# Development and Applications of the Finite Point Method to Compressible Aerodynamics Problems

E. Ortega
E. Oñate
S. Idelsohn

Monograph CIMNE Nº-143, June 2014

# Development and Applications of the Finite Point Method to Compressible Aerodynamics Problems

E. Ortega
E. Oñate
S. Idelsohn

Monograph CIMNE Nº-143, June 2014

**International Center for Numerical Methods in Engineering**
Gran Capitán s/n, 08034 Barcelona, Spain

INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING
Edificio C1, Campus Norte UPC
Gran Capitán s/n
08034 Barcelona, Spain
www.cimne.com

First edition: June 2014

**DEVELOPMENT AND APPLICATIONS OF THE FINITE POINT METHOD TO COMPRESSIBLE AERODYNAMICS PROBLEMS**
Monograph CIMNE M143

ISBN: 978-84-941686-7-3

Depósito legal: B-15424-2014

*to Natalia*

*“Simplicity is everything. After having exhausted all the difficulties, after having (…), and more (…), then simplicity emerges with all its charm, like art’s final seal. Whoever wants to obtain this immediately will never achieve it: you can’t begin with the end.(…) it’s no easy matter.”*

*Fryderyk Chopin (Eigeldinger, 1988)*

# Contents

# Acknowledgments

Many people have contributed to make this work come to fruition. First of all, I would like to express my gratitude to Prof. Eugenio Oñate for making this project possible, and also for his support and confidence all these years; and to Prof. Sergio Idelsohn, for his kindness, support and advice. I would also like to thank many friends and colleagues at CIMNE, with whom I have shared these years, and who many times have generously given me their time. To Dr. Roberto Flores, whose knowledge and support has helped me a lot. To Carlos Labra for many useful discussions and the help provided; to Eduardo Soudah, Hiram Badillo, Maurizio Bordone, Xavier Diego, Ariel Eijo, Mohammad Kouhi, Cuhautemoc Escudero and Nelson Lafontaine, among other good companions and friends. To the members of the GiD team, particularly Enrique Escolano. To Dr. Nestor Calvo, for being always willing to help; and to Chinapat Buachart, for the work done together. Thanks to many others who have contributed to this project, and it is difficult to recognize individually. I also dedicate an affectionate thought to Mercè Alberich, Rosa Olea and all the staff at CIMNE, who ease our daily work.

I would also like to express my gratitude to several persons who have not been directly involved in this work during these years, but are also a part of it. Particularly to Prof. Carlos Sacco, who initiated this project and guided me through my first steps in meshless methods. I am also grateful for his encouragement to continue my studies at CIMNE. Thanks to Prof. Mario D'Errico for my first lessons on aerodynamics, and first steps in teaching. To all my friends and professors at the IUA and the UNC in Argentina; also to Prof. Norberto Nigro, with whom I took the first CFD course, and especially to Prof. Carlos Paoletti, who gave me the opportunity to work on experimental aerodynamics, and taught me not only how to do research, but also to write it, in a course of action that has guided me ever since.

This research was initially funded by the European program of scholarships for Latin America Alβan (scholarship No E04D027284AR). Financial support and several inputs during this work have been also provided by the FP6 and FP7 European projects REMFI (contract No AST3-CT-2004-502895) and ALEF (contract No ACP7-GA-2009-211785). All these contributions are gratefully acknowledged.

Finally, thanks to my wife Natalia, for her love and generosity, for a lot of help, advice and encouragement. And thanks to our families and friends, for their unconditional love and support, and for giving always their best.

# Abstract

This work deals with the development and application of the Finite Point Method (FPM) to compressible aerodynamics problems. The research focuses mainly on investigating the capabilities of the meshless technique to address practical problems, one of the most outstanding issues in meshless methods.

The FPM spatial approximation is studied firstly, with emphasis on aspects of the methodology that can be improved to increase its robustness and accuracy. Suitable ranges for setting the relevant approximation parameters and the performance likely to be attained in practice are determined. An automatic procedure to adjust the approximation parameters is also proposed to simplify the application of the method, reducing problem- and user-dependence without affecting the flexibility of the meshless technique.

The discretization of the flow equations is carried out following well-established approaches, but drawing on the meshless character of the methodology. In order to meet the requirements of practical applications, the procedures are designed and implemented placing emphasis on robustness and efficiency (a simplification of the basic FPM technique is proposed to this end). The flow solver is based on an upwind spatial discretization of the convective fluxes (using the approximate Riemann solver of Roe) and an explicit time integration scheme. Two additional artificial diffusion schemes are also proposed to suit those cases of study in which computational cost is a major concern. The performance of the flow solver is evaluated in order to determine the potential of the meshless approach. The accuracy, computational cost and parallel scalability of the method are studied in comparison with a conventional FEM-based technique.

Finally, practical applications and extensions of the flow solution scheme are presented. The examples provided are intended not only to show the capabilities of the FPM, but also to exploit meshless advantages. Automatic *h*-adaptive procedures, moving domain and fluid-structure interaction problems, as well as a preliminary approach to solve high-Reynolds viscous flows, are a sample of the topics explored.

All in all, the results obtained are satisfactorily accurate and competitive in terms of computational cost (if compared with a similar mesh-based implementation). This indicates that meshless advantages can be exploited with efficiency and constitutes a good starting point towards more challenging applications.

# Resumen

En este trabajo se aborda el desarrollo del Método de Puntos Finitos (MPF) y su aplicación a problemas de aerodinámica de flujos compresibles. El objetivo principal es investigar el potencial de la técnica sin malla para la solución de problemas prácticos, lo cual constituye una de las limitaciones más importantes de los métodos sin malla.

En primer lugar se estudia la aproximación espacial en el MPF, haciendo hincapié en aquéllos aspectos que pueden ser mejorados para incrementar la robustez y exactitud de la metodología. Se determinan rangos adecuados para el ajuste de los parámetros de la aproximación y su comportamiento en situaciones prácticas. Se propone además un procedimiento de ajuste automático de estos parámetros a fin de simplificar la aplicación del método y reducir la dependencia de factores como el tipo de problema y la intervención del usuario, sin afectar la flexibilidad de la técnica sin malla.

A continuación se aborda el esquema de solución de las ecuaciones del flujo. La discretización de las mismas se lleva a cabo siguiendo métodos estándar, pero aprovechando las características de la técnica sin malla. Con el objetivo de abordar problemas prácticos, se pone énfasis en la robustez y eficiencia de la implementación numérica (se propone además una simplificación del procedimiento de solución). El comportamiento del esquema se estudia en detalle para evaluar su potencial y se analiza su exactitud, coste computacional y escalabilidad, todo ello en comparación con un método convencional basado en Elementos Finitos.

Finalmente se presentan distintas aplicaciones y extensiones de la metodología desarrollada. Los ejemplos numéricos pretenden demostrar las capacidades del método y también aprovechar las ventajas de la metodología sin malla en áreas en que la misma puede ser de especial interés. Los problemas tratados incluyen, entre otras características, el refinamiento automático de la discretización, la presencia de fronteras móviles e interacción fluido-estructura, como así también una aplicación preliminar a flujos compresibles de alto número de Reynolds. Los resultados obtenidos muestran una exactitud satisfactoria. Además, en comparación con una técnica similar basada en Elementos Finitos, demuestran ser competitivos en términos del coste computacional. Esto indica que las ventajas de la metodología sin malla pueden ser explotadas con eficiencia, lo cual constituye un buen punto de partida para el desarrollo de ulteriores aplicaciones.

# 1 INTRODUCTION

The development and applications of meshless methods[1] have grown substantially over the last decades. As an example, a recent consult to Elsevier's SciVerse[2] database shows that from 1995 to 2012 the growth curve of peer-reviewed publications in meshless methods per year fits to an ascending Gaussian bell, going from a few papers published around 1995 to 289 papers in 2012. The total number of journal articles and conference papers during these years is about 2580, distributed in areas of application such as engineering (39%), mathematics (22%), computer sciences (15%), physics and astronomy (8.4%) and material sciences (6.2%), among others (9.4%). These results are indicative of the interest of researchers in meshless techniques and explain the evolution that these methods have experienced in many fields of engineering and science.

The meshless procedures proposed to date have shown satisfactory accuracy and capabilities to deal with an ever-expanding field of applications, but there are further needs that call for immediate attention. In particular, these methods have today been unable to meet the requirements of practical problems, mainly due to difficulties inherent to the meshless technique, but also for a relative small number of applications. Hence, their capability in real scenarios remains vague an uncertain. With this important aspect in mind, the present work aims to contribute to the further development of the meshless technique.

## 1.1 Motivation and scope

The present work focuses on a particular meshless technique, the Finite Point Method, and aims at exploring its capabilities in the field of compressible flow problems. According to the challenges facing meshless techniques today,

[1] The term *meshless methods* will be used in this work exclusively to refer to meshless spatial discretization techniques employed for solving PDEs. Requisites for a discretization technique to be meshless are discussed in Section 2.2.1.

[2] SciVerse is an online database launched by the publishing company Elsevier B. V in 2010. This combines *ScienceDirect,* which provides fulltext journal articles, and *Scopus,* the largest abstract and citation database of peer-reviewed literature.

this investigation is carried out with special emphasis on practical applications, particularly in areas where meshless procedures can make useful contributions. The main guidelines and objectives driving this work are described next.

An important part of the research in meshless methods to date has been aimed to establish theoretical and practical criteria in order to isolate the aspects with a strongest impact on the accuracy and robustness of the methodology. These aspects, mainly concerned with the characteristics of the global and local discretization and the selection of proper approximation bases and weighting functions, have been studied and a significant progress has been made. However, the criteria and procedures adopted to ensure well-posed and accurate approximations often impose restrictions to the meshless method, which reduce its flexibility and make the technique more complex, expensive and not suitable for practical applications. The present work also focuses on these essential aspects, but under two main premises: not to impose restrictions penalizing meshless advantages, and simplifying the approximation settings in general application problems. To this end, the basic behavior of the FPM approximation is characterized in practical situations to obtain both, suitable ranges for the key parameters of the approximation (in view of implementing automatic adjustments) and numerical estimates allowing to infer the performance of the meshless method in real scenarios.

Regarding the discretization of the compressible flow equations, there are in the literature robust, reliable and well-established approaches that are difficult to improve and, of course, this is not the aim of this work. In this regard, the upwind flow solution approach presented in (Löhner, Sacco, Oñate & Idelsohn, 2002) is adopted with minor modifications, mainly related to the discrete forms of the equations. In addition, some alternative procedures (regarding convective stabilization and time integration) are also proposed to suit the requirements found in particular applications in a more efficient way. Various practical aspects of the basic FPM methodology, particularly accuracy and computational cost, are studied in this work in order to provide insight into the real capabilities of the meshless technique. All in all, the studies performed intend to fill the gap found in the meshless literature regarding practical applications, and providing further evidence helping to clarify the possibilities and competitiveness of the meshless approach.

Different three-dimensional applications of the FPM in the context of compressible flow problems are investigated in this work. The examples provided aim to demonstrate not only the basic capabilities of the meshless technique, but also to exploit meshless advantages in areas of research presenting opportunities for the development and promotion of meshless approaches. Automatic *h*-adaptive procedures, problems involving moving domain and fluid-structure interaction and viscous high-Reynolds number applications are some of the areas in which, from our view, meshless approaches can make useful contributions.

## 1.2 Organization

According to the guidelines and objectives given above, this work is divided into three main parts. After the literature review (Chapter 2), the first part deals with the construction and analysis of the FPM spatial approximation (Chapter 3). The second part is devoted to the development and assessment of the basic (inviscid) flow solution scheme (Chapters 4 and 5). Finally, practical applications and extensions of the basic solution methodology are carried out in the last part of this work (Chapter 6 to 9). The content of the chapters composing this investigation is summarized below.

Chapter 2 reviews the evolution of general meshless techniques, with focus on applications in the field of computational fluid dynamics. Among the various methods discussed, emphasis is given to the Finite Point Method in order to provide an adequate framework for the present research.

Chapter 3 presents the basic aspects of the FPM approximation and discusses important issues regarding the global and local problem discretization as well as the effects of the weighting functions on the accuracy of the method (mainly functional form and support size). Furthermore, an alternative procedure is proposed to build the local approximation. The objective is twofold. On the one hand, it aims at reducing the dependence on user-defined parameters and particular features of the spatial discretization (also providing automation of the procedures). On the other hand, it is also intended to improve the robustness of the FPM technique. The characteristics of the approximation are studied by different examples in order to provide numerical estimates that help to characterize the behavior of the FPM in typical application problems. Implementation details are also given through the different topics discussed.

Chapter 4 focuses on the basic (inviscid) flow solution approach. There, the theoretical bases of the governing equations and the numerical solution approach are described, and different aspects of the methodology are discussed. These refer to the resulting characteristics of the scheme and typical issues that arise in practice, for instance, in relation to the computational implementation. The application of boundary conditions in inviscid flow problems is also addressed in this chapter. Finally, several verification and validation test cases are provided in order to assess the basic performance of the numerical scheme (mainly accuracy).

Chapter 5 deals with a comparative analysis of the basic FPM procedure. This aims at completing the results obtained in the previous chapter with a more focused study of the accuracy and computational cost of the flow solver in the context of a practical application problem. The studies are performed in contrast with an equivalent edge-based FEM implementation, which can be assumed to be representative enough of conventional techniques used in practice. This will allow evaluating the feasibility and competitiveness of the meshless approach. In this regard, a simplification of the basic FPM technique is also proposed in this chapter in order to improve its efficiency.

The remainder of this work is dedicated to investigate particular applications of the proposed methodology in some of the areas where meshless solution approaches can make interesting contributions. These applications are described below.

Chapter 6 is devoted to the development of a meshless *h*-adaptive procedure designed to improve the spatial resolution and solution accuracy in an automatic manner. In order to drive the adaptive procedure two different error indicators are proposed: a typical feature-based indicator and a truncation error-based indicator. Since the latter has a closer relation with the discretization error, it is expected to provide a more effective improvement of the numerical solution. The performance of the different indicators and the adaptive scheme is studied by means of several numerical applications.

Chapter 7 presents applications of the FPM methodology to the solution of problems involving moving boundaries and fluid-structure interaction. To solve these problems, a variant of the basic solution scheme presented in Chapter 4 is proposed following Jameson's dual time steeping approach. The resulting scheme is tested on typical application problems involving aerodynamic bodies subject to prescribed movements in transonic flows. The adaptive technique presented in Chapter 6 is also applied in this context to enhance the resolving capability of moving discontinuities and localized flow features. In addition, a realistic fluid-structure interaction problem involving a wind tunnel model under inviscid transonic flow conditions is presented. The computational cost of the method is discussed through the examples provided.

Chapter 8 covers an application of the FPM to the solution of high-Reynolds number flow problems, a topic scarcely studied in the meshless literature. With a focus on attached boundary layer problems, the proposed analysis methodology aims to achieve both, an automatic meshless discretization of viscous layers (to exploit meshless advantages) and the construction of a robust numerical approximation in these zones. The study performed focuses on two-dimensional compressible flow problems but it is indicative of the possibilities to deal with more complex flow situations. A simple algebraic turbulence model is also implemented.

Chapter 9 extends the basic FPM methodology to the solution of two-dimensional shallow water problems. Due to the characteristic of the governing equations involved, this extension is carried out with minor modifications of the solution procedure developed in Chapters 4 and 8. The performance of the solution scheme is investigated by means of several numerical examples. In addition, a realistic problem involving tsunami simulation is also solved to give a better idea of the capabilities of the meshless approach in more challenging areas of application.

Chapter 10 presents the main achievements and conclusions of this investigation as well as the lines of research steaming from this work.

Finally, a list of the main publications arising during the development of this investigation is presented in the Appendix.

# 2 LITERATURE REVIEW

## 2.1 From scattered data fitting to PDEs

The first applications of scattered data fitting and interpolation emerged in the field of earth sciences with the purpose of handling spot spatial measurements. The earliest works date back at least two centuries ago (e.g. Gauss and Legendre's Least-Squares (LSQ)), or even before if other methodologies prior to the development of applied mathematics are considered. However, the techniques that gave origin to the meshless methods in use today for solving partial differential equations (PDEs) are more recent in time and were mainly developed over the past fifty years. Next, the most important contributions to the evolvement of meshless methods over the last half-century are described[3].

The point of departure for the development of meshless methods can be found in two techniques that appeared in the late sixties in the context of cartography and geostatistics. One of them is the kriging[4] method, proposed by Matheron to map surfaces using scattered sampled data (Matheron, 1969, 1970); the other is the interpolation procedure developed by Shepard (Shepard, 1968). In the kriging method, a weighted interpolation is used to minimize the estimated variance of an unknown point with respect to the weighted average of a collection of neighbors samples. Shepard, in turn, proposes a global (non-polynomial) interpolation method, but using normalized inverse-distance weighting functions to provide certain degree of data locality. These functions take unit values at the point of interest and approach zero far away from that point. In addition, the normalization applied makes the sum of the weights to be unity. This conferred interesting properties to the interpolation procedure which were intensively exploited

[3] The description provided in this work expands and complete the timeline presented in (Fasshauer, 2006).        .

[4] The name *kriging* was given in honor of the mining engineer D. G. Krige, whose master thesis (Krige, 1951) inspired the basic concepts underlying the theory developed by Matheron.

later[5]. Different modifications were proposed to the basic Shepard's method. For example, Franke and Neilson designed a weighting function with local support in order to reduce the influence of data points which are distant from the point of interest (Franke & Neilson, 1980). This improved the data locality of the original method and made it suitable for the analysis of larger data sets.

A few years after Shepard's method, Hardy introduced the concept of multiquadrics interpolation (MQs), where the interpolant is constructed as a linear combination of a radially symmetric basis function (acting as a weighting function) which depends solely on the position of the data points (Hardy, 1971). This model has been applied to a wide range of problems mainly in the field of geodesy, geophysics, geography and mapping. Later works due to Tarwater (Tarwater, 1985), Micchelli (Micchelli, 1986) and Madych and Nelson (Madych & Nelson, 1988, 1990) gave to MQs a solid mathematical foundation. Among other important advances, they demonstrated the invertibility of the resulting MQ system matrix and the convergence properties of the method.

Almost simultaneously with MQs, Harder and Desmarais proposed the method of Thin Plate Splines (TPS). This was originally intended to solve geometric surface interpolation problems in aircraft design (Harder & Desmarais, 1972). The method of TPS assumes that the data points are distributed on a thin elastic plate (or spline). As the points constrain the spline, the latter is forced to deform and the deformed shape gives the sought data fit. The best results are generally obtained by minimizing the bending energy of the spline (minimum curvature computing). The theory of TPS was mostly developed by Duchon (Duchon, 1976, 1977, 1978) and Meinguet (Meinguet, 1979, 1984).

An important and outstanding line of research was pioneered by Lancaster and Salkauskas in the 1980s. With roots in the Shepard's method and extensions due to McLain (Mclain, 1974, 1976), Franke and Nielson (Franke & Neilson, 1980) and a previous work by Lancaster (Lancaster, 1979), they developed the Moving Least-Squares Method (MLS) (Lancaster & Salkauskas, 1981). The MLS uses the classical polynomial LSQ technique, but introduces compact support weighting functions (e.g. inverse-distance or exponential functions) which are translated to specific points where the approximation is to be computed. This local approach has the advantage of avoiding the well-known polynomial oscillation problems often seen in classical methods (particularly in high-order approximations) and also facilitates the treatment of larger datasets. The MLS approximation is

[5] This property is recognized as *partition of unity*. In short, given an approximation domain $\Omega$ covered by overlapping subdomains $\Omega_\mathrm{i}$ with associated functions $w_i$, the latter constitute a partition of unity if satisfies that at a given point $\boldsymbol{x}$ $\Sigma_\mathrm{i}\, w_\mathrm{i}(\boldsymbol{x}) = 1$. Functions fulfilling this condition have been widely used in meshless methods.

continuous but, in contrast to the techniques previously seen, it is not interpolant[6] (Kronecker's delta property is not satisfied). Among other interesting features of MLS approximations, it is possible to demonstrate that if the approximation basis is complete in the polynomials of order *k*, polynomial functions up to the *k*-order can be reproduced in an exact manner[7]. In addition, the approximation constitutes a partition of unity (see Footnote 5) if the constant term is included in the approximation basis.

A landmark study published in 1982 by Franke had a large impact on the field of interpolation and data fitting (Franke, 1982). The publication presented a comparative analysis of the typical interpolation methods at that time, with emphasis on aspects such as accuracy, visual quality of the approximation, computer requirements (time and storage) and ease of implementation. The conclusions reached by Franke after a thorough analysis pointed to the methods using global interpolation as the most efficient. The Hardy's multiquadric method was the first one in Franke's ranking, immediately followed by Thin Plate Splines. Naturally, these results led to MQs, which were largely unexplored at that time, got under way and were generalized to other type of radial basis functions (e.g. TPS, Gaussian and cubic functions), giving origin to the development of different RBF-based methods, see for instance (Dyn, 1987, 1989). One important advantage of RBFs is that they rely only on Euclidean distances, which are trivially extended to arbitrary dimensions. This contrasts with piece-wise polynomial-based schemes, which usually involve the tensor product of the coordinate dimensions. Also, RBFs have a lower number of free parameters, which make the results less sensitive to particular problems and user settings. The computation of derivatives is simpler in RBFs because the coefficients of the approximation are constant and, among other features, the RBF approximation satisfies Kronecker's delta function and partition of unity properties. In relation to MLS, RBFs have the disadvantage that the approximation support is infinite; thus the solution of a global system of equations (typically ill-conditioned) is needed. In addition, consistency requirements are generally not satisfied, but an augmented (enhanced) basis can be used to this end. An updated comparison of MLS, RBF and kriging techniques can be found in (Krishnamurthy, 2005).

All these practical and theoretical developments set the foundations for linking scattered data fitting to PDEs. Mainly impulsed by the need to simplify conventional solution procedures, as well as to relieve problems

[6] Some modifications were proposed to enforce interpolation properties in MLS-based techniques, see for instance the Interpolating Moving Least-Squares method (IMLS) method of (Maisuradze & Thompson, 2003).

[7] As will be seen later, completeness and reproduction ability in the meshless literature are typically related to the concept of *consistency*. The use of this term in meshless methods has not a straightforward equivalence to that in differential equations (i.e. convergence to the continuous form).

related to domain discretization, researchers found in spatial approximation techniques working on scattered data sets some attractive alternatives. With a strong focus on RBFs and MLS-based techniques[8], meshless methods underwent a rapid development within an ever-growing field of applications. The more important lines of research are discussed next.

## 2.2 Meshless methods for PDEs

Numerous meshless techniques have been developed since 1990s for the solution of PDEs and the review articles describing the development, characteristics and applications of the different methods that can be found in the literature are numerous. For instance, a complete classification and description of the most relevant techniques is provided in (Fries & Matthies, 2004) and (Gu, 2005). There, theoretical, practical and comparative aspects of the methods are discussed (the former is more comprehensive). There are also some recent books that review the evolution of meshless techniques. In (Liu, 2003), the author systematically explores and establishes the theories, principles, and procedures leading to meshless methods, covering most of the existing techniques. Another book which can be seen as complementary to the latter is (Liu & Gu, 2005). In this work, more specific theoretical and implementation details are provided.

Taking into account that the evolution of meshless methods has been extensively described in the existing literature, we do not attempt here a new exhaustive review. Our aim is to present an adequate framework according to the motivation and objectives of the present investigation. Next, meshless methods are defined and classified, and the most relevant techniques are described according to the adopted spatial discretization approach. Then, the applications in the field of computational fluid dynamics are discussed separately and particular emphasis is given to the Finite Point Method, which is the methodology used in this work.

### 2.2.1 Initial considerations

In order to describe, analyze and understand meshless methods, first it is necessary to define what a meshless method is, and this leads one to wonder about the characteristics that make an approximation technique for PDEs to be a meshless technique. The answer to this question is not as trivial as it may seem because a multitude of particular situations in the different discretization approaches should be often accounted for. In spite of this, fortunately, the

[8] Even though widely used in data fitting and interpolation, kriging techniques have received a relatively minor attention in the field of PDEs.

different criteria adopted in the literature to define meshless methods concur on the general aspects which convey the nature of the meshless technique.

In this work, the criteria given in (Oñate, Idelsohn, Zienkiewicz & Taylor, 1996a) is adopted. There, three main characteristics of meshless methods are given. These can be summarized as follows (Sacco, 2002)

a. The discrete approximation of the unknown function and their derivatives must be defined *only* by the position of a set of points located within the analysis domain.

b. No volume or surface integration is needed to discretize the governing equations, or, if required, it should be independent of the approximation procedure chosen.

These concise criteria define a meshless method and make it possible to differentiate between mesh-based (conventional) approaches and meshless approaches. In addition to these statements, additional information is given in (Oñate, Idelsohn, Zienkiewicz & Taylor, 1996a) to clarify their application in particular situations. For instance, statement (a) does not mean that connectivity is not required in a meshless method. Actually, although typically local and subject to particular conditions, meshless techniques require some kind of connectivity between neighboring points[9]. Therefore, this statement only excludes methods that require some kind of spatial connectivity between points in which the final approximation depends on how these points are connected. Most of the typical conventional mesh-based methods (e.g. Finite Elements (FEM)) fall under this category; see other examples in (Oñate, Idelsohn, Zienkiewicz & Taylor, 1996a). On the other hand, statement (b), related to the discretization of the problem equations, does not exclude approaches based on integral formulations. In such cases, the quadrature procedure can be performed on background meshes, which are easy to generate and independent from the spatial approximation of the problem variables; thus preserving the meshless character of the methodology.

Based on the above criteria (a) and (b), it seems straightforward to classify the meshless techniques only taking into account the spatial approximation approach. In our view, although the different methods can be also classified according to the procedure used to discretize the problem equations (strong or integral forms), this characteristic does not seem to be a clear distinctive feature in meshless methods. The choice of strong or integral formulations seems to be more related to the nature of the problem to be solved and the customary approaches in the field of application.

[9] Additional considerations about point connectivity can be found in (Idelsohn & Oñate, 2006). There, several issues related to the changing nature of the connectivity, including updating and generation efficiency, are discussed. These aspects are closely related to the topological and quality requirements in a meshless approximation.

### 2.2.2 Overview of meshless methods

From the point of view of the spatial approximation techniques used in meshless methods, Radial Basis Functions, Moving Least-Squares (as well as other Weighted Least-Squares approaches)[10] and kernel interpolation are generally the most typical choices. These methods and some related approaches are described in this section.

#### 2.2.2.1 Radial basis functions

One of the first meshless methods, which paved the way for the research line connecting scattered data fitting to PDEs, was that due to Kansa (Kansa, 1986, 1990a, 1990b). This method adopted MQ radial basis functions for the spatial approximation of the problem variables and a collocation technique to discretize the governing equations (in strong form). Despite the fact that MQ is a global (infinite support) approximation method, and thus impractical for large scale problems, Kansa proposed to divide the analysis domain into overlapping subdomains. Hence, instead of a large global matrix, a small local problem is solved at each subdomain[11]. In this way, the local coefficients matrices are much better conditioned and the overall processing can be significantly accelerated by using parallel computing systems. The Kansa's method was successfully applied to different problems in computational mechanics; thus promoting the development of numerous meshless approaches.

Particularly in the context of RBFs, several methods using different basis functions have been proposed, e.g. (Fasshauer, 1997) and (Wendland, 1999). Among the typical are multi-quadrics (MQ), exponentials (Gaussian) and compactly supported RBFs ((Wendland, 1999) and (Wu, 1995)). The latter uses local approximation subdomains to reduce the computational cost and improve the conditioning of the resultant coefficients matrix. However, the accuracy of compactly supported RBF is lower in relation to functions having infinite support; moreover, global continuity is generally lost. The satisfaction of consistency conditions in RBF present some problems and sometimes the basis has to be enhanced with polynomial terms. Comparisons between MLS and RBF approximations are given in (Gu, 2005). It should be noticed that RBFs can be also employed in conjunction with MLS-based schemes in order to achieve interpolation properties or to increase the robustness of the procedures, see for example the Local Radial Point Interpolation method (LRPIM) (Liu & Gu, 2001b).

[10] This category also include methods using generalized finite difference approximation as this leads to very similar results, see (Liszka & Orkisz, 1980).

[11] Further developments in this line led to the definition of the popular compactly supported radial basis functions (CSRBFs), cf. (Schaback, 1995; Wendland, 1995).

#### 2.2.2.2 Moving and other Weighted Least-Squares techniques

In methods using MLS spatial approximations, one of the first to appear in the literature was the Diffuse Element Method (DEM) (Nayroles, Touzot & Villon, 1992). This method uses a MLS meshless approximation[12], generally in conjunction with polynomial bases, into Galerkin form equations (which requires defining a background mesh). Although this method offered some advantage over traditional FEM approaches, for instance regarding the continuity of the approximation functions, some problems mainly related to the application of essential boundary conditions, calculation of the derivatives and the accuracy of the quadrature procedure were observed. This led to improvements which crystallized in the Element Free Galerkin (EFG) method (Belytschko T., Gu & Yu, 1994). The latter adopted Lagrange multipliers to apply essential conditions[13] and proposed a more exact computation of derivatives and numerical integrals (using more integration points). The method was successfully applied to linear and non-linear elasticity, fracture and crack propagation problems, among others. Although the computational cost has been a disadvantage of the methodology (particularly when using high-order approximants), the EFG have become one of the most popular meshless techniques today; see for instance (Gu, 2005). Along the same line, another method called the Meshless Local Petrov-Galerkin (MLPG) was introduced in (Atluri & Zhu, 1998). In the MLPG, the Galerkin procedure is applied in a local manner, leading to weak forms of the equations which are solved in a local quadrature domain. Thus, no global background mesh is required. Further evolutions of the MLPG led to the Local Point Interpolation method (LPIM) (Liu & Gu, 2001a), which improves several of the deficiencies found in the MLPG (Gu, 2005).

Particular cases of the MLS approximation have also been proposed with the aim of reducing the computational cost (e.g. regarding the calculation of derivatives), but also to achieve particular properties of the approximation. These schemes, which are considered generically as Weighted Least-Squares (WLSQ), differ from each other mainly in the behavior of the weighting function and the manner in which it is applied in the approximation procedure. Examples of WLSQ schemes are the Fixed Least-Squares (FLS) and the Multiple Fixed-Least Squares (MFLS) methods; see other possibilities and its characteristics in (Oñate, Idelsohn, Zienkiewicz, Taylor & Sacco, 1996b). The employment of such approximations with collocation-based schemes is a typical approach in the meshless literature and a variety of

[12] Although the MLS approximation is not explicitly mentioned in the paper, the procedure described is exactly the same as in the MLS method; see (Duarte, 1995).

[13] In (Tongsuk & Kanok-Nukulchai, 2004), the MLS approximants in the EFG is replaced by kriging interpolation with the aim to address the issue of the enforcement of essential boundary conditions.

methods has been proposed. Among them, it is worth of mention the Finite Point Method (FPM) developed by (Oñate, Idelsohn, Zienkiewicz & Taylor, 1996a) and other similar approaches, e.g. the method due to (Cheng & Liu, 1999). These techniques retain most of the characteristics of the MLS (according to the weighting procedure chosen) and are also efficient and simple to implement. However, certain WLSQ approximations can present problems to enforce essential boundary conditions due to the lack of Kronecker's delta properties and continuity. Regarding the collocation procedure, the computational requirements are lower than those in methods requiring numerical integration, which results in a higher efficiency. Despite these advantages, ill-conditioning and instability problems (mainly due to the collocation procedure) must be addressed in certain applications. The approximation bases employed in WLSQ approaches can be polynomials or Taylor series expansions, see an overview in (Fries & Matthies, 2004).

#### 2.2.2.3 Integral kernel methods

Another group of methods using integral kernel estimates was inspired in the theory of wavelets, where a function can be represented by a combination of the dilatation and translation of a single wavelet. In these methods the approximation function is obtained through integration over the domain, where a kernel (or weighting) function (e.g. exponential or Gaussian or *n*-order splines) allows providing compact support and locality. The continuous form of the approximation needs to be discretized and solved by numerical integration[14]. Typical methods using integral kernel interpolation are the Smooth Particle Hydrodynamics (SPH) and the Reproducing Kernel Particle Method (RKPM). The SPH is a Lagrangian particle method[15] proposed in (Lucy, 1977) and (Gingold & Monaghan, 1977) to solve astrophysical problems[16]. The adoption of integral kernel approximations, derived from Monte Carlo statistical theory, was originally aimed at improving the accuracy of standard finite difference approximations in problems having a poor spatial resolution. An analysis of the kernel approximation and comparisons with other techniques can be found in (Monaghan, 1982). Some deficiencies in the basic SPH technique, mainly related to the dependence of the consistency on the features of the kernel function, the satisfaction of Kronecker's delta properties and the boundary treatment, led to the

[14] In spite of the integral character of kernel methods, some authors interpret them as a sort of particular collocation techniques, see for instance (Gu, 2005).

[15] In the context of methods such as the SPH, a particle is considered as an elemental fluid volume which moves in time under a set of forces acting on it. For a discussion on particle methods and their characteristics see (Idelsohn & Oñate, 2006).

[16] In these problems, the movement and interaction of objects in the space can be assimilated to the movement of particles in the interior of a fluid; thus, the classical equations governing the fluid behaviour can be employed.

development of the Reproducing Kernel Particle Method (RKPM) by (Liu, Jun & Zhang, 1995). In this method, a correction function is introduced into the approximation to satisfy consistency requirements and to improve the behavior of the approximation near boundaries. Both, SPH and RKPM are suitable for problems involving non-continuum media, large domain deformation, tracking of free surfaces and moving discontinuities. Further developments and applications of SPH and RKPM can be found in (Li & Liu, 2002), and a recent review on SPH is presented in (Liu & Liu, 2010). It is important to note that an equivalence between RKPM and MLS can be demonstrated; see (Belytschko, Krongauz, Organ, Fleming & Krysl, 1996). This allows to study and analyze these methods in a unified framework.

#### 2.2.2.4 Related approaches

Other approximation techniques, not included in the description above, are also worth of mention. A good starting point is methods based on partition of unity. These methods use functions that have this property as an extrinsic (independent) basis to build enriched approximations customized for particular purposes. The h-p clouds method (Duarte & Oden, 1996b) is a typical example of these techniques. There, basis functions of varying support size are used to perform *h*-adaptivity, and also the approximation basis can be enriched to perform *p*-adaptivity. These ideas were also applied in conjunction with conventional mesh-based methods but the resulting techniques cannot be regarded as truly meshless methods, at least in their original versions, cf. (Fries & Matthies, 2004). Some of these methods are the Partition of Unity Finite Element Method (PUFEM) (Babuška & Melenk, 1996), the Partition of Unity Method (PUM) (Babuška & Melenk, 1997) and the Generalized Finite Element Method (GFEM) (Strouboulis, Babuška & Copps, 2000). It is important to note that the paradigm of partition of unity provides a unified mathematical framework for the development and analysis of meshless techniques. Many of the existing methods (e.g. MLS) can be considered to be particular cases of partition of unity.

Additional approximation methods combining meshless ideas into conventional mesh-based schemes can be also found in the literature. For instance, the Particle-in-Cell (PIC) method (Sulsky, Zhou & Schreyer, 1995), where a stationary background grid is combined with Lagrangian moving particles and MLS approximations are used to transfer information between them (the equations are solved on the background grid). Along a similar line but from a different point of view, another example is the Free Mesh Method (FMM) (Yagawa & Yamada, 1996), which adopts a local problem discretization based on local Delaunay triangulation around the point of interest. This characteristic gives a higher locality to the computations, which makes the method suitable for implementing parallel solution strategies; see applications in (Yagawa & Furukawa, 2000).

With a focus on solid mechanics problems, we can also found the Natural Element Method (NEM) proposed in (Sukumar, Moran & Belytschko, 1998); cf. (Cueto, Sukumar, Calvo, Martínez, Cegonino & Doblaré, 2003) for an overview. In NEM, the spatial approximation is calculated by natural neighbor interpolation, a procedure based on Sibson functions[17] (Sibson, 1980) or simplified forms that reduce the computational cost (non-Sibsonian functions). The application of essential boundary conditions is straightforward in NEM (the shape functions satisfy the Kronecker's delta function and have partition of unity properties) and the numerical integration can be performed by using the surrounding Delaunay neighbors as integration cells; although this requires corrections due to the geometrical characteristics of the approximation support. To this end, the integration is usually performed by a decomposition of the integration cells into pieces that describe exactly the support of the shape functions (Atluri, Kim & Cho, 1999) or by nodal stabilized quadrature schemes, see (Chen, Wu, Yoon & You, 2001).

Other noteworthy particle methods exploiting meshless features into conventional schemes are the Meshless Finite Element Method (MFEM) (Idelsohn, Oñate, Calvo & Del Pin, 2003) and the Particle Finite Element Method (PFEM) (Idelsohn, Oñate & Del Pin, 2004). Although since their origin these methods have been closely related to the Lagrangian solution of fluid flows, applications to solid mechanics problems have been developed more recently. The basic aspects of MFEM and PFEM will be discussed in connection with flow problems in the next section.

### 2.2.3 Applications in computational fluid dynamics

Like in other areas of engineering and sciences, meshless applications in computational fluid dynamics (CFD) have developed mainly to overcome problems encountered in conventional mesh-based approaches, mostly related to grid generation, but also to simplify the implementation of specific solution procedures. Although the development of meshless methods as alternative (general) discretization techniques originated from the works of Kansa in the 1990s, there were previous developments that have played and still play an important role; for instance the Vortex Particle Method (VPM) and the Smooth Particle Hydrodynamics (SPH).

The VPM adopts a vorticity formulation of the Navier-Stokes equations which is solved by using vortex particles, convected with the flow in a Lagrangian manner; see some early applications in (Chorin, 1973) and (Chorin & Bernard, 1973). This method, still in use today, is regarded as one of the

[17] In Sibsonian approximations, the Voronoi cells surrounding a particle are used to define neighbor coordinates (shape functions) based on area (2D) or volume (3D) ratios which are used as trial and test functions in the Galerkin procedure.

precursors of the well-known panel methods, extensively employed in computational aerodynamics. A recent review of the VPM methodology and its applications can be found in (Barba, Leonard & Allen, 2005).

The other pioneer method was the SPH (Gingold & Monaghan, 1977). Although originally developed to solve astrophysical problems, it was rapidly applied to the solution of fluid flows. In the SPH, the discretization of the governing equations is based on discrete particles (elemental fluid volumes) and the local properties (density, velocity, etc.) are computed using a Lagrange description of the fluid in which the particles interact between them according to a given smoothing function. The pressure is obtained from the density using an equation of state and the acceleration is obtained from the pressure gradient and density; effects of physical viscosity can be also accounted for. The basic SPH technique can present problems with the definition of the smoothing length (that affects considerably the results) and the order of consistency of the approximation, which depends on the particular problem setting and simulation parameters. To address these deficiencies, a correction to the kernel function is proposed in (Liu, Jun & Zhang, 1995), which led to the RKPM. Another more recent solution, which includes some of the previous corrective methods, introduces an additional approximation basis into the kernel interpolation (Liu & Liu, 2006). In spite of these problems, the SPH has undergone a continuous development and evolution, and multiple applications can be found in the literature; see (Liu & Liu, 2010) for a recent review. The method shows great potential to deal with heterogeneous media, large domain deformation, moving or changing boundaries, free surfaces and contact problems. However, it should be noted that the SPH also presents some lack of conservation (typically mass) which leads to accuracy problems in particular applications. As shown in (Niedoba, Čermák & Jícha, 2013) significant improvements can be achieved by applying particular settings of the simulation parameters and specific treatment of boundary conditions, but this causes dependency on the problem and user-specified parameters.

