you really can hire me, please hire me, and remember you are hiring me, not my wife. That's my business.

Ferri: Well, if you insist, I'll hire you. And you will face the music."

And that was that. In less than 3 months Ferri had cut through the red tape and obtained the necessary visa for Moretti to start work at GASL on October of that year.

When Ferri heard that Moretti had arrived at Miami and was buying a car to drive to New York, he commented: "He'll never make it. He'll get lost somewhere between Miami and New York". Moretti did not get lost on his way to New York. But he got lost on Hempstead Avenue on the way to GASL and was 20 min late.

One day on his second year at GASL, 1959, Ferri walks in asking: "Where is Gino? Where is Gino? Gino we have a computer and I want you to compute some real-gas flow around a blunt body."

The computer was the Bendix G15. The Bendix G15, introduced in 1956, had 180 vacuum tube packs and 300 germanium diodes, its memory held 2160 words of 29 bits. The G-15's primary output device was a typewriter with an output speed of about 10 characters per second for numbers.

Moretti writes: "Using the Bendix G15 for any of Ferri's projects was pure madness. Nevertheless, I considered his order a challenge. Ferri, who was not a theorem writer but a practical man, an engineer in search of better ways to fly, wanted numbers, but numbers be could rely on. I offered him what he was looking for: *a reasonable mathematical and physical background and all the enthusiasm of a young, bold unprejudiced explorer.*"

Soon after that, around 1960, the Air Force was asking for a computational code to simulate the flow over the Dyno Soar, a blunt nose flat bottomed delta wing boost-glide vehicle somewhat like a small version of the space shuttle, re-entering

¹Actually Ferri and Ferrari were good friends and Ferri was saying this only in jest.

the atmosphere at an angle of attack of 30° . Ferri put pressure on Moretti to develop this code.²

Moretti writes: “No high-speed computer of that time could store all the necessary data and process the code in a reasonable time; not to mention the complexity of the flow on the lee side. I refused to collaborate and tried to persuade Ferri to give up. He did not hide his annoyance.”

This was the genesis for Moretti’s study of the blunt body problem in the early 1960s, with similar work conducted by Rusanov in the Soviet Union. In my opinion this seminal work signals the birth of Computational Fluid Dynamics. Moretti’s approach to numerical computations followed a simple recipe: the numerical computation should closely match the physical evolution. And it had two corollaries: (1) the numerical initial and boundary conditions should mirror the physical initial and boundary conditions; (2) the numerical discretization of the governing equations should take advantage of the wave phenomena that the equations describe.

If you want to read one paper to get a sense of Moretti’s approach to computational fluid dynamics, in my opinion the best paper to study is: *Transient and Asymptotically Steady Flow of an Inviscid, Compressible Gas Past a Circular Cylinder*, Polytechnic Institute of Brooklyn, April 1970. This work was a pure labor of love. It was only published as a Poly report, but is available in the internet at the Defense Technical Information Center, document accession number: AD0708898.

Gino Moretti was born on the 2nd of January of 1917 in Turin, the capital of the Piemont region of Italy. This is the mountain region in the northern part of Italy bordering France. Turin is a city with a glorious mathematical past [1]. It was the birthplace of one of Europe’s greatest mathematician Joseph-Louis Lagrange (1736–1813) and of Giuseppe Peano (1858–1932), the founder of the school of mathematical logic.

The drawing on Fig. 2 is from Moretti’s Illustrated Memoirs, a four volume collection of pen and ink drawings covering the years 1917–1948. This and many other sketches that follow are courtesy of the Moretti Family. The drawing depicts the Church of Mercy which is next to the apartment house, on the right, where Moretti was born.

Moretti starts his memoirs with this hideous drawing, Fig. 3, of “The First person I saw was a doctor in uniform.” It would make you want to return to the safety of the womb.

He then tells us that “...my parents decided that I was worth a picture”. The photo, Fig. 4, is of Moretti as a baby, but I do not know when it was taken. Later in his memoirs he comments: “Times were changing; I was growing in age (14) in length and, somewhere, in width (very embarrassing). After strenuous exams, I was admitted to the first year of Lyceum [High School]. The first day (my father having reluctantly abstained from walking me all the way to school) I showed up still in shorts, dressed like the soccer umpires. Everybody shouted “Dalli all’Arbitro!” (Kill the umpire!).” see Fig. 5.

²Fifteen years later, under contract to NASA Langley Research Center, Moretti, Frank Marconi and I developed a similar code to study the space shuttle at more moderate angles of attack.

Fig. 2 The Church of Mercy, Turin, Italy. Source: Gino Moretti's Illustrated Memoirs (courtesy of the Moretti Family)



Fig. 3 The first person I saw was a doctor in a uniform. Source: Gino Moretti's Illustrated Memoirs (courtesy of the Moretti Family)



Fig. 4 Moretti's baby photo. Source: Gino Moretti's Illustrated Memoirs (courtesy of the Moretti Family)



Fig. 5 Moretti's first day at the Lyceum. Source: Gino Moretti's Illustrated Memoirs (courtesy of the Moretti Family)



Fig. 6 Moretti's dad first car, a Fiat 509. Source: Gino Moretti's Illustrated Memoirs (courtesy of the Moretti Family)



At the beginning of the twentieth century, Turin was a city immerse in science and technology and from an early age Moretti was in love with trains and automobiles and this is reflected in his illustrated memoirs. Figure 6 is a sketch of his dad's first car; a Fiat model 509, which Moretti tell us had no less than 15 previous owners when his dad bought it.

When time came to make plans for a career, Moretti's dad, a magistrate, had a serious talk with Moretti. His dad ruled out architecture, music, journalism, and any of the arts. His rationale was that “too many talented people are competing” for these positions. Moretti's dad thought only of three dignified and worthwhile professions: justice, education, and the military. His dad thought that these professions were regulated and financed by the government with prestige, good pay, and a pension plus very long vacations. Moretti writes: “since I was clearly not interested in justice, dad suggested the military.” According to his dad “It will do you some good, strengthen your character and keep you in a safe place, if war ever comes again.”

Moretti decided to go into education and enrolled in the University of Turin in mathematics and physics. Among Moretti's professors was Alessandro Terracini whom he said “bred my love for geometry,” Francesco Tricomi from whom “I learned to love calculus,” and Guido Fubini a master of mathematical analysis. Of

Fubini he writes “The course on advanced calculus by Professor Fubini was great entertainment. He had a terrific reputation for clarity at the school of engineering, [but] at the school of mathematics he took his vengeance.”

Moretti learned descriptive geometry from Gino Fano (1871–1952). During the final exam, the first question Fano asked was on the theory of the Helicoid-Conoid. Moretti had studied Fano’s book in great detail, but of course the one chapter he had not reviewed for the exam was the chapter on the Helicoid-Conoid. Moretti writes: “I learned then that even a brilliant student cannot build up the theory of the Right Helicoid-Conoid during an oral exam. I also learned that I had a quick rate of recovery. Fuming, Fano drilled me on all the chapters of his book, appendices included, and I was just great”. Fano wanted to give Moretti 18 points for the exam, the lowest possible grade. Terracini protested. He said Moretti deserved 30 points, the highest possible. After arguing for 15 min, they settled on 26 points. Moretti felt dejected.

Moretti’s scientific career began in 1939 as assistant professor to Carlo Ferrari, Chair of the Aeronautical Department, Turin University. Ferrari then was a young, energetic, and highly feared teacher of classical mechanics. The students called him “The Panther”. Ferrari gave Moretti a research grant that paid 500 Lire, enough to survive half a month. Ferrari wanted Moretti to work on turbulence. Moretti writes: “He wanted to be the first to understand turbulence. He didn’t succeed, but neither did anyone else.” Moretti had no interest in turbulence. One day frustrated, trying to make sense of Prandtl’s monograph on turbulence, Moretti throws the monograph out the window, then, he writes, “I had to go downstairs to pick it up.”

On September 1st, 1939, Germany invaded Poland, England and France declared war and Moretti’s life was thrown into the maelstrom. Moretti was sent to the Russian front as part of the Italian corps of engineers to lay telephone lines.

The illustration in Fig. 7 shows how Moretti communicated in code with his wife Anita. Before leaving for the Russian front, around June 1942, Moretti made two copies of a map of Russia. In order to let Anita know of his whereabouts, Moretti coded longitudes and latitudes into nouns and adjectives that he would include on the 11th line of his letters to Anita. He wrote over a hundred letters to his wife while on the Russian front carefully letting her know where he was. All for naught since Anita lost her copy of the map shortly after he left. Moretti entered Russia through Minsk and Beresina where Napoleon’s retreating army had suffered great loses to the Russian army back in 1812, see Fig. 7. The sketch shows the path Moretti followed through Russia, from Minsk to Stalingrad.

As he is leaving for the Russian front Moretti agonizes about his future:

I hate wars.

In this one, I hoped for a victory of the “enemy”.

I was terrified by Hitler and disgusted with Mussolini.

My wife and my daughter were all I could ask for in life.

I was going to abandon them, perhaps forever.

But I had been with my soldiers for one year now – they had to go.

They were my friends. I had to give them moral support . . .

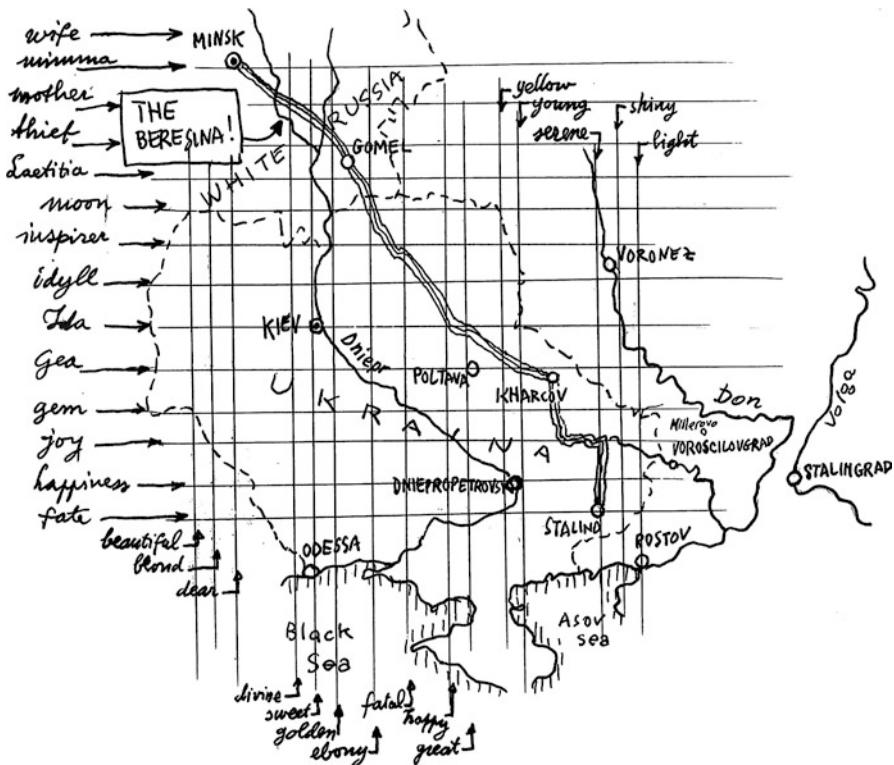


Fig. 7 Moretti's encrypted map designed to provide his location to his wife while in Russia (courtesy of the Moretti Family)

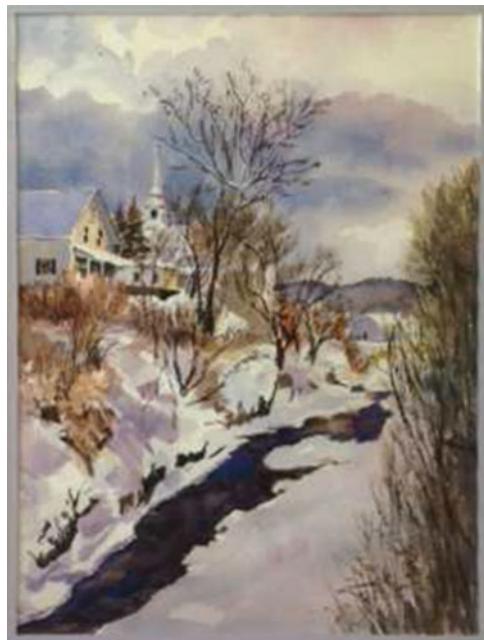
Later on he writes: "I worked 18–20 hours a day, often my meal was a biscuit or nothing, I had to organize and maintain all the telephone lines of a good chunk of the Italian army in Russia, I had to do everything despite the commanding officer, and the mail from home was ever so scarce ..."

From 1944 to 1945, with the war coming to an end in Italy, Moretti moves to Switzerland where he joins the University of Lausanne as a visiting professor. There he teaches calculus and later published a two volume textbook based on this course.

The day he arrived at Lausanne he was told that the Professor of Mathematics was under quarantine because of a case of scarlet fever. Could he take his place teaching? Moretti writes: "“yes” I said, simply because it is hard for me to say “no.” When should I begin? Tomorrow morning at 8:30, at the uptown campus. Thus began my life as a Mathematician."

His many lectures in Switzerland, Argentina, and the US are collected in several textbooks: Analisi Matematica, two volumes, 1952; Mecanica de los Fluidos, two volumes, 1955 (unpublished); Meccanica Razionale, two volumes, 1956; Metodos Matematicos de la Fisica, 1959; Functions of a Complex Variable, 1964.

Fig. 8 Watercolor (courtesy of the Moretti Family)



In the early 1990s, Moretti compiled four volumes of pen and ink drawings and annotations which he called *Gino's Illustrated Memoirs*. They cover the years 1917–1948. They remain unpublished. In the late 1990s Moretti wrote two short stories. One, *Cavalli 8 Uomini 40, Memorie di una guerra assurda*, is based on over 144 letters that he wrote to his wife Anita while in the Russian front. The title “Horses 8, Men 40” refers to the group of telephone line engineers of which he was a member. The other, *Volevo tanto fare il tramviere, ma mi hanno cambiato i tram*, describes his childhood and adolescence in Turin during an era of fast technical progress. Moretti also liked to paint, see Figs. 8 and 9, compose music, dabble in carpentry, and build train sets. He was a linguist, trained in classical Latin and Greek, fluent in Italian, Piedmontese, Spanish, and English, with some French, German and Russian. His love for languages extended to grammar of which he was a meticulous practitioner.

On the last day of my first course on Fluid Dynamics, Moretti said to me: “Manny, please stay after class.” When class was over I stayed in my desk. My mind wandering what I had done or failed to do...

Moretti: “Manny, the Navy’s Office of Naval Research has awarded me a contract to study the blunt body problem using the Navier-Stokes equations. I have 100 hours of computer time at the Courant Institute on their CDC 6600. I would like your help. Are you interested?”

Any sane person should have said “I appreciate your interest in me, but you should really look for a more qualified person.” I was sane, but I was also very young and didn’t know any better, so I responded: “Of course, I would love to help you, Professor Moretti. But, what is the blunt body problem?”