As mentioned before, Kansa's works in the 1990s paved the way for the sustained development of general meshless discretization techniques. According to the results of Franke's paper (Franke, 1982), Kansa proposed to use multiquadrics (MQ) spatial approximations functions in conjunction with a collocation scheme to solve strong form PDEs (Kansa, 1990a, 1990b). The methodology was applied with success to many elliptical, convective-diffusive and inviscid (Euler) flow problems. Since then, different applications in different fields were proposed, e.g. for boundary-value problems (Hon, Lu, Xue & Zhu, 1997), Burger's equation (Hon & Mao, 1998), incompressible viscous flows (Mai-Duy & Tran-Cong, 2001) and (Shu, Ding & Yeo, 2003), shallow water equations (Hon, Cheung, Mao & Kansa, 1999), flow in porous media (Šarler, Perko & Chen, 2004) and phase change problems (Kovačević, Poredoš & Šarler, 2003).

Although in general the results obtained were quite satisfactory, modifications and improvements of the methodology were proposed to fix problems regarding the solvability of the coefficients matrix (Fasshauer, 1997), instabilities of the collocation procedure (see an analysis in (Larsson & Fornberg, 2003)) and dependence of numerical results on the shape parameter of the basis functions. In this regard, theoretical attempts to determine optimal values did not succeed and, although some general guidelines were formulated (Kansa & Carlson, 1992), the proper problem setting remains to be determined largely by numerical experiments. Another problem in classical RBFs, which seriously penalizes practical applications, is related to the fact that a full-matrix, typically ill-conditioned, global system must be solved to obtain the approximation coefficients (the RBFs have infinite support). In order to avoid this complication, alternative approaches have been proposed mainly in the line of domain decomposition techniques (e.g. (Dubal, 1994; Kansa, 1990b), (Kansa & Hon, 2000) and (Li & Hon, 2004)) and local RBF approximations (e.g. (Lee, Liu & Fan, 2003) and (Shu, Ding & Yeo, 2003)). Particularly, the latter approach has been used with considerable success. For example, local RBFs have been used to develop an upwind-type flow solver (using Roe fluxes) with applications to two-dimensional steady and unsteady compressible flows (Shu, Ding, Chen & Wang, 2005). The results were compared with numerical computations obtained by a conventional Finite Volume method (FVM) and a LSQ-based meshless technique[18]. The conclusion was that the local RBF method is slightly more accurate than the FVM (using a comparable flow solver), and also than LSQ, if a regular arrangement of points is used. However, the LSQ-based technique showed better results in non-regular distribution of points.

Meshless methods using MLS-based spatial approximation (or other WLSQ approaches) in conjunction with the strong conservative form of the equations have been the most typical choices for fluid flow computations. These methods have their origins in the Generalized Finite Difference (GFDM) schemes developed in (Perrone & Kao, 1975) and (Liszka & Orkisz, 1980) with the purpose to extend Finite Differences techniques (FDM) to arbitrary irregular grids[19]. These ideas, and the need to relief the dependence on meshes for three-dimensional complex applications, led to propose a *gridless*[20] scheme based on LSQ polynomial approximation using linear basis functions (Batina, 1992). The proposed solution approach followed the typical compressible FD or FV flow solvers using central-type discretizations plus artificial dissipation terms and explicit time integration. The results presented,

[18] The method presented in (Sridar & Balakrishnan, 2003) was employed for the comparison.

[19] It is possible to demonstrate equivalence between GFDM and MLS-based approximations; see for instance (Duarte, 1995).

[20] As named by the author in his original paper.

for two-dimensional inviscid and laminar (low-Reynolds) viscous flows, showed satisfactory accuracy and also the feasibility of the approach.

Some years later, also in the line of Batina's work, an outstanding investigation on MLS and general WLSQ-based schemes was conducted in (Oñate, Idelsohn & Zienkiewicz, 1995) and (Taylor, Zienkiewicz, Oñate & Idelsohn, 1995). This led to the development of the Finite Point Method (FPM). Basically, this technique is characterized by WLSQ approximations on clouds of points and a discretization procedure based on point collocation[21]. The first applications of the FPM mainly focused on compressible flow problems, see for instance (Fischer, Onate & Idelsohn, 1995; Oñate, Idelsohn & Zienkiewicz, 1995; Oñate, Idelsohn, Zienkiewicz & Fisher, 1995). In addition, systematic studies intended to determine the effects on the approximation of the selection of points and weighting functions were also carried out using linear and quadratic polynomial bases (Fischer, 1996). Then, the properties of the approximation were studied in more detail in the context of convective-diffusive and incompressible flow problems, which gave the FPM a more solid base; cf. (Oñate, Idelsohn, Zienkiewicz & Taylor, 1996a) and (Oñate, Idelsohn, Zienkiewicz, Taylor & Sacco, 1996b). These works, along with (Oñate & Idelsohn, 1998), defined the main aspects of the FPM technique in use today. These are discussed in detail in the next section.

In the context of the gas kinetic theory, Deshpande and Gosh proposed the use of LSQ-based approximations, using a Taylor's series expansion basis, and a flux splitting scheme. The approach led to the development of the Least Squares Kinetic Upwind Method (LSKUM) (Ghosh & Deshpande, 1995). The basic aspects of the methodology and typical applications are presented in (Deshpande, Kulkarni & Ghosh, 1998) and (Deshpande, Anandhanarayanan, Praveen & Ramesh, 2002). Some practical applications to aircraft store separation and aerodynamic optimization can be found in (Harish & Pavanakumar, 2006) and (Srinarayana, Gonzalez, Whitney & Srinivas, 2006), respectively. In addition, developments intended to improve the efficiency of the method are presented in (Anandhanarayanan & Nagarathinam, 2005), where a parallel implementation is discussed.

Another approach in the line of research suggested by Batina was proposed in (Sridar & Balakrishnan, 2003). On the basis of a LSQ Taylor-based approximation and an upwind FD-type scheme, these authors proposed the Least-Squares Upwind Finite Difference Method (LSFD-U). In this work successful two-dimensional applications to inviscid flow problems are presented; the accuracy and convergence properties of the method are studied and conservation is assessed empirically. Applications to viscous flow

[21] Other finite point approaches following these lines can be found in the literature, see for instance (Liu & Su, 1996).

problems are presented in a more recent publication (Munikrishna & Balakrishnan, 2011) using Cartesian point distributions.

Following the technique developed in the LSFD-U method, Praveen and Deshpande proposed some modifications to the LSKUM, mainly intended to improve its accuracy and robustness. The new method was called Kinetic Meshless Method (KMM), see for instance (Praveen & Deshpande, 2003) and (Praveen & Deshpande, 2007). The idea was to use a modified LSQ scheme in which mid-point fictitious nodes are used to introduce upwind convective stabilization, thus avoiding the flux splitting employed before in the LSKUM. Further research on the properties of the resulting schemes has been conducted in (Praveen, 2004), where certain modifications are proposed to satisfy positivity conditions.

More recently, Katz and Jameson (see for instance (Katz, 2009)) developed a so-called volume approach to compute the meshless LSQ approximation coefficients using Taylor expansions. This procedure has some advantages regarding the reciprocity of points between adjacent approximation supports (derived from an edge-based data structure) and the memory storage requirements (adjacent nodes share the same metric coefficients). The flow solver is based on an upwind-type discretization scheme which can be adapted to different meshless discretizations and convective stabilization methods. Successful applications involving compressible inviscid and viscous (low-Reynolds number) flows are presented. Moreover, a convergence acceleration technique named *multicloud* and a meshless methodology for overset grids were proposed with the aim to exploit meshless capabilities. In another interesting work, the same authors explore the differences between RBFs, Taylor-based LSQ and polynomial-based WLSQ discretization approaches, using a similar flow solution scheme and comparing the results with an established FV method (Katz & Jameson, 2009a),. The numerical tests performed showed that the three methodologies have a comparable (and satisfactory) accuracy level for smooth flow problems. However, in flows involving shock discontinuities, the LSQ-based approaches work considerably better and, among them, the polynomial approximation improves over Taylor expansions (although the latter is computationally cheaper). The authors also noted some lack of accuracy to resolve the shock position with RBFs, although the satisfaction of Rankine-Hugoniot conditions was verified (at least at the same level than the other methods). This led to a considerable error in the aerodynamic forces computed with RBFs.

It is important to note here that none of the typical meshless schemes can be shown to be formally conservative. Except for some simple especial cases, the characteristics of the meshless coefficients make the total flux around an interior point generally be different from zero and, thus, the schemes do not satisfy telescopic flux collapsing. In spite of the fact that the lack of formal conservation is an important theoretical drawback in meshless methods, their practical applications do not seem to be negatively affected and the results are still accurate enough. In view of the relevance of this matter, further investigations are being carried out. For example, an alternative approach to enforce conservation by construction of the meshless coefficients was recently proposed; see (Chiu, Wang & Jameson, 2011).

The employment of weak forms equations in computational fluid dynamics has been somewhat relegated with respect to strong forms, especially in meshless applications, where additional difficulties arise due to the characteristics of the spatial support. Methods based on weak forms relay on a sound mathematical basis and have advantages regarding the order of consistency of the approximation and the stability of the resulting scheme. These facts make them very attractive; however, there are also some disadvantages mainly related to the need of numerical integration (less efficient and more complicated in a meshless context) and the application of essential boundary conditions[22]. In spite of this, methods such as the EFG, and particularly the MLPG, have been also successfully employed to solve fluid flow problems; see for instance (Lin & Atluri, 2001) and more recently (Wu, Tao, Shen & Zhu, 2010). In these works, a MLS-based approximation is used with local residual equations, which are integrated on simple local domains, thus avoiding the global background grid and giving flexibility to the meshless technique. The applications of this method are mainly in the field of incompressible viscous flow problems.

Also in the line of weak formulations, it is worth mentioning other approaches in the literature that combine meshless features into conventional mesh-based schemes (or viceversa). As mentioned before, within Lagrangian particle methods, of particular interest due to its flexibility is the Meshless Finite Element Method (MFEM) (Idelsohn, Oñate, Calvo & Del Pin, 2003) and its successor, the Particle Finite Element Method (PFEM) (Idelsohn, Oñate & Del Pin, 2004). These methods simulate a set of particles (elemental fluid volumes) flowing under forces acting on it. Since the problem data structure depends on the position of the particles and, therefore, on the problem solution, it must be regenerated at each step along a time marching procedure. This is done in an efficient manner by using an extended Delaunay tessellation (Idelsohn, Calvo & Oñate, 2003) to build a global mesh connecting the particles. Using this partition, the spatial approximation in MFEM is constructed with non-Sibsonian functions in polygonal or polyhedral elements, while standard FEM functions in triangular or tetrahedral elements are adopted in PFEM. The boundaries of the analysis domain are also efficiently recovered during the simulation by using an alpha-shape technique (cf. (Akkiraju, Edelsbrunner, Facello, Fu, Mücke & Varela,

[22] To overcome this drawback, alternative approaches based on Lagrange multipliers, penalty formulations or coupling with finite elements in a boundary strip have been proposed, see for instance (Fries & Matthies, 2004).

1995)). The problem equations in both, MFEM and PFEM, are discretized and solved using a conventional FEM approach. These characteristics provide a great flexibility to the methods to deal with moving or changing domain problems, while inheriting beneficial characteristics of mesh-based schemes. The applications of PFEM mainly focus on incompressible free-surface fluid flows, fluid-structure interaction, multiphase problems and also solid mechanics, see examples in (Oñate, Idelsohn, Del Pin & Aubry, 2004), (Idelsohn, Oñate, Del Pin & Calvo, 2006), (Idelsohn, Marti, Limache & Oñate, 2008), (Larese, Rossi, Oñate & Idelsohn, 2008), (Oñate, Idelsohn, Celigueta & Rossi, 2008) and (Idelsohn, Mier-Torrecilla & Oñate, 2009).

In spite of the fact MFEM and PFEM cannot be regarded as meshless methods according to the criteria adopted in Section 2.2.1, additional considerations related to the characteristics of the discretization clearly differentiate these methodologies from conventional mesh-based approaches. For example, the domain partition is not unique, it changes in time during the computation and can be generated in bounded time[23] (the cost is linear with the number of particles). Moreover, there is certain flexibility in the construction of the numerical approximation as the method is not restricted to a specified element shape.

## 2.3 The Finite Point Method

The basic aspects of the FPM are discussed first in this section. Then, the evolution of the method in the field of fluid flow problems is described with some detail in order to frame the research conducted in this work.

### 2.3.1 Characteristics of the FPM approximation

It can be said that the basic aspects shaping the present FPM were established after some previous works in (Oñate & Idelsohn, 1998). In this paper, the authors adopt a particular case of WLSQ approximation named Fixed Least Squares (FLS). This approach uses a compact support weighting function (Gaussian type), fixed in the local approximation domain (local cloud), which reaches its maximum value (unity) at the point where the approximation is computed (star point) [24,25]. This weighting function does not change in the local cloud and the minimization procedure yields constant approximation

[23] Note that this is not always possible in conventional mesh generation, since an unknown number of iterations and/or manual work may be needed to meet minimal quality requirements.

[24] This point always belongs to the local approximation domain.

[25] The nomenclature used follows the concepts introduced in Generalized Finite-Differences schemes.

coefficients, which simplifies considerably the computation of derivatives and reduces the computational cost[26]. As regards the problem discretization, it is performed in the FPM by point collocation on the strong form of the equations. This procedure, which is the most typical in meshless methods, is very efficient and easy to implement.

The FLS approach used in the FPM has also some disadvantages. On the one hand, the approximation is only accurate at the star point of the cloud, where the weighting function is located (the error increases with distance). On the other hand, the approximation is multivalued since different overlapping clouds yield different values for the weighting function at a same point. Nevertheless, these disadvantages do not impose any serious restriction on the FPM because of the discretization procedure applied. Note that since the spatial approximation is only necessary at the star point of the cloud in the collocation procedure, these problems are easily resolved (although the lack of continuity of the approximation has also some additional consequences which will be discussed later in this work).

### 2.3.2 Applications in fluid flow problems

The FPM technique has been successfully applied in numerous fluid flow problems, where many capabilities of the method have been discovered, but also some limitations and opportunities for further improvements. The most relevant works are reviewed next.

Among the first publications in the FPM, the early lines of research and applications are summarized in (Fischer, 1996). In that work, convective-diffusive problems were studied using LSQ and WLSQ polynomial approximations with linear and quadratic bases. The study focused on the effects of the number of points and weighting functions on the local approximation, which helped to understand the basic behavior of the FPM. The accuracy of the approximation was also investigated using one-dimensional clouds of points. The results showed that the FPM approximation leads to discrete derivative forms similar to those obtained with central difference approximations, which are second-order accurate. However, the accuracy degrades to first-order for non-symmetric clouds, depending on the characteristics of the weighting function employed[27]. Some preliminary criteria about the selection of points conforming the local clouds were also defined with the aim to improve the ill-conditioning of the minimization

[26] The discrete derivatives can be are comparable to those obtained by diffuse approximations in MLS (Nayroles, Touzot & Villon, 1992).

[27] As will be discussed further through the present work, the dependence of the spatial approximation on these parameters difficulties to perform reliable analysis intended to obtain theoretical bounds for the main characteristics of discrete approximation.

problem. The flow solver employed in that work was based on a two-step Taylor-Galerkin scheme with explicit artificial dissipation. The numerical examples involved inviscid subsonic, transonic and supersonic two-dimensional problems, but a viscous low-Reynolds number test case was also provided. In general, the results obtained in this work were satisfactory and demonstrated that the introduction of weighting in the LSQ minimization leads to superior results (linear basis were used). However, in some problems involving transonic flows, a low convergence rate and inaccurate shock positions were also observed. This behavior was related to possible effects of the spatial distribution of the points and the non-conservative character of the method. It should be noticed that in these examples small local cloud of points were used and, probably, this was not sufficient to ensure the necessary overlap between approximation subdomains to propagate the information through the domain correctly.

In a similar line of research, a residual stabilization technique derived in terms of flux balancing in a finite domain, known as Finite Increment Calculus (FIC) (Oñate, 1996, 1998), was introduced. The results were satisfactory and comparable to the ones obtained previously with explicit artificial dissipation terms, but with the advantage that the stabilization in FIC is introduced in a consistent manner, see applications in (Oñate, Idelsohn, Zienkiewicz, Taylor & Sacco, 1996b) and (Oñate & Idelsohn, 1998).

Among these developments, the issue of point generation, which is of capital importance for meshless methods, was addressed in (Löhner & Oñate, 1998). Based on an advancing front technique, these authors showed that point discretizations suitable for meshless computations can be generated more efficiently by avoiding the usual quality checks needed in conventional mesh generation. In this way, highly competitive generation times were achieved in comparison with traditional meshers. This showed, for the first time, that meshless methods are a feasible alternative to alleviate discretization-related problems in conventional solution approaches.

Incompressible two-dimensional fluid flow problems were first studied in (Oñate, Sacco & Idelsohn, 2000). The proposed solution scheme was a semi-implicit pressure splitting algorithm (known as fractional step) stabilized through the FIC technique. A detailed analysis of this solution scheme was carried out in (Sacco, 2002). Outstanding achievements from that work have given the FPM a more solid base; among them, the definition of local and normalized approximation bases, a procedure for constructing local clouds of points based on local Delaunay triangulation, and a criterion for evaluating the quality of the resultant approximation. The numerical applications presented focused mainly on two-dimensional (viscous and inviscid) incompressible flows, but a three-dimensional application example was also provided.

A preliminary application of the FPM in a Lagrangian framework, presented in (Idelsohn, Storti & Oñate, 2001), is also worth of mention. Despite the interesting results obtained for incompressible free-surface flows, this line of research was not continued under the FPM and later formulations were exclusively based on Eulerian flow descriptions.

The first application of the FPM to the solution of three-dimensional compressible flow problems was presented in a pioneer work by (Löhner, Sacco, Oñate & Idelsohn, 2002). Here, we can highlight two remarkable contributions: a reliable and general procedure for constructing local clouds of points (based on a Delaunay technique) and a well-suited scheme for solving the flow equations. In this scheme, the discrete flux derivatives are written along edges connecting the cloud's points as a central difference-like expression plus an upwind bias term which provides convective stabilization. An approximate Riemann solver of Roe and van Leer flux vector splitting are used for this purpose. This approach is more accurate (also more expensive) than artificial dissipation methods and, additionally, does not require the definition of geometrical measures in the local cloud and problem dependent parameters. The time integration of the equations is performed through a multi-stage explicit scheme in the line of Runge-Kutta methods. The numerical applications presented focus on three-dimensional inviscid flow problems, one of them a practical computation test case involving a half-span F117 aircraft. It is important to note that, contrary to the tendency observed in (Fischer, 1996), no accuracy and convergence problems were observed in this work. Though not conservative, the proposed methodology allowed capturing the flow discontinuities correctly. In this regard, similar results were also observed in other works; for example in (Ortega & Sacco, 2003), where satisfactory results for steady and unsteady transonic flow problems were achieved using a time-marching solution scheme with explicit artificial dissipation (comparable to that used by Fischer). These results suggest that the problems observed by Fischer may be presumably more related to the selection of points conforming the local clouds rather than the non-conservative character of the discrete meshless scheme.

Following similar approaches in terms of spatial discretization and solution schemes, many other FPM techniques intended to specific purposes can be found in the literature. Many of them are discussed later through this work in the context of particular applications.

# 3 THE FINITE POINT METHOD

The Finite Point Method (FPM) is a meshless discretization technique intended to solve partial differential equations (PDEs) on scattered distributions of points (cf. Section 2.3 for an overview). In brief, the approximation process in the FPM can be summarized as follows. For each point in the analysis domain Ω (star point), an approximated solution is locally constructed by working on a subset of surrounding supporting points, which also belong to the problem domain (local cloud of points). The approximation is computed as a linear combination of the cloud unknown nodal values (or parameters) and certain metric coefficients. These are obtained by solving a Weighted Least-Squares (WLSQ) problem at the cloud level, in which the distances between the nodal parameters and the approximated solution are minimized in a least-squares sense. Once the approximation metric coefficients are known, the problem governing PDEs are sampled at each star point by using point collocation. The continuous variables (and their derivatives) are replaced in the sampled equations by the discrete approximated forms, and the solution of the resulting system allows calculating the unknown nodal values. Hence, the approximated solution satisfying the governing equations of the problem can be obtained.

The various aspects of the FPM technique outlined above are discussed in detail in this chapter. In Section 3.1 the construction of the local approximation and the characteristics of the basis and weighting functions are described. The equations discretization procedure is presented in Section 3.2, and Section 3.3 analyzes the sensitivity of the FPM approximation to adjustable parameters governing the functional shape of the weighting functions and features of the local clouds. This analysis aims to determine suitable ranges of the parameters and also the capability of the latter to improve the local approximation using automatic adjustment procedures. Next, the accuracy of the FPM approximation is investigated numerically in Section 3.4, in order to obtain accuracy measures attainable in practical test cases. In Section 3.5, an automatic procedure to adjust the relevant approximation parameters is proposed with the aim to simplify the problem settings as well as to increase robustness and accuracy. Several issues arising in practice in relation to the global and local FPM discretization are discussed in Section 3.6 and, finally, implementation aspects concerning the organization of the FPM data structure used in this work are discussed in Section 3.7 to complement and round off the description of the method.

## 3.1 Construction of the local approximation

Let us assume that φ($\boldsymbol{x}$) is an unknown field function defined in an analysis domain $\Omega \in \Re^{1,2,3}$ which is discretized by a set of $n$ scattered points. Let $\boldsymbol{x}_i$ be a point of interest in which the function and their derivatives are to be evaluated (star point) and $\boldsymbol{x}_j$ a collection of surrounding points providing an adequate support for the local approximation. Assume further that all these points belong to the domain Ω and conform a local cloud $\Omega_i$ (Figure 1), in such a way that $\Sigma\Omega_i$ constitutes a covering of the problem domain.

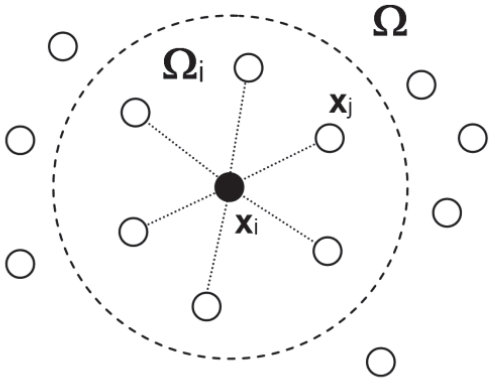

Figure 1. Global and local approximation domains in the FPM.

Then, an approximation of φ($\boldsymbol{x}$) in $\Omega_i$ can be expressed by

$$\varphi(\boldsymbol{x}) \approx \hat{\varphi}(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\alpha} \tag{3.1}$$

where $\boldsymbol{p}(\boldsymbol{x})$ ($m$ x 1) is an approximation basis vector (typically polynomial) and $\boldsymbol{\alpha}$ ($m$ x 1) is a vector of unknown approximation coefficients. Sampling φ($\boldsymbol{x}$) at each point within $\Omega_i$ gives the following system of equations

$$\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_{np} \end{bmatrix} \cong \begin{bmatrix} \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_{np} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_{np}^T \end{bmatrix} \boldsymbol{\alpha} = \mathbf{P}\boldsymbol{\alpha} \tag{3.2}$$

where $\varphi_j = \varphi(\boldsymbol{x}_j)$ are the values of the unknown function at each point in $\Omega_i$ (nodal parameters), the tilde ( $\hat{}$ ) indicates approximated values at the same points and $\boldsymbol{p}_j^{\mathrm{T}} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)$. Then, the approximation coefficients can be obtained from Eq. (3.2) by solving $\boldsymbol{\alpha}=\mathbf{P}^{-1}\boldsymbol{\varphi}$, provided that $np=m$ and matrix **P** is non-singular[28] (inversion is required because $\boldsymbol{\varphi}$ is not known in advance). Replacing the coefficients $\boldsymbol{\alpha}$ into Eq. (3.1), the approximation for the star point (and any point $\boldsymbol{x}$ in $\Omega_i$) can be obtained by

$$\hat{\varphi}(\boldsymbol{x}_i) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i)\,\boldsymbol{P}^{-1}\boldsymbol{\varphi} \tag{3.3}$$

In spite of the fact that Eq. (3.3) interpolates the function $\varphi(\boldsymbol{x})$ at the sampling points, and this gives interesting properties to the approximation function, the condition $np=m$ significantly penalizes the meshless methodology (the number of points in the local clouds cannot be chosen arbitrarily). As matrix **P** is no longer squared if $np>m$, an alternative solution approach is required.

In order to overcome this restriction, an approximate solution can be obtained by Least-Squares (LSQ) techniques. The idea is to compute a set of approximation coefficients $\boldsymbol{\alpha}$ that minimize the sum of the squared approximation error to $\varphi(\boldsymbol{x})$ in $\Omega_i$. In addition, a weighting function $w(\boldsymbol{x})$ can be introduced in the minimization problem to enforce data locality[29] around the point of interest were the approximation is sought (Weighted Least-Squares). Following this procedure, a discrete functional can be defined as

$$J = \sum_{j=1}^{np} w_i(\boldsymbol{x}_j)\left[\hat{\varphi}_j - \varphi_j\right]^2 = \sum_{j=1}^{np} w_i(\boldsymbol{x}_j)\left[\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)\boldsymbol{\alpha} - \varphi_j\right]^2 \tag{3.4}$$

in which $w_i(\boldsymbol{x}_j) = w(\|\boldsymbol{x}_j\text{-}\boldsymbol{x}_i\|)$ is a compact support weighting function fixed at the star point of the cloud. This procedure, known as Fixed Least-Squares[30] (Oñate & Idelsohn, 1998), can be considered as a particular case of the Moving Least-Squares method introduced in (Lancaster & Salkauskas, 1981). The weighting function used in Eq. (3.4) typically takes the value of unit nearby the star point and decreases with distance, vanishing at the cloud boundaries. As discussed later in this chapter, the functional form of the weighting function can be defined with great flexibility, but it usually has an impact on the resultant approximation.

[28] The solvability of the inverse matrix does not depend so much on the basis functions chosen, but mainly on the characteristics of the distribution of points in the local cloud; see for instance (Duarte, 1995).

[29] Standard (unweighted) LSQ-based approximations usually present a considerably lack of accuracy when the number of points is large. Comparisons of LSQ and WLSQ-based meshless approximations can be found in (Fischer, 1996).

[30] Different WLSQ-based approaches having particular characteristics can be obtained according to the manner in which the weighting function is applied in the minimization process, see for instance (Oñate, Idelsohn, Zienkiewicz, Taylor & Sacco, 1996b) for an overview of the main possibilities.

Going back to the WLSQ problem, the cloud approximation coefficients can be obtained by minimizing the functional (3.4) with respect to $\boldsymbol{\alpha}$. This leads to the following set of equations

$$\begin{aligned}
\frac{\partial J}{\partial \alpha_1} &= 2\sum_{j=1}^{np} w_i(\boldsymbol{x}_j) p_1(\boldsymbol{x}_j)\left[\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)\boldsymbol{\alpha} - \varphi_j\right] = 0 \\
\frac{\partial J}{\partial \alpha_2} &= 2\sum_{j=1}^{np} w_i(\boldsymbol{x}_j) p_2(\boldsymbol{x}_j)\left[\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)\boldsymbol{\alpha} - \varphi_j\right] = 0 \\
&\vdots \\
\frac{\partial J}{\partial \alpha_m} &= 2\sum_{j=1}^{np} w_i(\boldsymbol{x}_j) p_m(\boldsymbol{x}_j)\left[\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)\boldsymbol{\alpha} - \varphi_j\right] = 0
\end{aligned} \tag{3.5}$$

which are known as *normal equations* in the LSQ literature. The system (3.5) can be rewritten in vector form as

$$\sum_{j=1}^{np} w_i(\boldsymbol{x}_j)\boldsymbol{p}(\boldsymbol{x}_j)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)\boldsymbol{\alpha} = \sum_{j=1}^{np} w_i(\boldsymbol{x}_j)\boldsymbol{p}(\boldsymbol{x}_j)\varphi_j \tag{3.6}$$

and introducing matrices

$$\mathbf{A} = \sum_{j=1}^{np} w_i(\boldsymbol{x}_j)\boldsymbol{p}(\boldsymbol{x}_j)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j) \quad (m \,\mathrm{x}\, m) \tag{3.7}$$

$$\mathbf{B} = \left[w_i(\boldsymbol{x}_1)\boldsymbol{p}(\boldsymbol{x}_1), w_i(\boldsymbol{x}_2)\boldsymbol{p}(\boldsymbol{x}_2), \ldots, w_i(\boldsymbol{x}_{np})\boldsymbol{p}(\boldsymbol{x}_{np})\right] \quad (m \,\mathrm{x}\, np) \tag{3.8}$$

Eqs. (3.6) can be recasted in

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{B}\boldsymbol{\varphi} \tag{3.9}$$

from which the approximation coefficients can be calculated

$$\boldsymbol{\alpha} = \mathbf{A}^{-1}\mathbf{B}\boldsymbol{\varphi} \tag{3.10}$$

The solution of Eq. (3.10) requires matrix **A** (usually known as *moment matrix*) to be positive-definite and, consequently, non-singular. Similar to matrix **P** in Eq. (3.3), matrix **A** can become very ill-conditioned and difficult to invert with accuracy depending on the basis and weighting functions and the geometry of the local cloud (especially in multidimensional problems). Hence, these variables in the approximation should be subject to particular conditions to keep the approximation error under control[31]. This will be discussed later in Sections 3.3 and 3.6.

[31] A discussion on the relations between the point distribution, the conditioning of the resultant minimization problem and error estimates in MLS-based approximations can

For the time being, supposing that the approximation coefficients can be calculated from Eq. (3.10), the numerical approximation at the star point of the cloud is obtained by introducing $\boldsymbol{\alpha}$ into Eq. (3.1) and sampling at the same point. This leads to

$$\hat{\varphi}_i = \hat{\varphi}(\boldsymbol{x}_i) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i)\,\mathbf{A}^{-1}\mathbf{B}\,\boldsymbol{\varphi} = a_{\mathrm{ij}}\varphi_{\mathrm{j}} \tag{3.11}$$

where summation over repeated indices is assumed. It is important to note that matrices **A** and **B** and, consequently, the approximation coefficients $\boldsymbol{\alpha}$, are constant in $\Omega_{\mathrm{i}}$ because of the fixed weighting function chosen.

Note that according to the LSQ character of the approximation, $\hat{\varphi}(\boldsymbol{x})$ does not interpolate the nodal unknown parameters $\varphi(\boldsymbol{x})$, i.e. the FPM approximation does not satisfy the Kronecker's delta property. In this regard, it should be noted that $\hat{\varphi}(\boldsymbol{x})$ is the true approximation for which the governing equations and boundary conditions will be enforced and $\varphi(\boldsymbol{x})$ are simply internal nodal values employed to construct such numerical approximation.

The derivatives of the approximation with respect to the spatial coordinates $x_{\mathrm{k}}$ are simply obtained by differentiating the approximation basis, because matrices **A** and **B** are constant in $\Omega_{\mathrm{i}}$. Hence, the first derivatives results

$$\frac{\partial \hat{\varphi}_i}{\partial \boldsymbol{x}_k} = \frac{\partial \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_i)}{\partial \boldsymbol{x}_k}\mathbf{A}^{-1}\mathbf{B}\,\boldsymbol{\varphi} = b_{\mathrm{ij}}\varphi_{\mathrm{j}} \tag{3.12}$$

and higher-order derivatives can be obtained similarly, up to the order of differentiability of the approximation basis chosen.

The adoption of a fixed weighting function in the local cloud notably simplifies the computation of the derivatives and reduces the computational cost. However, the accuracy of the approximation can degrade considerably far from the point where the weighting function is located and, therefore, the validity of the local approximation is generally restricted to the star point of the cloud (note that collocation techniques are appropriate for that purpose). Moreover, the fact that the same point can belong to different overlapped clouds and, thus, it is affected by different weights, makes the approximation coefficients to be multivalued, i.e. $a_{\mathrm{ij}} \neq a_{\mathrm{ji}}$ and $b_{\mathrm{ij}} \neq b_{\mathrm{ji}}$ (continuity of the global approximation cannot be achieved). This entails some issues regarding implementation aspects (e.g. data structures, storage requirements and computational efficiency) and the properties of the resulting discrete schemes (typically positivity and conservation). These are discussed further in the context of the flow solution scheme (mainly in Chapters 4 and 5).

be found in (Zuppa, 2003). The same results can be applied to other WLSQ techniques under particular situations.

Relevant aspects in the FPM technique, such as the approximation basis, the weighting functions and important characteristics of the FPM approximation are addressed in the next sections.

### 3.1.1 The basis of approximation

Although any kind of basis functions can be used to construct the numerical approximation, the typical choice in the FPM is to adopt complete polynomial bases up to a certain order in $\Re^d$. These basis are simple to compute (not expensive) and flexible enough to adapt to different problem solutions. As mentioned before, it is of crucial importance for the accurate solution of the minimization problem that the moment matrix **A** can be inverted with accuracy. In this respect, the approximation basis plays a role.

The sensitivity of the conditioning of the moment matrix to certain characteristics of the approximation basis was analyzed in (Sacco, 2002). There, the use of dimensionless local bases is proposed to alleviate the ill-conditioning of matrix **A** caused by large and dissimilar entries in the **p** matrices. As shown in (Günther, 1998), this procedure does not affect the resulting approximation function and improves the accuracy and robustness of the solution. Following a similar line, in this work the following complete polynomial bases are employed:

2nd-order approximation bases (2D and 3D cases)

$$\begin{aligned} \boldsymbol{p}^T(\boldsymbol{x}) &= \left[1\; x_1\; x_2\; x_1^2\; x_1x_2\; x_2^2\right] & m &= 6 \\ \boldsymbol{p}^T(\boldsymbol{x}) &= \left[1\; x_1\; x_2\; x_3\; x_1^2\; x_1x_2\; x_1x_3\; x_2^2\; x_2x_3\; x_3^2\right] & m &= 10 \end{aligned} \tag{3.13}$$

3rd-order approximation base, 3D case (m=20)

$$\begin{aligned} \boldsymbol{p}^T(\boldsymbol{x}) = \Big[ & 1\; x_1\; x_2\; x_3\; x_1^2\; x_1x_2\; x_1x_3\; x_2^2\; x_2x_3\; x_3^2 \\ & x_1^3\; x_1^2x_2\; x_1^2x_3\; x_1x_2^2\; x_1y_2x_3\; x_1x_3^2\; x_2^3\; x_2^2x_3\; x_2x_3^2\; x_3^3 \Big] \end{aligned} \tag{3.14}$$

4th-order approximation base, 3D case (m=35)

$$\begin{aligned} \boldsymbol{p}^T(\boldsymbol{x}) = \Big[ & 1\; x_1\; x_2\; x_3\; x_1^2\; x_1x_2\; x_1x_3\; x_2^2\; x_2x_3\; x_3^2 \\ & x_1^3\; x_1^2x_2\; x_1^2x_3\; x_1x_2^2\; x_1y_2x_3\; x_1x_3^2\; x_2^3\; x_2^2x_3\; x_2x_3^2\; x_3^3 \\ & x_1^4\; x_1^3x_2\; x_1^3x_3\; x_1^2x_2^2\; x_1^2x_2x_3\; x_1^2x_3^2\; x_1x_2^3\; x_1x_2^2x_3 \\ & x_1x_2x_3^2\; x_1x_3^3\; x_2^4\; x_2^3x_3\; x_2^2x_3^2\; x_2x_3^3\; x_3^4 \Big] \end{aligned} \tag{3.15}$$

being the local scaled coordinates of a point $\boldsymbol{x} = (x_1\; x_2\; x_3)$ defined as

$$x_k = \frac{\tilde{x}_k - \tilde{x}_k}{d_i} \quad k = 1,3 \tag{3.16}$$

where the tilde denotes the spatial point coordinates[32] and $d_i$ is a characteristic distance calculated at each cloud by

$$d_i = \max\left(\left\|\tilde{\boldsymbol{x}}_j - \tilde{\boldsymbol{x}}_i\right\|\right) \qquad \forall\, \tilde{\boldsymbol{x}}_j \in \Omega_i \tag{3.17}$$

Alternatively, the characteristic distance (3.17) can also be defined for each spatial coordinate. This normalizes the cloud support and alleviates the ill-conditioning of the moment matrix if the anisotropy of the cloud is elevated[33].

The introduction of a local scaled basis simplifies the computation of the approximation function and their derivatives at the star point $\boldsymbol{x}_i$ as it is located at the origin of the local system ($\boldsymbol{x}_i$=(0,0,0)). Consequently, the approximation basis and its first derivatives evaluated at $\boldsymbol{x}_i$ results

$$\begin{aligned} &\boldsymbol{p}^T(\boldsymbol{x}_i) = \left[1\,0\,0\,...\,0\right] \\ &\frac{\partial \boldsymbol{p}^T(\boldsymbol{x}_i)}{\partial x_1} = \left[0\; d_i^{-1}\;...\,0\right] \quad , \quad ... \end{aligned} \tag{3.18}$$

and a similar procedure can be applied for the derivatives in the other spatial directions $x_{2,3}$. In this way, it is possible to write the approximation metric coefficients at the star point as

$$a_{\mathrm{ij}} = \mathbf{C}_{1,\mathrm{j}} \quad , \quad b_{\mathrm{ij}}^k = \frac{1}{d_i}\mathbf{C}_{(k+1),\mathrm{j}} \tag{3.19}$$

where $\mathbf{C} = \mathbf{A}^{-1}\mathbf{B}$ is a ($m$x$np$) matrix and index j = 1,np. The approximation coefficients for higher-order derivatives can be obtained analogously.

### 3.1.2 The weighting function

The introduction of a compact support weighting function into the minimization problem allows to focus on the information in the close neighborhood of the star point and, consequently, to enhance the local character of the approximation. There exist many possibilities for choosing the functional form of a weighting function satisfying the conditions given

[32] Note that dimensionless coordinates are used to construct the local bases. The non-dimensionalization procedure is explained later in Section 4.1.2.

[33] This situation is typically found in the discretization of viscous layers in high-Reynolds number problems. The construction of robust approximations in such clouds of points is addressed in Chapter 8.

before in Section 3.1. In the FPM, the following normalized Gaussian weighting function is typically adopted

$$w_i(\boldsymbol{x}_j) = \frac{e^{-\left(\frac{d_j}{\alpha}\right)^k} - e^{-\left(\frac{\beta}{\alpha}\right)^k}}{1 - e^{-\left(\frac{\beta}{\alpha}\right)^k}} \tag{3.20}$$

where $d_\mathrm{j} = \|\boldsymbol{x}_j\text{-}\boldsymbol{x}_i\|$, $\alpha = \beta/w$ and $\beta = \gamma\, d_i$ ($\gamma>1.0$). This function takes unit value at the star point and decreases monotonically with the distance, approaching zero at the cloud boundary points. The support of this function is isotropic and the parameters *w, k* and $\gamma$ determine its functional shape. These are free parameters that should be properly adjusted because they have a major impact on the approximation. This is studied in Section 3.3.

### 3.1.3 Consistency of the approximation

In the context of PDEs, consistency refers to the ability of a discretized form to approach the continuous equations of the problem when h→0 and Δt→0. Along with stability, consistency is a requisite for the solution to be convergent (Lax-Richtmyer theorem for linear equations); therefore, the discrete approximation must satisfy certain consistency requisites.