Fig. 9 Watercolor (courtesy of the Moretti Family)

And that day my life was changed forever. I had come looking for Ferri, but Moretti found me.

Gino Moretti died on March 15, 2015 in Williston, Vermont, surrounded by family. A few years before his death he had written a letter to his grandchildren and great grandchildren. A portion of it was included in his obituary: “The Quintessential Maverick. Mr. Maverick was a Texan farmer who never branded his cattle. When they got mixed with other herds and had to be rounded up they were called “Maverick’s”; metaphorically the name is used to characterize “a person not labeled as belonging to any one party, faction, etc. who acts independently”. A few months after my arrival to the States my new colleagues said that I was a maverick; forty-nine years later I see that their guess was very good. I taught and practiced mathematics, fluid mechanics and computational techniques, wrote memoirs, short stories and some verses, played piano and composed a few pages of music, did some work as a cabinet maker, and filled scores of pages with sketches, watercolors and pastels. A genius can do many such things perfectly and deserves a place of honor in history; a maverick just does them as well as he can. I tell you, however, I enjoyed everything I did and I had a very happy life. I wish I could have shared more time with you; I always enjoyed your visits even if the “generation gap” is great and it becomes greater and greater every day. I want to tell you my recipe for happiness: Be interested in whatever you do, do not be afraid of doubting but use it for stimulating new discoveries; pay attention to what old people say but use your judgment to act... in short, be mavericks!”

Reference

1. Caparrini, S.: A walk through Mathematical Turin. *Math. Intell.* **32**(4), 59–68 (2010)

A Renaissance Man: Personality and Generosity of a Dear Friend

Maurizio Pandolfi

Renaissance Man (Fig. 1) refers to a person capable of excelling in a broad range of intellectual fields. A curious mind, open to new thoughts and ideas. At the same time, a concrete and pragmatic person, while projected in the future.

Gino Moretti was Italian, not only by birth, but also by character, education, and culture. He had all the characteristics that fit into the definition of a Renaissance Man.

I met Gino when in June 1970 he gave a seminar at my University, the Politecnico di Torino. He spoke about the time dependent technique used in the prediction of the transonic flow about a cylinder (Figs. 2a and b).

I liked very much the presentation and the liveliness of his talk. After some questions and answers on unsteady flows, we formed a connection with each other. My great esteem and friendship with him first, and with his family later, was born in that occasion.

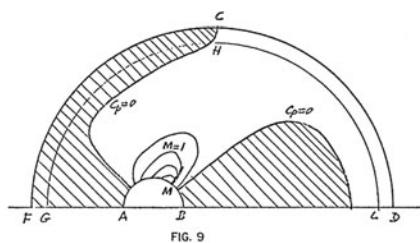
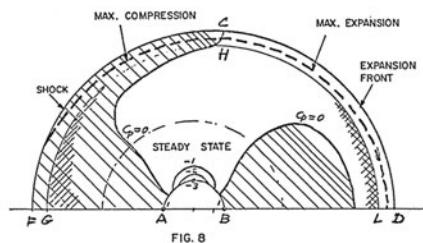
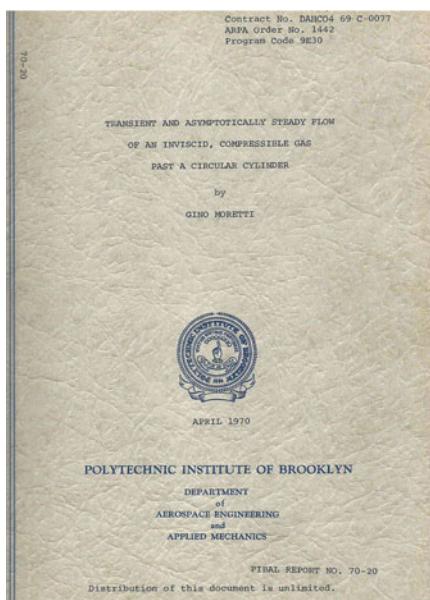
He offered me a postdoctoral fellowship, at the Polytechnic Institute of Brooklyn, in the Research Center in Farmingdale. I talked about this proposal at home. My wife Esa agreed. I accepted.

The 25th of November of that year, I recall well the date, I flew to New York. I was excited, even a bit timorous. It was the first time I crossed the Atlantic, facing such a new experience.

Gino and his wife Anita (Fig. 3) gave me full hospitality at their home, as an old family friend.

The next day was Thanksgiving, a gray and cold day. We traveled North, to the New York upstate and had an American traditional lunch in an Italian restaurant. Very much appreciated. I felt well comfortable. They were nice and

M. Pandolfi (✉)
Politecnico di Torino, Torino, Italy
e-mail: maurizio.pandolfi@polito.it

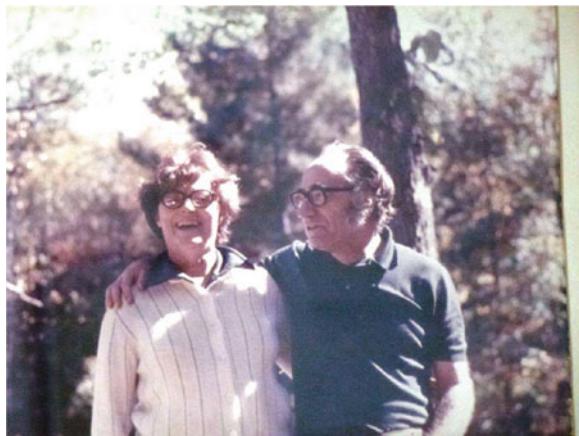
Fig. 1 Gino Moretti

(a)

(b)

Fig. 2 PIBAL REPORT 70-20 (Courtesy of Moretti family). (a) Cover page. (b) Transonic flow about a cylinder

kind. Also curious to know something more about my family, Esa, Selina, Paola, who were supposed to join me later.

Fig. 3 Anita and Gino

On the next Monday I began to live and work in Farmingdale. Gino gave me the text of a code on the 2D Blunt Body and asked me to understand what was inside. In Torino, I had worked experimentally on shock tubes. I knew—poorly—a bit of FORTRAN. I began to learn it on the field. I was surprised because most of the subroutines were labeled with girls' names, his daughters and fantasy names. The first impact was hard. I was frightened. Then things began to run smoothly. I started to practice and I learned a new job. After some months, he offered me to be a consultant at the Grumman Aerospace in Bethpage, close to Farmingdale. There I met two young researchers: Manuel Salas and Frank Marconi. They became very good friends.

I am not going to talk here about the important contributions of Gino Moretti to CFD. On his merits in science, widely known in our community, other friends have already commented or they will do it. Rather, I would prefer to recall the aspects of his personality that I had the opportunity to deeply know and appreciate.

Gino attended the Faculty of Mathematics at the University of Torino. He was a brilliant student. He attended his classes diligently. Some of them were boring, others he liked very much, as the lectures given by Prof. Francesco Tricomi (Figs. 4a and b), the teacher he preferred most. Tricomi has been coauthor with Prof. Carlo Ferrari (Figs. 5a and b) of an important book on transonic aerodynamics. Later Gino became Ferrari's assistant at the Politecnico. They held each other in great esteem for all their life.

However, most of his best and closest friends were at the Faculty of Arts and Philosophy. He used to spend hours with them, in long debates on themes not exactly belonging to sciences or mathematics. Gino told me that he liked particularly three of them (Fig. 6). Luigi Pareyson and Giorgio Colli, two opposite personalities (Fig. 7) who became well known philosophers, and Vittorio Cravetto (Fig. 8), poet and writer, who reached high positions in the RAI, the Italian Broadcasting Corporation. Gino was very impressed by the writing ability of Vittorio. He regarded him with respect and wonder. He wanted probably to compete, secretly, with him in

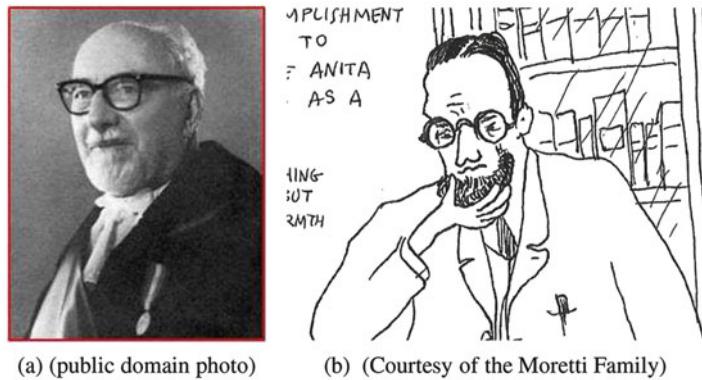


Fig. 4 (a) Giacomo Francesco Tricomi. (b) Younger Tricomi, as drawn by Gino

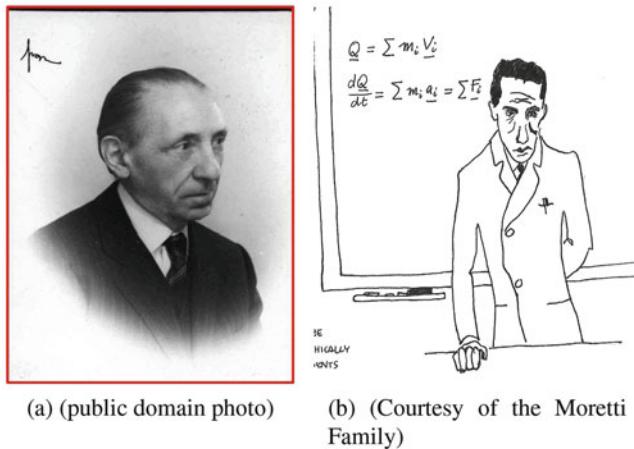


Fig. 5 (a) Carlo Ferrari (b) Younger Ferrari (the Panther) drawn by Gino

writing. He had no motivation to do it, because his own level in writing was already very high. He had excellent capabilities, strong imagination and fluent prose.

I was amazed when he drafted manuscripts. He wrote easily and fast. Never, or only seldom, he retraced his own steps in revisiting the text. The first version was very much close to the final one. His handwriting was neat and precise. No deletions. From the beginning he had very clear and detailed on his mind the global structure of what he was going to write as well as any final subtle use of adjectives.

My first impact with Gino's writings was through articles, reports or proposals. Only later I had the chance to know his *Illustrated Memoirs*. They are strips, as the ones you may find on the last pages of newspapers. Drawings with Indian ink, very expressive of both the people and the environment represented. Witty short comments on the side, written in English because the Memoirs were addressed to American readers.

Fig. 6 Three great friends and the small Gino, admiring them (Courtesy of the Moretti Family)

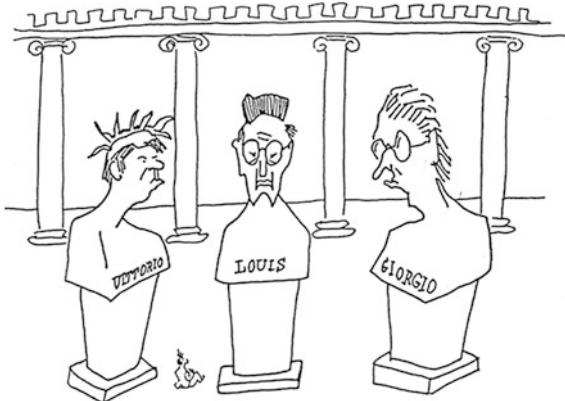


Fig. 7 Luigi Pareyson and Giorgio Colli (Courtesy of the Moretti Family)

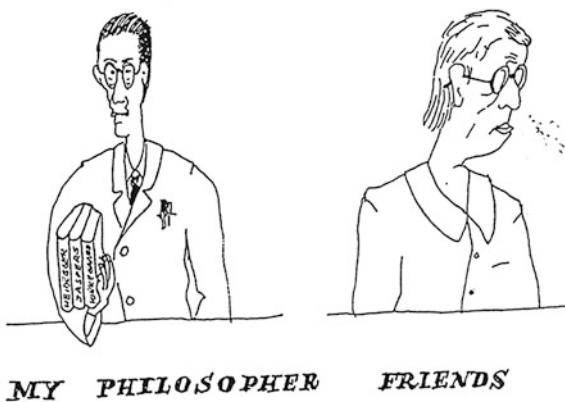


Fig. 8 Vittorio Cravetto (Courtesy of the Moretti Family)



He wrote (and drew), four books of Illustrated Memoirs:

- from his birth to the end of the *lyceum* (1917–34)
- years of the university (1935–39)
- falling in love, wedding, Russia campaign (1940–43)
- return home, up to the leave to Argentina (1943–48)

The *Memoirs* are permeated by subtlety and sagacity and by all the feelings Gino was able to pour in them. Two are the points I mostly appreciate. First, his memory in retaining an extraordinary amount of details about his past life, persons and events. Then his full domain of the English language, mainly acquired after his arrival in the States, well after his youth. My English is quite limited, thus I can't really judge his. But those who are well able to do so, have expressed very positive evaluations. Only the accent still betrayed his origin from Torino. After all, he wrote and spoke perfectly the Italian regional dialect from his birth place.

Among the different *renaissance* activities, the *Memoirs* are the ones I appreciate the most.

Later, some Italians friends (I was one of them) asked him to add an Italian version of the *Memoirs*. The comments could have been more understandable and enjoyable, particularly from the readers in Torino. The city and its people are so deeply and subtly described there! He did something less and something more. He didn't touch the original *Memoirs*, simply re-wrote the side comments as real autobiographical books, but this time in Italian.

The first book is titled: *Volevo tanto fare il tramviere, ma mi hanno cambiato I tram* (*I so wanted to be a trolley driver but they changed the trolleys*) (Fig. 9a). It refers to the more lively and brighter period of his life. The old Torino society is perfectly depicted. The second book refers to the period after the university. Gino, as adult man, faced full responsibilities in the political and social environment that later brought the country to the war. *Cavalli 8, uomini 40* (*Horses 8, men 40*) (Fig. 9b) is the title of this book. The inscription referred to the allowed maximum train-wagon load and was painted on the side of the trains sent to Russia and transporting horses or infantry. He had to leave for Russia as an officer right after his wedding to Anita. Ida, their first daughter, was a newborn. The book recalls the sufferings of the Russia campaign and collects letters he wrote regularly to Anita. Tragedies of a war, he defined *absurd* in the subtitle. The long way back to Torino, through Eastern Europe evokes the famous book *La tregua*, by Primo Levi, also from Torino. The book reminds how these meaningless and absurd conflicts result in suffering and tragedies. Finally, and fortunately, the happy reunion with Anita and Ida. This book was so different from the first one. Different were the periods and the responsibilities faced by the author. I received this book from Gino as a gift but I can't find it anymore. I probably lent it to some friend. I don't remember who. Unfortunately it's sold out.