In the meshless context, it is generally assumed that the degree of consistency of the approximation is the order of the polynomial which can be represented in an exact manner. In this respect, consistency is closely related to completeness and reproduction conditions[34]. However, these relations are not always observed in practice, where different issues belonging to the approximation procedure and particular problem settings can affect the order of consistency of the method. Despite this, it is important to show that at least the FPM approximation (3.11) can reproduce the functions included in the approximation basis, i.e.

$$\hat{\boldsymbol{\varphi}}(\boldsymbol{x}) = a_{ij}\mathbf{P} = (\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\,\mathbf{A}^{-1}\mathbf{B})\,\mathbf{P} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \tag{3.21}$$

[34] Following (Belytschko, Krongauz, Organ, Fleming & Krysl, 1996), completeness means that the basis chosen can approximate the function with arbitrary order of accuracy. If linear polynomials and their derivatives can be approximated in an exact manner, any smooth function can be approximated with arbitrary accuracy as the grid is refined; thus, linear completeness is equivalent to linear consistency. On the other hand, reproduction conditions are achieved when the approximation can reproduce the functions included in the basis chosen with arbitrary accuracy; if *p*-order polynomials can be reproduced, the approximation has *p*-order consistency. It should be noticed that a difference is made between reproduction conditions and consistency in (Liu & Gu, 2005). The former is assumed to involve any class of basis functions while consistency is only related to polynomials.

with matrix **P** given by Eq. (3.2).

Following (Taylor, Zienkiewicz, Oñate & Idelsohn, 1995), matrices **P** (Eq. (3.2)) and **B** (Eq. (3.8)) can be replaced into Eq. (3.21) yielding

$$\begin{aligned}\hat{\boldsymbol{\varphi}} &= \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\,\mathbf{A}^{-1}\left[w_1\boldsymbol{p}(\boldsymbol{x}_1), w_2\boldsymbol{p}(\boldsymbol{x}_2), \ldots, w_{np}\boldsymbol{p}(\boldsymbol{x}_{np})\right]\begin{bmatrix}\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_1)\\ \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_2)\\ \vdots\\ \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_{np})\end{bmatrix} \\ &= \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\,\mathbf{A}^{-1}\left(\sum_{j=1}^{np} w_i(\boldsymbol{x}_j)\boldsymbol{p}(\boldsymbol{x}_j)\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}_j)\right)\end{aligned} \qquad (3.22)$$

where the term between brackets corresponds to the moment matrix **A**, as defined by Eq. (3.7). Hence,

$$\hat{\boldsymbol{\varphi}} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\,\mathbf{A}^{-1}\mathbf{A} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \qquad (3.23)$$

which corroborates that the FPM approximation can reproduce the functions included in the approximation basis.

The reproduction condition (3.21) entails another important characteristic of the discrete approximation. Note that if the zero-order polynomial ($m$ = 1) is included in the basis, Eq. (3.23) implies that

$$\sum\nolimits_j a_{ij} = 1 \quad \forall\, \boldsymbol{x}_i \in \Omega \qquad (3.24)$$

and the approximation functions constitute a partition of unity. Moreover, the derivative approximation coefficients satisfy that

$$\sum\nolimits_j b_{ij} = 0 \quad \forall\, \boldsymbol{x}_i \in \Omega \qquad (3.25)$$

thus conforming a *partition of nullities*. These properties are of great interest for any numerical approximation and provide a framework for the development and analysis of approximation methods. Moreover, these characteristics can be exploited in the design of numerical solution schemes. This will be useful later in Chapter 4.

## 3.2 Discretization of the equations

The discretization of the problem governing equations is performed in the FPM by using point collocation on the strong form. This method is efficient and easy to implement and also fits the characteristics of the FLS approach.

Considering that a differential problem in a domain Ω with boundary $\Gamma = \Gamma_t \cup \Gamma_\varphi$ is stated according to

$$\begin{aligned} A(\varphi)-b=0 &\quad \text{in } \Omega \\ B(\varphi)-t=0 &\quad \text{in } \Gamma_t \\ \varphi-\bar{\varphi}=0 &\quad \text{in } \Gamma_\varphi \end{aligned} \tag{3.26}$$

the collocation procedure consist on sampling Eqs. (3.26) at each star point in Ω by replacing the continuous variables and their derivatives by the discrete counterparts computed at the cloud level. This leads to

$$\begin{aligned} A(\hat{\varphi})-b=0 &\quad \text{in } \Omega \\ B(\hat{\varphi})-t=0 &\quad \text{in } \Gamma_t \\ \hat{\varphi}-\bar{\varphi}=0 &\quad \text{in } \Gamma_\varphi \end{aligned} \tag{3.27}$$

where the approximate solution $\hat{\varphi}$ is a function of the unknown nodal parameters φ and the cloud metric coefficients (see Eqs. (3.11) and (3.12)). Then, the discrete system (3.27), which is typically sparse (due to the compact support of the FPM approximation) and non-symmetric, can be solved for the nodal parameters, from which the approximated values satisfying the problem equations can be obtained (Eq. (3.11)).

Note that the collocation procedure described above facilitates the enforcement of essential boundary conditions (recall that the FPM approximation has no delta property) and, moreover, spatial numerical integration is not required. Thus, the efficiency is higher if compared to weak form methods, but there are no advantages concerning the order of differentiability of the approximation (it should be at least the order of the equations to be solved). In spite of this, the main drawback of collocation procedures is related to the ill-conditioning of the resultant equations system. This can have a major effect on the robustness and the overall accuracy of the meshless technique, see for example (Cheng & Cheng, 2008).

In order to address robustness and accuracy problems in collocation methods, different approaches have been presented in the literature. Many of them work on the local properties of the approximation; for example, enforcing certain conditions such as positivity by means of the selection of the cloud's points, or through a local manipulation of the weighting function (Jin, Li & Aluru, 2004). Other approaches obtain good results by introducing stabilization terms into the governing equations, particularly for Neumann boundary points; see for instance (Oñate & Idelsohn, 1998; Oñate, Idelsohn, Zienkiewicz, Taylor & Sacco, 1996b). A similar approach in this line is presented in (Boroomand, Tabatabaei & Oñate, 2005), where conditions on the selection of points and the approximation basis are additionally discussed.

It must be noted that in time dependent problems which are solved explicitly in time (like those addressed in this work), the collocation procedure generally leads to a well-conditioned (mass-consistent like) system whose global matrix is diagonally dominant. Thus, no particular problems related to the collocation procedure are found in such cases, and the equations are generally solved by Jacobi or Gauss-Seidel iterations with a low computational cost. This is the approach followed in this work.

## 3.3 Setting free parameters

As observed in the preceding sections, the FPM approximation involves free parameters which need to be properly specified. These parameters are those governing the functional form of the weighting function (γ, *w* and *k*) and the number of points in the local clouds (*np*). Moreover, the order of approximation of the polynomial basis can be changed according to the PDEs to be solved or to improve the solution characteristics. Although these free parameters introduce dependency in the problem solution (a feature which is not always desirable), the local adjustment of the parameters can also help to improve the approximation and address robustness and accuracy issues, thus extracting the best of the numerical approach.

This section presents a numerical investigation aimed at studying the sensitivity of the approximation to the free parameters. This will allow determining suitable ranges for setting the parameters in typical application problems, and also to investigate the capability to improve the local approximation by means of automatic adjustments. The test case employed in the investigation as well as the sensitivity analyses are presented below.

### 3.3.1 Test problem

The test problem proposed involves the approximation of a smooth test function in a cubic analysis domain. The latter is discretized by an unstructured homogeneous distribution of 9650 points, which aims to be representative of the distribution of points usually obtained away from bodies (where the solution is generally smooth) in typical 3D aerodynamic problems. The test function employed is a scalar function defined by

$$\varphi(\boldsymbol{x}) = 64x_1\left(1-x_1\right)x_2\left(1-x_2\right)x_3\left(1-x_3\right) \tag{3.28}$$

and the approximation error is evaluated by the following error norm

$$e_\varphi = \left(\frac{\sum_i (\hat{\varphi}_i - \varphi_i)^2}{\sum_i (\varphi_i)^2}\right)^{1/2} \qquad i = 1, n \tag{3.29}$$

where approximated and exact values of the test function are employed. The error in the derivatives is measured similarly, but using approximate and exact values of the divergence of the test function. Boundary effects are accounted for in the error measurements with the aim to make the results closer to those expected in practice. A view of the analysis domain and the approximated problem solution is presented in Figure 2.

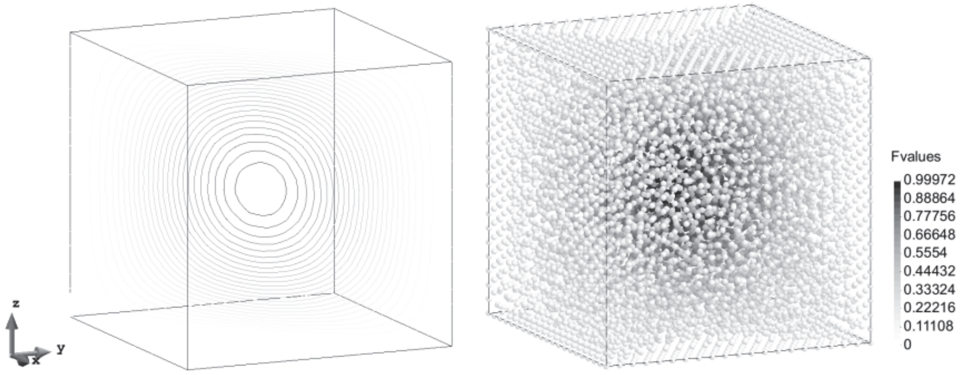

Figure 2. Approximation of a smooth test function in a cubic domain (n = 9650).

### 3.3.2 Effects of parameter γ

The parameter γ of the weighting function (see Eq. (3.20)) is analyzed firstly. Assuming that γ >1, this parameter gives more weight to the cloud boundary points by enlarging the support of the weighting function (see Figure 3). In order to investigate the effects of γ, several test cases involving local clouds with different number of points are solved with linear and quadratic approximation bases. The parameters *w* = 3 and *k* = 2 are assumed to be fixed.

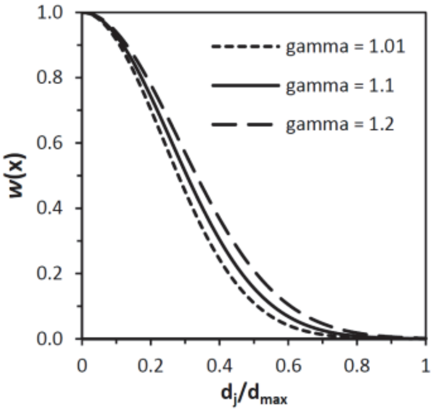

Figure 3. Effects of the parameter γ on the functional form of the weighting function (*w* = 3 and *k* = 2).

The numerical results presented below in Figure 4 show that the error in the test function increases with γ as a result of the loss of locality of the approximation. Conversely, the error in the derivatives improves slightly because more information of far points is introduced into the minimization problem (this can be compared to enlarging the stencil of points). It is also observed that the sensitivity of the error to the parameter γ increases when *np* ≈ *m* and decreases with the number of points in the local cloud.

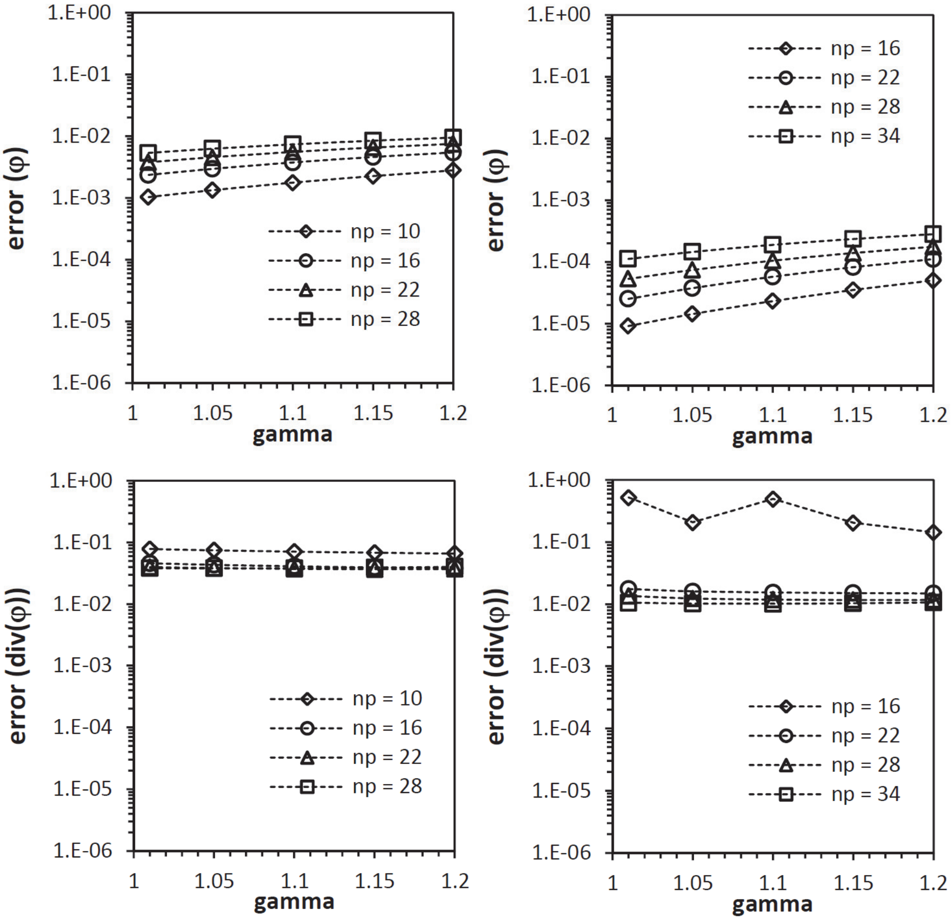

Figure 4. Effects of the parameter γ on the approximation for different *np* computed with linear (left) and quadratic (right) bases ($n = 9650$, $w = 3$ and $k = 2$).

The experiments performed suggest that in typical application problems (where *np*>*m*) the parameter γ can be set to a constant value slightly higher than unit in the whole problem domain (e.g. γ = 1.01). This choice minimizes the error in the approximated function without affecting the derivatives. However, the performance observed in practice indicates that a further increase of the parameter γ can be advantageous to improve the accuracy and robustness of the approximation when sudden changes in the density of points happen in the problem domain (especially in problems governed by conservation laws). This causes an enlargement of the overlapping zone between clouds which has proven effective where rapid changes in the nodal spacing occurs (usually closer to bodies), and does not affect significantly the approximation in areas where the discretization is homogeneous. In such cases, satisfactory results can be obtained by setting a constant value γ in the whole domain in the range of $1.05 < \gamma < 1.2$. It should be noticed that the sensitivity of the approximation to the parameter $\gamma$ is be very low (its effects are mainly localized at the extremes of the cloud) and, therefore, this parameter is not well suited for implementing automatic adjustment procedures to improve the local approximation.

The anomalous behavior observed in the derivatives when using quadratic bases in clouds with $np = 16$ (Figure 4, right bottom), may be due to the fact that the number of points in the local support is not adequate for setting a proper approximation in the range of parameters adopted (mainly due to ill-conditioning of the moment matrices). Additional test performed indicate that this effect becomes more important when the order of the basis increases. Thus, the minimum number of points per cloud required in such cases is higher. Examples of this will be provided later in this section.

### 3.3.3 Effects of parameters *w* and *k*

The next free parameters considered for study in Eq. (3.20) are $w$ and the exponent $k$. These parameters change the distribution of weights (shape) along the entire local support as shown in Figure 5. In spite of the fact that their effects on the numerical approximation can be assimilable, the parameter $w$ has proven more effective in practice to make small adjustments in the approximation (particularly in relation to the derivatives). Therefore, the sensitivity of the numerical solution is investigated for the parameter $w$ while $k$ is assumed constant. This approach, which can be compared to adopt a basic functional shape determined by $k$ which can be modified by varying $w$, simplifies the sensitivity analysis and is also preferred to implement automatic adjustment procedures. Next, the accuracy of the numerical approximation is studied for different values of $w$ using linear and quadratic bases. Local clouds with a variable number of points and constant parameters of the weighting function $k = 2$ and $\gamma = 1.1$ are considered.

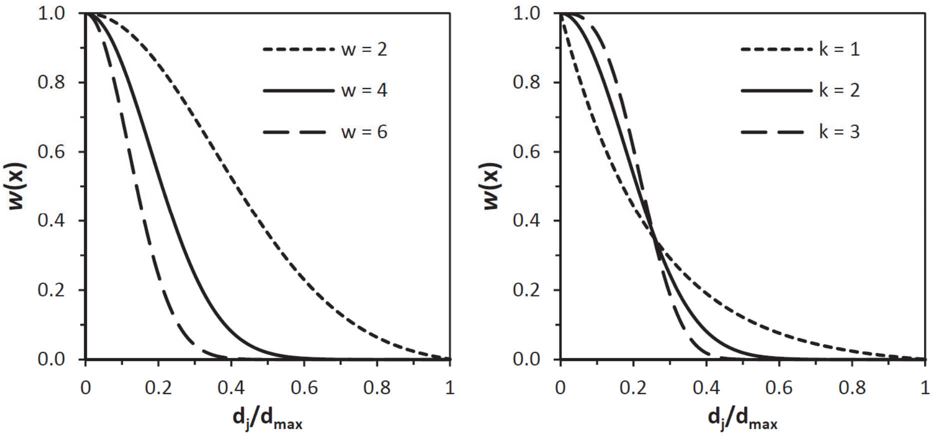

Figure 5. Effects of the parameters $w$ and $k$ on the functional form of the weighting function. Constant values $k = 2$ and $w = 4$ are used, respectively ($\gamma = 1.1$).

As observed in Figure 5 (left), an increase in parameter *w* reduces the weight assigned to the cloud supporting points. As a consequence the numerical approximation tends to interpolate the unknown nodal values, i.e $a_{ij}\rightarrow 0$ for $j\neq i$ while $a_{ii}\rightarrow 1$ (the shape function tends to the Dirac's delta function). Thus, the error in the approximation to the test function decreases with *w*, but the opposite behavior is observed in the derivatives (see results in Figure 6). In the latter case, there is less information of the vicinity of the star point in the minimization problem and, hence, the computed derivatives lose accuracy.

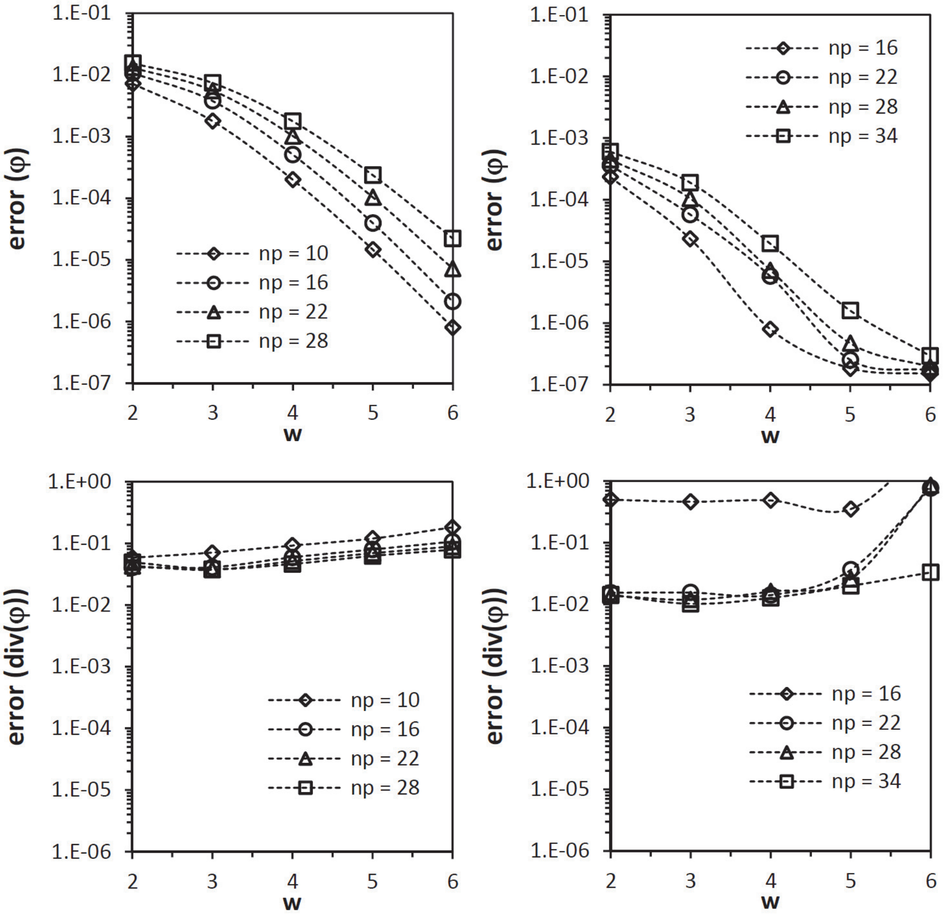

Figure 6. Effects of the parameter *w* on the approximation for different *np* computed with linear (left) and quadratic bases (right) ($n = 9650$, $k = 2$ and $\gamma = 1.1$).

It is worth noting that the improvement of the error in the test function provided by the parameter *w* is limited as the problem becomes more and more ill-conditioned while *w* increases. This makes it difficult to invert the moment matrices with accuracy beyond a given threshold and the numerical approximation deteriorates quickly. Figure 7 illustrates this effect for an isolated cloud centered on the analysis domain (isotropic cloud). Linear and quadratic approximation bases are employed.

Taking into account the numerical experiments performed (and other tests not reported here), a suitable maximum value for *w* is determined for a range of

$3.0 \leq w_{max} \leq 4.5$. The flexibility of this parameter and its effects on the numerical approximation make it appropriate for implementing automatic adjustment procedures. This will be addressed later in this chapter.

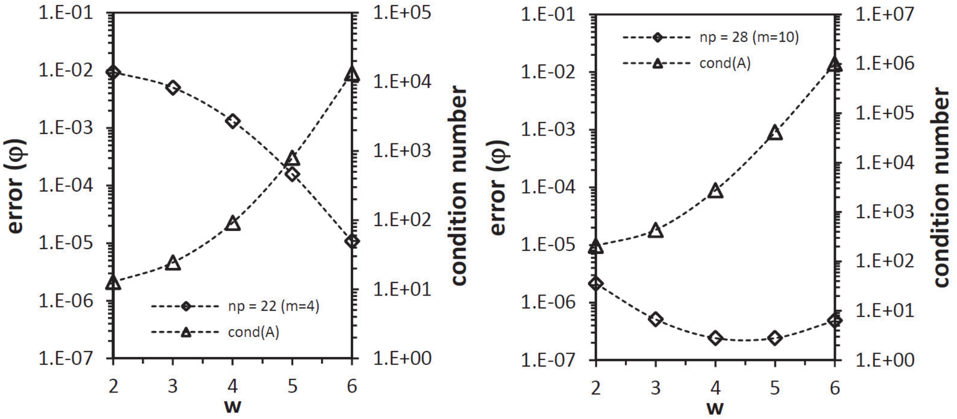

Figure 7. Effects of *w* on the approximation error and conditioning of matrix **A** computed for an isolated cloud for linear and quadratic bases ($k = 2$ and $\gamma = 1.1$).

The last free parameters are the number of points in the local clouds and the order of the approximation bases; these can be studied to a large extent through the examples provided above.

### 3.3.4 Effects of the number of points *np*

Concerning the number of points in the local clouds, the error in the test function increases with *np* as it is affected by smoothing, but a slight improvement is observed for the derivatives (they profit from enlarged stencils). Particularly for the latter, the results show that the impact of *np* is low when the number of points is above a given threshold ensuring the proper conditioning of the minimization problem (see for instance Figure 6). In this regard, if the order of the basis (or parameter *w*) is raised, the condition number of the moment matrices grows and larger supports are needed to achieve a satisfactory behavior (the sensitivity to *np* increases). Likewise, the same behavior occurs when irregular point distributions (not accounted for in this example) are faced. In practice, more points should be introduced in these clouds to counteract ill-conditioning and endow the approximation with robustness. Figure 8 shows the relation between *np* and the condition number of matrix **A** obtained for different basis functions in an isotropic cloud.

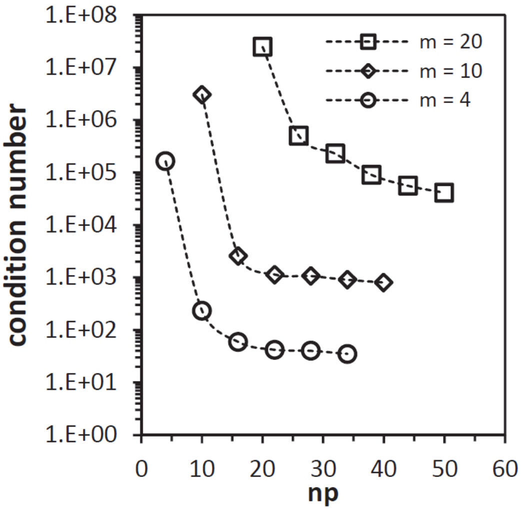

Figure 8. Effects of parameter *np* on the conditioning of the matrix **A** computed for an isolated cloud using linear, quadratic and cubic bases ($k$ = 2 and $\gamma$ = 1.1).

According to these results, *np* can be used to give stability to the minimization problem. Its adjustment will be addressed in conjunction with *w*, because the latter can produce, up to some limit, a similar effect, i.e. enlarging the physical cloud support (but with a lower impact on the computational cost).

### 3.3.5 Effects of the approximation basis

Regarding the order of the approximation bases, similar trends are observed for the combination of parameters under study. An increase in the order of approximation has a positive impact on the function and derivatives errors (Figure 4 and Figure 6), while a reduction improves the conditioning of the moment matrices considerably (Figure 8). According to these results, two main possibilities arise in view of a local adjustment of the approximation basis (*p*-adaptivity). First, enrich the approximation in order to gain accuracy[35]; secondly, smooth the solution to make the problem solvable when a lack of robustness is detected (generally at the expense of a local loss of accuracy). Given the requirements of robustness and computational efficiency in the present work, the first possibility is not attractive, mainly due to the computational cost (larger clouds are needed) and a certain lack of robustness that is normally observed in higher-order approximations. The second possibility, instead, may be useful for addressing problems arising in general discretizations. However, in a previous application along this line (cf. (Ortega, Oñate & Idelsohn, 2007)), the efficiency of this kind of *p*-adaptive procedure proved to be low compared to the adjustment of the weighting function (or

[35] *p*-adaptivity is a field of research of great interest where meshless procedures have demonstrated extensive capabilities and potential; see a review of the most typical methodologies in (Nguyen, Rabczuk, Bordas & Duflot, 2008).

*np*). Moreover, a significant loss of accuracy was often observed in the affected clouds. According to these previous experiences, such adaptive procedure is not considered to be well-suited for the present requirements. Hence, the approximation basis is assumed constant. The selection of the proper order of the approximation basis is discussed in the next section.

## 3.4 Accuracy of the FPM approximation

The accuracy and convergence of a numerical solution depend to a large extent on the discretization error. The typical approaches to determine theoretical bounds for the latter are generally based on the assessment of the truncation terms arising in the discrete approximations. Due to the complexity involved, these studies are often carried out on regular discretizations (generally 1D) and additional assumptions must be introduced when there are free parameters in the problem.

Following this line, regular 1D distribution of points and symmetric local clouds were used to study the accuracy of the FPM derivatives in (Fischer, 1996). The results indicate that quadratic (and even linear) approximations lead to discrete forms equivalent to those obtained by central differencing schemes, which are of second-order accuracy. However, the study also shows that for non-symmetric clouds (or irregular distribution of points) the accuracy of the method tends to first-order[36] and the effective value attained relies heavily on the weighting function settings. If the locality of the weighting function is increased, the approximation tends to recover second-order accuracy. Comparable results are also achieved in other studies, see for instance (Cheng & Cheng, 2008), where theoretical estimates are obtained using a MLS technique[37] in multidimensional FPM approximations.

These analyses are useful to characterize the basic performance of the approximation; however, the limitations adopted in the formal derivation can restrict the scope of the results, and thus the extrapolation to realistic application cases. In order to obtain accuracy measures applicable to typical 3D approximation problems, a grid convergence analysis is performed in this section. Furthermore, the selection of a proper approximation basis is discussed from the viewpoint of attainable accuracy and computational cost.

[36] The deterioration of spatial convergence rates in irregular stencils of points is not a particular issue in meshless approximations and is also observed in conventional mesh-based methods. See for instance a numerical comparison between convergence rates obtained with a meshless approach in contrast with other mesh-based techniques in regular and irregular distribution of points in (Katz, 2009).

[37] In this work, the authors show that the discretization error (related to the problem solution) is also proportional to the condition number of the resulting discrete system obtained by collocation. This makes the global solution accuracy attainable in practice to be further reduced.

### 3.4.1 Convergence analysis

The accuracy of the FPM approximation is investigated here through a grid convergence study. The test problem adopted is similar to that employed in Section 3.3. The analysis domain is initially discretized by an unstructured homogeneous distribution of 1527 points with averaged point spacing $h = n^{-1/3}$ = 8.68E-2. Then, three successively refined discretizations are generated by scaling the grid spacing in a uniform manner. The resultant averaged refinement ratio is $r = h_{k+1}/h_k = (n_{k+1}/n_k)^{-1/3}$ = 1.9 (standard deviation = 0.05). The parameters of the weighting functions are set to $k = 2$, $\gamma = 1.1$ and $w = 3.5$ for all the simulation cases, and clouds with 16, 22 and 56 points are used with linear, quadratic and cubic approximation basis, respectively. Table 1 presents the characteristics of the discretizations.

| grid # | n | h |
|---|---|---|
| **4** | 1527 | 8.68E-2 |
| **3** | 9650 | 4.70E-2 |
| **2** | 65817 | 2.48E-2 |
| **1** | 490528 | 1.27E-3 |

Table 1. Details of the problem discretizations.

The evolution of the approximation error with the grid refinement is studied for the first derivatives (divergence) of the test function (3.28) using the quadratic error norm given by Eq. (3.29). The convergence rates observed for linear and quadratic bases are quite good, although values slightly lower than expected are obtained for the cubic basis (see Figure 9). This result can be explained by the fact that the moment matrices become more ill-conditioned (and prone to numerical errors) as the order of the basis increases. Moreover, larger local supports increase the likelihood of geometrical issues affecting the approximation. These effects are more notorious in coarse grids outside the asymptotic range of convergence.

The dependence of the approximation error on the number of points in the local cloud is illustrated in Figure 10 for the cubic basis. Although the trends obtained are similar to those already seen for linear and quadratic basis (see for example Figure 6), the effects of *np* are more marked in this case. This makes it difficult to adjust the optimal free parameters for the clouds, and the resulting approximation is usually less robust.

It is important to take into account that in the solution of PDEs, the error reported in Figure 9 is only a part of the total error. Other sources of inaccuracies, e.g. due to the discretization scheme adopted and the solution of the resulting algebraic system, could increase the final solution error. This will be studied in Chapter 5 in the context of particular application problems.

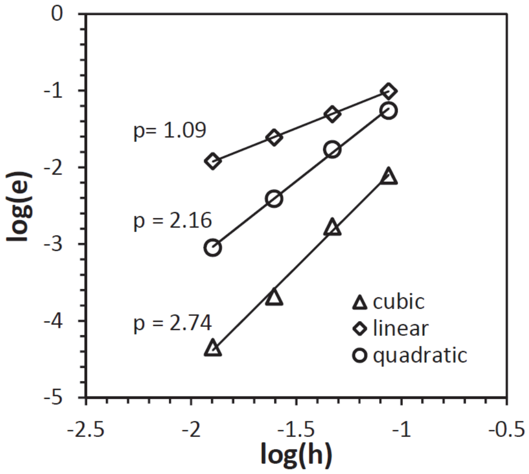

Figure 9. Observed order of accuracy for the derivatives of the test function for linear, quadratic and cubic approximations in clouds with 16, 22 and 56 points respectively ($k = 2$, $\gamma = 1.1$ and $w = 3.5$).

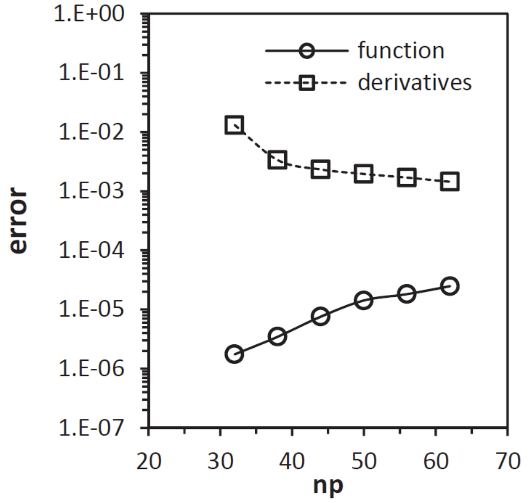

Figure 10. Evolution of the approximation error as a function of the number of points in the local clouds computed with grid #2 and a cubic basis ($k = 2$, $\gamma = 1.1$, $w = 3.5$).

### 3.4.2 Remarks on the order of the approximation bases

According to the results presented in the previous section, the number of free parameters and other particularities involved in a given solution process (e.g. geometrical) make it difficult to determine the suitability of higher-order approximations in general applications. For example, the attainable accuracy in an approximation can be lower than that expected for a higher-order basis due to the conditioning of the moment matrices. Although this behavior can be improved by enlarging the local clouds, thus restoring the accuracy to some extent, this involves an extra cost in the computations which sometimes does not compensate for the accuracy gained. Therefore, a decision should be made in each particular application on the basis of the attainable accuracy-cost ratio and the problem requirements.

Regarding the computational cost, numerical experiments performed with the finest grid (#4) show that the size of the local cloud, required to achieve a (near) optimal solution with a given approximation basis, grows more rapidly as the order of the basis increases. As seen in Figure 11, *np* grows with slope $m^{0.57}$ when going from linear to quadratic bases, but the rate increases to $m^{1.1}$ from quadratic to quartic bases. As a consequence, the number of operations needed for frequent computations such as gradients (which behave linearly with *np*) increases at the same rate and, therefore, the extra CPU-time required can be considerable in large practical problems (especially in explicit time marching computations). Note that an enlargement of the local clouds also increases the size of the data structure and thus the memory storage requirements. This fact can also impose limitations in large-scale problems.

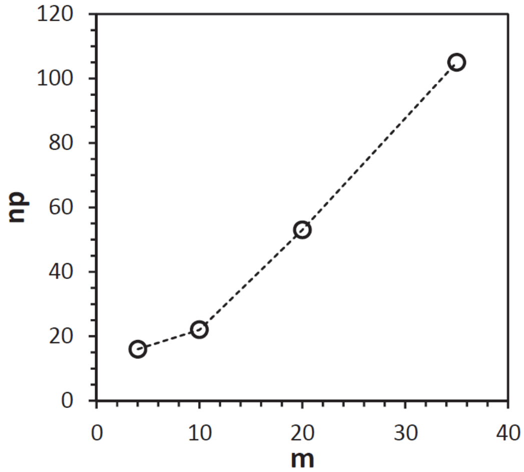

Figure 11. Near optimal clouds computed for grid #4 (n = 490529) using linear, quadratic, cubic and quartic approximation bases with m = 4, 10, 20 and 35, respectively (k = 2, $\gamma = 1.1$, w = 3.5).

Having in mind these considerations, accuracy-cost ratios can be improved with some confidence when moving from linear to quadratic basis because the enlargement of the support size and the extra computational cost involved are low. However, for higher-order approximations, the cloud size and the computational cost grow considerably and the accuracy gains are sometimes uncertain (high-order bases are more likely to suffer instabilities). This means that the attainable accuracy-cost ratios can be not advantageous. In such cases, *h*-refinement seems to be more suitable approach to increase the accuracy.

In spite of the fact that the performance of higher-order approximations may vary depending on the problem[38] and the approximation settings chosen,

[38] For example, more favorable results supporting the use of higher-order bases are obtained for Poisson's type problems in (Ortega, Oñate & Idelsohn, 2007). In that work there are two main differences. First, an automatic procedure for adjustment of

according to our experience in the field of applications addressed in this investigation, quadratic polynomial bases lead to the best compromise between attainable accuracy, robustness and computational cost. This is the choice adopted throughout this work.

## 3.5 Iterative local adjustment of the parameters

For a given point discretization, the characteristics of the FPM approximation depend significantly on the parameters of the weighting functions and the number of points in the local clouds (the approximation basis is considered to be fixed). In the previous sections, the effects of these parameters have been analyzed to determine suitable ranges for practical applications as well as possibilities to improve the local approximation by performing adaptive adjustments. In this respect, the results obtained indicated that the parameters $k$ and $\gamma$ (related to the basic shape of the weighting function and the extension of its local support) may be considered to be constant without detriment to the approximation in particular situations. However, the parameter $w$ and the number of points in the local clouds have shown a large impact on the approximation and also a capacity to improve the characteristics of the latter. Although suitable ranges for these parameters were determined in the previous analyses, specific values should be defined for each problem under consideration. The approach in this work is to define these parameters according to some optimal (general) values and then, based on the characteristics of the local discretization thus obtained, perform iterative adjustments at each cloud to get the best of the numerical approximation. The adjustment procedure adopted is described next for the three-dimensional general case (this is performed at the same time the local approximation is computed). In addition, a more robust methodology for solving the minimization problem, which can be used in cases where the usual approach fails, is presented.

### 3.5.1 Iterative construction of the approximation

Given a star point $\boldsymbol{x}_{\mathrm{i}}$ and a list of surrounding points sorted in ascendant distances, an initial local cloud is set with np $\approx$ 30 (closest points) and the

the local approximation is used, and this guarantees (in some sense) a satisfactory robustness and accuracy. Second, and more important, the problem is solved in an implicit manner. This implies that most of the issues discussed about the computational cost are not relevant as the assembly of the equations is performed only once (approximately 70% of the total simulation time reported corresponds to the solution of the discrete equations). Therefore, the extra cost involved when using higher-order approximations is not really important in such kind of computations and the accuracy-cost ratio results more advantageous.

parameters of the weighting function (Eq. (3.20)) are chosen according to: $k$ = 2, $\gamma$ = 1.1 and $w = w_{\max} \approx 3.5$. Then, the approximation coefficients are computed by Eq. (3.10) and the metric coefficients (function and derivatives) are obtained according to Eqs. (3.11) and (3.12). The resulting approximation is accepted if

i. cond(**A**) $\leq \kappa_{\max}$

and, in addition

ii. $\sum_j a_{ij} - 1.0 \leq \varepsilon$ and $\sum_j b_{ij}^k \leq \varepsilon$

iii. $\max\left(\left|\hat{\varphi} - \varphi\right|, \left|\hat{\varphi}_{,k} - \varphi_{,k}\right|\right) \leq \delta$ (for k = 1,3)

In the conditions above, $k$ denotes the spatial coordinates directions and $\kappa_{\max}$, $\varepsilon$ and $\delta$ are constants specified to achieve a satisfactory balance between accuracy and computational cost. The first requirement (i) imposes an upper bound to the condition number of matrix **A** in order to have a certain guarantee of the inversion process. This is necessary because often numerical results can be obtained even for matrices very close to singular, but with a total lack of accuracy (particularly when working with double-precision floating point arithmetic). Thus, a relatively large condition number can be chosen, e.g. a value $\kappa_{\max}$ = 1.0E6 is adopted in this work (based on the infinite norm of matrix **A)**. Once the approximations coefficients have been obtained, the second requirement (ii) checks partition of unity properties. The fulfillment of the second condition is essential for the implementation of the flow solver that will be proposed later. To this end, a typical value $\varepsilon$ = 1.0E-4 can be adopted. The last requirement (iii) limits the approximation error and this can be seen as a proof of consistency. According to the procedure suggested in (Löhner, Sacco, Oñate & Idelsohn, 2002), the nodal parameters in Eqs. (3.11) and (3.12) are set according to a known spatial function, and the deviation of the approximated values from the latter are measured at the star point of the cloud $\boldsymbol{x}_i$. In this work a simple linear function $\varphi(\boldsymbol{x}) = x_1 + x_2 + x_3$ is used with $\delta$ = 1.0E-4. Note that the local scaled coordinates are dimensionless and of order unity at the cloud level, see Eq. (3.16).

In the case that one of the preceding requirements is not satisfied, the approximation is improved iteratively. In each iteration the parameter $w$ is decreased setting $w = w^i = \alpha w^{i-1}$ ($\alpha \approx 0.75$, $w^0 = w_{\max}$, i: iteration counter) and the numerical approximation is calculated again. This procedure continues until all the requirements are satisfied or $w$ reaches a minimum admissible value $w_{\min} \approx 2$. A few iterations are usually enough to improve the approximation in distorted clouds of points (if $w_{\max}$ is large). Finally, if an admissible approximation is not achieved, new points are inserted in the cloud (six at a time and in ascendant distances from the star point) and the procedure starts again. A maximum number of points about $np_{\max}$=50 is specified for quadratic approximations in 3D clouds[39].

In spite of the fact that the limitations applied to $w_{\min}$ and $np_{\max}$ can prevent some of the requirements (i-iii) from being satisfied in a certain cloud, the resultant local approximation is in general improved. This increases the accuracy and, in particular, the robustness of the overall solution process, enabling the FPM to deal with large (relatively low quality) arbitrary point distributions in practical application problems.

A further enhancement to robustness, though sometimes resulting in a higher computational cost when constructing the approximation, is to avoid direct inversion of matrix **A** in Eq. (3.10) using an alternative approach based on QR factorization. This procedure, already used with success in (Ortega, Oñate & Idelsohn, 2007), allows obtaining suitable results where matrix **A** inversion fails. The idea is to switch to the QR solver the first time that the approximation fails to meet any of the requirements (i-iii). Then, the adjustment of the parameters follows the same basic steps described above. This increments the chances of success and sometimes allows reducing the resulting number of points in the local clouds (and the impact on the computational cost of the simulations). The QR-based solution procedure is described below.

### 3.5.2 QR solution of the minimization problem

QR factorization is a robust and accurate method for solving the minimization problem. The aim of using the QR technique is to get an acceptable solution in cases where direct inversion of the matrix **A** fails, without having to modify significantly the geometrical structure of the cloud. The QR solution procedure can be described as follows.

If matrix **P** has rank *m* and *np* > *m*, it can be factored as

$$\mathbf{P} = \mathbf{Q}\,\mathbf{R} \tag{3.30}$$

where matrix **Q** (*np* x *m*) is orthogonal ( $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ ) and matrix **R** (*m* x *m*) is upper triangular with positive diagonal elements $R_{ii} > 0$. A similar procedure, based on columns pivoting, can be applied for cases in which matrix **P** is rank deficient or near rank deficient.

[39] An enlargement of the local support beyond that limit has not proved to be beneficial because of two main reasons. First, as previously discussed, the computational cost grows rapidly with the cloud size. Second, the potential improvement in accuracy is often small (in part because the added points are relatively far from the star point). Moreover, in problems whose solution has discontinuities and other high-gradient flow features, excessive smearing can be found in large local approximation supports.

In order to apply the QR factorization for solving the WLSQ problem (3.9), it is necessary to obtain an equivalent unweighted problem. To this end, the next factorization is proposed

$$\tilde{\mathbf{W}}(\boldsymbol{x}) = \sqrt{\mathbf{W}(\boldsymbol{x})} \quad \text{such that} \quad \tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{W}} = \mathbf{W} \tag{3.31}$$

where **W**($\boldsymbol{x}$)=diag($w_{ij}$) and matrix **P** is modified according to

$$\tilde{\mathbf{P}} = \tilde{\mathbf{W}}\mathbf{P} \tag{3.32}$$

Then, it is possible to write an equivalent minimization problem as

$$\left(\tilde{\mathbf{P}}^{\mathrm{T}}\tilde{\mathbf{P}}\right)\boldsymbol{\alpha} = \left(\tilde{\mathbf{P}}^{\mathrm{T}}\tilde{\mathbf{W}}\right)\boldsymbol{\varphi} \tag{3.33}$$

whose equivalence with Eq. (3.9) is verified by

$$\begin{gathered}
\tilde{\mathbf{P}}^{\mathrm{T}}\tilde{\mathbf{P}}\boldsymbol{\alpha} = \tilde{\mathbf{P}}^{\mathrm{T}}\tilde{\mathbf{W}}\boldsymbol{\varphi} \\
(\tilde{\mathbf{W}}\mathbf{P})^{\mathrm{T}}\tilde{\mathbf{W}}\mathbf{P}\boldsymbol{\alpha} = (\tilde{\mathbf{W}}\mathbf{P})^{\mathrm{T}}\tilde{\mathbf{W}}\boldsymbol{\varphi} \\
\mathbf{P}^{\mathrm{T}}(\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{W}})\mathbf{P}\boldsymbol{\alpha} = \mathbf{P}^{\mathrm{T}}(\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{W}})\boldsymbol{\varphi} \\
\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}\boldsymbol{\alpha} = \mathbf{P}^{\mathrm{T}}\mathbf{W}\boldsymbol{\varphi} \\
\mathbf{A}\boldsymbol{\alpha} = \mathbf{B}\boldsymbol{\varphi}
\end{gathered} \tag{3.34}$$

Next, the modified matrix (3.32) is factorized, i.e. $\tilde{\mathbf{P}} = \mathbf{QR}$, and replaced in the equivalent unweighted problem given by Eq. (3.33). This leads to

$$\begin{gathered}
(\mathbf{QR})^{\mathrm{T}}\mathbf{QR}\boldsymbol{\alpha} = (\mathbf{QR})^{\mathrm{T}}\tilde{\mathbf{W}}\boldsymbol{\varphi} \\
\mathbf{R}^{\mathrm{T}}(\mathbf{Q}^{\mathrm{T}}\mathbf{Q})\mathbf{R}\boldsymbol{\alpha} = \mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{W}}\boldsymbol{\varphi}
\end{gathered} \tag{3.35}$$

where $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$ due to the orthogonality property. Multiplying by $(\mathbf{R}^{\mathrm{T}})^{-1}$

$$\mathbf{R}\boldsymbol{\alpha} = \mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{W}}\boldsymbol{\varphi} \tag{3.36}$$

and the unknown coefficients $\boldsymbol{\alpha}$ can be finally obtained

$$\boldsymbol{\alpha} = \mathbf{R}^{-1}\left(\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{W}}\right)\boldsymbol{\varphi} \tag{3.37}$$

Here matrix **R** is generally well-conditioned and its inverse is easy to obtain with accuracy, even for the cases when matrix **P** is near rank-deficient.

The described procedure allows getting shape functions of acceptable quality in cases where these cannot be obtained via direct inversion of matrix **A**. However, the solution of the minimization problem by QR factorization entails an increment in the computational cost, which can be up to twice as that in the solution via matrix **A** inversion if *np* » *m* (Demmel, 1997). This extra cost may be not very significant in the overall time when the percentage of problematic clouds in the domain is low, but this cannot be ensured in large practical problems and, therefore, the application of the QR-based procedure should be considered in detail. For example, in a practical test case involving a half-span wing-body configuration with approximately 2.8M points (this problem will be presented later in Chapter 4), the iterative procedure triggered in about 70% of the total clouds (30% of the latter did not fulfill the requirements (i-iii)). Then, the increment in the computational cost can be considerably if QR factorization is used.

## 3.6 Domain and local discretization in the FPM

The spatial discretization requirements found in meshless methods can be relatively low when compared with those in conventional mesh-based techniques. However, the spatial distribution of the discrete points composing the analysis domain and the local clouds continue playing an important role. In spite of the fact that procedures intended to adjust the local approximation (such as that presented in the previous section) help to minimize discretization-related issues, the quality of the resultant approximation is still proportional to the quality of the distribution of points on which it is based.

The domain and local discretization procedures used in this work are described below. Details regarding the computational cost of the methods are given in the context of a practical test case in Section 5.

### 3.6.1 Domain discretization

Meshless approximations have advantages regarding the problem discretization because conformity, topology and the final quality of the partition generally are not a matter of concern. Thus, suitable point distributions filling the analysis domain can be generated more quickly and efficiently by using dedicated algorithms. As it was first shown in (Löhner & Oñate, 1998), suitable point discretizations can be generated at least one order of magnitude faster than in mesh-based methods by avoiding many of the costly checks required to ensure the validity of the resultant volume partition (an advancing-front based technique was adopted in that work). Besides the generation time, meshless discretizations may also have advantages regarding the preparation of the computational models. As simpler CAD representations can be often employed for point generation, the developing time can be reduced considerably. Therefore, any discretization technique employed with meshless solvers will profit from these characteristics. In our experience, a highly effective technique is that developed by Calvo (Calvo, 2005)[40].

In this method, the point generation starts from a closed boundary grid (internal isolated surfaces are also allowed) and after an initial triangulation new points are inserted in the centre of empty spheres filling Ω through an optimization driven point insertion procedure; see (Idelsohn, Calvo N. & Oñate, 2003). This incremental quality technique, based on unconstrained Delaunay tetrahedralization, allows achieving a quality point discretization with approximated cost O($n$). The resulting discretization is suitable for FPM computations and does not require further optimization (cosmetic) procedures (though it can benefit from them). It should be noted that this technique has additional advantages regarding model preparation; e.g. the point spacing inside the domain does not need to be specified, as it is automatically assigned by computing linear variations between input grids. Moreover, a desired grid spacing can be locally enforced by introducing internal surface grids (*h-layers*) which can be isolated and non-conformal. These elements are removed from the final discretization but the nodes are conserved. This gives ample ability to the generation technique to cope with the discretization of complex geometries, but not at the expense of a higher model complexity.

## 3.6.2 Local cloud construction

Given a global point discretization of the analysis domain, there are different criteria in the literature that can be applied to define the size, shape and spatial structure of local clouds suitable for FPM approximations. Some of the techniques proposed belong to geometrical intuitive considerations related to the cloud symmetry, and are mainly intended to have a proper support in the different spatial directions (e.g. selection of points by quadrants, variance of point distances, etc.). Other techniques introduce mathematical concerns with basis on the structure of the matrices involved in the minimization problem, focusing for instance on conditioning and invertibility features (Han & Meng, 2001; Liu, Li & Belytschko, 1997). Mixed geometrical and mathematical considerations are also employed. Among them, criteria based on the overlap within approximation subdomains, or related with the stability of the point collocation procedure and positivity conditions have been proposed; see for instance (Liszka, Duarte & Tworzydlo, 1996), (Xiaozhong, Gang & Aluru, 2004) and (Boroomand, Tabatabaei & Oñate, 2005). While effective in

[40] This method is coded into a software package named MeshSuite (see http://www.cimec.org.ar), which can be linked as a library into the computation code or can be used as an external program through command line arguments or a graphical interface. Recently, this methodology has been incorporated into the GID pre-post processing software developed by CIMNE (GID, 2013).

particular contexts, most of these procedures often fail to meet other important practical requirements. In this regard, the robustness, computational cost and the ability of the procedure to deal with general geometries with no need of manual input should be also of the major concern.

A technique striking a good balance between all these requirements was proposed in (Löhner, Sacco, Oñate & Idelsohn, 2002), and we follow the main lines defined therein. The construction of the local clouds and the numerical approximation is this work is described below.

Given a point discretization bounded by a triangulation with associated geometrical data, for each star point $\boldsymbol{x}_\mathrm{i}$ in $\Omega$:

a. A set of neighboring points within a given radius is sought by using a spatial search algorithm based on bins. These points are employed to generate an initial local cloud.

b. The points in the initial cloud are filtered in order to match boundary restrictions (needed to ensure physical compatibility in the local cloud). Basically, if a ray from $\boldsymbol{x}_\mathrm{i}$ to another point $\boldsymbol{x}_\mathrm{j}$ in the local cloud pierces a boundary, the point $\boldsymbol{x}_\mathrm{j}$ is discarded. If the resultant number of admissible points is lower than a given threshold (our experience in 3D problems has shown 120 points to be enough) the search radius is increased and the procedure returns to step (a).

c. An unconstrained Delaunay grid is generated with the admissible points and the layer of nearest neighbors of $\boldsymbol{x}_\mathrm{i}$ is retained and stored[41]. This guarantees the necessary overlapping of the clouds in order to cover the complete analysis domain[42]. Furthermore, as will be seen throughout this work, the information concerning the first layer of neighboring points for each star point is very useful to implement several computational procedures.

After generating the first layers of neighboring points, the numerical approximation is computed (recall that complete quadratic polynomial bases are used). For each star point $\boldsymbol{x}_\mathrm{i}$ in $\Omega$:

d. A local cloud is initialized with its layer of nearest neighbors[43]. If the number of points is lower than $\mathrm{np}_{\mathrm{min}}$ (about 30 in 3D), further points are added from an auxiliary neighbors list. This auxiliary list is

[41] When boundary points are involved in the local cloud, a pass of alpha-shape is used to clean the resultant cloud connectivity with a very low extra cost.

[42] Non-overlapping clouds of points typically lead to a loss of accuracy and robustness, especially in problems involving conservation laws. The partial loss of information from the surroundings occurring in disconnected clouds can explain this behavior to a large extent. In this case, the normal propagation of variables inside the problem domain and the fulfillment of CFL conditions can be at risk.

[43] Note that always $\mathrm{np}_{\mathrm{initial}} = \max(\mathrm{np}_{\mathrm{min}};\text{number of Delaunay nearest neighbors})$.

constructed by adding points according to layers and ordering them by increasing distances from $\boldsymbol{x}_i$.

e. The numerical approximation is solved by applying the automatic procedure described in Section 3.5. If additional points must be introduced in the local cloud, these are obtained from the auxiliary neighbors list generated in step (d).

The procedure described above has proven reliable in general problems in which the data available are the coordinates of the points and a boundary grid delimiting the analysis domain. However, the steps involved require some operations per node that are computationally intensive (e.g. ray-face intersection, local triangulation, sorting and matrix inversion in the minimization problem). Although these are totally independent and important speed-ups can be achieved by running in parallel, it is customary to simplify the procedures whenever possible. For instance, the information about connectivity between points known from a previous domain discretization stage can be employed to speed-up the computations. This is particularly used in fixed domain problems or when moderate movement/deformation is involved. In these cases, the graph of nearest neighbors can be obtained from the global cloud connectivity[44] after applying a filtering stage to enforce boundary restrictions (step (b)). In this way, a considerably time saving can be achieved by avoiding a local Delaunay gridding at each cloud.

Regarding the filtering stage (step (b)), in simple domains this can be performed with a low computational cost by using only visibility criteria; see for instance (Ortega, Oñate & Idelsohn, 2007). In more complex cases involving multiple surfaces and corners, this simple procedure may lead to indeterminacies for some points and more robust intersection algorithms should be employed. The implementation in the present work proceeds as follows. Firstly, closest boundary faces (triangles) to $\boldsymbol{x}_i$ are sought within the radius of the cloud by using bins. These faces are filtered according to a visibility criterion and, finally, every ray from $\boldsymbol{x}_i$ to another point in the local cloud is checked for intersection[45] with a face by using a brute force approach. As only a few nearby faces are usually involved (if the discretization is not very coarse), the computational cost is not high.

It should be noticed that when information on the global connectivity of points is available, only visibility criteria can be applied to keep the computational requirements low (the global connectivity ensures to some extent the compatibility of the nearest points). If no connectivity information is available, for example after performing local adaptive refinement, the ray-face intersection procedure should be employed.

[44] The construction and storing of the required data structures can be performed very efficiently from the global cloud connectivity; see for instance (Löhner, 2001).

[45] The intersection algorithm employed follows the lines given in (Sunday, 2001).

## 3.7 FPM data structure

The data structure used to store the problem information plays an important role in the efficiency of the numerical implementation. A suitable definition of this structure can help to reduce memory storage requirements and facilitate the access to data during execution. In this section some details about the data structure used in this work are given in order to complete the description of the basic FPM approach.

The basic information required in typical computations can be organized into point-based arrays of dimension (#,npoint) when the data is determined *a-priori* (e.g. point coordinates, problem variables, physical properties, etc.) and linked lists can be used if the data length is not known in advance or varies during the simulation. In the latter case two vectors are employed in this work; one is used to store the data (an estimation of the maximum size should be given) and the other allows to access the data stored in the latter by using the initial and final position of each item. See for instance (Löhner, 2001) for a general description of linked lists and efficient algorithms for generating and organizing typical data structures.

Using linked lists, in this work the local cloud connectivity is stored in a vector `clouds(maxsize)`. There, the star point and the surrounding neighbors for each cloud are ordered consecutively in memory. This vector is initially dimensioned as `maxsize=npmax*npoint` where `npmax` is a user-specified constant (typically 50). However, additional free space can be reserved in cases in which the total number of points in the domain can increase, for instance when performing *h*-adaptivity. The vector `clouds` is accessed by means of two vectors, `np_clouds(npoint+1)` and `np_first(npoint)`. The vector `np_clouds` stores the initial and final position in `clouds` of the data concerning each cloud in the domain. The vector `np_first` only allows retrieving the data in the nearest first layer of a point `ipoin` (this is frequently required during execution). Examples of use are given below.

```
!   Local cloud for "ipoin"
    do j = np_cloud(ipoin)+1,np_cloud(ipoin+1)
       jpoin = clouds(j)
    end do

!   First layer of "ipoin" including the star point
    do j = np_cloud(ipoin)+1,np_ first(ipoin)
       jpoin = clouds(j)
    end do

!   First layer "ipoin" excluding the star point
    do j = np_cloud(ipoin)+2,np_first(ipoin)
       jpoin = clouds(j)
    end do
```

The metric information corresponding to each cloud is stored similarly. To this end, an array shapes(ns,maxsize) is used for the coefficients $a_{ij}$ and $\boldsymbol{b}_{ij}$ in each cloud. If only the approximation and its first derivatives are stored, then ns=4. Another useful information, computed when constructing the clouds and used repeatedly during computations, is the distance from the star point to each point of the cloud. The inverse distances are stored in a vector inv_dist(maxsize) (with exception of the star point). For example,

```
!   Metric coefficients and inverse distances for cloud "ipoin"
    do j = np_cloud(ipoin)+1,np_cloud(ipoin+1)
       aij   = shapes(1,i)    ! shape function
       bij_1 = shapes(2,i)    ! derivatives
       bij_2 = shapes(3,i)
       bij_3 = shapes(4,i)
       inv_dist_ij = inv_dist(j)  ! 1 / ||xj-xi|| j ≠ ipoin
    end do
```

In spite of the fact that the final length of the vectors does not differ much from the estimated one, these can be packed and resized after generation. This allows to remove blanks between clouds where *np<npmax* and reduces the required memory storage. However, if it is expected that the data will change dynamically during the computations (e.g. as occurs when *h*-adaptive procedures are employed), it may be useful to conserve the free space to accommodate new data or modify the existent clouds.

Using the data structure described above, typical operations such as the computation of the gradients are performed as follows

```
!   gradient of a scalar variable var(n) at a point "ipoin"
    gradvar(1:3) = 0.0
    do j = np_cloud(ipoin)+1,np_cloud(ipoin+1)
       jpoin = clouds(j)
       vpoin = var(jpoin)
       gradvar(1) = gradvar(1) + shapes(2,j)*vpoin
       gradvar(2) = gradvar(2) + shapes(3,j)*vpoin
       gradvar(3) = gradvar(3) + shapes(4,j)*vpoin
    end do
```

It is important to point out that a suitable ordering of the points is required to achieve a satisfactory performance during execution. The ordering required should be in accordance to the manner in which the data will be retrieved and, to this end, the information of the local cloud connectivity (or global connectivity if available) can be used.

The procedure adopted here follows the main lines given in (Flores, Ortega & Oñate, 2011). Starting with a first point (arbitrary), consecutive numbers are assigned to its first layer neighbors. Once all of them have been renumbered, the procedure is repeated starting with the nodes just renamed and renumbering their neighbors. The cycle is repeated until all the nodes in the mesh have been assigned a new numbering. When performing operations on the clouds (which are processed in the same order as during the renumbering) the chances of a cache miss are reduced because neighboring nodes are stored close to each order in the memory. The result of this renumbering procedure may have some dependence on the choice of the first node. This can be reduced by running several renumber passes taking as first node at each iteration the last one from the previous pass (Löhner, 2001).

## 3.8 Concluding remarks

The basic aspects of the FPM methodology have been studied in this chapter with attention to the particular characteristics of the method which have an impact on the robustness, accuracy and computational cost of practical applications.

Issues regarding the sensitivity of the numerical solution to adjustable parameters in the approximation were examined first. These parameters, mainly related to those defining the shape of the weighting function, the number of points in the local clouds and the order of the polynomial basis, have been studied in order to determine their effects on the numerical solution, as well as suitable ranges of application and their ability to improve the local approximation using automatic adjustments. The analyses performed determined suitable ranges for the parameters and indicated those that can be adjusted to improve the local approximation. These are the parameter *w* of the weighting function (which affects the weight distribution along the local support) and the number of points in the local clouds (*np*). The proposed adjustment procedure assumes that a user-specified initial setting is given (with the aim to be optimal in the specified ranges), and modifies the local parameters to guarantee satisfactory accuracy and robustness. This procedure, which is carried out automatically during the construction of the local clouds, has demonstrated to simplify the problem setting while increasing the robustness of the resultant FPM approximation.

A grid convergence analysis was performed in order to investigate the accuracy of the FPM approximation. The study showed that the convergence rates of the method are within the expected and, in addition, allowed to identify issues related to the approximation settings and the geometrical characteristics of the clouds that can affect considerably the accuracy of the approximation. These results were not only useful to infer the performance attainable in practice, but also to decide on the suitability of higher-order approximations. In this concern, the numerical tests suggest that quadratic bases can achieve the best compromise between accuracy, robustness and computational cost. The use of *p*-adaptive procedures to improve the local approximation has been considered less efficient than the adjustment of parameters *w* and *np*. Thus, in this work it was decided to use constant basis functions.

Various aspects of the global point discretization and the construction of local clouds have been also discussed in this chapter, as well as numerical implementation details. This complements and rounds off the description of the basic FPM methodology.

# 4 INVISCID FLOW SOLVER

A Finite Point methodology for solving the inviscid compressible flow equations is presented in this chapter. The flow solver is based on a central-type differencing scheme, biased by the introduction of an upwind numerical flux which provides convective stabilization. Following the ideas in the so-called Godunov-type methods, the numerical flux is calculated by means of an approximate Riemann solver, for which the Roe solution is employed. This choice leads to a robust scheme with enhanced wave capture capabilities, but only first-order accurate. The spatial accuracy is increased through slope-limited MUSCL reconstruction (van Leer's approach). The time integration of the resulting semi-discrete equations is performed explicitly by means of a multi-stage time marching scheme. Convergence acceleration techniques are also employed in order to speed-up the computation of steady flows.

This flow solver is robust and accurate, and meets well the requirements of general application problems. However, there are particular situations, often arising in practice, where high-accuracy results are not mandatory, and the computational time is of major concern. In such cases, faster, lower-fidelity schemes are usually preferred. In order to address this need, two additional stabilization schemes are also proposed. These schemes only differ from the previous upwind approach in the definition of the numerical flux; thus they are implemented with easiness into the solution algorithm described above.

This chapter is organized as follows. The inviscid flow equations are presented first in Section 4.1 and the spatial discretization scheme is described in Section 4.2. There, theoretical foundations and practical implementation aspects of the low-order scheme and the higher-order extension are discussed. Also in the same section, two alternative stabilization schemes, appropriate for faster computations, are proposed. The time discretization of the equations is described in Section 4.3, and final remarks on accuracy, conservation issues and computational implementation of the FPM flow solver are given in Section 4.4. In order to complete the description of the basic solution scheme, and focusing on inviscid flow problems, the application of boundary conditions is addressed in Section 4.5. Finally, numerical examples involving typical verification and validation test cases are provided in Section 4.6 to illustrate the basic performance (especially accuracy) of the methodology.

## 4.1 The Euler equations

Inviscid compressible flows can be modeled by the Euler equations. These are a system of first-order hyperbolic equations stating mass, momentum and energy conservation, that can be written in several equivalent forms. Assuming an Eulerian frame of reference, these equations can be expressed in conservative differential form by

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}^k}{\partial x_k}=\mathbf{0} \tag{4.1}$$

where **U** is the state vector (which contains the conserved quantities) and $\mathbf{F}^k$ is the advective flux vector in the spatial direction $x_k$. These vectors are given by

$$\mathbf{U}=\begin{bmatrix}\rho\\ \rho u_i\\ \rho e_t\end{bmatrix}\quad,\quad \mathbf{F}^k=\begin{bmatrix}\rho u_k\\ \rho u_i u_k+\delta_{ik}p\\ (\rho e_t+p)u_k\end{bmatrix} \tag{4.2}$$

where $\rho$, $p$ and $e_t$ denote, respectively, the density, pressure and total energy (internal+kinetic) of the fluid; $u_i$ is the $i$-component of the velocity vector, $\delta_{ik}$ is the Kronecker delta and indices $i$ and $k$ = 1,3 for 3D cases. The following state relation for a perfect gas closes the system of equations

$$p=\rho(\gamma-1)\left[e_t-\tfrac{1}{2}u_k u_k\right] \tag{4.3}$$

in which $\gamma = C_p/C_v$ is the specific heats ratio (typically $\gamma = 1.4$ for air). The solution of Eq. (4.1) in a closed domain $\Omega$ with boundaries $\Gamma$ requires additional proper initial and boundary conditions. These are discussed later in Section 4.5.

### 4.1.1 Quasi-linear form of the Euler equations

Taking advantage of the fact that the flux vectors in Eqs. (4.2) can be written only in terms of the conservative variables **U**, it is possible to rewrite

$$\frac{\partial \mathbf{F}^k}{\partial x_k}=\frac{\partial \mathbf{F}^k}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_k}=\mathbf{A}^k\frac{\partial \mathbf{U}}{\partial x_k} \tag{4.4}$$

where $\mathbf{A}^k$ is the Jacobian matrix of the flux vector $\mathbf{F}^k$ ($A_{ij}^k = \partial \mathbf{F}_i^k/\partial \mathbf{U}_j$). Introducing Eq. (4.4) into Eq. (4.1), the quasi-linear form of the Euler equations is obtained

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}^k \frac{\partial \mathbf{U}}{\partial x_k} = \mathbf{0} \qquad (4.5)$$

where $\mathbf{A}^k = \mathbf{A}(\mathbf{U},x_k,t)$[46]. Due to the fact that the equation system is hyperbolic, the jacobian matrices **A** have real eigenvalues and are diagonalizable, i.e. they have a complete set of linearly independent eigenvectors. Hence, it is possible to perform the following factorization

$$\mathbf{A} = \mathbf{R}\,\Lambda\,\mathbf{R}^{-1} \qquad (4.6)$$

where **R** is a matrix whose columns are the right eigenvectors of the matrix **A**, the rows of $\mathbf{R}^{-1}$ are their left eigenvectors and Λ is a diagonal matrix whose entries are the eigenvalues of **A**. It is important to note that the Jacobian matrices $\mathbf{A}^k$ cannot be diagonalized simultaneously, but it is possible to do so with any linear combination of these matrices. Therefore, given an arbitrary vector direction $\hat{\boldsymbol{n}}$ it is possible to state

$$\mathbf{A}_{\boldsymbol{n}} = \mathbf{A}^k n^k = \mathbf{R}_{\boldsymbol{n}}\,\Lambda_{\boldsymbol{n}}\,\mathbf{R}_{\boldsymbol{n}}^{-1} \qquad (4.7)$$

in which the matrices of eigenvectors and eigenvalues are calculated for the Jacobian matrix in the direction of the arbitrary vector $\hat{\boldsymbol{n}}$, i.e. $\mathbf{A}_{\boldsymbol{n}}$. The Jacobian matrices $\mathbf{A}^k$ and their associated eigenvectors and eigenvalues matrices can be calculated analytically if the fluid constitutive relations are specified. Explicit expressions for these matrices can be found in the literature; see for instance (Hirsch, 1990).

The quasi-linear form of the Euler equations (4.5), combined with the decomposition (4.7), allows a complete description of the flow problem in terms of propagating waves (characteristic approach). This description is essential to describe the mathematical properties of the problem and plays an essential role in the design of numerical solution schemes, especially in schemes based on upwind type discretizations.

### 4.1.2 Non-dimensional form of the equations

Dimensionless forms of the equations are advantageous for two main reasons. On one hand, they simplify the computations and use of the results, since only the relevant flow and problem parameters are involved. Furthermore, they

[46] It is interesting to note that the flux vectors are a homogeneous function of degree one ($\mathbf{F}^1(\alpha\mathbf{U})= \alpha\mathbf{F}^1(\mathbf{U})$ being α an arbitrary constant). Thus, they satisfy the identity $\mathbf{F}^1(\mathbf{U})=\mathbf{A}^1\mathbf{U}$ and this allows writing the flux vector (4.4) introducing the Jacobian matrix inside the spatial derivatives. In spite of the fact that this procedure results in a mathematically equivalent problem, the equivalence of the discrete forms is not guaranteed. Some numerical schemes use this alternative form of equations.

allow to homogenize the magnitude of the different terms, which improves the numerical behavior.

In order to obtain the dimensionless equations, each variable in Eq. (4.1) is divided by a collection of proper variables or constant terms whose product has the same dimension. The dimensionless variables adopted here are

$$\tilde{t} = \frac{t\,c_\infty}{L} \;,\; \tilde{x}_i = \frac{x_i}{L} \;,\; \tilde{u}_i = \frac{u_i}{c_\infty} \;,\; \tilde{p} = \frac{p}{\rho_\infty c_\infty^2} \;,\; \tilde{e}_t = \frac{e_t}{c_\infty^2} \;,\; \tilde{T} = \frac{T}{T_\infty} \tag{4.8}$$

where $L$ is a characteristic length of the problem and $\rho_\infty$, $T_\infty$ and $c_\infty = (\gamma R T_\infty)^{1/2}$ are the undisturbed (upstream) density, temperature and speed of the sound, respectively. Note that the dimensionless velocities coincide with the Mach number along the different coordinate directions ($M_{1,2,3}$).

Replacing the dimensionless variables (4.8) into Eqs. (4.1), these results

$$\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{t}} + \frac{\partial \tilde{\mathbf{F}}^k}{\partial \tilde{x}_k} = \mathbf{0} \tag{4.9}$$

with the following set of state and flux vectors

$$\tilde{\mathbf{U}} = \begin{bmatrix} \tilde{\rho} \\ \tilde{U}_2 \\ \tilde{U}_3 \\ \tilde{U}_4 \\ \tilde{\rho}\tilde{e}_t \end{bmatrix} \quad , \quad \tilde{\mathbf{F}}^k = \begin{bmatrix} \tilde{U}_k \\ \tilde{U}_1\tilde{u}_k + \delta_{1k}\tilde{p} \\ \tilde{U}_2\tilde{u}_k + \delta_{2k}\tilde{p} \\ \tilde{U}_3\tilde{u}_k + \delta_{3k}\tilde{p} \\ \tilde{h}\,\tilde{u}_k \end{bmatrix} \tag{4.10}$$

being $\tilde{h} = \left(\tilde{\rho}\tilde{e}_t + \tilde{p}\right)$ the total fluid enthalpy. Note that the Eqs. (4.9) are similar to Eqs. (4.1), but all the variables are now dimensionless. The application of this procedure to the thermodynamic ideal gas relations leads to

$$\begin{aligned} \tilde{p} &= \tilde{\rho}(\gamma-1)\left[\tilde{e}_t - \tfrac{1}{2}\tilde{u}_k\tilde{u}_k\right] \\ &= \tilde{\rho}\left(\frac{1}{\gamma}\right)\tilde{T} \quad \text{(e.o.s)} \end{aligned} \tag{4.11}$$

In this work, dimensionless freestream density and temperature are considered to be unity. Thus, the far-field pressure and the reference state vector result

$$\tilde{p}_\infty = 1/\gamma \tag{4.12}$$

$$\tilde{\mathbf{U}}_\infty = \begin{bmatrix} 1 \\ \mathrm{M}_{\infty 1} \\ \mathrm{M}_{\infty 2} \\ \mathrm{M}_{\infty 3} \\ \dfrac{1}{\gamma(\gamma-1)} + \dfrac{\mathrm{M}_\infty^2}{2} \end{bmatrix} \tag{4.13}$$

where the components of the freestream Mach number in the coordinate directions $x_\mathrm{k}$ can be obtained for typical body axes by

$$\begin{aligned} M_{\infty 1} &= M_\infty \cos\alpha \cos\beta \\ M_{\infty 2} &= -M_\infty \sin\beta \\ M_{\infty 3} &= M_\infty \sin\alpha \cos\beta \end{aligned} \tag{4.14}$$

being α and β the angles of attack and sideslip which define the attitude of the body under study. Hereafter, the tilde will be dropped from the non-dimensional variables for the sake of clarity. Thus, all the variables and equations will be in non-dimensional form unless otherwise specified.

## 4.2 Spatial discretization

The semi-discrete problem in the FPM is obtained from the strong form of the flow equations (4.9) by replacing the continuous spatial variables by their discrete approximations (Eqs. (3.11) and (3.12)). Then, it is possible to write for each point $\boldsymbol{x}_\mathrm{i}$ in Ω

$$\frac{\partial \hat{\mathbf{U}}_i}{\partial t} = -\frac{\partial \mathbf{F}_i^k}{\partial x} = -\sum_{j\in\Omega_i} b_{ij}^k \mathbf{F}_j^k \tag{4.15}$$

where $\hat{\mathbf{U}}_i = a_{ij}\mathbf{U}_j$ (sum in j) is the discrete approximation at $\boldsymbol{x}_\mathrm{i}$ of the conservative variables vector, $\mathbf{F}_i^k = \mathbf{F}^k(\mathbf{U}_i)$ is the $k$-component of the convective flux (see Eqs. (4.10)), and $a_{ij}$ and $\boldsymbol{b}_{ij}$ are the cloud metric coefficients given by Eqs. (3.19).

Taking advantage of the partition of unity properties of the FPM (cf. Section 3.1.3), it is possible to express

$$\sum_{j\in\Omega_i} b_{ij}^k = b_{ii}^k + \sum_{j\neq i} b_{ij}^k = 0 \quad \rightarrow \quad b_{ii}^k = -\sum_{j\neq i} b_{ij}^k \tag{4.16}$$

Hence, replacing Eq. (4.16) in Eq. (4.15), the semi-discrete system results

$$\frac{\mathrm{d}\hat{\mathbf{U}}_i}{\mathrm{dt}} = -\sum_{j \neq i} b_{ij}^k \left[ \mathbf{F}_j^k - \mathbf{F}_i^k \right] \qquad \forall j \in \Omega_i \tag{4.17}$$

where the term involving the derivatives of the convective flux vector can be shown to be equivalent to a central approximation at point $\boldsymbol{x}_\mathrm{i}$[47].

### 4.2.1 The low-order solution scheme

Being the discrete convective term in Eq. (4.17) equivalent to a central-type discretization, this can be affected by spurious instabilities caused by odd-even decoupling at the grid level. Typical approaches to prevent this behavior usually involve upwind discretizations or, alternatively, use some kind of numerical diffusion to counter the decoupling mode[48]. In order to implement either approach in an unified manner, a more suitable form of Eq. (4.17) is obtained by scaling by a half the stencil of points used for its calculation (Praveen, 2004). This leads to

$$\frac{\mathrm{d}\hat{\mathbf{U}}_i}{\mathrm{dt}} = -2\sum_{j \neq i} b_{ij}^k \left[ \mathbf{F}_{ij}^k - \mathbf{F}_i^k \right] \qquad \forall j \in \Omega_i \tag{4.18}$$

where $\mathbf{F}^\mathrm{k}_\mathrm{ij}$ is a numerical flux computed at the midpoint of the ray (edge) connecting the star point $\boldsymbol{x}_\mathrm{i}$ to another point $\boldsymbol{x}_\mathrm{j}$ in $\Omega_\mathrm{i}$ (see Figure 12). The numerical flux introduced in Eq. (4.18) provides a means to stabilize the central-type discretization employed for the convective terms.

The basic solution approach adopted in this work is based on an upwind definition of the numerical flux. The scheme follows the general lines proposed in (Löhner, Sacco, Oñate & Idelsohn, 2002), where the flux is computed by solving an approximate Riemann problem along the edge connecting the points $\boldsymbol{x}_\mathrm{i}$ and $\boldsymbol{x}_\mathrm{j}$ (flux-difference splitting). This allows the flux

[47] A total equivalence of this term with a central finite-difference representation can be demonstrated in the 1D case, using a regular distribution of points and symmetric clouds with 3 points, cf. (Fischer, 1996).

[48] The stability of numerical schemes of the form of Eq. (4.17) can be also studied in the more general analysis framework of Local Extremum Diminishing (LED) schemes (Jameson, 1993, 1995). According to this theory, the local maxima of the flux does not increase and local minima does not decrease if the coefficients $\boldsymbol{b}_{ij}$ are all non-negative (positivity condition). From this point of view, in the FPM the positiveness of the metric coefficients, and thus the non-oscillatory character of the resulting scheme, is not guaranteed. The introduction of additional stabilization terms (or upwinding) can be seen as a modification of the weight coefficients to satisfy the positivity condition. This approach facilitates the analysis and design of non-oscillatory discrete schemes.

function to distinguish the contributions arising from negative and positive waves, thus providing the required upwinding direction for each equation. Moreover, this choice has the advantage that the approximate Riemann solver provides information concerning the exact solution of the problem, giving robustness and excellent shock capturing properties to the numerical scheme. Among the different possibilities that exist for flux splitting, the approximate Riemann solver of Roe is adopted here (Roe, 1981). Relevant aspects about approximate Riemann solvers and the solution procedures are given below.

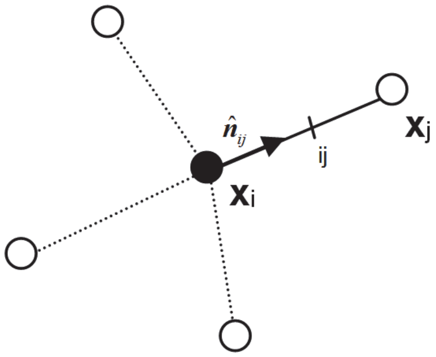

Figure 12. Computation of the numerical flux in a cloud of points.

### 4.2.1.1 Approximate Riemann problem

The Riemann problem is one of the few problems involving the unsteady 1D Euler equations which has exact solution. Considering any pair of points $x_i$ and $x_j$, this problem can be defined by the following initial condition

$$\mathbf{U}(x,t_0)=\begin{cases}\mathbf{U}_\mathrm{L}=\mathbf{U}_i & x\le x_{ij}\\ \mathbf{U}_\mathrm{R}=\mathbf{U}_j & x>x_{ij}\end{cases} \tag{4.19}$$

where $\mathbf{U}_\mathrm{L}$ and $\mathbf{U}_\mathrm{R}$ are constant state vectors (Godunov's approach) and $x_{ij}$ is a point located midway between $x_i$ and $x_j$. For t > $t_0$, the interaction between the initial states gives rise to the three basic types of flow discontinuities; a shock wave, an expansion fan and a contact discontinuity, whose development can be solved analytically. Figure 13 sketches the Godunov's problem at t = $t_0$.

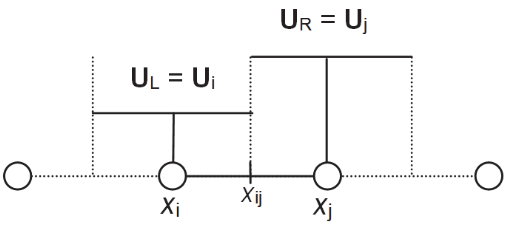

Figure 13. Piecewise constant initial condition for the Riemann problem at $x_{ij}$.