Gino wrote more works that were never published. I saved two of these manuscripts: one describing the time he spent at the university and a second focused on the following years in Argentina. The first deals with a young man, perseveringly curious, in the *renaissance* acceptation of the concept. A pleasant reading about a



Fig. 9 (a). Volevo tanto fare il tramviere... (Courtesy of Moretti Family). (b) Cavalli 8, Uomini 40 (Courtesy of Moretti Family)

life with no worries, fully projected towards the future. In the second one, Gino had reached a full maturity. He was lecturing mathematics in the Superior Institute of Aeronautics and University of Cordoba. He wrote scientific books in mathematics and mechanics. The life in Cordoba wasn't what Gino expected for him and his family and he moved to the Atomic Center in San Carlos de Bariloche, hoping for better working conditions. Even there not everything went smoothly. The experience in Argentina had not been an easy one. Inflation destroyed most of their savings and forced the family to emigrate once more. This time to North America, to New York.

These were the only two autobiographical works. The other writings are mostly novels and tales on fantasy themes. Nevertheless, some link to his life can be almost always found there.

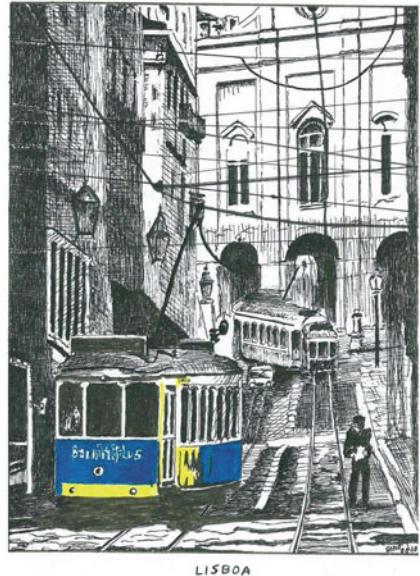
Years later he abandoned words and began to express himself through drawings (Figs. 10 and 11), with Indian ink or charcoal stick. Different from the *Memoirs* sketches, these works were more complex and complete. He also challenged himself with watercolor paintings. He used to send season's greetings on witty cards reproduced from his originals (Figs. 12 and 13). Once he sent us a full set of cards on his beloved Torino. They were beautiful. I wanted to keep them all. But I mailed most of them to friends, as I knew they would have appreciated such cards. I still have a few though.

Along with writings and drawings, Gino pursued avidly music. With the money earned with private lessons to lyceum students, Gino, a bit older than them, insatiably bought records, vinyl at that time, in the store La voce del Padrone



Fig. 10 Anita and Gino walking in Burlington, VT (Courtesy of the Moretti Family)

Fig. 11 Lisboa (Courtesy of the Moretti Family)



(Fig. 14), the Italian branch of the British His Master's Voice. Arnaldo or Dado Corio, another very good fellow of him in Mathematics, at the university, author of diffused books in Algebra my generation used during our lyceum years, loved classic music, as did Gino. Dado shared with him the choice of the records. Beethoven's Fifth Symphony was recorded on about five records. Each side lasted only about 3 min. The cost of the records corresponded to 15 h of tutoring! And the pleasure of listening to classic music was also disrupted, not only by the quality of recording at that time, but also by the recurring change of needles on

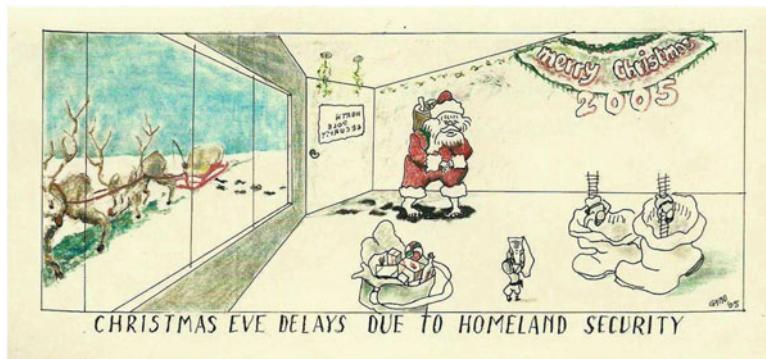


Fig. 12 Greetings 2005 (Courtesy of the Moretti Family)



Fig. 13 Greetings 2006 (Courtesy of the Moretti Family)

the gramophones. Gino was an amateur musician, but clever and very sensible. He practiced often in Argentina. Music played with friends was an usual amusement in Bariloche. In Freeport, my wife Esa played with him four hands piano pieces (Fig. 15). She remembers a beautiful Schubert's Sonata. Talking to Esa, Gino often regretted to have missed the opportunity to become an orchestra director. That was one of his great ambitions.

He was also an amateur composer: short pieces for piano and songs on his far away Torino. “*A good musicality and contents reflecting a nostalgic character of the sensitiveness of the author*” has been the comment from a professional in the field.

Gino Moretti was a generous man. A generosity shown when, in spite of problems and hard difficulties he found along his life, he was supported by his highly positive mind in recovering fresh courage and reconstructing a new future for all his family. These are also aspects of a *renaissance man*.

The restless love to Anita in the hard and tragic times in the 1940s is a great expression of such generosity. Their life went through events fully different from the soft life of my generation, more fortunate than Gino and Anita's. The war, the

Fig. 14 Gino and Dado at la Voce del Padrone (Courtesy of the Moretti Family)



Fig. 15 Four hands piano (Courtesy of the Moretti Family)



10 months of the campaign of Russia, the troublesome and suffered return. The 20 months of exile in Lausanne. The unsuccessful and painful attempt of Anita to join him at the Swiss border, when she was badly wounded by a German gun (Fig. 16). Then, soon after the war, tough economic circumstances in affording an academic career in his country. The courage of emigrating to Argentina to allow for his family a decent and less hard life. The tough times in Cordoba and the hard move to Bariloche. Finally, the economic and social earthquake of the country, prompted the decision to restart from scratch a new life, once more. A new emigration, new unknowns. He felt highly responsible for his family with five daughters. He afforded difficulties and efforts with a great generosity.



Fig. 16 Anita and Ida wounded by a German gun (Courtesy of the Moretti Family)



Fig. 17 Gino, Paola Laura, Patrizia, Esa and Maurizio

The same generosity that helped myself in my *American adventure*, order of magnitude lighter than his life, when he conveyed to me his scientific and human experience.

Again, the generosity offered to many Italian fellows working on CFD, was found at his research company GMAF (Gino Moretti And Friends), he founded after his retirement at the PIB.

I was lucky to meet Gino Moretti 46 years ago. My family had acquired great friends, Gino, Anita and their daughters. Today I embrace Gino's girls with much affection, as we all together remember and honor here their father (Fig. 17).

Gino Moretti at GASL, Brooklyn Poly, and the Later Years

Pasquale M. Sforza

1 Moretti Arrives in the U.S. and Joins GASL

In June 2000 a meeting of the “Friends of Ferri” was convened in Melville, NY to commemorate the 25th anniversary of the death of the esteemed aerodynamicist Antonio Ferri. Among the contributed reminiscences was one by Gino Moretti entitled “My Years at GASL with Ferri” It started out as follows:

“I spent the entire month of March 1958 traveling on the eastern corridor, up and down from New York to Boston, then to Princeton and again to New York, visiting industrially oriented companies and universities, without a pre-organized plan, from day to day following the suggestion of the last person I had visited... The last day in March, early in the morning, I was introduced to Ferri... I was somewhat intimidated (Note: It may be difficult for those who knew Moretti to imagine him being intimidated, but his respect for Dr. Ferri’s genius was deeply felt). Ferri’s directness and irreverence impressed me... That is how I came to the States.”