The solution of the Riemann problem fully describes (in space and time) the ratio of the variables across the discontinuities, their spatial position and speed of propagation. Hence, the interface flux $\mathbf{F}_{\mathrm{ij}}=\mathbf{AU}(x_{\mathrm{ij}})$ can be obtained; see for instance (Laney, 1988). The latter can be written as

$$\mathbf{F}_{\mathrm{ij}} = \tfrac{1}{2}\left(\mathbf{F}_L + \mathbf{F}_R\right) - \tfrac{1}{2}\left|\mathbf{A}\right|\left(\mathbf{U}_R - \mathbf{U}_L\right) \tag{4.20}$$

where |**A**| is the absolute value of the Jacobian matrix , which can be obtained as $|\mathbf{A}|=\mathbf{R}|\Lambda|\mathbf{R}^{-1}$ (cf. Section 4.1.1). It is important to note that, due to the similarity of the solution of the Riemann problem, the numerical flux (4.20) is constant along a line $x = x_{\mathrm{ij}}$ in the (x,t) plane.

Unfortunately, the solution of the Riemann problem allowing to compute Eq. (4.20) in an exact manner requires solving a system of non-linear algebraic equations which can be very expensive for applications such as in Eq. (4.18). Thus, a more affordable approach is to use *approximate* Riemann solutions, where the non-linear flux function **F**(**U**) is locally linearized in the governing equations (4.5), i.e. the Jacobian matrix **A** is considered to be constant.

In order to perform such linearization, a plane connecting the states $\mathbf{U}_{\mathrm{L}}$ and $\mathbf{U}_{\mathrm{R}}$ can be used (Laney, 1988). That is,

$$\mathbf{F}(\mathbf{U}) \approx \mathbf{F}(\mathbf{U}_{\mathrm{L}}) + \mathbf{A}_{\mathrm{RL}}\left(\mathbf{U} - \mathbf{U}_{\mathrm{L}}\right) \tag{4.21}$$

where the matrix $\mathbf{A}_{\mathrm{RL}}$ satisfies

$$\mathbf{F}(\mathbf{U}_{\mathrm{R}}) - \mathbf{F}(\mathbf{U}_{L}) = \mathbf{A}_{\mathrm{RL}}\left(\mathbf{U}_{\mathrm{R}} - \mathbf{U}_{\mathrm{L}}\right) \tag{4.22}$$

Eq. (4.22) involves (dim+2) equations concerning the components of the flux vector **F**(**U**) and $(\mathrm{dim}+2)^2$ unknowns, which are the components of the Jacobian matrix $\mathbf{A}_{\mathrm{RL}}$. Thus, there are an infinite number of planes containing the states $\mathbf{U}_{\mathrm{L}}$ and $\mathbf{U}_{\mathrm{R}}$. Fortunately, based on a result derived from the mean value theorem in the scalar case, it is possible to replace the secant plane by averaged tangent planes. Hence, it can be supposed that

$$\mathbf{A}_{\mathrm{RL}} = \mathbf{A}\left(\mathbf{U}_{\mathrm{RL}}\right) = \mathbf{A}\left(\mathbf{U}_{\mathrm{L}}, \mathbf{U}_{\mathrm{R}}\right) \tag{4.23}$$

in which $\mathbf{U}_{\mathrm{RL}}$ is an intermediate state between $\mathbf{U}_{\mathrm{L}}$ and $\mathbf{U}_{\mathrm{R}}$. Then, introducing the averaged Jacobian matrix (4.23) into Eq. (4.22), the unknowns of the problem are reduced to the (dim+2) components of the intermediate state $\mathbf{U}_{\mathrm{RL}}$ and a well-posed system is obtained. Moreover, the statement (4.23) has another important advantage as any expression based on **A**(**U**) remains true for $\mathbf{A}(\mathbf{U}_{\mathrm{LR}})$, by replacing **U** with the intermediate state $\mathbf{U}_{\mathrm{LR}}$. This allows to use the expressions of the Jacobian matrix, the eigenvectors and eigenvalues derived for the general non-linear case to the linearized problem.

#### 4.2.1.2 Roe averages

In order to find an intermediate state vector $\mathbf{U}_{\mathrm{LR}}$ suitable for calculating the averaged Jacobian matrix, Roe proposed the following conditions (Roe, 1981)

i. $\mathbf{A}(\mathbf{U}_{\mathrm{L}},\mathbf{U}_{\mathrm{R}}) \to \mathbf{A}(\mathbf{U}) \quad \text{for} \quad \mathbf{U}_{\mathrm{L}},\mathbf{U}_{\mathrm{R}} \to \mathbf{U}$

ii. $\mathbf{A}(\mathbf{U}_{\mathrm{L}},\mathbf{U}_{\mathrm{R}})$ is diagonalizable with real eigenvalues

iii. $\mathbf{A}(\mathbf{U}_{\mathrm{L}},\mathbf{U}_{\mathrm{R}})(\mathbf{U}_{\mathrm{R}} - \mathbf{U}_{\mathrm{L}}) = \mathbf{F}(\mathbf{U}_{\mathrm{R}}) - \mathbf{F}(\mathbf{U}_{L}) \quad \text{for any } \mathbf{U}_{\mathrm{L}},\mathbf{U}_{\mathrm{R}}$

These requirements, collectively named as *property U* in the original paper, imply (i) consistency of the resulting scheme with the Eqs. (4.5), (ii) hyperbolicity of the linear scheme, and the satisfaction of conservation and Rankine-Hugoniot conditions (iii); cf. (Hirsch, 1990). These requisites make the approximated solution obtained matches the exact Riemann solution when the states $\mathbf{U}_{\mathrm{L}}$ and $\mathbf{U}_{\mathrm{R}}$ are connected by a single discontinuity (shock or contact). However, expansion fans cannot be reproduced as the waves in the approximate solution have zero spread. In such cases, the solution will admit the appearance of expansion shocks, which violate the entropy conditions. Some procedures to overcome this behavior are discussed in Section 4.2.1.2.2.

The procedure adopted by Roe to compute the intermediate variables and the averaged Jacobian matrix focuses on the fulfillment of the requirements (i-iii), but also on the efficiency of the solution process. With these aims, he proposed a transformation of the original problem, recognizing that the state and flux functions are quadratic in a new set of variables given by

$$\boldsymbol{w} = \rho^{1/2}\left[1,u,h\right]^{\mathrm{T}} \tag{4.24}$$

which notably simplifies the solution of Eqs. (4.22), cf. (Roe, 1981). The matrix $\mathbf{A}(\mathbf{U}_{\mathrm{LR}})$ thus obtained is similar to the typical Jacobian matrix $\mathbf{A}$, but changing the variables by the following density averages

$$\begin{aligned} \rho_{ij} &= \sqrt{\rho_j \rho_i} \\ u_{ij} &= \frac{\sqrt{\rho_j}u_j + \sqrt{\rho_i}u_i}{\sqrt{\rho_j} + \sqrt{\rho_i}} \\ h_{ij} &= \frac{\sqrt{\rho_j}h_j + \sqrt{\rho_i}h_i}{\sqrt{\rho_j} + \sqrt{\rho_i}} \end{aligned} \tag{4.25}$$

where $h$ is the specific total enthalpy of the fluid, $u$ denotes the velocity and subscripts $i$ and $j$ denote the left (L) and right (R) states, respectively. Additionally, the average speed of sound results

$$c_{ij}=\sqrt{(\gamma-1)\left(h_{ij}-\tfrac{1}{2}u_{ij}u_{ij}\right)} \tag{4.26}$$

Note that the Roe variables can be calculated in a computationally more efficient way with the help of an auxiliary parameter $r$ = $(\rho_i/\rho_j)^{1/2}$ (Hirsch, 1990). The latter allows rewriting Eqs. (4.25) as

$$\rho_{ij}=r\rho_i \quad , \quad u_{ij}^k=\frac{r\,u_j^k+u_i^k}{r+1} \quad , \quad h_{ij}=\frac{r\,h_j+h_i}{r+1} \tag{4.27}$$

In order to obtain the positive Jacobian matrix required for constructing the numerical flux in Eq. (4.20), the factorized form of Eq. (4.6) can be employed along with the Roe's averaged variables. An efficient procedure to perform this calculation for the multidimensional case is presented below.

#### 4.2.1.2.1 Calculation of the Roe's numerical flux

In the multidimensional case, the upwind numerical flux (4.20) should be computed along edges connecting pairs of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the local cloud (see Eq. (4.18)). This directional flux can be expressed as

$$\mathbf{F}_{ij}^k=\tfrac{1}{2}\left(\mathbf{F}_j^k+\mathbf{F}_i^k\right)-\tfrac{1}{2}\left|\mathbf{A}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)\right|\left(\mathbf{U}_j-\mathbf{U}_i\right)\hat{n}_{ij}^k \tag{4.28}$$

being $\hat{\boldsymbol{n}}_{ij}$ a unit vector in the direction of the edge $\boldsymbol{l}_{ij}=\boldsymbol{x}_j\text{-}\boldsymbol{x}_i$ and $|\mathbf{A}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)|$ the absolute value of the Roe matrix calculated in the same direction. According to the hyperbolicity properties of the Euler equations (see Section 4.1.1), Eq. (4.28) can be computed by

$$\left|\mathbf{A}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)\right|=\mathbf{R}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)\left|\mathbf{\Lambda}_{\hat{n}}(\mathbf{U}_i,\mathbf{U}_j)\right|\mathbf{R}_{\hat{n}}^{-1}(\mathbf{U}_i,\mathbf{U}_j) \tag{4.29}$$

with the diagonal matrix of eigenvalues given by

$$\Lambda=diag\left\{|\lambda_1|,|\lambda_2|,|\lambda_3|,|\lambda_3|,|\lambda_3|\right\} \tag{4.30}$$

where

$$\begin{aligned}\lambda_1&=\hat{u}_{ij}+\tilde{c}_{ij}\\ \lambda_2&=\hat{u}_{ij}-\tilde{c}_{ij} \qquad \text{with} \quad \hat{u}_{ij}=\tilde{u}_{ij}^k\,\hat{n}^k\\ \lambda_3&=\hat{u}_{ij}\end{aligned} \tag{4.31}$$

and the expressions for the remaining right and left eigenvectors can be found in the literature, see for instance (Hirsch, 1990). Note that all the variables in Eq. (4.29) correspond to the averaged Roe variables calculated for the constant states $\mathbf{U}_i$ and $\mathbf{U}_j$. Then, the upwind (diffusive) contribution of the numerical flux (4.28) can be obtained by

$$\mathbf{D} = \left|\mathbf{A}_{\hat{n}}(\mathbf{U}_{ij})\right|(\mathbf{U}_j - \mathbf{U}_i) = \mathbf{R}_{\hat{n}}^{-1}(\mathbf{U}_{ij})\left|\mathbf{\Lambda}_{\hat{n}}(\mathbf{U}_{ij})\right|\mathbf{R}_{\hat{n}}(\mathbf{U}_{ij}) \cdot \Delta\mathbf{U}_{ij} \quad (4.32)$$

in which $\Delta\mathbf{U}_{ij} = (\mathbf{U}_j - \mathbf{U}_i)$ is the difference state vector.

In spite of the fact that explicit expressions are available for the construction of the upwind term (4.32), its computation involves matrix-matrix and matrix-vector multiplications which can demand a considerable computational effort. In order to carry out these calculations more efficiently, a closed expression is derived in (Turkel, 1988). This can be written as

$$\mathbf{D} = \Pi_1\,\Delta\mathbf{U} + \Pi_2 \begin{bmatrix} 1 \\ \tilde{u}_{ij}^{(1)} \\ \tilde{u}_{ij}^{(2)} \\ \tilde{u}_{ij}^{(3)} \\ \tilde{h}_{ij} \end{bmatrix} + \Pi_3 \begin{bmatrix} 0 \\ \hat{n}^{(1)} \\ \hat{n}^{(2)} \\ \hat{n}^{(3)} \\ \hat{u}_{ij} \end{bmatrix} \quad (4.33)$$

where

$$\begin{aligned} \Pi_1 &= |\lambda_3| \\ \Pi_2 &= \left(\frac{\sigma_1 - |\lambda_3|}{c^2}\right)\Psi_1 + \frac{\sigma_2}{c}\Psi_2 \\ \Pi_3 &= \frac{\sigma_2}{c}\Psi_1 + \left(\sigma_1 - |\lambda_3|\right)\Psi_2 \end{aligned} \quad (4.34)$$

with

$$\begin{aligned} \Psi_1 &= \frac{1}{\gamma - 1}\left(q\Delta\mathbf{U}^{(1)} - \tilde{u}_{ij}^{(1)}\Delta\mathbf{U}^{(2)} - \tilde{u}_{ij}^{(2)}\Delta\mathbf{U}^{(3)} - \tilde{u}_{ij}^{(3)}\Delta\mathbf{U}^{(4)} + \Delta\mathbf{U}^{(5)}\right) \\ \Psi_2 &= -\hat{u}_{ij}\Delta\mathbf{U}^{(1)} + \hat{n}^{(1)}\Delta\mathbf{U}^{(2)} + \hat{n}^{(2)}\Delta\mathbf{U}^{(3)} + \hat{n}^{(3)}\Delta\mathbf{U}^{(4)} \end{aligned} \quad (4.35)$$

and

$$\sigma_1 = \tfrac{1}{2}\left(|\lambda_1| + |\lambda_2|\right), \sigma_2 = \tfrac{1}{2}\left(|\lambda_1| - |\lambda_2|\right), q = \tfrac{1}{2}\,\tilde{u}_{ij}^k\,\tilde{u}_{ij}^k \quad (4.36)$$

Note that the expressions above can be recast in one and two-dimensional forms by cancelling all the variables in the remaining Cartesian components.

### 4.2.1.2.2 Entropy correction

In the particular case that a transition through a sonic point occurs in a flow expansion, the Roe solver will allow the appearance of an expansion shock (pressure and density decrease across the shock), a solution which does not comply with the second principle of thermodynamics. In order to overcome this unphysical behavior, most numerical techniques locate the sonic expansion (wave speeds→0) and diffuse the shock into expansion fans. This can be achieved by limiting the minimum value of the wave speed, i.e. limiting the eigenvalues to a constant value $\lambda_{\min} > 0$. In general, the minimum wave speed is set to a certain small fraction of the spectral radius of the Jacobian matrix $\rho(\mathbf{A})$. This can be implemented by enforcing (Turkel, 1988)

$$\begin{aligned} |\lambda_1| &= \max\left[\, |\lambda_1|\,, \alpha_2 \rho(\mathbf{A})\,\right] \\ |\lambda_2| &= \max\left[\, |\lambda_2|\,, \alpha_2 \rho(\mathbf{A})\,\right] \quad \text{with} \quad \rho\left(\mathbf{A}\right) = \left|\hat{u}_{ij}\right| + \tilde{c}_{ij} \qquad (4.37) \\ |\lambda_3| &= \max\left[\, |\lambda_3|\,, \alpha_1 \rho(\mathbf{A})\,\right] \end{aligned}$$

where the arbitrary constants can be set approximately to $\alpha_1 \approx 0.1$ and $\alpha_2 \approx 0.2$. It is important to note that, in addition to the possible appearance of expansion shocks, vanishing eigenvalues can cause misbehaviors and instabilities in other flow situations found in practical computations. For instance, the acoustic eigenvalues $\lambda_1$ and $\lambda_2$ vanish at sonic points and the eigenvalue $\lambda_3$ goes to zero at stagnation points (or boundary layers approaching the surface wall). These situations may require some special setting of the arbitrary constants to avoid instabilities and/or excessive numerical diffusion (particularly if $\alpha_1$ is large within boundary layers).

A more accurate correction is proposed in (Harten & Hyman, 1983). There, the eigenvalues are limited according to

$$\begin{aligned} |\lambda_k| &= \frac{\lambda_k^2 + \delta^2}{2\delta} \quad \text{if } |\lambda_k| < \delta \\ \delta &= \max\left[0, (\lambda_k - \lambda_k^L), (\lambda_k^R - \lambda_k)\right] \quad k = 1,3 \end{aligned} \qquad (4.38)$$

where $\lambda_k$ corresponds to the eigenvalues (4.31) calculated for the Roe averaged variables and $\lambda^L_k$ and $\lambda^R_k$ are the eigenvalues calculated at the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, which define the left and right constant states of the Riemann problem. Although this method leads to slightly better results, the improvement observed in practical problems no always compensates for the extra cost. This is the reason why the limiting (4.37) is adopted in this work. Other entropy corrections can be found in the literature, see a comparative study among the most typical methods in (Kermani & Plett, 2001).

### 4.2.2 Increasing the spatial accuracy

The upwind scheme proposed in the previous section is non-oscillatory but it can only achieve first-order accuracy. Given that this does not meet the requirements in practical computations (at least second-order accuracy is needed when the flow is smooth), the spatial accuracy of the scheme should be increased. As observed by van Leer, the accuracy of Godunov type schemes results only from the projection stage, thus it can be improved by replacing the zero-order extrapolation ($\mathbf{U}_L$=$\mathbf{U}_i$ and $\mathbf{U}_R$=$\mathbf{U}_j$) used to compute the upwind flux at $\boldsymbol{x}_{ij}$ with higher-order estimates (Van Leer, 1977, 1979). Methods using this approach are known in the literature as MUSCL[49] (Monotone Upstream-centered Schemes for Conservation Laws).

Following the MUSCL approach, backward and forward-biased higher-order estimates at the interface point $\boldsymbol{x}_{ij}$ ($\mathbf{U}_i^+$,$\mathbf{U}_j^-$) are constructed by using information of neighboring cells. Then, these approximated values are introduced in Eq. (4.28) replacing the constant states ($\mathbf{U}_i$,$\mathbf{U}_j$)[50], i.e.

$$\mathbf{F}_{ij}^k = \tfrac{1}{2}\left(\mathbf{F}^k(\mathbf{U}_j^-) + \mathbf{F}^k(\mathbf{U}_i^+)\right) - \tfrac{1}{2}\left|\mathbf{A}_{\hat{n}}(\mathbf{U}_i^+, \mathbf{U}_j^-)\right|\left(\mathbf{U}_j^- - \mathbf{U}_i^+\right)\hat{n}_{ij}^k \quad (4.39)$$

When the flow is smooth, the introduction of the extrapolated states in Eq. (4.39) reduces the jump in the variables through the interface. This makes the upwind dissipative contribution to vanish while the scheme recovers its central character (second-order accuracy). However, this procedure cannot guarantee an oscillation free solution around discontinuities. As stated by Godunov's theorem, monotonicity cannot be achieved if the order of approximation of the scheme is higher than linear (Godunov, 1959). Therefore, the accuracy should be reduced in such cases.

In the context of MUSCL extrapolation, the order of the scheme can be modified by the introduction of non-linear slope limiters into the reconstruction process. Basically, these limiters recognize local extrema of the solution and automatically switch the high-order extrapolation to a zero-order extrapolation (or an admissible value in between). This mechanism allows changing to the low-order scheme around discontinuities while preserving the order of accuracy where the flow is smooth. Next, the extrapolation procedure is described and two possibilities to calculate the limiters are presented. Practical implementation aspects are also discussed.

[49] This name was coined after the first code developed by van Leer implementing this methodology (Van Leer, 1979).

[50] In practice, the computation of the extrapolated flux vectors at the interface ($\mathbf{F}^k$ ($\mathbf{U}_i^+$) and $\mathbf{F}^k$($\mathbf{U}_j^-$)) can be avoided by assuming zero-order extrapolation in this part of the upwind flux, i.e. ($\mathbf{F}^k$ ($\mathbf{U}_i$) and $\mathbf{F}^k$($\mathbf{U}_j$). This simplification reduces the computational cost and has not demonstrated to have an effect on the solution accuracy.

#### 4.2.2.1 Higher-order accurate estimates

Assuming that the function **U**(x) is smooth enough in the vicinity of $x_{\mathrm{ij}}$, it is possible to state the following leftward-biased reconstructions (Laney, 1988)

- Linear centered reconstruction

$$\mathbf{U}(x) \approx \mathbf{U}_i + \frac{\mathbf{U}_j - \mathbf{U}_i}{\Delta x}\left(x - x_i\right) \tag{4.40}$$

- Linear backward reconstruction

$$\mathbf{U}(x) \approx \mathbf{U}_i + \frac{\mathbf{U}_i - \mathbf{U}_{i\text{-}1}}{\Delta x}\left(x - x_i\right) \tag{4.41}$$

- Quadratic backward reconstruction

$$\begin{aligned}\mathbf{U}(x) \approx \mathbf{U}_i &- \frac{\mathbf{U}_j - \mathbf{U}_i - (\mathbf{U}_i - \mathbf{U}_{i\text{-}1})}{24} + \frac{\mathbf{U}_j - \mathbf{U}_i}{\Delta x}\left(x - x_i\right) \\ &+ \frac{\mathbf{U}_j - \mathbf{U}_i - (\mathbf{U}_i - \mathbf{U}_{i\text{-}1})}{2\Delta x^2}\left(x - x_i\right)\left(x - x_j\right)\end{aligned} \tag{4.42}$$

Then, defining $\Delta x = x_{\mathrm{j}} - x_{\mathrm{i}}$ and taking $x = x_{\mathrm{ij}} = x_{\mathrm{i}} + \Delta x/2$, it is possible to combine the expressions (4.40), (4.41) and (4.42) into

$$\mathbf{U}_i^+ = \mathbf{U}_i + \frac{1}{4}\left[\left(1-\eta\right)\left(\mathbf{U}_i - \mathbf{U}_{i-1}\right) + \left(1+\eta\right)\left(\mathbf{U}_j - \mathbf{U}_i\right)\right] \tag{4.43}$$

where $\mathbf{U}_{\mathrm{i}}^{+}$ is the backward approximation to $\mathbf{U}(x_{\mathrm{ij}})$. The parameter η controls the degree of the extrapolation, for instance η = -1 leads to a second-order backward approximation for $\mathbf{U}_{\mathrm{i}}^{+}$ and η = 1 or η = 1/3 lead to second-order centered and a third-order approximation, respectively.

Rightward-biased approximations for $\mathbf{U}_{\mathrm{j}}^{-}$ can be obtained similarly by

- Linear centered reconstruction

$$\mathbf{U}(x) \approx \mathbf{U}_j + \frac{\mathbf{U}_j - \mathbf{U}_i}{\Delta x}\left(x - x_j\right) \tag{4.44}$$

- Linear forward reconstruction

$$\mathbf{U}(x) \approx \mathbf{U}_j + \frac{\mathbf{U}_{j+1} - \mathbf{U}_j}{\Delta x}\left(x - x_j\right) \tag{4.45}$$

- Quadratic forward reconstruction

$$\mathbf{U}(x) \approx \mathbf{U}_j - \frac{\mathbf{U}_{j+1} - \mathbf{U}_j - (\mathbf{U}_j - \mathbf{U}_i)}{\Delta x} + \frac{\mathbf{U}_j - \mathbf{U}_i}{\Delta x}\left(x - x_j\right) + \frac{\mathbf{U}_{j+1} - \mathbf{U}_j - (\mathbf{U}_j - \mathbf{U}_i)}{2\Delta x^2}\left(x - x_i\right)\left(x - x_j\right) \tag{4.46}$$

which merge at the interface point into

$$\mathbf{U}_j^- = \mathbf{U}_j - \frac{1}{4}\left[\left(1-\eta\right)\left(\mathbf{U}_{j+1} - \mathbf{U}_j\right) + \left(1+\eta\right)\left(\mathbf{U}_j - \mathbf{U}_i\right)\right] \tag{4.47}$$

being $\mathbf{U}_j^-$ a forward approximation to $\mathbf{U}(x_{ij})$. Like in the previous case, the parameter η allows changing the order of the reconstruction and, thus, the accuracy of the resulting scheme. Figure 14 sketches backward reconstructions for the left state of the Riemann problem at $x_{ij}$. Forward extrapolations perform similarly for the right state.

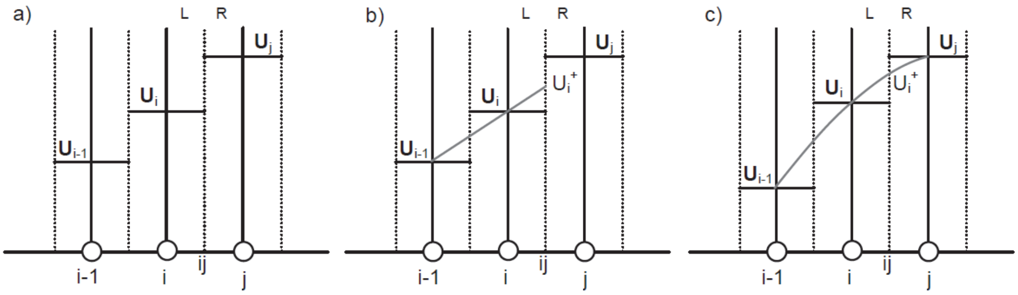

Figure 14. Leftward-biased extrapolations of the variables at the point $x_{ij}$ ($\mathbf{U}_i^+$). From left to right: zero-order, linear and quadratic extrapolations.

As it can be noticed in Eqs. (4.43) and (4.47), higher-order reconstructions need information from adjacent points $x_{i-1}$ and $x_{j+1}$ located to the left and right of the interval in which the approximate Riemann problem is solved. This information can be easily obtained in regular structured point distributions, but the procedures are not straightforward in unstructured discretizations. In such cases, several techniques have been proposed in the literature; see for instance (Löhner, 2001; Lyra & Morgan, 2002). Following these ideas, the unknown state variables $\mathbf{U}_i^+$ and $\mathbf{U}_j^-$ are computed from a centered approximation to the $\nabla\mathbf{U}$ at the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, i.e.

$$\begin{aligned} \Delta\mathbf{U}_i^- &= \mathbf{U}_i - \mathbf{U}_{i-1} = 2\boldsymbol{l}_{ji} \cdot \nabla\mathbf{U}_i - \left(\mathbf{U}_j - \mathbf{U}_i\right) \\ \Delta\mathbf{U}_j^+ &= \mathbf{U}_{j+1} - \mathbf{U}_j = 2\boldsymbol{l}_{ji} \cdot \nabla\mathbf{U}_j - \left(\mathbf{U}_j - \mathbf{U}_i\right) \end{aligned} \tag{4.48}$$

where the vector $\boldsymbol{l}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ links the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and the solution gradients are directly computed from the actual FPM solution using Eq. (3.12). Note that it is not necessary that the additional virtual points coincide with another point in the discretization, and even can fall outside the analysis domain when Eqs. (4.48) are applied near boundaries. Although no special treatment is required in this case, virtual points falling outside the fluid domain may bring about inaccuracies in the extrapolation process. As discussed below, this misbehavior can be mitigated in the limiting stage.

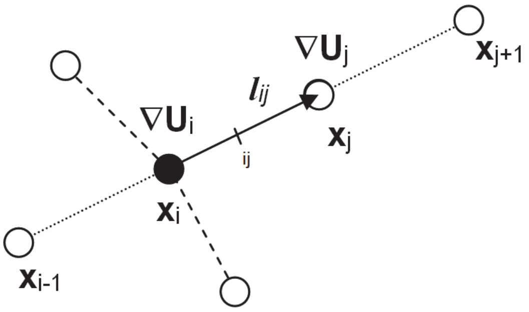

Figure 15. Multidimensional reconstruction of the variables along an edge.

#### 4.2.2.2 Limiting

The reconstruction given by Eqs. (4.43) and (4.47) does not guarantee a non-oscillatory solution near discontinuities and, thus, non-linear limiters are introduced to control the extrapolation. Taking information from the local gradients of the solution (through the difference vectors $\mathbf{\Delta U}_{ij}$, $\mathbf{\Delta U}_i^-$ and $\mathbf{\Delta U}_j^+$), the limiters check and correct the slopes involved in the reconstruction process following several criteria derived from TVD or LED conditions for non-oscillatory schemes; see for instance (Hirsch, 1990; Laney, 1988; Lyra & Morgan, 2002). Generally, the limiter functions approach unity when the differences at each side of the interface are similar (e.g. $\mathbf{\Delta U}_i^- \approx \mathbf{\Delta U}_{ij}$) and vanish when they are very different. These functions can be computed at each point by using the primitive, conservative or characteristic variables. The latter choice achieves the best results, but it is more expensive as it requires variables conversion. In the present work the limiters at each point $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, i.e. $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, are calculated using the conservative variables (note that these are vectors of dimension 5 in 3D).

Once the limiters are computed, the MUSCL extrapolation procedure given by Eqs. (4.43) and (4.47) is modified according to

$$\begin{aligned} \mathbf{U}_i^+ &= \mathbf{U}_i + \frac{\mathbf{s}_i}{4}\left[(1-\eta)(\mathbf{U}_i - \mathbf{U}_{i-1}) + (1+\eta)(\mathbf{U}_j - \mathbf{U}_i)\right] \\ \mathbf{U}_j^- &= \mathbf{U}_j - \frac{\mathbf{s}_j}{4}\left[(1-\eta)(\mathbf{U}_{j+1} - \mathbf{U}_j) + (1+\eta)(\mathbf{U}_j - \mathbf{U}_i)\right] \end{aligned} \tag{4.49}$$

where $\mathbf{0} \le \mathbf{s} \le \mathbf{1}$. Note that if the limiters are equal to the unity, Eqs. (4.49) lead to a high-order extrapolation of the variables $\mathbf{U}_i^+$ and $\mathbf{U}_j^-$ at the interface and, thus, a higher-order accurate scheme is obtained. When the limiters are equal to zero, the constant extrapolation is recovered and it leads to a low-order monotone scheme.

Two different approaches for computing the limiting functions are proposed next: the van Albada limiter (Van Albada, Van Leer & Roberts Jr, 1982), which is the standard approach in this work; and the minmod limiter, which is more restrictive and leads to a more robust though diffusive solution.

### 4.2.2.2.1 Van Albada limiter

The van Albada limiter at points $\boldsymbol{x}_\mathrm{i}$ and $\boldsymbol{x}_\mathrm{j}$ can be expressed by

$$\begin{aligned} \mathbf{s}_i &= \max\left[0\,,\frac{2\Delta\mathbf{U}_i^-\Delta\mathbf{U}_{ij}+\varepsilon}{(\Delta\mathbf{U}_i^-)^2+(\Delta\mathbf{U}_{ij})^2+\varepsilon}\right] \\ \mathbf{s}_j &= \max\left[0\,,\frac{2\Delta\mathbf{U}_j^+\Delta\mathbf{U}_{ij}+\varepsilon}{(\Delta\mathbf{U}_j^+)^2+(\Delta\mathbf{U}_{ij})^2+\varepsilon}\right] \end{aligned} \tag{4.50}$$

where ε is a very small positive constant which avoids possible divisions by zero when the surrounding flow field is smooth. In this work ε ≈ 1.0E-5 is adopted (this value may have small effects on the numerical results).

### 4.2.2.2.2 Minmod limiter

The minmod limiter, based on the minimum modulus difference, allows increasing the robustness; however, the solution results more diffusive. This can be written as

$$\begin{aligned} \boldsymbol{s}_i &= \begin{cases} \operatorname{sgn}(\Delta\mathbf{U}_i^-)\dfrac{\min\left[\left|\Delta\mathbf{U}_i^-\right|,\left|\Delta\mathbf{U}_{ij}\right|\right]}{\max\left[\left|\Delta\mathbf{U}_i^-\right|,\left|\Delta\mathbf{U}_{ij}\right|\right]} & \text{if } \operatorname{sgn}(\Delta\mathbf{U}_i^-)=\operatorname{sgn}(\Delta\mathbf{U}_{ij}) \\ 0 & \text{otherwise} \end{cases} \\ \boldsymbol{s}_j &= \begin{cases} \operatorname{sgn}(\Delta\mathbf{U}_j^+)\dfrac{\min\left[\left|\Delta\mathbf{U}_j^+\right|,\left|\Delta\mathbf{U}_{ij}\right|\right]}{\max\left[\left|\Delta\mathbf{U}_j^+\right|,\left|\Delta\mathbf{U}_{ij}\right|\right]} & \text{if } \operatorname{sgn}(\Delta\mathbf{U}_j^+)=\operatorname{sgn}(\Delta\mathbf{U}_{ij}) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.51}$$

With the objective to speed-up the computations, it is possible to use a simplified form of the minmod limiter by a direct replacement in Eq. (4.39) of the following limited difference (Flores, Ortega & Oñate, 2011)

$$\Delta\mathbf{U}_{ij}^{\text{limited}} = \begin{cases} \Delta\mathbf{U}_{ij} - \max\left[\min\left(\Delta\mathbf{U}_i^-, \Delta\mathbf{U}_j^+\right), 0\right] & \textit{if} \;\; \Delta\mathbf{U}_{ij} > 0 \\ \Delta\mathbf{U}_{ij} - \min\left[\max\left(\Delta\mathbf{U}_i^-, \Delta\mathbf{U}_j^+\right), 0\right] & \textit{otherwise} \end{cases} \tag{4.52}$$

#### 4.2.2.2.3 Practical remarks

As mentioned before, the extrapolation procedure given by Eqs. (4.48) can introduce information from outside the fluid domain in clouds having a point on the solid boundary. This may eventually cause inaccuracies in the computation of the limiters (thus, on the extrapolated variables and upwind fluxes), which often lead to a more diffusive solution near the boundaries (especially in coarse discretizations). In order to overcome this behavior to some extent the limiters of the affected nodes can be set to zero. In this way, the flux only takes information from the interior node of the edge, where the loss of accuracy is much smaller (Flores, Ortega & Oñate, 2011).

Another problem commonly observed in practice is related to the non-linearity introduced by the limiters in the solution process, which can penalize considerably the convergence of the solution to the steady state. As observed in (Lyra & Morgan, 2002), when the solution is very close to the steady state, the action of the limiters does not provide a tangible gain in accuracy. Thus, these can be frozen to overcome the limitation on the convergence rate. As suggested in (Flores, Ortega & Oñate, 2011), the limiters can be frozen when the residual has decreased (at least) three orders of magnitude with respect to its initial value; or it can be made progressively along the simulation. The latter choice allows reducing the computational cost as the problem approaches the steady state.

Finally, it is sometimes found in practical analyses that the computation tends to be unstable during the first steps of the simulation. This is often caused by geometries having non streamlined features, for which the initial conditions are far from the steady solution. In such cases, the robustness can be improved by running a certain number of steps with the low-order scheme (i.e. setting all the limiters to zero) at the beginning of the simulation. This is usually enough to reach conditions closer to the steady state, so that the high order scheme converges smoothly. Note that this procedure may not be applicable in cases where the transient solution is of interest.

### 4.2.3 Two reduced-fidelity models

The central-type discretization employed for the convective terms in Eq. (4.18) can also be stabilized by the introduction of artificial diffusion terms. As a matter of fact, if a numerical flux as defined in Eq. (4.20) is introduced into the semi-discrete scheme (4.18), working out the resulting expression it is possible to obtain

$$\frac{\mathrm{d}\hat{\mathbf{U}}_i}{\mathrm{dt}} = -\sum_{j\neq i} b_{ij}^k \left[ \mathbf{F}_j^k - \mathbf{F}_i^k - \mathbf{d}_{ij}^k \right] \qquad \forall j \in \Omega_i \tag{4.53}$$

which is equivalent to the direct addition of an explicit diffusion term $\mathbf{d}_{ij}$ into the centered scheme. The diffusion term in Eq. (4.53) may be, for example, the upwind flux contribution given by Eq. (4.32) in the Roe-MUSCL approach, or any diffusive term helping to counteract the odd-even decoupling mode of the central scheme.

The possibility to use different types of explicit diffusion terms in Eq. (4.53) gives flexibility to the numerical approach and allows meeting more effectively the especial requirements found in practical problems. For example, in many frequent applications of fluid flow solvers, the accuracy of the solution can be somewhat relegated in comparison to the computational cost, which becomes a variable of major concern. This situation can be found, for instance, when fast preliminary data production is required for design and analysis. In such cases, the use of simpler diffusion models can save considerable computation time and still provide accurately enough solutions.

With this aim, two additional classic stabilization approaches are proposed. The first method combines a scalar dissipation model with MUSCL-limited differences. The second approach uses Jameson's second and fourth order dissipation terms and a pressure-based flow detector. The low-fidelity stabilization models are described below and comparisons between these schemes and the Roe-MUSCL approach are presented in Section 5.4.

### 4.2.3.1 Scalar dissipation scheme

In this scheme the positive Roe matrix in Eq. (4.32) is replaced with an averaged spectral radius computed in the direction of the edge. The diffusive flux contribution results

$$\mathbf{d}_{ij}^k = \lambda_{ij} \left( \mathbf{U}_j^- - \mathbf{U}_i^+ \right) \hat{n}_{ij}^k \tag{4.54}$$

where $\lambda_{ij} = | u_{ij}^k \cdot \hat{n}_{ij}^k | + c_{ij}$ is the spectral radius of the Jacobian matrix along the unit vector $\hat{\boldsymbol{n}}_{ij}$ in the direction of the edge $\boldsymbol{l}_{ij} = \boldsymbol{x}_\mathrm{j} - \boldsymbol{x}_\mathrm{i}$ . The fluid velocity $\boldsymbol{u}_{ij}$ and speed of sound $c_{ij}$ are simple aritmetic edge-averaged values, and the extrapolated state variables $\mathbf{U}_\mathrm{i}^+$ and $\mathbf{U}_\mathrm{j}^-$ are computed through Eqs. (4.49). When limiters are zero (e.g. around discontinuities), the dissipative operator (4.54) is second-order and the accuracy of the resulting scheme is first-order, which guarantees monotonicity. If the flow is smooth and the limiters are equal to unity, the dissipative operator is fourth-order (see for instance (Löhner, 2001)). This provides background dissipation without affecting the order of accuracy of the resulting scheme.

### 4.2.3.2 Switched 2nd and 4th-order differences scheme

The dissipation term in this scheme is composed by a blend of second and fourth order differences (Jameson, Schmidt & Turkel, 1981). Accordingly, the diffusive flux in Eq. (4.53) is expressed as

$$\mathbf{d}_{ij}^{k} = \lambda_{ij}\left[\varepsilon_2\left(\mathbf{U}_j - \mathbf{U}_i\right) - \varepsilon_4\left(\Delta_j - \Delta_i\right)\right]\hat{n}_{ij}^{k} \tag{4.55}$$

where $\lambda_{ij}$ and $\hat{\boldsymbol{n}}_{ij}$ shall be interpreted as defined above, $\varepsilon_2$ and $\varepsilon_4$ are non-linear weights to be determined and $\Delta$ is a difference operator computed by

$$\Delta_i = \sum\nolimits_j \left(\mathbf{U}_j - \mathbf{U}_i\right) \tag{4.56}$$

where the summation includes the cloud nearest (Delaunay) neighbors of $\boldsymbol{x}_\mathrm{i}$. The weights $\varepsilon_2$ and $\varepsilon_4$ can be obtained as

$$\begin{aligned} \varepsilon_2 &= \alpha_2\theta_{ij} \quad , \quad \theta_{ij} = \max(\theta_i, \theta_j) \\ \varepsilon_4 &= \max(0, \alpha_4 - \varepsilon_2) \end{aligned} \tag{4.57}$$

being $\alpha_2$ and $\alpha_4$ user-defined dissipation coefficients and $\theta$ a nodal flow detector. The latter can be evaluated in terms of the nodal pressures as

$$\theta_i = \frac{\left|\sum\nolimits_j \left(p_j - p_i\right)\right|}{\sum\nolimits_j \left(p_j + p_i\right)} \tag{4.58}$$

where index $j$ also extends over the nearest neighbors of $\boldsymbol{x}_\mathrm{i}$.

The flow detector given by Eq. (4.58) is proportional to the curvature of the pressure field and allows introducing diffusion in a selective manner. It takes values near zero if the flow is smooth, and tends to unity around discontinuities in the pressure field. Hence, the non-linear weight $\varepsilon_2$ vanishes if the flow is smooth and only the fourth order contribution remains active. This provides some background dissipation (proportional to $\alpha_4$) without affecting the order of accuracy of the scheme. Contrary, in the presence of discontinuities the flow detector tends to unity. This activates the second-order diffusion term (controlled by $\alpha_2$) and makes the resultant scheme to be first-order. At the same time, the non-linear weight $\varepsilon_4$ is switched-off to avoid counteracting the diffusive effect of the second-order operator. This avoids the appearance of further oscillations and instabilities.

The action of the non-linear weights $\varepsilon_2$ and $\varepsilon_4$ is in some sense similar to that produced by the limited differences in Eq. (4.54). However, this scheme incorporates user-defined dissipation coefficients $\alpha_2$ and $\alpha_4$, which allow a finer control on the amount of dissipation introduced. Furthermore, as nodal gradients are not required in Eq. (4.55), the computational cost is lower.

## 4.3 Time discretization

The time discretization of the semi-discrete scheme (4.18) is carried out explicitly by means of a multi-stage method. Assuming that the vector of conservative variables **U** is known at time $t = t^n$, the right hand side of Eq. (4.18) ($\mathbf{R}_i(\cdot)$) can be calculated for each star point. Then, the solution advances in time according to the following *s-stage* scheme

$$\begin{aligned} \hat{\mathbf{U}}_i^{(0)} &= \hat{\mathbf{U}}_i^n \\ &\vdots \\ \hat{\mathbf{U}}_i^{(m)} &= \hat{\mathbf{U}}_i^{(m-1)} - \alpha_m \Delta \mathrm{t}_i \mathbf{R}_i\left(\mathbf{U}_j^{(m-1)}\right) \\ &\vdots \\ \hat{\mathbf{U}}_i^{n+1} &= \hat{\mathbf{U}}_i^{(s)} \end{aligned} \tag{4.59}$$

where $\Delta t_i$ is an allowable local time increment and $\alpha_m$ are integration coefficients depending on the number of stages employed (*s*). For two, three and four-stages schemes these coefficients can be set as

- 2 stages → $\alpha_1 = 1/2$ and $\alpha_2 = 1.0$
- 3 stages → $\alpha_1 = 3/5$ , $\alpha_2 = 3/5$ and $\alpha_3 = 1.0$
- 4 stages → $\alpha_1 = 1/4$ , $\alpha_2 = 1/3$ , $\alpha_3 = 1/2$ and $\alpha_4 = 1.0$

and other possibilities can be found in (Jameson, 1993).

As can be observed in Eqs. (4.59), the residual vector $\mathbf{R}_i$ is a function of the cloud nodal parameters $\mathbf{U}_j$, which do not coincide with the approximated problem solution ($\hat{\mathbf{U}}_j$) because the FPM does not interpolate nodal data. Therefore, a non-symmetrical linear system must be solved at the end of each time integration stage to recover the nodal parameters. According to Eq. (3.11), this system is

$$\sum\nolimits_{\forall j \in \Omega_i} a_{ij} \mathbf{U}_j = \hat{\mathbf{U}}_i \quad i = 1, n \tag{4.60}$$

where the mass-consistent like matrix resulting from the assembly of the approximation coefficients $a_{ij}$ is diagonally dominant and typically well-conditioned. Hence, its solution can be obtained with little computational cost with a few Gauss-Seidel iterations.

It should be noticed that the dissipation terms in $\mathbf{R}_i$ can be frozen at the first integration stage, thus reducing the computational cost without affecting the accuracy and robustness of the scheme.

### 4.3.1 Stability requirements and time step computation

The time step employed in explicit integration algorithms must be bounded by some stability criterion, which can be obtained, for instance, from a linear stability analysis (e.g. von Neumann analysis). In the numerical computation of conservation laws, such as the Euler equations, the fulfillment of an additional condition known as CFL (Courant-Friedrichs-Lewys) is also required. In short, this condition states that the numerical domain of dependence (the stencil of points involved in $\mathbf{R}_i$) must contain the physical domain of dependence (bounded by the waves of the system in the x-t plane). This statement simply translates into an inequality restricting the maximum distance that any wave can travel in a single time step (Laney, 1988). Thus, the inequality which bounds the time step for linear stability and the CFL condition can be merged and meet simultaneously.

In FPM discretizations, a general stability criterion is difficult to derive because the distribution of points in a cloud can be highly random, and the metric coefficients not only depend on the spatial position of the points, but also on other parameters of the approximation (see Section 3.4). Thus, approximated stability limits inferred from simpler discretizations should be employed (this also happens with most discretization approaches).

In this work, with basis on well-known results for simple regular one-dimensional point distributions, the local time increment is computed by

$$\Delta \mathrm{t}_i = \mathcal{C} \min\left( \frac{\left\| \boldsymbol{l}_{ji} \right\|}{\left| u_i^k \hat{n}_{ij}^k \right| + c_i} \right) \qquad \forall j \in \Omega_i \tag{4.61}$$

where $\hat{\boldsymbol{n}}$ is a unit vector in the direction of the edge $\boldsymbol{l}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ and $c$ is the speed of the sound. The Courant number $\mathcal{C}$ should be generally restricted to take values above the unity, although multi-stage schemes can usually provide an enlarged stability margin.

The adoption of a local time step $\Delta \mathrm{t}_i$ in Eq. (4.59) increases the speed of convergence of the solution to the steady state for stationary problems (the information is propagated faster throughout the domain). However, if the solution of the problem is time dependent, a global time step must be adopted. This global time step is defined as

$$\Delta t_{\mathrm{global}} = \min\left(\Delta t_i\right) \qquad \forall \boldsymbol{x}_i \in \Omega \tag{4.62}$$

### 4.3.2 Implicit residual averaging

In order to increase the convergence speed to steady state solutions, the maximum permissible time step can be enlarged by using implicit residual averaging (Jameson & Baker, 1987). In this method, a Laplacian smoothing is performed on the discrete residuals of Eq. (4.59) to extend the support of the approximation, thus increasing the allowable Courant number. The smoothed residual can be written as

$$\overline{\mathbf{R}}_i = \mathbf{R}_i + \varepsilon \sum_j \left(\overline{\mathbf{R}}_j - \overline{\mathbf{R}}_i\right) \tag{4.63}$$

where $\boldsymbol{\varepsilon}$ is a specified smoothing parameter and the summation is performed over the nearest neighboring points of $\boldsymbol{x}_\mathrm{i}$. An exact solution of the system (4.63) is not generally required for satisfactory convergence acceleration. Hence, an approximated solution of Eq. (4.63) can be found with a little computational cost by performing a few Jacobi iterations, i.e.

$$\overline{\mathbf{R}}_i^{(k+1)} = \frac{\mathbf{R}_i + \varepsilon \sum_j \left(\overline{\mathbf{R}}_j^{(k)} - \overline{\mathbf{R}}_i^{(k)}\right)}{1 + \sum_j 1} \tag{4.64}$$

where $k$ is the iteration counter. In typical applications, satisfactory results can be obtained by performing three passes of the scheme (4.64) with $\varepsilon \approx 0.1$. Furthermore, it is not necessary to smooth residual at each time integration stage. For example, in the case of the 4th-stage scheme, smoothing just the first and third stages is usually enough to double the allowable Courant number. It is important to note that residual smoothing does not affect the problem steady state solution.

## 4.4 Final remarks on the FPM flow solver

Some important aspects regarding the accuracy and conservation properties of the method, as well as the general lines followed in the computational implementation are addressed next. The aim is to provide complementary information about the characteristics of the methodology and the performance expected in practice.

### 4.4.1 Accuracy of the FPM solver

The accuracy of the FPM solver depends mainly on the accuracy of the spatial approximation to the flux derivatives and the accuracy of the numerical flux, that also depends on the reconstruction procedure applied.

Regarding the spatial accuracy of the FPM approximation, it can be shown that the FPM yields second-order accurate first derivatives in regular distributions of points (see Section 3.4), but degrades for irregular distributions. There, the accuracy tends to first-order and the effective value attained depends considerably on the weighting function employed; see for instance (Fischer, 1996) and (Cheng & Cheng, 2008). Therefore, second-order accuracy can be regarded as an upper limit for the flux derivatives.

The accuracy of the numerical flux (Eq. (4.28)) affects the order of accuracy of the scheme because this flux acts as a bias term in the central approximation. Thus, depending on the extrapolation employed in the reconstruction process and the characteristics of the underlying discretization[51], the accuracy of the resulting scheme may change. As an upper bound of the attainable accuracy, it can be assumed that when the flow is smooth and second-order extrapolation (or higher) is adopted, second-order accuracy may be achieved. Note that the scheme reverts to the centered approximation for a vanishing upwind term (see for instance Eq. (4.53))[52].

According to these results, the spatial accuracy ($p$) of the FPM solver can be considered to be at best second-order in smooth flows. In practice, it is expected that $1 < p < 2$ depending on the characteristics of the local clouds and the weighting function settings. A numerical investigation aimed at determining the attainable accuracy in a representative practical application problem is presented in Chapter 5.

### 4.4.2 Conservation

Conservation properties of the FPM solver cannot be formally proved for general application cases. This theoretical drawback, which affects most meshless approaches, is a direct consequence of the lack of symmetry properties of the metric coefficients and their dependence on the distribution

[51] The irregularity of the stencil of points employed in the reconstruction process and the estimates adopted for the virtual points (Eq. (4.48)) can affect the formal order of the reconstruction scheme, even in smooth flows.

[52] (Sridar & Balakrishnan, 2003) found accuracy estimates for a quite similar scheme (but using Taylor-series-based LSQ approximations) to be $O(h^{(l+1)-p})$, for $p$-order flux derivatives and $l$-degree terms retained in the Taylor expansion (assuming $k$-order reconstruction with $k \geq l$). This is consistent with the estimation in the present work.

of points in the local clouds and weighting function settings. These facts make that, for instance, the typical telescopic flux properties cannot be demonstrated in such schemes.

However, despite the lack of proofs guaranteeing the conservation property for meshless schemes at the discrete level, the empirical evidence (which is extensive in the literature) does not reveal the typical problems which are usually attributed to a lack of conservation. This numerical evidence, though important, is not sufficient by itself and the implications of lack of conservation should be studied further. Following this line there are some recent works in which conservation and other desirable properties are enforced by the construction of the approximation metric coefficients; see for instance (Chiu, Wang & Jameson, 2011). It must be stressed, however, that these modified formulations yield solutions which coincide with the non-conservative schemes. It is therefore open to debate if formal conservation is strictly required, and if a lack thereof has a detrimental impact on the accuracy of the meshless numerical solution.

### 4.4.3 Computational implementation and optimization

The methodology described in the previous sections is coded in Fortran 77/90 and parallelized through OpenMP directives. The implementation has been tailored for reduced cache misses during execution, satisfactory parallel performance in multi-core and multi-CPU computers and minimal storage requirements. The reduction of cache-misses is achieved by an efficient reordering of nodes (see Section 3.7) intended to maximize, during the computations, the use of data located as close as possible in memory. The design of the data structures follows this objective and is also aimed to reduce memory contention and indirect addressing in loops. In certain cases, this leads to the duplication of data causing a memory overhead, but the computational efficiency is enhanced noticeably. As regards parallel efficiency, especial emphasis was placed on reducing data dependence. Efficient parallel implementations are easy in the FPM due to its inherent low data dependence. Evidence of this will be given in the studies presented in the next chapter, where additional enhancements of the FPM data structure intended to improve the performance of the method are also proposed.

## 4.5 Boundary conditions

The solution of the Euler equations (4.1) requires the definition of proper initial and boundary conditions in order to complete the description of the problem to be solved.

The initial conditions initialize the flow variables and enable to start the explicit calculation (Eq. (4.59)). Typically, the initial flow values are taken from the far-field state ($\mathbf{U}(\boldsymbol{x},t_0)=\mathbf{U}_\infty$), but particular initial solutions can be adopted in specific time dependent problems.

Regarding boundary conditions, its specification is not trivial and should be done observing the mathematical behavior of the equations. In general, information (waves) entering the computational domain must be prescribed (e.g. according to the far-field state $\mathbf{U}_\infty$) and waves leaving the computational domain must be able to move freely. The propagation of information or waves can be easily analyzed using a characteristic-based approach; therefore studies based on this description of the flow are generally used to apply boundary conditions, see for instance (Hirsch, 1990; Laney, 1988). Procedures in which the mathematical behavior of the flow is not accounted for can lead to the appearance of wave reflections and instabilities at the boundaries, thus affecting the convergence and accuracy of the solution.

In this work, the boundary conditions adopted for inviscid problems are far-field (freestream) conditions on outer boundaries $\Gamma_\infty$ and slip conditions on solid (airtight) boundaries $\Gamma_W$ ($\Gamma=\Gamma_W\cup\Gamma_\infty$). In the case of far-field boundaries, the flux in the normal direction at each point $\boldsymbol{x}_i \in \Gamma_\infty$ is prescribed by solving an approximate Riemann problem between $\mathbf{U}_i$ and the far-field state $\mathbf{U}_\infty$. Over body boundaries, slip wall conditions are applied by cancelling the flux components in the surface normal direction. The procedures followed for applying these conditions are described in the succeeding text.

### 4.5.1 Far-field conditions

Freestream conditions are enforced at each point $\boldsymbol{x}_i \in \Gamma_\infty$ by applying a corrected normal flux that accounts for the exchange of information with the far-field. This flux is obtained from the solution of a Riemann problem between the states $\mathbf{U}_i$ and $\mathbf{U}_\infty$. Using the Roe solver, this is calculated by

$$\mathbf{F}^*_{\hat{n}_i} = \tfrac{1}{2}\left(\mathbf{F}_{\hat{n}_i}(\mathbf{U}_\infty)+\mathbf{F}_{\hat{n}_i}(\mathbf{U}_i)\right) - \tfrac{1}{2}\left|\mathbf{A}_{\hat{n}_i}(\mathbf{U}_\infty,\mathbf{U}_i)\right|\left(\mathbf{U}_i-\mathbf{U}_\infty\right) \quad (4.65)$$

where the matrix $|\mathbf{A}_n(\mathbf{U}_\infty,\mathbf{U}_i)|$ is the positive Roe matrix computed in the direction of a unit boundary normal vector $\hat{\boldsymbol{n}}_i$[53] (pointing inside the fluid domain). The fluxes in the same direction are given by

$$\mathbf{F}_{\hat{n}_i}(\mathbf{U}) = \mathbf{F}^k(\mathbf{U})\,\hat{n}_i^k \quad (4.66)$$

[53] The normal vector at a point $\boldsymbol{x}_i$ is computed as an arithmetic average of the normals of the boundary elements sharing the point.

where **U** stands for freestream or point valued state conditions. Then, the convective fluxes in the different coordinate directions can be modified to account for the normal flux (4.65) according to

$$\bar{\mathbf{F}}^k(\mathbf{U}_i) = \mathbf{F}^k(\mathbf{U}_i) - \left(\mathbf{F}_{\hat{n}_i}(\mathbf{U}_i) - \mathbf{F}^*_{\hat{n}_i}\right)\hat{n}_i^k \tag{4.67}$$

where the overbar indicates corrected values. The fluxes (4.67) are applied on outer-boundary nodes in the solution process and there is no need to enforce any value on the respective state vectors $\mathbf{U}_\mathrm{i}$. This kind of boundary condition notably reduces the boundary perturbations (particularly if the analysis domain is not large enough) and avoids wave reflections. Furthermore, this method simplifies the boundary treatment because no distinction has to be made between inflow/outflow or subsonic/supersonic boundaries when generating the problem input files.

### 4.5.2 Slip and symmetry wall conditions

Slip and symmetry wall conditions are enforced by cancelling the boundary normal flux at each point $\boldsymbol{x}_\mathrm{i} \in \Gamma_\mathrm{w}$. This can be achieved by setting

$$u_i^k \hat{n}_i^k = 0 \tag{4.68}$$

where $\hat{\boldsymbol{n}}_i$ and $\boldsymbol{u}_i$ are the boundary normal and fluid velocity vectors, respectively. The condition (4.68) can be enforced at each time step in the conservative variables by setting[54]

$$\bar{\mathbf{U}}_i^{k+1} = \mathbf{U}_i^{k+1} - \left(\mathbf{U}_i^{k+1}\hat{n}_i^k\right)\hat{n}_i^k \quad \text{for } k = 1,3 \tag{4.69}$$

In problems in which the initial condition is far from the problem solution (particularly when high freestream Mach numbers or not streamlined bodies are involved), the application of condition (4.68) can lead to some instability during the initial time steps of the simulation. This behavior can be improved by relaxing the condition (4.68) through the introduction of a parameter κ ∈ [0,1] in Eq. (4.69) (Lyra & Morgan, 2002). This leads to

$$\bar{\mathbf{U}}_i^{k+1} = \mathbf{U}_i^{k+1} - \kappa\left(\mathbf{U}_i^{k+1}\hat{n}_i^k\right)\hat{n}_i^k \quad \text{for}\, k = 1,3 \tag{4.70}$$

[54] At solid boundaries, all the convective flux components through the surface vanish and only the pressure contribution remains. Thus, this condition on the fluxes can be used to enforce slip boundaries instead of using condition (4.68) on the velocity. Both alternatives have been studied in this work but no important differences were found. Hence, the application of Eq. (4.69) on the conservative variables was chosen as this simplifies the calculations (corrected velocities may also be needed for other uses during execution). In addition, the condition on fluxes may not ensure the fulfillment of Eq. (4.68) in an exact manner.

where the parameter κ is varied from zero at the beginning of the computation to unity over a certain number of time steps. This allows applying the solid boundary condition progressively as the near-body solution tends to match smoothly the conditions at the boundary[55].

It is important to note that the application of solid wall conditions can present problems on certain geometric features (e.g. sharp corners) where the surface normal is not well-defined. There, two cases must be distinguished. If the solid surface is concave at the edge, the condition (4.68) can be replaced by

$$\bar{\mathbf{U}}_i^{k+1} = \left(\mathbf{U}_i^{k+1} \hat{t}_i^k\right) \hat{t}_i^k \quad \text{for } k = 1,3 \tag{4.71}$$

which forces the velocity vector to be parallel to the edge ($\hat{\boldsymbol{t}}_i$ is the edge tangent direction). If the surface is convex (e.g. trailing edges), the direction of the velocity is not known a-priori and thus cannot be prescribed. In such cases, an alternative procedure is to let the velocity move freely (not subject to any particular condition). This approach is discussed further below.

### 4.5.3 Treatment of trailing edge points

The enforcing of wall conditions on points along convex sharp edges (such as the trailing edge of airfoils and wings), presents a problematic situation because the direction of the velocity is not known along these edges. Additionally, another problem must be faced in the FPM regarding the construction of proper clouds for these points. Note that in such cases, the application of the criteria discussed before in Section 3.6 can lead to distorted asymmetrical clouds of points which may violate the CFL condition (it is quite possible that the physical domain of dependence is not totally included in the cloud of points). As a consequence, the proper propagation of information through the domain can be affected, and thus numerical instabilities may appear. In order to fix these problems, the following approach is proposed.

In relation to the construction of clouds for trailing edge points, no boundary restrictions are applied to the points in these clouds. Thus, the resulting cloud structure is generally symmetrical and only depends upon the distribution of neighboring points (see Figure 16). Moreover, no condition is prescribed on the velocity or fluxes at these points, i.e. the flow variables are allowed to move freely. In this way, any point located along a trailing edge is considered like a point in the interior fluid domain. Numerical experiments performed show that this treatment allows the trailing edge points to automatically adapt their behavior according to the local solution field. Moreover, this permits the

[55] Note that Eq. (4.70) can be also used to set transpiration boundary conditions. There, κ may be a constant or a solution dependent parameter.

fluid to leave the trailing edge smoothly (in accordance with Kutta conditions) avoiding instabilities and non-physical numerical solutions. If the discretization is not very coarse, a satisfactory approximation to rear stagnation points can be also obtained.

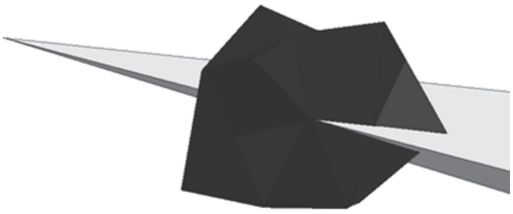

Figure 16. Cloud of points for a star point located at the trailing edge of a wing.

## 4.6 Application examples

Several compressible flow calculations are presented in this section with the aim of illustrating the performance of the FPM flow solver. The first examples, presented in Sections 4.6.1 and 4.6.2, are typical one and two-dimensional verification test cases. Next, three-dimensional calculations are presented in Section 4.6.3 and the results are compared with available experimental data. These examples are intended to demonstrate the applicability of the method in more realistic situations.

### 4.6.1 Shock tube problem

The shock tube problem is a one-dimensional Riemann problem proposed by Sod in 1978 (Sod, 1978). The fact that this problem has both analytical and experimental solution makes it one of the most popular benchmarks for numerical schemes. The problem consists of a closed tube divided by a diaphragm into two compartments (left and right). Each compartment contains a gas at rest with a given pressure and density. The simulation begins when the diaphragm is suddenly removed. At this point, the interaction between the left and right fluid states instantly generates a shock wave, a contact discontinuity and an expansion fan, which propagate within the tube.

This problem is solved firstly in a one-dimensional closed domain Ω = (0,1) subject to the following initial conditions

$$\mathbf{U}\left(\mathrm{x},\mathrm{t}_0\right)=\begin{cases}\mathbf{U}_\mathrm{L}=\left(1,0,2.5\right)^\mathrm{T} & x\leq 0.5\\ \mathbf{U}_\mathrm{R}=\left(0.125,0,0.025\right)^\mathrm{T} & x>0.5\end{cases}\tag{4.72}$$

which give a pressure ratio across the diaphragm $p_L/p_R = 10$. Accordingly, the intensity of the shock is moderate and the flow regime after the expansion is subsonic. The computational domain is discretized by a regular distribution of 100 points and quadratic approximation bases are used with clouds having 5 points. Third-order MUSCL extrapolation is adopted in conjunction with the Van Albada limiter and the solution is advanced in time by means of a four-stage integration scheme. The results, computed for a time t = 0.2 seconds after the rupture of the diaphragm, are compared with analytical results (see for instance (Laney, 1988)) in Figure 17.

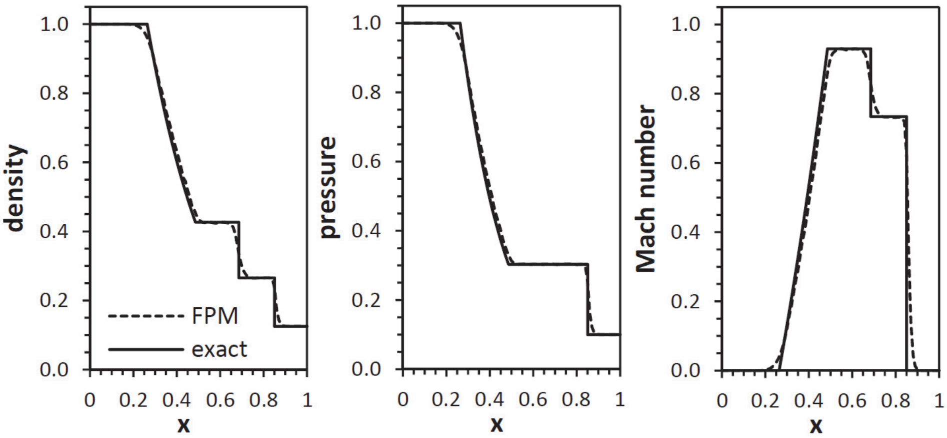

Figure 17. Comparison of FPM and exact solutions. One-dimensional shock tube problem $p_L/p_R = 10$ and t = 0.2 s. ($n$ = 100).

Next, the shock tube problem is solved for an initial pressure ratio across the diaphragm $p_L/p_R$ = 100. In this case, supersonic flow is obtained after the expansion. The initial conditions are given by

$$\mathbf{U}(\mathrm{x}, \mathrm{t}_0) = \begin{cases} \mathbf{U}_\mathrm{L} = (1, 0, 2.5)^\mathrm{T} & x \leq 0.5 \\ \mathbf{U}_\mathrm{R} = (0.125, 0, 0.0025)^\mathrm{T} & x > 0.5 \end{cases} \tag{4.73}$$

The analysis domain is discretized by a regular distribution of 200 points and, like in the previous test case, quadratic bases in clouds of 5 points are employed. Third-order MUSCL extrapolation is also adopted, but using the minmod limiter. This helps to smooth some small oscillations that appear with the van Albada limiter without affecting the accuracy of the computations. Exact and numerical solutions for a time t = 0.15 seconds after the rupture of the diaphragm are compared in Figure 18.

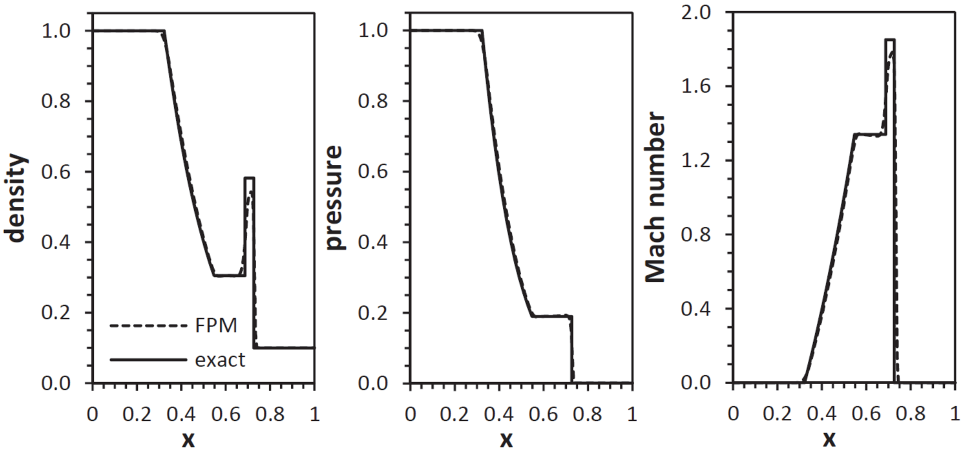

Figure 18. Comparison of FPM and exact solutions. One-dimensional shock tube problem $p_L/p_R = 100$ and t = 0.15 s. ($n = 200$).

An additional two-dimensional computation is presented next for the initial conditions defined in Eq. (4.72). The problem domain $\Omega = (0,1)x(0,1)$ is discretized by a structured distribution of 100 points in each spatial direction. Quadratic approximation bases are used with clouds having 15 points and slip-wall conditions are applied along the boundaries. The time integration is performed by a four-stage scheme, and a third-order MUSCL extrapolation with the Van Albada limiter is chosen. The numerical results are compared with the exact one-dimensional solution in Figure 19. As in the previous examples, a satisfactory agreement between the computed FPM and exact solutions is achieved.

### 4.6.2 Airfoil test cases

A subsonic flow around a NACA 0012 airfoil set at zero incidence angle ($\alpha = 0^o$) and freestream Mach number $M_\infty = 0.2$ is solved firstly to investigate the accuracy of the numerical scheme. To this end, the convergence of the drag coefficient with the grid refinement is investigated.

In order to carry out the analysis, four model discretizations with representative point spacing $h_1<h_2<h_3<h_4$ are generated from a coarse model ($h_4$) by scaling the grid sizes in a uniform manner. In this way, discretizations ranging from 3364 to 7162 points are obtained with an average refinement ratio $r = h_{k+1}/h_k \approx (n_{k+1}/n_k)^{-1/2} = 1.13$ (std. deviation 0.02). The boundary conditions applied are freestream Riemann conditions on the outer domain boundary and slip condition on the airfoil contour. The numerical solutions are obtained by using third-order MUSCL extrapolation with van Albada limiter and a 4-stages time marching scheme. As regards the FPM discretization, quadratic approximation bases are employed in clouds with 15-21 points. A view of the finest discretization (grid #1) near the airfoil is presented in Figure 20.

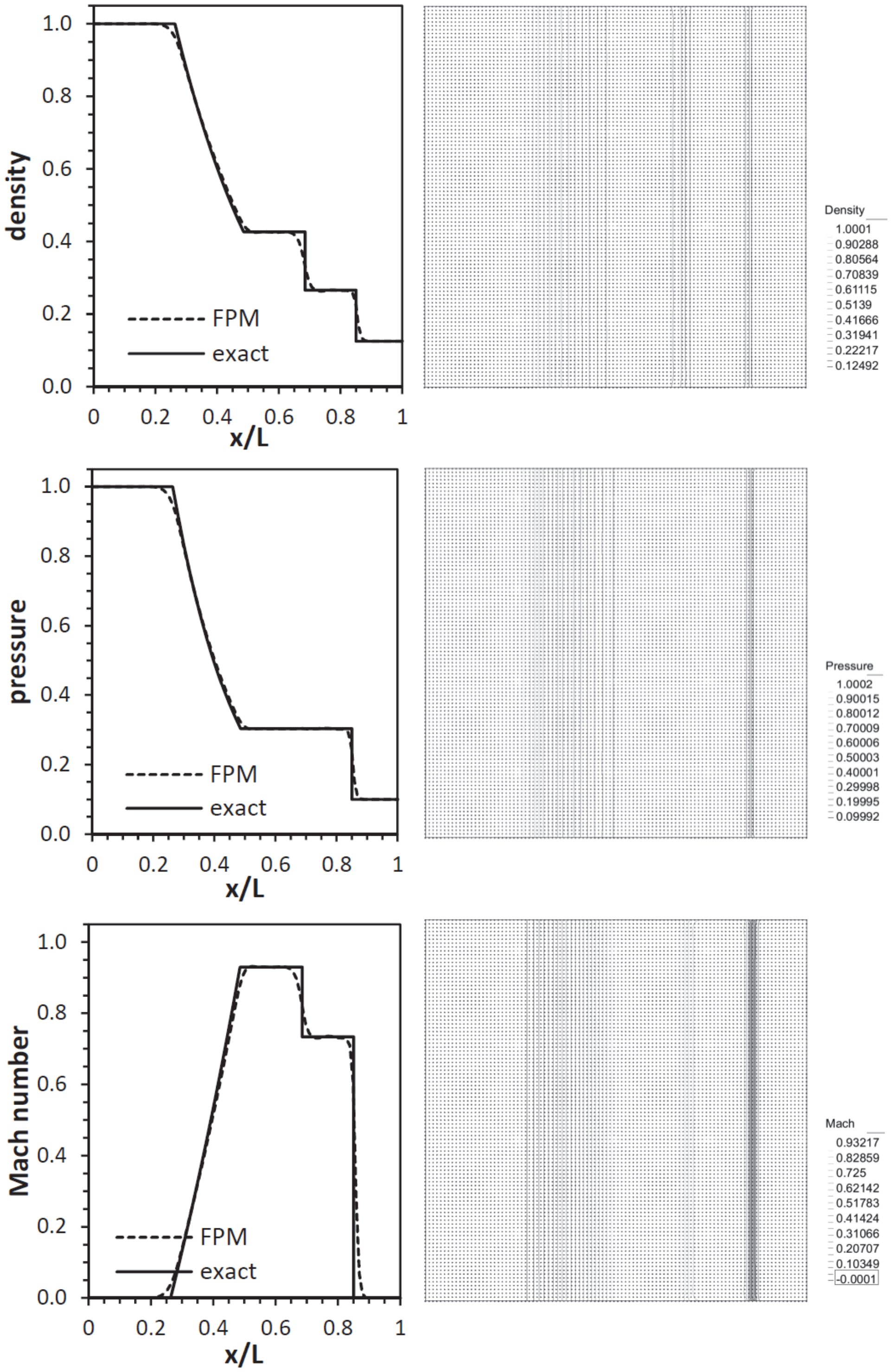

Figure 19. Two-dimensional simulation of the shock tube problem ($p_L/p_R = 10$ and t = 0.2 s.). Density, pressure and Mach number isolines are shown on the right.

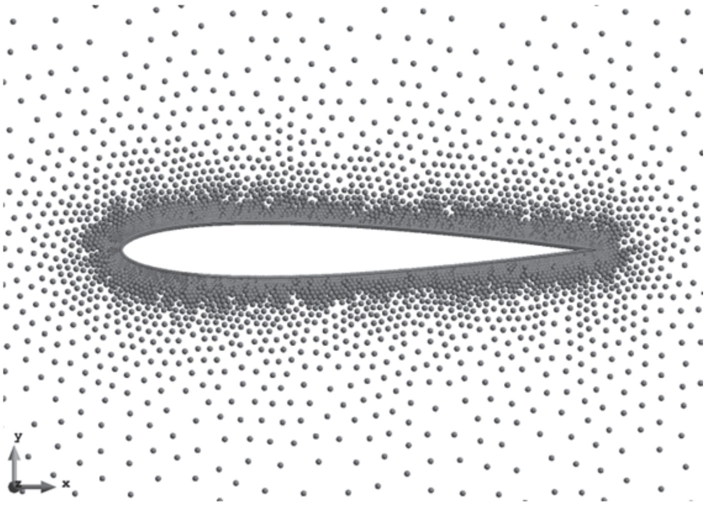

Figure 20. Spatial discretization in the proximity of the airfoil NACA 0012 (n=7162).

The evolution of the drag error with the grid refinement is examined first in order to assess the observed accuracy of the scheme (mainly discretization error). Note that since the drag in a subsonic inviscid flow is zero, the drag error in this example is equal to the calculated drag. The results presented in Figure 21 indicate a convergence rate p = 1.72, which is not far from the theoretical order of accuracy of the scheme. The solution on the coarsest grid is dropped because it is outside the asymptotic range of convergence.

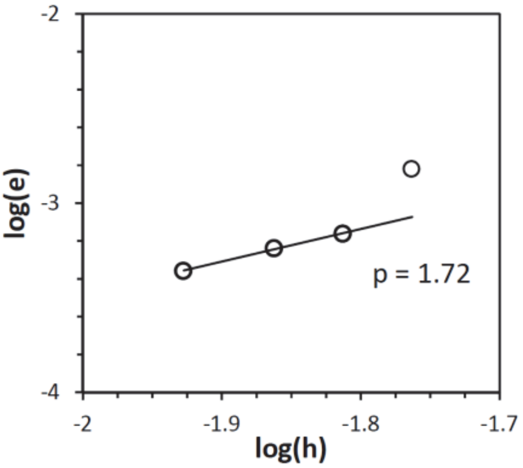

Figure 21. Drag convergence. NACA 0012, $\alpha = 0°$ and $M_\infty = 0.2$.

The convergence of the aerodynamic forces and density residual on the finest grid is depicted in Figure 22. There, the calculated aerodynamic forces can be seen as actual errors in forces (symmetric airfoil in subsonic inviscid flow). The density residual is evaluated by means of a quadratic (L2) norm. Next, Figure 23 shows a comparison between the computed coefficient of pressure (Cp) along the airfoil with potential flow results (corrected by compressibility using the Karman-Tsien rule). The agreement obtained is satisfactory.

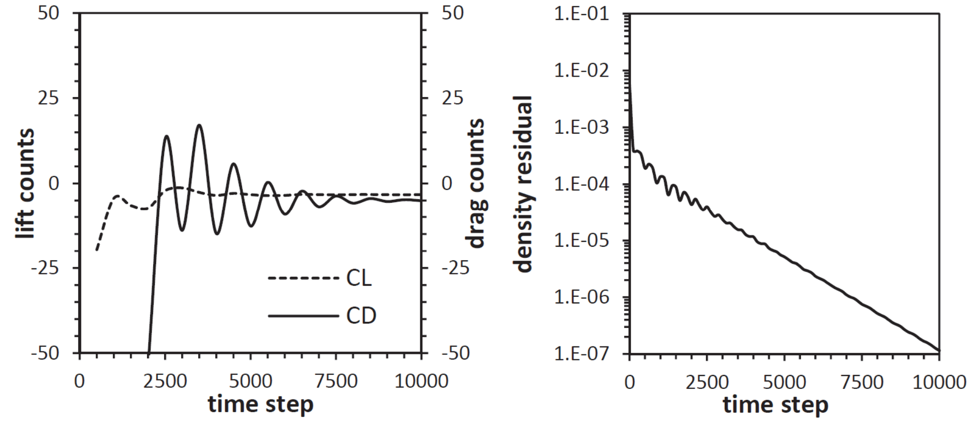

Figure 22. Iterative convergence for the finest grid solution (grid #1). NACA 0012, α = 0º and M$_\infty$ = 0.2.

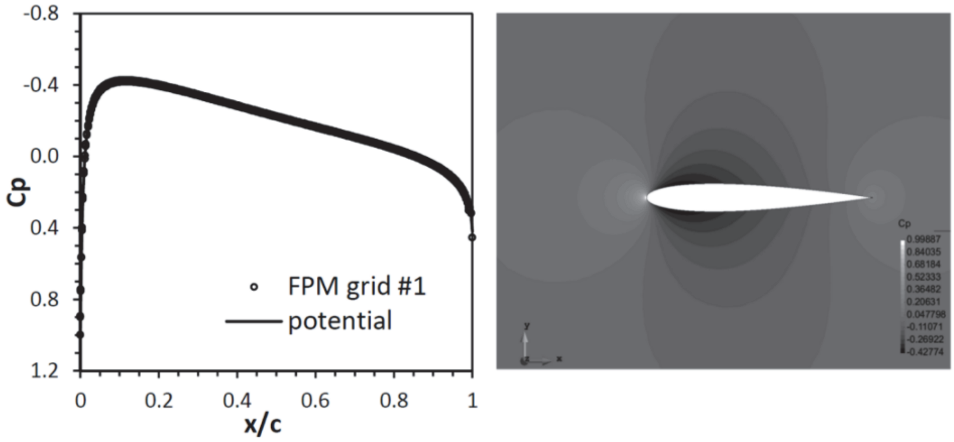

Figure 23. Comparison of pressure distribution computed on grid #1 with potential flow results. NACA 0012, α = 0º and M$_\infty$ = 0.2.

The flow field around a RAE 2822 airfoil is solved next. The airfoil is set at an incidence angle α = 3º and the freestream Mach number is M$_\infty$ = 0.75. The computational domain is discretized by an unstructured distribution of 7126 points (see Figure 24) and the boundary conditions and problem settings are similar to those employed in the previous airfoil example.

The pressure distribution computed along the airfoil is compared in Figure 25 with a reference numerical solution presented in (Rizzi & Viviand, 1981). This result was obtained using an unstructured finite volume method with a mesh having 5071 points. Despite some small oscillations on the leading edge suction side observed in the FPM solution, the general agreement with the reference results is satisfactory. The convergence of the aerodynamic forces and density residual is presented in Figure 26.

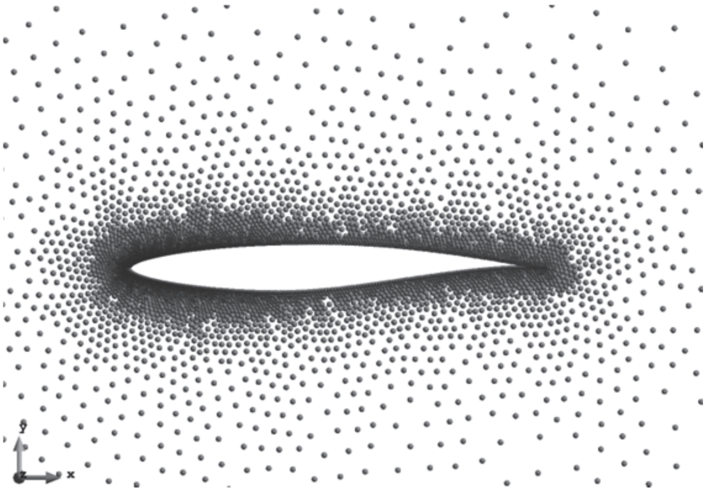

Figure 24. Spatial discretization in the near-field of the airfoil RAE 2822 (n=7126).