At that time Dr. Ferri was Head of the Department of Aeronautical Engineering and Applied Mechanics at the Polytechnic Institute of Brooklyn. For a discussion of the history of aerospace engineering at Brooklyn Poly, see Ref. [1]. Dr. Ferri was also President of General Applied Science Laboratories (GASL), a company he founded with Lee Arnold, Head of NYU’s Aeronautical Engineering Department, and with the encouragement and participation of Theodore von Karman. GASL was heavily engaged in the industrial side of advanced aerospace engineering research and over the years its traditions have been carried on by a number of Brooklyn Poly and NYU alumni and it is now part of Orbital ATK.

P.M. Sforza (✉)
University of Florida, Gainesville, FL, USA
e-mail: sforzapm@ufl.edu

As a result of that meeting in March 1958, Dr. Ferri hired Moretti as a Senior Scientist at GASL thereby starting a spirited relationship between them. With Dr. Ferri's help, Moretti was able to leave Argentina where he had been a professor at universities in Cordoba and Bariloche for the previous eight or so years, and emigrate to the U.S.

2 Moretti Joins Brooklyn Poly

In 1964 Dr. Ferri and a small group of professors left the Polytechnic Institute of Brooklyn to establish a high-speed laboratory at the University Heights campus of New York University in the Bronx, New York. At that time Professor Martin Bloom was on leave from Brooklyn Poly and serving as Vice President of GASL. He was called back to the Polytechnic Institute of Brooklyn and appointed Head of the Department of Aerospace Engineering and Applied Mechanics, and another new era in developing aerospace engineering education at Brooklyn Poly had begun. In 1965 the writer joined the faculty of Brooklyn Poly and participated as a new Assistant Professor in the transfer of the Polytechnic Institute of Brooklyn Aerodynamics Laboratories (PIBAL) facilities to their new home at Polytechnic's Long Island Graduate Center (LIGC) in suburban Farmingdale, New York, 12 miles northeast of their original location in Freeport, New York. Brooklyn Poly's new Preston R. Bassett Research Laboratory, formally dedicated in 1966, was constructed with financial assistance from NASA and provided attractive surroundings for consolidation and improvement of the PIBAL facilities (see Fig. 1).

Professor Bloom was instrumental in bringing Moretti to these new Brooklyn Poly facilities to assist in the development of a major effort in computational fluid mechanics conceived by Bloom and Professor Stanley Rubin. As Moretti put it in his reminiscences of Ferri:

“I opted for moving to Brooklyn Poly and a more sedate life as a teacher. It did not work out like that but I do not regret my years at Poly either... The NYU campus was too far from home, I never liked commuting and Ferri was not a letter writer. He joined Tricomi and Ferrari as the third of my mentors to whom I owe gratitude as long as I live.”

An idea of the state of the on-site computer facilities at Brooklyn Poly's new research laboratory is illustrated in Fig. 2. There were several women who carried out computations for the faculty and thereby became knowledgeable about computing from its earliest stages. Catherine Fahy, a wonderfully capable and meticulous member of that computational staff, worked as a programmer with Gino for many years. A landmark symposium on computational aerodynamics was held in June, 1972 to commemorate the 25th anniversary of PIBAL. Given the state of computing facilities at the time it is not surprising that one of the two papers presented by Moretti at that conference (see Fig. 3) was aptly titled



Fig. 1 The Preston R. Bassett research laboratory of the polytechnic institute of Brooklyn

Fig. 2 Computer (Bendix G-15) installation at the Preston R. Bassett laboratory in 1966



“Computational Aerodynamics Using Mini Computers”. This conference was the basis for the founding of the Journal of Computers and Fluids by Martin Bloom and Stanley Rubin.

PROGRAM

I. 8:45 A.M.	II. 8:50 A.M.
L. I. Sedov Academy of Sciences, U.S.S.R. (title to be announced)	Patrick J. Roache Ecodynamics Research Associates, Inc. A Semidirect Method with Non-Iterative Boundary Coupling for Viscous Flows
R. T. Jones NASA Ames Research Center History of Airfoil Development and Some Problems for the Future	F. Walkden, P. Caine and P. Laidler University of Salford On a Technique for Computing Higher than First Order Accurate Solutions of the Supersonic Flow Equations
M. H. Bloom Director, Aerodynamics Laboratories Account of Aerodynamics Laboratories, 1954-1979, and Citation of Founding Director, A. Ferri	E. N. Tinoco, J. D. McLean and F. T. Johnson The Boeing Aerospace Company and J. M. Luckring NASA Langley Research Center Free Vortex Flow Analysis Using Panel Methods
W. Ballhaus and R. Bailey NASA Ames Research Center Computational Aerodynamics Using Large Computers	T. de Neef Delft University of Technology and G. Moretti Polytechnic Institute of New York Shock Fitting for Everybody
G. Moretti Polytechnic Institute of New York Computational Aerodynamics Using Mini Computers	Manuel D. Salas NASA Langley Research Center Flow Properties for a Spherical Body at Low Supersonic Speeds
A. Jameson Courant Institute of Mathematical Sciences and	F. G. Blottner Sandia Laboratories Variable Grid Scheme for Discontinuous Grid Spacing and Derivatives
J. South NASA Langley Research Center Recent Developments in Subsonic and Transonic Aerodynamics	U. Ghia and K. N. Ghia University of Cincinnati and S. G. Rubin and P. K. Khosla Polytechnic Institute of New York Study of Incompressible Flow Separation Using Primitive Variables
R. Peyret Université de Paris J. C. Le Balleur and H. Viviani ONERA, France Numerical Studies in High Reynolds Number Aerodynamics	M. O. Soliman and A. J. Baker University of Tennessee/Knoxville Accuracy and Convergence of a Finite Element Algorithm for Laminar Boundary Layer Flow
E. Krause Aerodynamische Institute, Aachen, Germany Strive for Accuracy-Improvement of Predictions	B. Grossman Grumman Aerospace Corp. A Modern View of Vortical Singularities in Conical Flow
R. T. Davis University of Cincinnati and S. G. Rubin Polytechnic Institute of New York Non-Navier-Stokes Viscous Flow Computations	P. K. Khosla and S. G. Rubin Polytechnic Institute of New York A Fast Iterative Algorithm for Multi-Dimensional Flow Problems
T. Taylor Aerospace Corporation Recent Developments in the Application of Spectral Methods	These papers are scheduled to appear in a Symposium in Print in "Computers and Fluids".
J. D. Whitfield, S. R. Pate, W. F. Kemzey and D. L. Whitfield Sverdrup/ARO, Inc. The Role of Computers in Future Aerodynamic Testing	A special issue of the International Journal, "Computers and Fluids", will highlight these papers.
V. M. Kovpenko and N. N. Yanenko Institute of Theoretical and Applied Mechanics, Novosibirsk, U.S.S.R. Numerical Method of Solution of Viscous Gas Equations on Moving Grids	

Fig. 3 The program for the symposium on computers in aerodynamics, held at the Polytechnic's Long Island center on June 4 and 5, 1979

At Brooklyn Poly, Moretti started research with a promising young Brooklyn Poly student named Manuel Salas. Salas wrote in the preface to his book, *A Shock-Fitting Primer* (CRC Press, NY, 2009):

"This book has its origins in a letter from Gino Moretti to me, dated January 1, 1990. Moretti had started writing a book on one-dimensional problems that he planned as the first volume of a two-volume set. Two-dimensional problems would be covered in the second volume.

Moretti started the letter with “I rejoice in the idea that, with your help, this book may finally be completed.” We collaborated on the manuscript for a couple of years, but unfortunately other work commitments pulled me away and the manuscript was put in a drawer.”