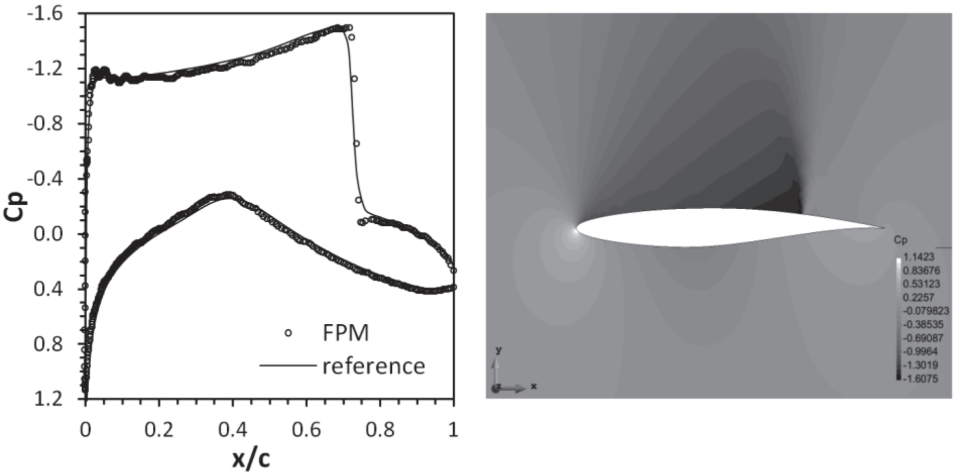

Figure 25. Comparison of pressure distribution along the airfoil. RAE 2822, α = 3º and $M_\infty = 0.75$.

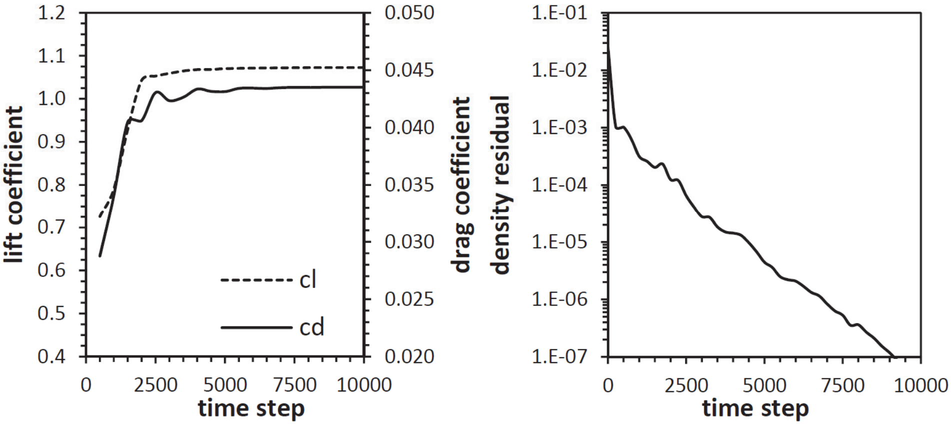

Figure 26. Convergence of the computed lift and drag (wave) forces and the density residual. RAE 2822, α = 3º and $M_\infty = 0.75$.

### 4.6.3 Three-dimensional test cases

The first three-dimensional test case involves transonic flow over the ONERA M6 wing. This is a typical benchmark developed by the ONERA Aerodynamics Department in 1972 with the objective to provide experimental support for numerical studies (see (Schmitt & Charpin, 1979)). The available experimental results cover a wide range of subsonic and transonic flows. In this example the test case #2308 is chosen. The model is set to an incidence angle $\alpha = 3.06^o$, the freestream Mach number is $M_\infty = 0.84$ and the Reynolds number is Re = 11.7E6 (an inviscid solution is computed here). The most relevant data about this test case can be also found in (NASA, 2013).

The computational model employed in this example includes a symmetry vertical plane along the wing centerline and a hemispherical outer freestream boundary, located 10 half-spans away from the wing. An *h-layer* of points is located inside the domain to force a slow variation of the point spacing near the wing (see Section 3.6.1). The discrete model generated has 512141 points. Quadratic approximation bases are employed for computing the FPM approximation in clouds having 30-45 points. The flow solver uses a third-order MUSCL extrapolation scheme with the van Albada limiter and a three-stage time integration scheme is employed.

Calculated Cp and Mach number fields on the wing and the symmetry plane are shown in Figure 37. The point discretization on the wing is also displayed.

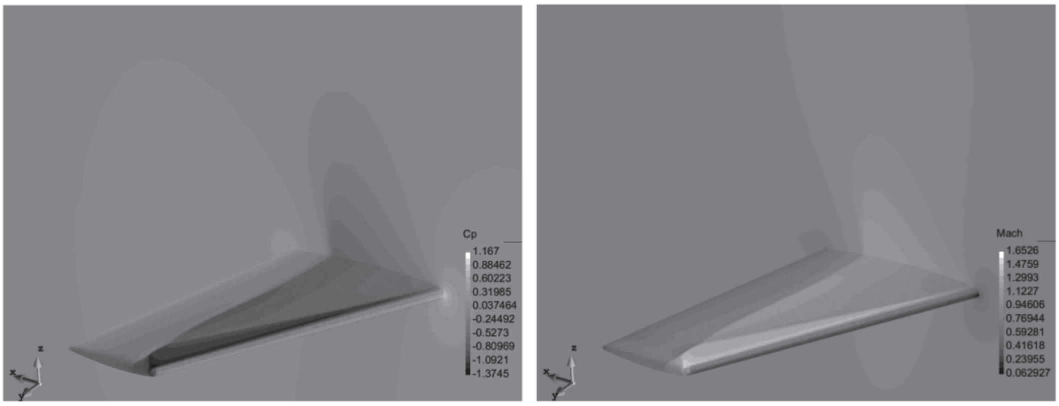

Figure 27. Pressure and Mach number on the upper surface of the wing and the symmetry plane. ONERA M6 wing, $M_\infty = 0.84$ and $\alpha = 3.06^o$.

A comparison between numerical and experimental Cp distributions at several sections along the wing is shown in Figure 28. In accordance with the available experimental data (Schmitt & Charpin, 1979), these sections are located at the following spanwise stations: $\eta = 0.2, 0.44, 0.65, 0.8, 0.9, 0.95$ and 0.99 being $\eta = 2y/b$. The agreement with the experimental data achieved (Figure 28) is satisfactory for the flow model adopted in the simulation. As expected, the inviscid solution gives a shock wave which is slightly stronger than the real shock and is located close behind the latter. A separated flow region behind the shock wave at η=0.99 (upper side of the wing) can also be observed. There, naturally, experimental and computed results do not match. The evolution of the normal force and density residual is shown in Figure 29.

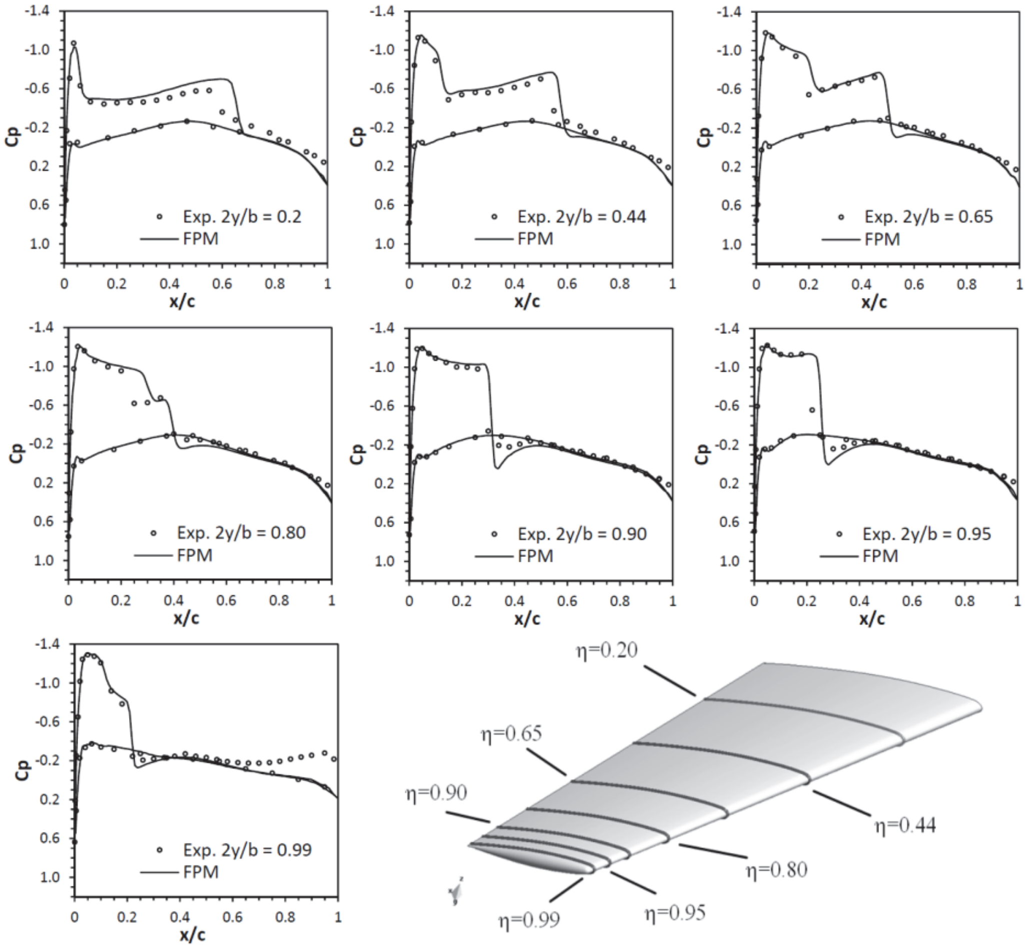

Figure 28. Comparisons between computed and experimental Cp distributions along several sections on the wing. ONERA M6 wing, $M_\infty$=0.84 and α=3.06º.

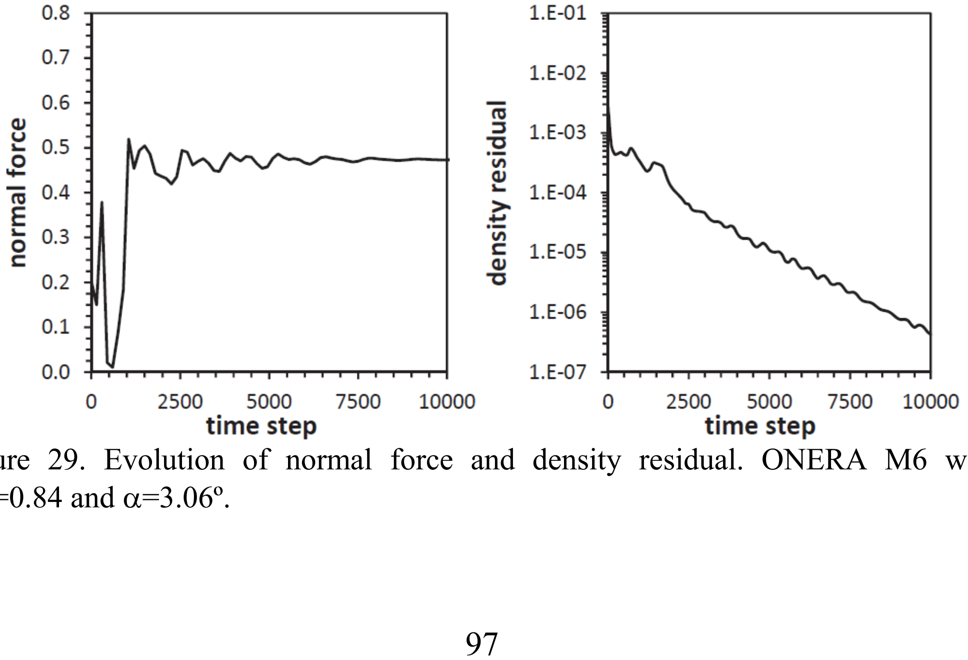

Figure 29. Evolution of normal force and density residual. ONERA M6 wing, $M_\infty$=0.84 and α=3.06º.

The next example involves the computation of transonic flow over a NACA L51F07 wing-body configuration (Loving & Estabrooks, 1951). The model has a sweepback wing, with NACA 65A006 airfoil, and axially symmetric fuselage. Its rear part is attached to a sting, which supports the test model in the wind tunnel. Taking advantage of geometry and flow symmetry, only a half-span body is solved in this example (see Figure 30).

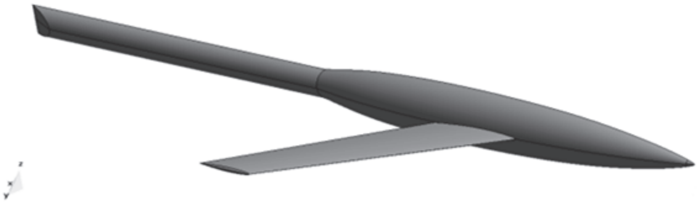

Figure 30. Analysis test model NACA RM L51F07.

The discretization employed consists of an unstructured distribution of 512553 points and quadratic approximations are built on clouds with 30-45 points. The flow solver settings are similar to that employed in the previous example. The pressure distribution computed for a freestream Mach number $M_\infty = 0.9$ and an angle of attack $\alpha = 4^o$ is presented in Figure 31.

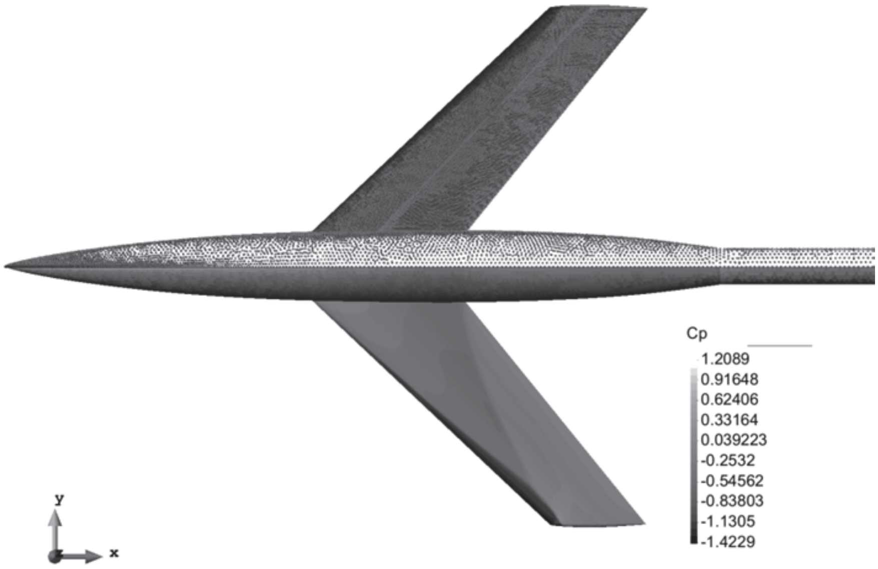

Figure 31. Pressure contours on the NACA wing-body (the surface discretization is displayed in the upper mirror image). $M_\infty = 0.90$ and $\alpha = 4.0^o$.

Comparisons with experimental results provided in (Loving & Estabrooks, 1951) are displayed for two streamwise wing stations located at $\eta = 0.4$ and 0.8 in Figure 32. The longitudinal pressure distribution along the fuselage symmetry plane is also compared in Figure 33. Like in the previous test case, only minor differences (probably due to the inviscid assumption adopted) can be observed between numerical and experimental results. This test case is further studied in the next chapter, where a detailed assessment of the method is presented.

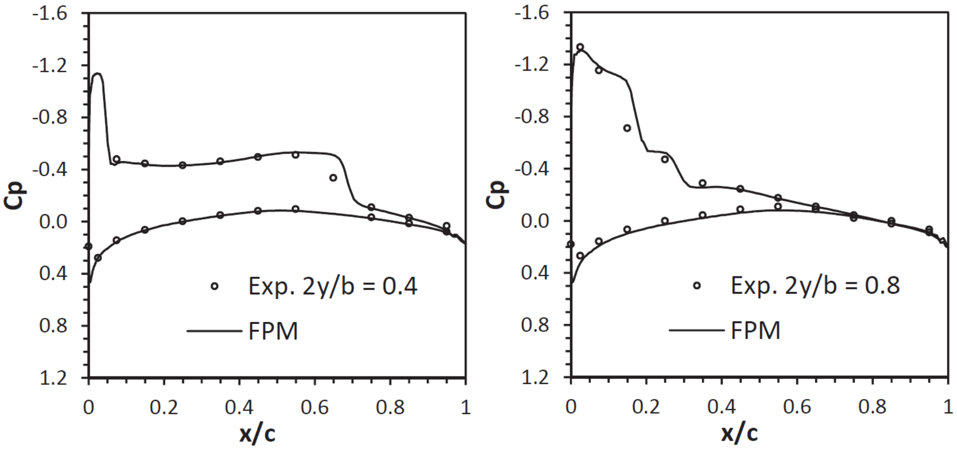

Figure 32. Comparison between computed and experimental Cp distribution along spanwise stations 2y/b = 0.40 and 0.8. NACA wing body, $M_\infty = 0.90$ and $\alpha = 4.0º$.

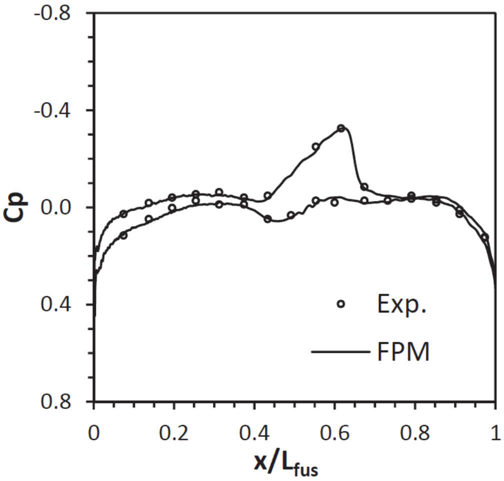

Figure 33. Comparison between computed and experimental Cp distribution along the fuselage symmetry plane. NACA wing-body configuration, $M_\infty = 0.90$ and $\alpha = 4.0º$.

## 4.7 Concluding remarks

A Finite Point method for solving the inviscid compressible flow equations has been presented in this chapter and theoretical and implementation aspects were discussed. Basically, the flow solver proposed is based on a central-type differencing scheme, biased by the introduction of an upwind numerical flux. The latter is obtained by the approximated Riemann solver of Roe, and limited MUSCL extrapolation is used to increase its spatial accuracy. Two additional stabilization schemes, based on added artificial diffusion, were also introduced to allow faster computations. The time integration of the resulting semi-discrete equations is performed explicitly by means of a multi-stage scheme and a residual smoothing technique is introduced for convergence acceleration purposes.

The computational procedures and code implementation have been developed with a focus on efficiency, to ensure that the present solution approach is adequate to address practical problems involving arbitrary geometries. The overall methodology, which follows standard techniques in computational gasdynamics, yielded a robust and accurate core solver, which constitutes a satisfactory basis for implementing enhanced solution approaches and flow models.

Several numerical applications involving typical verification and validation test cases have also been presented in this chapter in order to illustrate the performance of the methodology. The analyses presented have been intended primarily to contrast the FPM computations with exact and experimental solutions, and a basic accuracy assessment was also performed for the two-dimensional case. The results obtained in the test cases presented were satisfactory but not complete. More specific analyses focused on accuracy and computational cost of the basic flow solution scheme, and various extensions and improvements of the latter, are developed further in the next chapters.

# 5 ACCURACY AND PERFORMANCE ASSESSMENT

The field of applications of meshless methods has expanded considerably in the last years and many advantages over conventional discretization approaches have been revealed. However, while efforts have mostly focused on new and improved developments, little progress has been done in comparative studies with conventional techniques to evaluate accuracy, numerical implementation issues, computational efficiency, robustness and other aspects of practical interest. As a consequence, further research is needed to characterize the behavior of meshless methods in practical application scenarios.

With this in mind, the objective of this chapter is twofold. First, to complete the results obtained in the previous chapter with a more focused analysis of the accuracy and computational cost of the FPM flow solver (in the context of a practical application problem). Second, to carry out the assessment in comparison with a conventional technique in order to have a reference to evaluate the feasibility and competitiveness of the meshless approach. Note that competitiveness is not a minor issue; very often the complexity, low efficiency and lack of robustness found in meshless implementations negate the specific advantages of the approaches (e.g. simplified model preparation, easy implementation of adaptivity and domain deformation, etc.). Therefore, a satisfactory performance of the basic technique must be achieved so that meshless advantages can be exploited efficiently. With this purpose, a simplification of the FPM scheme is also proposed in this chapter to reduce the performance gap with respect to conventional grid-based algorithms.

The conventional technique employed in the comparative assessment is a classical Finite Element (FEM) unstructured edge-based solver named PUMI (Flores, Ortega & Oñate, 2011). This is chosen because, regardless of the intrinsic differences due to the spatial discretization approaches, PUMI and the FPM solver follow very similar solution strategies. This allows a more direct comparison of the core algorithm properties eliminating the effects arising from specific implementation choices. The analysis performed are based on the solution of an inviscid three-dimensional compressible flow problem (representative of practical applications) and focuses, primarily, on evaluating the accuracy, computational cost and parallel performance of the FPM.

This chapter is organized as follows. First, the solution methodologies adopted in the FPM and PUMI solvers are discussed and compared in Section 5.1, and a modification of the meshless technique, aimed to improve its efficiency, is presented in Section 5.2. Second, a comparative assessment is carried out in Section 5.3 to evaluate the solution accuracy, computational cost and parallel performance of the FPM in relation to the FEM solver. Finally, the suitability of the reduced-fidelity stabilization models proposed in the previous chapter is examined in comparison with the basic (upwind) flow solution approach in Section 5.4.

## 5.1 Flow solution approach in FPM and FEM

The strategies adopted to solve the flow equations in FPM and the FEM code PUMI follow similar lines. However, the numerical implementations differ as a consequence of specific features of the spatial discretizations. These differences bring about advantages and disadvantages of one method with respect to the other concerning both, the computational requirements and the properties of the discrete schemes. Both solution approaches are compared in this section in order to identify meshless implementation issues affecting the FPM competitiveness.

The inviscid flow governing equations and the FPM solution approach have been discussed in detail in the previous chapter. Next, the basic aspects of the FEM solver are briefly presented in order to introduce the necessary elements for the analysis and comparison of the flow solution strategies.

### 5.1.1 Basic aspects of the FEM approach

Starting from the weak Galerkin form of the governing equations (4.1), the semi-discrete problem is expressed in PUMI in terms of an edge-based data structure as follows

$$\sum\nolimits_j \mathbf{M}_{ij} \frac{\mathrm{d}\mathbf{U}_j}{\mathrm{dt}} = -\sum_e \left( d_{ij}^k \mathbf{F}_{ij}^k + c_i^k \mathbf{F}_i^k \right) \tag{5.1}$$

where $\mathbf{M}$ is the mass matrix of the system, j denotes nodal values, $\mathrm{d}^{\mathrm{k}}_{\mathrm{ij}}$ and $\mathrm{c}^{\mathrm{k}}_{\mathrm{i}}$ are metric coefficients, $\mathbf{F}_{ij}^k = \mathbf{F}_i^k + \mathbf{F}_j^k$ is a numerical flux function and the summation $e$ extends over all the physical edges of the mesh sharing the point $\boldsymbol{x}_{\mathrm{i}}$. At each edge $\boldsymbol{x}_{\mathrm{j}}$-$\boldsymbol{x}_{\mathrm{i}}$, the metric coefficients are computed in terms of the standard FE shape functions (linear tetrahedra); cf. (Flores, Ortega & Oñate, 2011).

It is important to note that only half of the ***d*** coefficients need to be stored because these coefficients are antisymmetric ($d^k_{ij}$=-$d^k_{ji}$). Hence, the storage requirements and the number of operations required to assemble Eq. (5.1) are noticeably reduced. Moreover, as a consequence of the symmetry properties of the metric coefficients, discrete conservation can be easily demonstrated in this kind of schemes through telescopic collapse of the fluxes.

#### 5.1.1.1 Convective stabilization

The definition of the numerical flux in PUMI also adopts the approximate Riemann solver of Roe and slope-limited MUSCL reconstruction of the flow variables along the edges. The formulation is similar to that employed for the meshless solver (see Sections 4.2.1 and 4.2.2). However, as nodal gradients are not available in the present case, a recovery of the nodal derivatives is required in Eq. (4.48); cf. (Flores, Ortega & Oñate, 2011).

#### 5.1.1.2 Time integration

The time integration in PUMI is performed by means of a multi-stage scheme similar to that described by Eq. (4.59). Aimed at avoiding the solution of a system of equations at each stage, the consistent mass matrix is replaced by its lumped counterpart. Local time stepping and residual smoothing are also employed in PUMI to accelerate the convergence to the steady state. The diffusion terms in Eq. (4.59) are likewise only updated at the first integration stage in order to reduce the computational cost.

### 5.1.2 Comparative summary

The main characteristics as well as the advantages (↑) and disadvantages (↓) of the FPM and PUMI solution approaches are summarized in Table 2.

<table>
<tr><th></th><th colspan="2">FPM</th><th colspan="2">PUMI</th></tr>
<tr><td rowspan="2"><b>Spatial discretization</b></td><td></td><td>Set of points obtained from a boundary triangulation by using an unconstrained Delaunay gridding technique.</td><td></td><td>FE tetrahedral mesh.</td></tr>
<tr><td>↑</td><td>Point discretizations can be obtained efficiently by using dedicated algorithms.</td><td>↓</td><td>Topological and quality mesh requirements make domain discretization costlier.</td></tr>
<tr><td><b>Flow governing equations</b></td><td colspan="4">Inviscid compressible flow (Euler equations)</td></tr>
<tr><td><b>Discrete equations</b></td><td colspan="2">Obtained from the strong form of the balance equations through point collocation.</td><td colspan="2">Obtained from the weak Galerkin form of the balance equations.</td></tr>
<tr><td rowspan="2"><b>Numerical approximation</b></td><td colspan="2">Weighted Least-Squares on clouds of points (complete polynomial bases of order 2 are employed).</td><td colspan="2">Standard FEM linear approximation on tetrahedra.</td></tr>
<tr><td>↓</td><td>Cloud generation increases the complexity when constructing the approximation coefficients.</td><td>↑</td><td>The construction of the approximation coefficients has a lower computational cost.</td></tr>
<tr><td rowspan="3"><b>Equations assembly and final discrete forms</b></td><td>↓</td><td>A higher connectivity per node is required.</td><td>↑</td><td>Lower connectivity per node.</td></tr>
<tr><td>↓</td><td>The lack of symmetry properties of the metric coefficients implies higher storage and number of operations per node to compute gradients.</td><td>↑</td><td>Edge coefficients are antisymmetric. Storage requirements and number of operations is reduced to half.</td></tr>
<tr><td>↓</td><td>Conservation at the discrete level cannot be demonstrated due to the properties of the metric coefficients.</td><td>↑</td><td>Conservation at the discrete level is corroborated due to the antisymmetric properties of the edge coefficients (telescopic flux properties).</td></tr>
<tr><td rowspan="2"><b>Convective stabilization</b></td><td colspan="4">Upwind fluxes computed using a Roe solver with slope-limited MUSCL extrapolation</td></tr>
<tr><td>↑</td><td>Nodal gradients are available.</td><td>↓</td><td>Nodal gradients must be obtained by recovery.</td></tr>
<tr><td rowspan="3"><b>Time integration</b></td><td colspan="4">Explicit, Runge-Kutta type, time marching scheme.</td></tr>
<tr><td>↓</td><td>A consistent-mass like system should be solved at the end of each stage (see Eq. (4.60))</td><td>↑</td><td>The consistent-mass matrix is replaced by its diagonal counterpart to avoid solving a system.</td></tr>
<tr><td colspan="4">Convergence acceleration is achieved by local time stepping and residual smoothing techniques.</td></tr>
<tr><td><b>Spatial accuracy</b></td><td colspan="2">The expected order of accuracy is p≤2 for smooth flow solutions (but with an additional dependence on the local cloud parameters).</td><td colspan="2">Second-order accuracy (p=2) can be expected for smooth flow solutions.</td></tr>
<tr><td><b>Time accuracy</b></td><td colspan="4">Dependent on the number of time integration stages employed.</td></tr>
<tr><td rowspan="3"><b>Coding</b></td><td colspan="4">Fortran 77/90 parallelized through OpenMP directives.</td></tr>
<tr><td>↑</td><td>Due to low data dependence, loops can be easily parallelized with efficiency.</td><td>↓</td><td>Higher data dependence.</td></tr>
<tr><td>↓</td><td>Higher memory storage requirements.</td><td>↑</td><td>Lower memory storage requirements.</td></tr>
</table>

Table 2. Comparison between FPM and PUMI (FEM) methodologies.

## 5.2 FPM efficiency issues and improvements

As shown in the previous section, many aspects of the FPM involve a higher computational complexity, which may not be compensated for by the benefits resulting from the exploitation of meshless features. Hence, it becomes necessary to improve the efficiency of the basic meshless solver.

Taking into account the aspects that increase the computational cost of the FPM (see Table 2), it can be seen that the stencil of points involved in the assembly of the nodal residual plays an important role[56]. To illustrate this, the basic semi-discrete scheme of Eq. (4.18) is rewritten as a flux central-difference like term plus an explicit diffusion term given by Eq. (4.32) (as in Section 4.2.3). This leads to

$$\begin{aligned}\frac{\mathrm{d}\hat{\mathbf{U}}_i}{\mathrm{dt}} &= -\sum_{j\neq i} b_{ij}^k \left[\mathbf{F}_j^k - \mathbf{F}_i^k\right] + \sum_{j\neq i} b_{ij}^k \left|\mathbf{A}_{\hat{n}}(\mathbf{U}_i^+, \mathbf{U}_j^-)\right|\left(\mathbf{U}_j^- - \mathbf{U}_i^+\right)\hat{n}_{ij}^k \\ &= \mathcal{F}_i + \mathcal{D}_i\end{aligned} \tag{5.2}$$

where the convective and diffusive flux contributions are assembled by summing the contributions of the edges connecting $\boldsymbol{x}_{\mathrm{i}}$ with each of the points in the local cloud. This connectivity typically includes the first layer of Delaunay neighbors (physical edges in a FE mesh) plus additional points located outside this layer (numerical edges). Due to the fact that only points connected by physical edges are used to assemble Eq. (5.1) in FEM[57], the overhead incurred by the FPM can be large depending on the number of numerical edges in each cloud. Although there is limited room to reduce the connectivity depending on the approximation bases chosen (some restrictions will be discussed later), this approach is not flexible enough. Therefore, a more general solution is necessary to reduce the extra cost. This will be addressed in the next section.

Among the other differentiating aspects of the FPM highlighted in Table 2 which could increase the computational cost, it is worth mentioning that the need to solve the system (4.60) when advancing the equations in time does not make a substantial difference. In addition, numerical experiments have demonstrated that this step could be skipped assuming $\mathbf{R}(\mathbf{U}) \approx \mathbf{R}(\hat{\mathbf{U}})$ in Eq. (4.59) without causing a negative impact on the problem solution. This behavior can be explained by the fact that the FPM approximation yields a

[56] The construction of the meshless approximation is not accounted for as it is only performed once in typical computations.

[57] From *h*-refinement arguments it is possible to estimate 14 neighbor points (on average) for a tetrahedral mesh. However, the number of points may be higher in practice according to the characteristics of the mesh.

very good estimate of the nodal parameters, i.e $\mathbf{U}(\boldsymbol{x}_i) \cong \hat{\mathbf{U}}(\boldsymbol{x}_i)$, for well-behaved solutions.

## 5.2.1 Cloud connectivity issues

Some features of the FPM approximation can be exploited with the aim of reducing the additional cost caused by the cloud connectivity. In particular, as the influence of the cloud points in the local approximation decreases progressively with the distance from the star point (as a result of weighting, see Figure 34), a reduced stencil could be used to compute the derivatives without a significant detriment on accuracy.

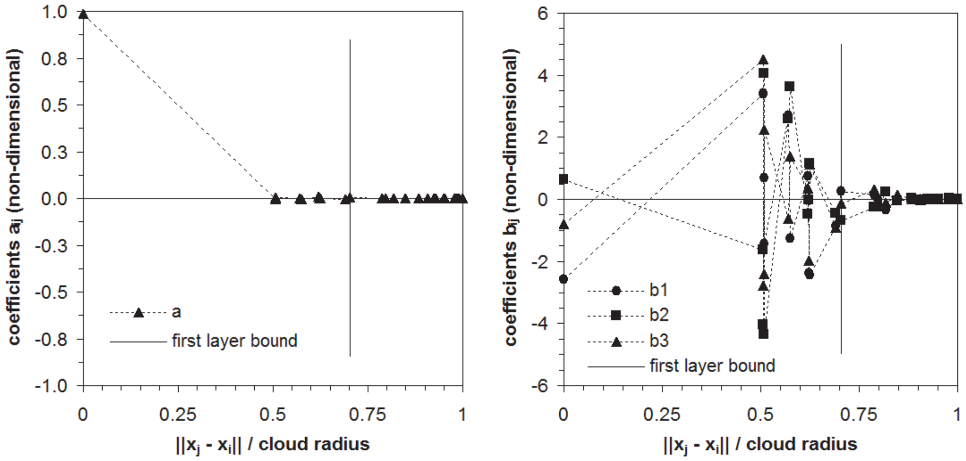

Figure 34. Variation of the cloud metric coefficients *a* and ***b*** (Eqs. (3.11) and (3.12)) with the distance from the star point. Results computed for a typical interior cloud with 30 points and quadratic approximation basis.

This possibility was explored by using a point stencil based on the Delaunay nearest neighbors of the star point. The numerical tests performed on typical 3D clouds indicated that the approximation error in the derivatives computed in this way is marginal in most cases but can be up to 20% higher than the standard approximation in severely distorted clouds of points (which are only a small percentage of the total cloud count). This result supports the employment of reduced stencils for computations in which high accuracy is not required. For example, explicit artificial diffusion terms like those in Eq. (5.2) are suitable candidates for this simplified treatment.

### 5.2.1.1 Simplifying the diffusive flux contribution

If the stencil of points used to compute the derivatives appearing in the diffusion terms is restricted to the nearest neighbors, the computations can be efficiently carried out by means of a data structure similar to that employed in edge-based FEM solvers. Using this data structure (which can be constructed from the cloud connectivity), the diffusive terms can be assembled by performing a loop over the edges and scattering contributions to the nodes. At each star point, this procedure leads to

$$\mathcal{D}_i = \sum\nolimits_e d_{ij} \left| \mathbf{A}_{\hat{n}}(\mathbf{U}_i^+, \mathbf{U}_j^-) \right| \left( \mathbf{U}_j^- - \mathbf{U}_i^+ \right) \tag{5.3}$$

where $e$ denotes the edges in the first layer neighbors of $\boldsymbol{x}_i$. As the FPM approximation is not symmetric, two weight coefficients, $d_{ij}$ and $d_{ji}$, are required per edge. These are computed by

$$d_{ij} = b_{ij}^k \hat{n}_{ij}^k \quad , \quad d_{ji} = -b_{ji}^k \hat{n}_{ij}^k \quad \text{with} \quad k = 1,3 \tag{5.4}$$

where the $\boldsymbol{b}$ coefficients are proportionally scaled to recover the partition of nullities property in the reduced stencil (see Section 3.1.3). The tests performed have shown that the application of Eq. (5.3) does not affect the robustness and accuracy of the scheme, while the computational cost is drastically reduced due to the lower cloud connectivity and the avoidance of redundant operations.

#### 5.2.1.2 Treatment of the convective terms

In contrast to the diffusive contribution, accuracy is essential for the computation of the convective fluxes in Eq. (5.2). Thus, the reduced stencil is no longer applicable and all numerical edges would need to be accounted for in the edge data structure. This would have no clear advantage in the number of operations involved and may even present some drawbacks with respect to the original formulation regarding storage requirements and memory accesses during assembly. These reasons justify the fact that the full point-based structure is maintained to compute the convective fluxes at the expense of some degree of memory overhead. Therefore, after applying the partition of nullities property, the convective contribution can be expressed by

$$\mathcal{F}_i = -\sum\nolimits_j b_{ij}^k \tilde{\mathbf{F}}_j^k \tag{5.5}$$

where the summation is performed over all the points in the local cloud.

#### 5.2.1.3 An application example

The performance of the improved residual assembly (Eqs. (5.3) and (5.5)) is studied in relation to the original formulation (Eq. (5.2)) by comparing the CPU-times required to evaluate the residual vector 10 times in discretizations having different numbers of points (high-order limited MUSCL extrapolation is employed). The computations are performed in a desktop computer with an Intel Core2 Quad Processor Q9550 @ 2.83 GHz running on a Windows 32-bit operating system. The results show that the runtime is almost halved with the improved formulation (see Figure 35). The parallel scaling is also satisfactory, although not as linear as with the original formulation, presumably due to memory accesses when scattering edge contributions to the nodes. This behavior deserves further study.

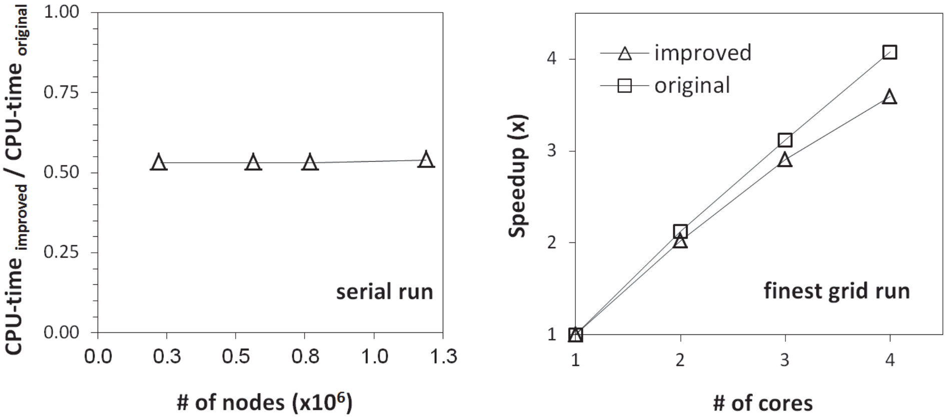

Figure 35. Runtime and speed-up comparison between the original and improved residual vector assembly.

It is important to note that the use of different stencils for the convective and diffusive fluxes requires a flow solver with explicit artificial diffusion, or which can be written in this form (e.g. the Roe-MUSCL approach here employed). For solvers where a separate diffusive term is not present (e.g. exact Riemann or vector splitting methods like van Leer) the reduced stencil could adversely degrade the accuracy of the computed fluxes. Therefore, the method here presented is limited to solvers with explicit artificial diffusion terms. This is not a serious limitation because this class of solvers encompasses a large number of successful implementations currently in use.

### 5.2.2 Use of lower order approximation bases

Given the fact that cloud connectivity is the main factor affecting the FPM performance, it can be assumed that the cost can be reduced by lowering the approximation base, and thus the required spatial support (recall that complete quadratic bases are used in this work). However, numerical experiments show that smaller approximation supports tend to give rise to problems due to non-overlapping clouds of points, particularly when there are sudden changes in the grid density through the domain. This behavior, as discussed in previous chapters, is avoided by including in each local cloud the nearest Delaunay neighbors of the star point. Since this implies that at least 14 points[58] are required for an interior cloud (this number is generally higher in practice), lower-order bases do not permit to reduce the stencil of points significantly.

[58] From *h*-refinement arguments, see Footnote 57.

Therefore, the potential time savings may little compensate for accuracy and robustness losses.

## 5.3 Comparative assessment of the FPM

The performance of the basic FPM solver is studied in this section in comparison with the FEM solver PUMI. Next, the computational test case employed in the assessment is described first, and a grid convergence analysis is performed to analyze the solution accuracy. Finally, computational cost and parallel performance of the methods are investigated on modest multi-core (shared-memory) CPUs.

### 5.3.1 Model spatial discretization

The wing-body configuration presented in Section 4.6.3 is selected to carry out the comparative analyses. The intended studies require a series of comparable model discretizations with point spacing $h_1<h_2< ...<h_n$ for each computational code. In order to obtain these models, the same input boundary triangulation is employed and several internal surfaces are defined within the analysis domain to control the point spacing variation (see Figure 36).