As far as teaching courses at Poly was concerned, Moretti’s temperament was more suited to advanced graduate courses and guided studies with post-doctoral scientists, although his seminars were always a source of interest and enjoyment by students as well as faculty members. He developed a good working relationship with scientists at Grumman’s research department collaborating with Bernie Grossman, who joined Brooklyn Poly’s faculty for some time before moving on to Virginia Tech and Frank Marconi, who ultimately did his doctoral work under Moretti’s guidance. Moretti’s ability to inspire researchers attracted outstanding post-docs from Europe like Maurizio Pandolfi, Tom DeNeef, Marcello Onofri, Mauro Valorani, Antonio Lippolis, among others. Moretti’s irreverent (recall that Moretti was impressed by Ferri’s irreverence) sense of humor was another facet of his character that attracted young research students. For example, who could forget his report entitled “A Pragmatic Analysis of Discretization Procedures for Initial-and Boundary-Value Problems in Gas Dynamics and Their Influence on Accuracy, Or: Look Ma, no Wiggles!” (Polytechnic Institute of New York, POLY-AE/AM Report No. 74-15, September 1974).

In 1986 the AIAA Fluid Dynamics Award was presented to Gino Moretti “For pioneering work in computational fluid dynamics, and the application of innovative numerical techniques to complex problems in gas dynamics. By stressing the need for consistency between computational models and fluid physics, he has set a standard for all those involved in CFD.” Around this time Moretti retired from Brooklyn Poly as Professor Emeritus and devoted his time to GMAF, Inc. (Gino Moretti and Friends, Inc.), a consulting company he started in 1979 and nurtured until dissolving it in 1993. In 1987 his 70th birthday was celebrated by friends at a restaurant on Long Island and subsequently *The Journal of Computers and Fluids* (vol. 17, issue 1, 1989) dedicated an entire issue to honoring him.

Moretti’s outlook on computational methods and on the importance of scientific leaders is best expressed by his own words in reviews of two books for AIAA Journal [2, 3].

3 Some Personal Memories of Moretti

Gino called himself a maverick, but to me he was a Renaissance man. Years ago he was building an electric organ, and whenever I’d visit Gino and Anita back on Casino Street in Freeport, I’d ask him how it was coming along. This was a delicate issue because it was truly a daunting project and Gino would immediately become agitated and, with much gesticulation, recite a litany of complaints about the whole idea. Equally long ago he told me how, on a trip to Italy, he had purchased a number of inexpensive suitcases, loaded them with beautiful tiles, and with Anita’s

help toted the weighty parcels home to Freeport. Then, of course, being Gino, he masterfully laid the tiles himself. I noticed early on that he addressed the keyboard of the computer as if he were playing the piano, poised, delicate, sometimes emphatic, and always expressive. I could imagine him typing away on his beloved Olivetti portable typewriter in just the same way (he made a note in his memoirs of his first typewriter—it was 1929).

He had a rich academic career as a student and displayed his artistic talents in drawing portraits of his professors. He once made for me a charcoal sketch of Professor Boggio who would sleep at the blackboard, a picture I still cherish. Of course, to go along with the drawing Gino would give a priceless impersonation with all of the sounds and gestures. A pen sketch of the house in Torino which was the birthplace of the mathematician Lagrange has a place on the walls of my home as does a watercolor sketch of fields by a river in New England. As he wrote his illustrated memoirs he would share them with me and other friends during our weekly Friday luncheon at Ubaldo's restaurant in Farmingdale. His enactment of the memoirs was delightful and there couldn't be a better venue in which to enjoy them.

Later, his writing talents led to the publication (in Italy) of “Volevo tanto fare il tramviere—ma mi hanno cambiato i tram” and “Cavalli 8, Uomini 40—memorie di una guerra assurda” (which includes some of his fine pen drawings at the head of each chapter). The covers of these two books, drawn by Moretti, are illustrated in Fig. 4. The former expressed his early love of trams and his desire to grow up to be a tram driver but he “changed trams.” The latter was his remembrances of being a soldier in the Italian army in “an absurd war.” The title derives from a departing railroad car which had printed on its side the legal capacity: “8 horses or 40 men.” I was particularly happy with this second book because I had suggested the title and the drawing that graces the cover of the book, Anita waving goodbye to Gino as his military train left the station.

When Gino and Anita left Freeport, New York, for Burlington, Vermont, Gino and I began a correspondence that lasted for many years. During those years Gino displayed another of his talents, writing short stories which he permitted me to edit. The last time I saw Gino and Anita together was in a restaurant in the Montreal airport. One of his daughters was kind enough to drive them to meet me there as I was on my way to a meeting in Quebec. This was in September 2001, and we had no idea what was coming. I saw Gino again in September 2006 on the occasion of his receiving an award from the Italian government for his scientific achievements. We had a wonderful time there, especially on a boat ride down the Tiber to Ostia. Of course, the boat broke down on the return trip and we enjoyed the lower reaches of the Tiber by night.

Joining forces with a Renaissance man like Gino was an operatic trip whose overtures and arias one would have been a fool to pass up.

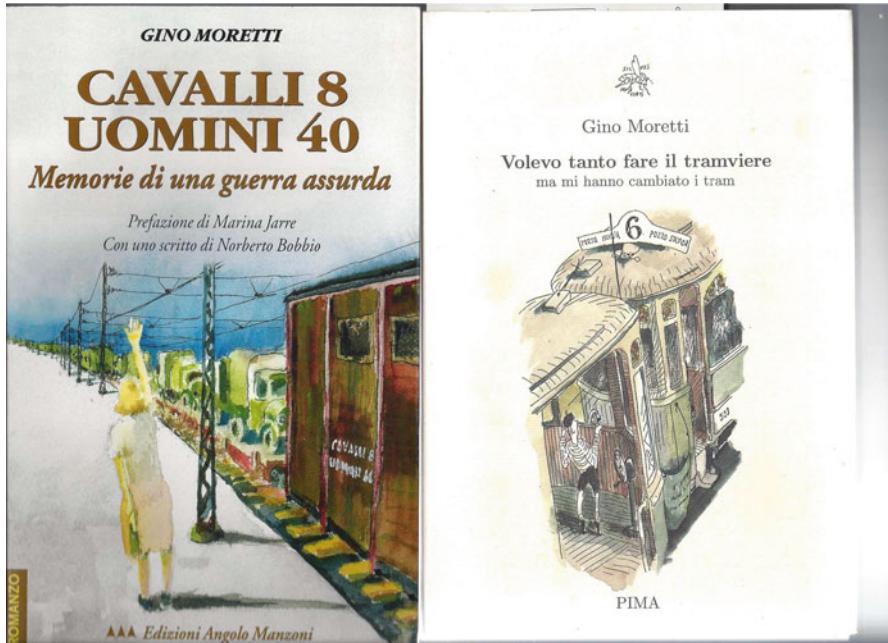


Fig. 4 The covers of Moretti's two published memoirs (courtesy of the Moretti family)

References

1. Sforza, P.M.: A history of aerospace engineering at the polytechnic institute of Brooklyn. AIAA 2009-1164, AIAA 45th Annual Meeting, Orlando, FL (2009)
2. Moretti, G.: Book review. *AIAA J.* **33**(10) (1995)
3. Moretti, G.: Book review. *AIAA J.* **34**(6) (1996)

Gino Moretti, the Maestro of Gas Dynamics

Robert MacCormack

Gino Moretti was an exceptionally multi-talented man. My recollections of him can only be but one piece of a complex mosaic of him. He was extremely respectful for the physics of fluids and intolerant for sloppy or wasteful numerical simulations of fluid phenomena. To get a complete portrait of Gino Moretti, if it is at all possible, would take an extraordinary large number of people who interacted with him from diverse viewpoints. Most of my observations about this great man come from attending meetings where he spoke on computational fluid dynamics.

I believe that I first met this formidable man at a conference on computational fluid dynamics held in Farmingdale at the Polytechnic Institute of New York during the 1970s. All I remember of my presentation was me yelling “Yes!” down from the podium at its conclusion and Gino yelling “No!” as forcefully back at me from the audience. Perhaps I was yelling “No” and he was yelling “Yes.” I can’t remember which for sure. The subject of our disagreement was probably on the numerical treatment of shock waves. How could it have been otherwise? Later one evening during the conference a small ensemble played classical music for our entertainment. Not one to be moved by classical music, I looked around absently and saw Prof. Moretti sitting a few rows ahead of me thoroughly absorbed in his enjoyment in the sound flowing toward us. I envied him that.

I saw him again later at a meeting held in Champaign Urbana at a meeting at the University of Illinois. Dean Chapman, then Director of Astronautics at NASA Ames, my boss three levels above me and a very persuasive champion of computational fluid dynamics, even predicting its supplanting wind tunnels, presented a talk assessing the future of aeronautical design with a viewgraph showing a complete aircraft embedded within a grid containing a million mesh points, unheard of at that time. The next speaker was Gino Moretti. He walked

R. MacCormack (✉)
Stanford University, Stanford, CA, USA
e-mail: rwmacc@aol.com

slowly to the podium and introduced himself as a poor shopkeeper from Brooklyn and needed only six mesh points between the body and shock wave, stating “Obesity is the product of affluence and lack of self control.”

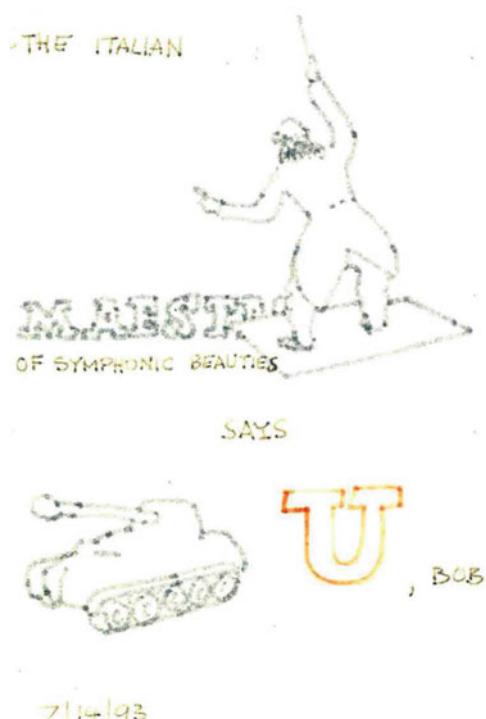
At a meeting at the University of Wisconsin—Madison, Hans Otto Kreiss presented a numerical simulation of multiple shock waves crisscrossing one another. It was very impressive, except the waves showed considerable smearing. Moretti got up to comment on the presentation and said something to the effect that “Has all my work been for naught?” It was said with wounded passion, as in an operatic scene where a dagger has been thrust through his heart. It was very moving. The audience was silent. Moretti then described his respect for the accurate representation of shock waves and the numerical procedures required to represent them correctly. Kreiss was awestruck, defending his simulation by indicating in effect that the forest was well represented although the individual trees were not intended to be. I also got up to comment, trying to bridge the gap between these two renowned fluid dynamicists.

Shock waves are discontinuities within the flow. In gas dynamics, inviscid flow, they are strictly so and even in viscous flow, at high Reynolds numbers, they are nearly so. Few calculations need to go to the extreme expense of placing mesh points spaced at free mean path distances for their resolution. Gino Moretti devised numerical procedures for treating them precisely as discontinuities, using two mesh points co-located at the shock, one at freestream conditions, just ahead of the shock wave, and the other determined from the geometry of the shock wave and the Riemann shock jump relations. This is not easy to do; it takes genius, particularly for embedded shock waves or moving ones. But it adds precision, eliminates waste and Gino Moretti was the master of it. He was the Maestro of Gas Dynamics and composed numerical programs of Symphonic Beauty. (See Fig. 1, an original Moretti work of art expressing thanks to me for stating the obvious about his work in 1993.)

There were two tribes: the *shock fitters* led by Moretti and his students and the *shock capturers*, followed by mostly everyone else. The term shock capturing was termed by Harv Lomax, head of computational fluid dynamics at NASA Ames. It is a clever term, indicating action where there is none. Instead, by doing nothing, a steep compression, with wiggles fore and aft, appears within the mesh over three to five mesh points representing a shock wave, sloppy, but good enough for most, but not for “Look Ma, No Wiggles!” Moretti and his followers. The two tribes were at war in the 1970s and 1980s. I believe that I witnessed an armistice one day in the kitchen of Manny Salas during a conversation between Phil Roe and Gino. Roe in a stroke of genius devised an approximate Riemann solver, which I hesitate to say could *capture* a shock discontinuity over just two mesh points. Because Roe’s procedure adhered to characteristics theory, strongly advocated by Moretti, I believe that Gino then allowed a truce, but not a capitulation, to take finally place between the two tribes.

The last time I saw Gino was in his birth city of Turin several years ago. I spoke at the Politecnico di Torino and he was there visiting old friends throughout Italy before, as he said, “they passed on or lost their minds.” I felt like the flood control

Fig. 1 The Italian maestro of symphonic beauties says thank you, Bob 7/14/1993



engineer who was asked to give a seminar on his work upon entering heaven and was told to remember that Noah was in the audience. I was apprehensive, remembering my first encounter with him at PINY in Farmingdale. It went better this time, not perfect, but better. He was still the master. Italy has produced many great masters, including Leonardo da Vinci and Michelangelo. I never met them, but I was fortunate to meet Gino Moretti.

There Was a Conference in Europe

Philip Roe

There was a conference in Europe, I think one of the GAMM meetings, where a guy was being an annoying nuisance. Let us call him Dr. X. Dr. X's paper had not been accepted, but he was using the question period after the other talks to announce his results. Dr. X tried to do this after Gino's talk. He had understood correctly that the point he wanted to make was one that Gino would agree with, but he did not realize that Gino would regard the point as trivial and not worth making. Gino, of course, was acting as his own chairman, and had signaled to Dr. X that he would take his question, but then immediately realized what was going on.

As soon as Dr. X began to speak, Gino started to scan the audience for the next question, making it very obvious that he considered this to be much more important than paying any attention to Dr. X. In the course of this search he contrived to shift his stance so that as Dr X completed what he had to say, Gino was facing almost directly away from him. When Dr. X stammered to a halt, Gino spoke a single curt and dismissive word. "Good!". Then he extended his arm magisterially to his chosen questioner.

Now this story has a sequel. A couple of years later, there was a conference in Oxford, and the conference dinner was held in the dining hall of one of the ancient colleges. High ceiled, old portraits hanging from stone walls. I was telling this story to the people at my table, and I really tried to channel Gino's spirit. When I described Gino's concluding gesture, I did my best to mime it as well; a large sweep of the arm expressing contempt for Dr. X, leaving no doubt who was in charge, and lifting my head to establish eye contact with the next questioner.

P. Roe (✉)
University of Michigan, Ann Arbor, MI, USA
e-mail: philroe@umich.edu

To my astonishment, what happened next was that the head waiter of the college came from the far side of the hall at a brisk run and anxiously wished to know what it was that I required. He seemed to think that perhaps the wine had not met my exacting standards, but whatever it was, he would certainly do his best to fix it.

So for one brief moment I actually WAS Gino, at least that part of Gino that automatically gained attention and respect. I have often felt tempted to try that technique on waiters again, but actually I never have. I think that only Gino deserved to walk in Gino's shoes.