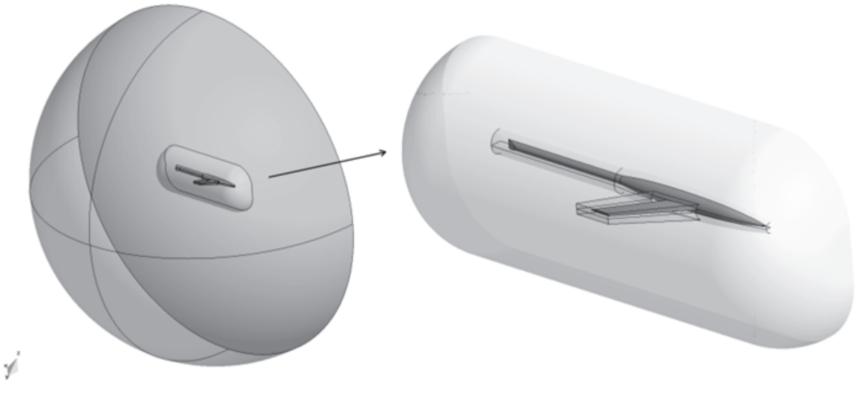

Figure 36. Definition of spacing control surfaces within the analysis domain.

The volume discretization is generated by an unconstrained Delaunay method in the FPM (see Section 3.6.1) and the mesher GiD (GiD, 2013), which is based on an advancing-front technique, is used with PUMI. In order to obtain successively refined discretizations, the grid sizes assigned over the geometrical entities which define the model (boundaries and internal control surfaces) are scaled by a constant factor. The refinement ratio was set to 2/3 to keep the computational cost low, although further size adjustments were needed at specific body locations to get smooth transitions in the boundary mesh. The level of similarity achieved between both sets of discretizations was satisfactory; some details are given in Table 3.

| grid # | # of nodes |  | body surfaces |  |
|---|---|---|---|---|
|  | FPM | PUMI | triangles | point |
| 1 | 2860318 | 2910146 | 694482 | 348073 |
| 2 | 1664535 | 1558418 | 268842 | 135183 |
| 3 | 854095 | 835316 | 146972 | 72265 |
| 4 | 427900 | 457575 | 83102 | 42086 |

Table 3. Details of the NACA test model discretizations.

Once a consistent set of successively refined grids is obtained, a representative grid refinement ratio is defined to carry out the grid convergence analysis. The following estimate is adopted

$$r = h_{k+1} / h_k = \left( n_{k+1} / n_k \right)^{-1/3} \tag{5.6}$$

where subscripts $k$ and $k$+1 indicate two successive refinement levels (fine and coarse respectively) and $n$ is the number of grid points. In cases when the grid refinement is not uniform throughout the domain, Eq. (5.6) should be applied with care. In this work, this approximation was found acceptable because the grids are refined systematically between the body and the control surfaces by scaling the desired point spacing in that area in a consistent manner. The refinement ratios computed by applying Eq. (5.6) to both sets of grids are r = 1.235 (std. deviation 0.033) for the FPM and r = 1.228 (std. deviation 0.005) for PUMI. Figure 37 shows a close view of the surface grids near the wing tip.

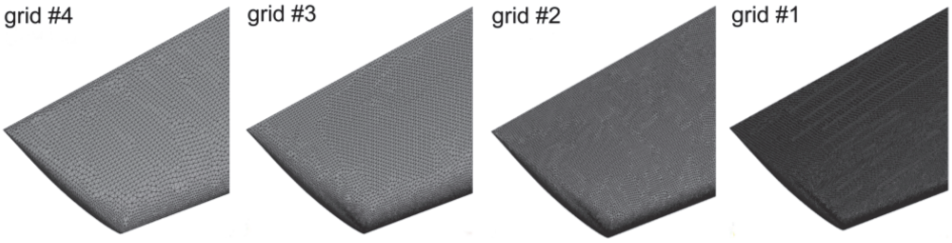

Figure 37. View of the boundary discretization near the wing tip for grids 1-4.

### 5.3.2 Flow and problem settings

The simulation condition adopted in the analyses assumes an inviscid fluid flow with freestream Mach number M = 0.9 and angle of attack $\alpha = 4^{\circ}$. The boundary conditions employed are flow symmetry on a plane across the fuselage centerline, freestream Riemann conditions on the outer hemispherical boundary and slip condition on the model surface.

The computations are performed using third-order MUSCL extrapolation with the van Albada limiter. A four-stage time integration scheme with local time-stepping and implicit residual smoothing (with two Jacobi iterations) is chosen. The Courant number is set to unity and the number of time steps in the problem solution is fixed to 10K for all the runs performed. This is an arbitrary value because the study is only intended to compare the performance of both methods. The coarser grids converge for a lower number of time steps while the finest grid could benefit from more iterations. In any case, a residual drop of at least four orders of magnitude is achieved for all the grids, which is deemed acceptable for a preliminary solution. A density residual L2 norm is used to monitor the solution convergence.

### 5.3.3 Grid convergence analysis

Exact solutions are not usually available in practical engineering problems, thus reliable estimates are required to evaluate the solution accuracy. A typical approach is to obtain these estimates from solutions on grids with different levels of refinement by means of Richardson's extrapolation (see for instance (Roy, 2005)). Assuming two discrete problem solutions $f_1$ and $f_2$, obtained on grids with point spacing $h_1$ and $h_2$ respectively ($h_1<h_2$), a higher-order estimate of the continuum solution (value at h→0) can be obtained by

$$f_0 = f_1 + \frac{f_1 - f_2}{r^p - 1} \tag{5.7}$$

where both $f_1$ and $f_2$ are assumed to lie in the asymptotic range of convergence, *p* is the order of accuracy of the method and *r* is the grid refinement ratio. In the same framework of Richardson's extrapolation, the order of the numerical scheme can also be estimated by using three different discrete solutions. Assuming a constant refinement factor, the order of accuracy can be calculated by

$$p = \frac{1}{\ln(r)} \ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) \tag{5.8}$$

It should be noticed that Eq. (5.8) is sensitive to the way in which the refined grids are generated and the estimated refinement ratio, see for instance (Roy, 2003; Salas, 2006). As this may affect the calculated results, a verification of the procedure is also performed in this section.

Once the accurate solution estimate, the grid refinement factor and the observed order of accuracy of the scheme are known, the accuracy of the numerical solution can be evaluated. This is usually performed by measuring the discretization error. Among other approaches pursuing this end, the Roache's grid convergence index (GCI) (Roache, 1994) has become quite popular. This indicator provides an error band for the discrete solution with respect to an accurate estimate $f_0$, accounting for the order of accuracy of the scheme and the grid refinement ratio. The GCI is computed for a fine grid solution $f_k$, using a solution $f_{k+1}$ on a coarser grid, by the following expression

$$GCI_{k,k+1}(\%) = \frac{F_s}{r^p - 1}\left|\frac{f_{k+1} - f_k}{f_k}\right| \times 100 \tag{5.9}$$

where F$_s$ is a safety factor normally set to 1.25 when more than 3 grids are compared. The GCI also allows verifying if the solutions are in the asymptotic range of convergence. In such a case, values computed over three consecutive refined grid solutions must satisfy the following relations

$$GCI_{23} = r^p GCI_{12} \quad \text{or} \quad \frac{GCI_{23}}{r^p GCI_{12}} = 1.0 \tag{5.10}$$

#### 5.3.3.1 Solution accuracy

The methodology described above is applied to the numerical solutions obtained with FPM and PUMI. The analysis is performed by examining the evolution of the body normal force coefficient (C$_N$) with grid refinement. Converged solutions computed for grids 1-4 are presented in Table 4 and the discrete data points are displayed in Figure 38 as a function of the grid size normalized with the point spacing of the finest grid (h$_1$). The higher-order solution estimates (computed on the three finest grids), and experimental results from (Loving & Estabrooks, 1951) are also displayed in Figure 38.

| grid # | FPM | | PUMI | |
|---|---|---|---|---|
| | h (n$^{-1/3}$) | C$_N$ | h (n$^{-1/3}$) | C$_N$ |
| 1 | 7.045E-03 | 0.35864 | 7.004E-03 | 0.35737 |
| 2 | 8.438E-03 | 0.35653 | 8.625E-03 | 0.35185 |
| 3 | 1.054E-02 | 0.35369 | 1.062E-02 | 0.34399 |
| 4 | 1.327E-02 | 0.35202 | 1.298E-02 | 0.34110 |

Table 4. Normal force coefficients and representative grid sizes computed for FPM and PUMI. NACA wing-body, $M_\infty$=0.9 and $\alpha$=4°.

It can be seen in Figure 38 that the extrapolated solutions present a slight discrepancy (about 1.5%, ~5 lift counts). This can be due to differences in the discrete models employed and uncertainties in the estimated grid refinement ratio, which in turn affects the estimation of $p$ (see Eq. (5.8)). In addition, the level of convergence of the time-marching scheme could have some influence on the results obtained (details in this respect are given later). It must be stressed, however, that the difference lies mostly on the extrapolated values; while the computed force coefficients for the highest refinement level show a very good agreement between both solvers. A difference between numerical and experimental force coefficients (about 10%) is also observed in Figure 38. Probably, this is mostly due to the inviscid flow assumption adopted in this work, with the wind tunnel corrections also playing a role.

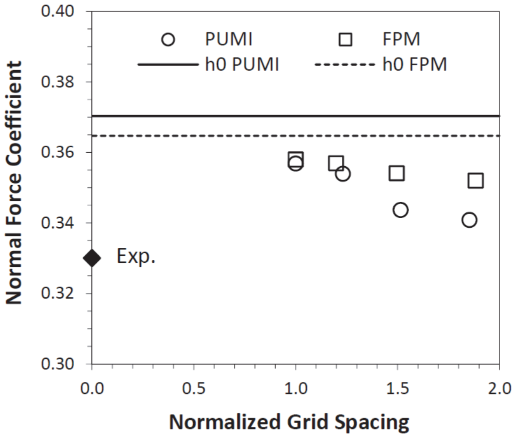

Figure 38. Normal force convergence. NACA wing-body, $M_\infty$=0.9 and α=4º.

According to the results presented above (see Table 4), the observed order of accuracy computed by Richardson's extrapolation is p = 1.4059 for the FPM and p = 1.7281 for PUMI. Although second-order spatial accuracy is theoretically possible, lower convergence rates are expected due to the non-smoothness of the flow, the limiting process, boundary conditions effects and the characteristics of the discretizations employed. The additional dependence the FPM has on the characteristics of the local approximation can explain the lower value achieved in comparison with PUMI.

A measure of the discretization error in the numerical solutions is computed by means of the GCI (Eq. (5.9)) and the results are presented in Table 5. As expected, the error bound decreases for the solutions on grids 3 to 1 but this behavior is not regular for grids 4 to 2. The checks performed by applying the relation (5.10) indicate that the solutions on the three finest grids are in the asymptotic range of convergence but the coarser grid, for both PUMI and FPM, is not. This could be the reason for the anomalous behavior in the coarser solutions.

| | FPM | PUMI |
|---|---|---|
| **$GCI_{12}$(%)** | 1.97 | 4.58 |
| **$GCI_{23}$(%)** | 2.67 | 6.64 |
| **$GCI_{34}$(%)** | 1.58 | 2.58 |
| **check 1-2-3** | 1.014 | 1.016 |
| **check 2-3-4** | 0.443 | 0.273 |

Table 5. Grid convergence index for discretizations sets 1-4 ($F_s$=1.25). NACA wing-body, $M_\infty$=0.9 and α=4º.

An additional verification of the observed order of accuracy of the schemes is performed by assuming second-order extrapolation to compute the estimates $f_0$. The convergence rates and error levels in Figure 39 are in agreement with the results obtained above.

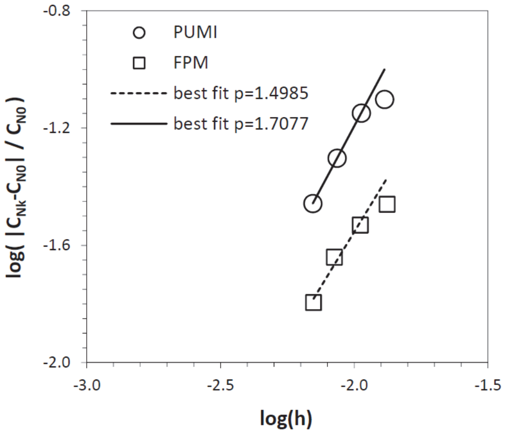

Figure 39. Evolution of the error estimates with the grid refinement level. NACA wing-body, $M_\infty$=0.9 and α=4º.

## 5.3.3.2 Iterative convergence

The iterative convergence of the solutions obtained for the test cases presented above is quite similar in both methods, although slightly better in PUMI (see Figure 40). It is also observed that the residual level achieved for the fixed number of steps chosen (10K) increases with grid refinement. This behavior, which can be explained due to the fact that more time steps are needed to propagate the information through the solution domain as the grid becomes finer, may cause incomplete convergence in the runs with the finest grids. In order to verify this, an analysis with 30K time steps is performed using the finest grid. Figure 41 shows that the normal force variations (over intervals of 100 steps) are within ±1 lift count approximately after step 5000 and oscillate, with decaying amplitude, around a mean value which does not change substantially with respect to the final converged solution. Hence, in spite of the fact that more than 10K iterations being required for a fully converged solution, the residual level achieved is considered acceptable.

It should be noted that more effective solution approaches, e.g. multigrid, would be required for satisfactory convergence in large-scale problems. In the present work neither convergence optimization (e.g. through freezing of limiters, Courant number and residual smoothing parameters adjustment, etc.) nor multigrid implementations have been considered. The performance is studied only in relative terms and, as shown in (Katz & Jameson, 2009b), it is expected that both methodologies can benefit equally from improved solution techniques like multigrid, thus achieving comparable performance gains.

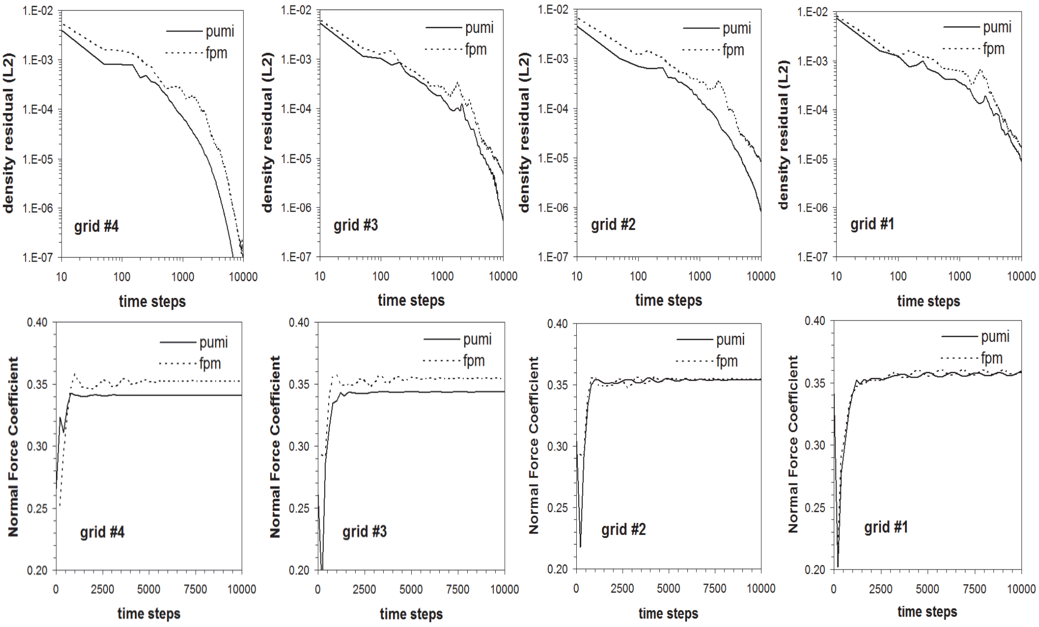

Figure 40. Iterative convergence for the NACA wing-body at $M_\infty$=0.9 and α=4º.

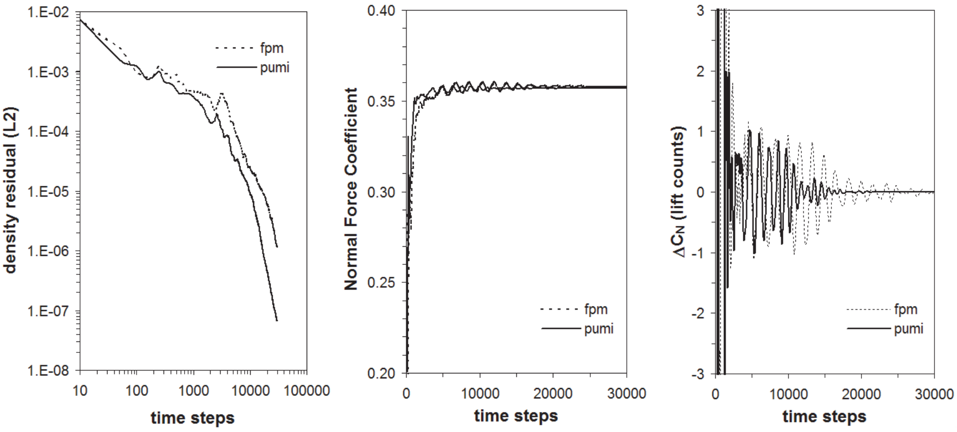

Figure 41. Convergence history computed for the finest grid. NACA wing-body at $M_\infty$=0.9 and α=4º.

Finally, pressure distributions along a wing section (40% half-span) are compared (Figure 42). A more diffusive solution (smoother shocks) is observed in PUMI but this trend disappears as the grids are refined, thus achieving a very good agreement for the finest grid. The reduced diffusion in the FPM (possibly due to the use of a reduced stencil for the computation of the artificial fluxes) implies that a more accurate solution might be achieved for a given grid spacing. On the other hand, the reduced amount of artificial diffusion may be the cause of a lower convergence speed (cf. Figure 40 and Figure 41). Overall, there seems to be a trade-off between reduced diffusivity (in FPM) and faster convergence to the steady state (FEM). The net effect is that both methods seem roughly comparable in this respect. The agreement between numerical and experimental results in Figure 42 is satisfactory.

Surface pressure contours computed with PUMI and the FPM are shown in Figure 43 for grid sets 1-4.

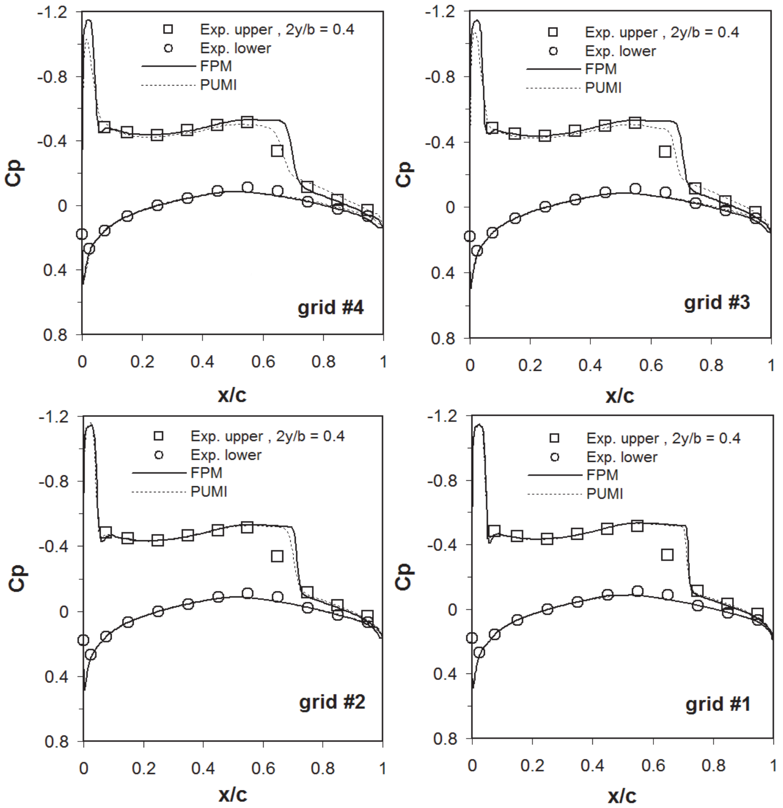

Figure 42. Pressure distributions computed at a 40% half-span wing section. NACA wing-body, $M_\infty$=0.9 and α=4°.

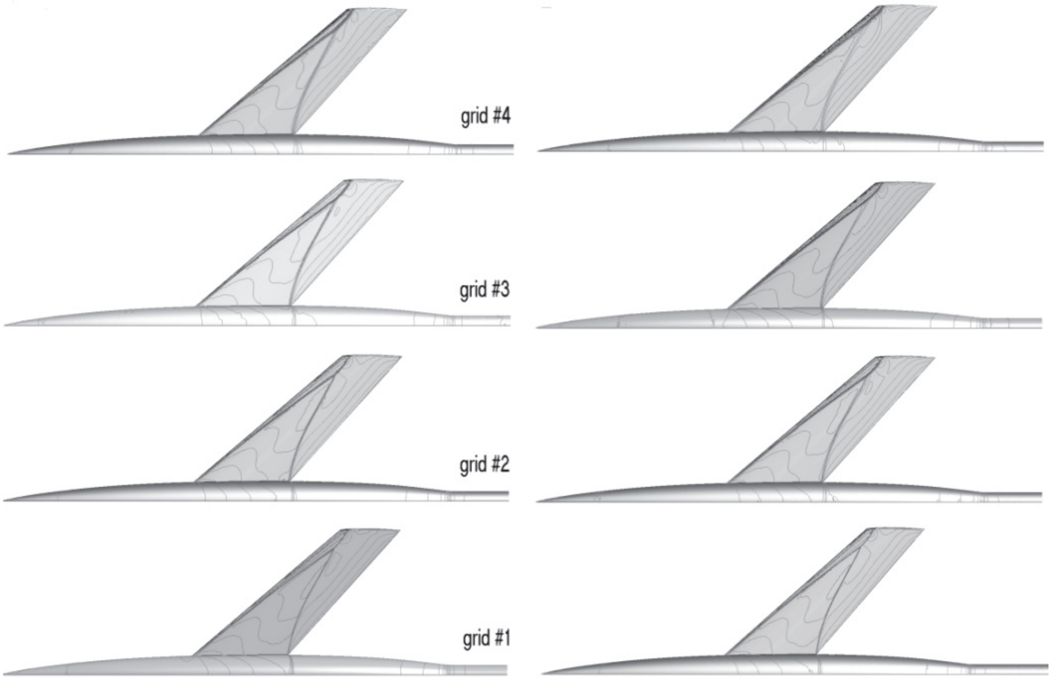

Figure 43. Surface pressure contours computed for grid sets 1-4. Left: FPM, Right: PUMI. NACA wing-body, $M_\infty$=0.9 and α=4°.

### 5.3.4 Performance assessment

The computational performance of the basic FPM solver with respect to FEM approach (PUMI) is examined in this section. The computational codes are compiled with Intel Fortran v10.1 with flags *–O3 –ipo –xT –no-prec-div –funroll-loops* and single precision real data. The runs are performed on a cluster node having 2 Intel Quad-core Xeon E5410 processors @ 2.33 GHz with 2 x 6Mb L2 cache. The system has 32 Gb global ram and runs under Linux operating system. No processor scheduling is specified.

In order to facilitate the performance comparison, the analysis is divided into pre-process and flow computation stages. The pre-process stage includes a series of operations which are similar in PUMI and the FPM (e.g. reading of input files, construction of boundaries and geometrical information and data renumbering) as well as other tasks which differ radically. The latter concern the computation of the spatial approximation and its related data structures. Hence, the difference between the CPU-times measured for the complete pre-process stage can be considered an approximate measure of the relative cost of constructing the meshless and the FEM-based approximations. Note that the global cloud connectivity (unconstrained Delaunay) provided by the point generator is employed to speed-up the clouds construction in the present examples. The flow computation stage concerns the Euler time marching solution and the CPU-time measurement is taken over 100 time steps running with the high-order MUSCL scheme. The simplified methodology proposed in Section 5.2 is employed for the FPM.

It is important to note that the analysis is conducted using modest hardware resources having a limited performance, particularly as regard parallel scalability (multi-core CPU in single-socket board). Consequently, the results should be considered only as a measure of the relative performance of the FPM with respect to a classical FEM technique running in this kind of hardware and implementing similar solution strategies (as far as this is possible). It is expected that the use of higher-performance hardware (multi-processors) would yield comparable gains for both codes.

#### 5.3.4.1 Pre-process stage

The CPU-times involved in the pre-process stage are compared in Figure 44 (the data for PUMI corresponds to serial runs as this part of the code is not parallelized). The serial runtime in the FPM is higher than in PUMI and the differences grow markedly with the problem size. However, important time savings are achieved when running in parallel. The speed-ups attained (Figure 44 (right)) show a fair scalability when running up to 4 CPU-cores, with higher speed-ups achieved for finer grids (probably due to the increase in the amount of work to be processed, which makes the parallel overhead less important in relative terms). It can be also noted that the parallel performance changes noticeably when more than 4 cores are employed. This behavior is typical in multi-core single-processor systems as those used in this work. It reflects the fact that when several threads are spanned, some processor resources must be forcibly shared (specifically, the L2-cache and main memory) exacerbating memory contention issues. This fact, coupled with the constant main-memory bandwidth (which depends only on the number of physical processor sockets and therefore does not scale with thread count) makes the performance gains beyond the 4-thread limit very inconsistent.

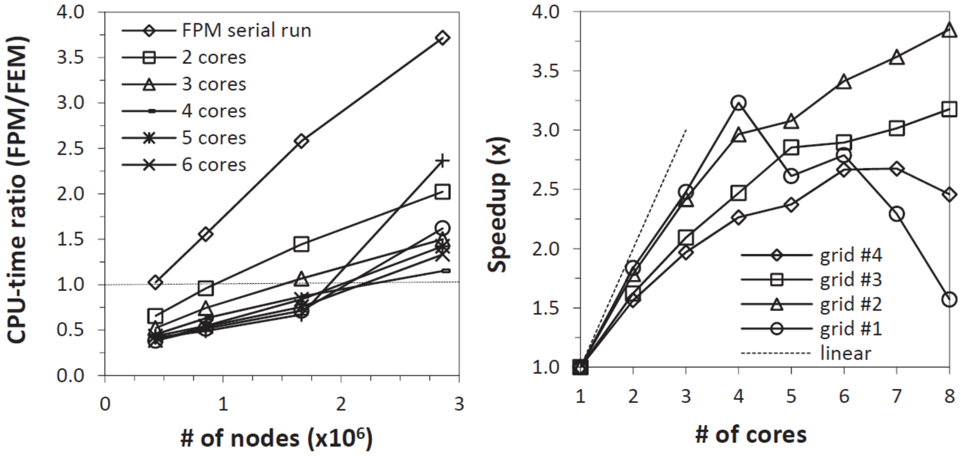

Figure 44. CPU-time ratio and parallel multi-core performance of the FPM in the pre-process stage. NACA wing-body problem.

### 5.3.4.2 Flow computation stage

The flow computation stage is evaluated by running 100 high-order Euler steps on different numbers of cores. The CPU-time ratios (Figure 45), obtained by dividing the FPM runtime by the time measured in PUMI for each grid and number of cores, show comparable computational cost in both methods. This result corroborates the effectiveness of the modifications introduced in Section 5.2. Furthermore, it is observed that the runtime in the FPM reduces faster than in PUMI as the number of cores increases (see Figure 46). The FEM solver achieves peak performance with 4 cores, presumably due to memory bandwidth saturation. Unlike PUMI, the higher data locality and totally independent computations in the FPM make it possible for the scaling to continue over 4 cores, albeit at a reduced rate. It should be noted that the scaling of the FEM solver can be improved by implementing some form of domain decomposition, bringing the performance to the same level as the FPM solver[59]. The advantage of the meshless

[59] The FEM solver has been parallelized in a very simple way because it was designed to run in machines with a small number of cores.

approach is that such kind of optimization is not required, given the independent nature of the computations.

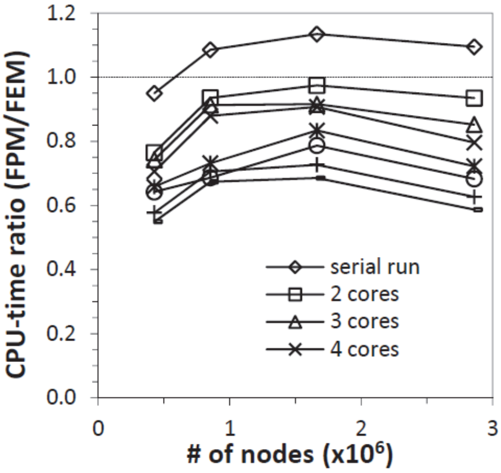

Figure 45. FPM runtime ratios measured for 100 high-order Euler cycles. NACA wing-body, M=0.9 and α=4º.

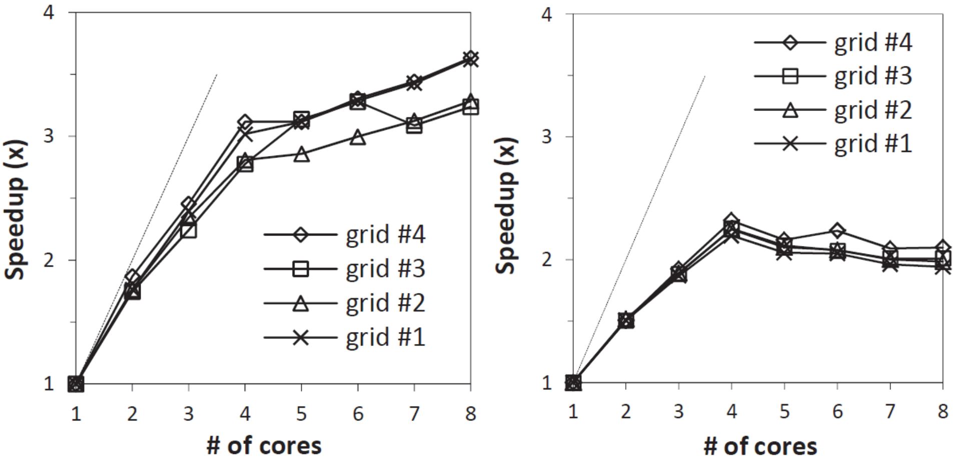

Figure 46. Speed-up comparison of the FPM (left) and PUMI (right) based on 100 high-order Euler cycles. NACA wing-body, M=0.9 and α=4º.

The evolution of the normal force error computed for the finest grid using 4 CPU-cores is compared in Figure 47 (the time values are normalized with the total time required in the FEM run). The results show that the CPU-time required to achieve the same error level is quite similar in both methods. The oscillatory behavior seen before (Figure 40) is also evidenced as the solution approaches an asymptotic behavior. The attained error levels are also within the error bands predicted in the previous section (see Table 5).

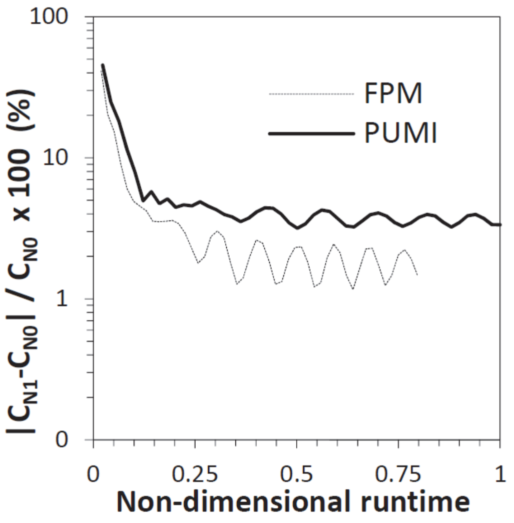

Figure 47. Evolution of the normal force error for the finest grid resolution. NACA wing-body, M=0.9 and α=4º.

### 5.3.4.3 Memory usage

Memory storage requirements are compared in Figure 48. The higher memory usage in the FPM responds to a higher connectivity per node (which increases the storage requirements per cloud) and the need to use two different data structures, with redundant information, to enhance computing performance. Moreover, the lack of symmetry of the approximation operators in the FPM does not allow reducing the storage requirements like in PUMI. Even though further optimizations are possible, some distinctive properties of the FPM, such as the higher connectivity per node and the lack of symmetry properties of the metric coefficients, mean that some degree of memory overhead cannot be avoided. The impact of the storage requirements on the applicability of the FPM in large-scale problems needs to be studied further.

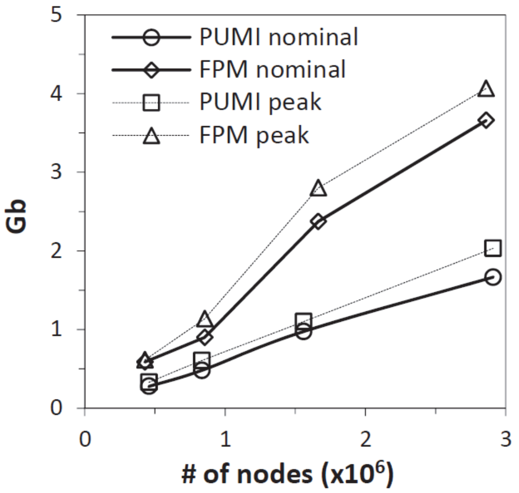

Figure 48. Memory storage requirements in FPM and PUMI.

### 5.3.4.4 Domain discretization

The CPU-time involved in the generation of the point discretizations used with the FPM is presented in Figure 49. The discretizations are generated using a desktop computer with an AMD Opteron 246 processor @ 1.99 GHz running on a Windows 64-bit system. The runtimes presented correspond to the overall generation time, which includes the initial Delaunay of the input boundary grids required to start the interior volume discretization through the driven point insertion procedure (see Section 3.6.1).

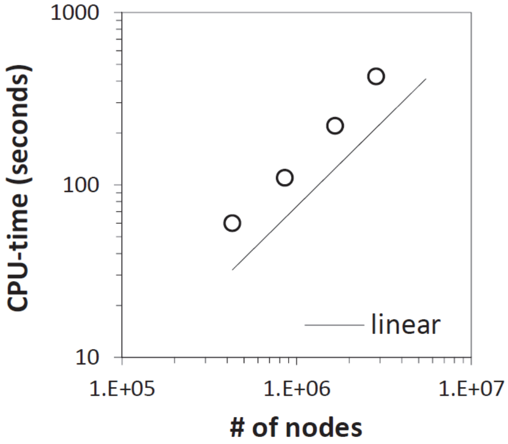

Figure 49. CPU-times required for the point discretization of the NACA wing-body with different refinement levels.

It should be noted that the CPU-times required for the FEM mesh generation using GID are not presented in Figure 49. The fact that the advancing front technique is more expensive than the Delaunay method and, in addition, cosmetic procedures are required to guarantee a suitable mesh, make the generation time in GID to be hardly competitive with the point discretization technique used in the FPM.

## 5.4 Assessment of FPM reduced-fidelity models

The performance of the artificial diffusion models presented in Section 4.2.3 is studied here in comparison with the basic Roe-MUSCL solver. With this aim, numerical computations are performed with the finest grid discretization (#1) using the same flow conditions and solver settings employed in the preceding sections. In the comparisons, third-order MUSCL extrapolation with the van Albada limiter is used in both, the Roe-MUSCL scheme and the limited-differences scalar model. The dissipation coefficients in the Jameson's model are set to $\alpha_2 = 0.25$ and $\alpha_4 = 0.05$.

The iterative convergence of the methods is studied first and the results are compared in Figure 50. As regards residuals convergence, the three methods exhibit a quite similar behavior. The evolution of the normal force is also comparable, although on average, slightly lower values are obtained with the artificial diffusion models (about 1.5% with respect to the Roe-MUSCL scheme). Note that incomplete solution convergence, causing the same oscillatory behavior seen earlier (Figure 40), is also observed for the reduced-fidelity schemes under testing.

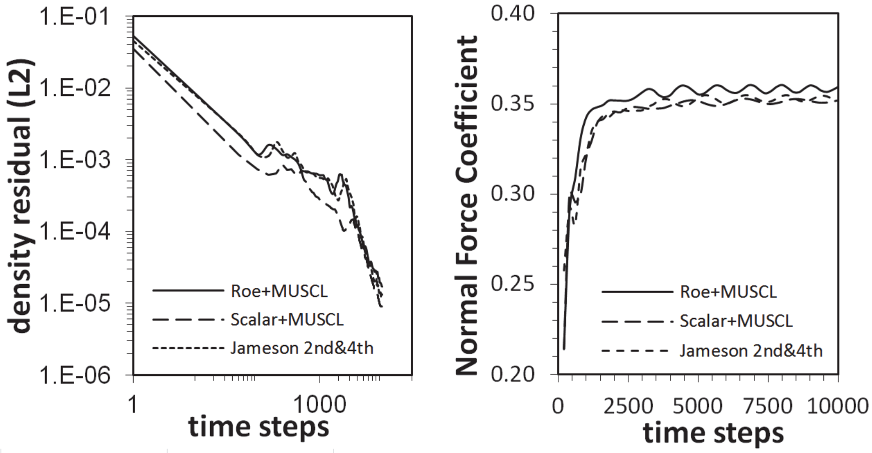

Figure 50. Comparison of iterative convergence between the FPM reduced-fidelity models and the Roe-MUSCL solver. NACA wing-body (grid #1), M=0.9 and $\alpha$=4º.

The pressure distributions computed along a section at 40% of the wing semi-span have also been compared (Figure 51). It can be observed that the scalar solution has a higher level of diffusivity, which causes some spread of the rear shock wave. In the case of the Jameson's solution, the shock position matches that in the original MUSCL-Roe scheme and it is sharply captured, although with considerable overshoot. This behavior can be improved by fine tuning of the dissipation parameter $\alpha_2$ and $\alpha_4$ or by using an enhanced flow detector. However, these adjustments are beyond the objectives in the present analysis.

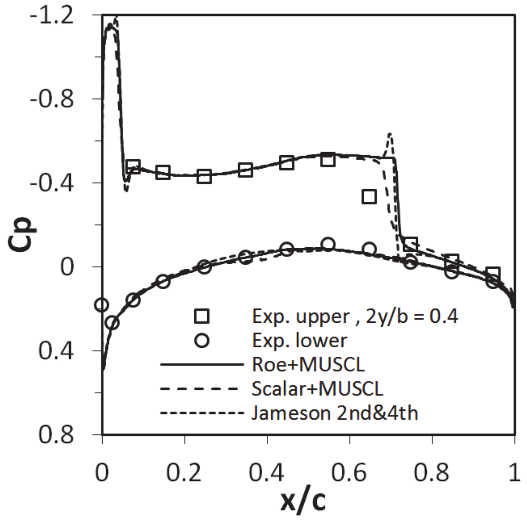

Figure 51. Comparison of FPM pressure distributions for the NACA wing-body at a wing station 2y/b=0.4 using grid #1. $M_\infty$=0.9 and $\alpha$=4º.

The computational cost of the three schemes is compared in Table 6. The CPU-times are taken over 100 Euler steps and the hardware platform is that used in Section 5.3.4 (running on 4 CPU-cores). The relative cost of the different methods shows that notable speed-ups can be achieved in the computations by using the reduced-fidelity models. Particularly in the Jameson's scheme, the computing time is reduced 42% with respect to the Roe-MUSCL solver and, as shown in Figure 50 and Figure 51, the solution accuracy is not significantly affected. Even though the Jameson's model has the disadvantage that diffusion coefficients must be specified, the ratio accuracy/cost achieved makes it a good candidate when fast computations are required. Furthermore, the employment of a more precise switch can help eliminate the overshoots as well as reduce the solution's sensitivity to the user-defined diffusion coefficients.

| | **CPU-time (secs.)** | **Relative cost (%)** |
|---|---|---|
| **Roe+MUSCL (original)** | 523.6 | 100 |
| **Scalar + MUSCL** | 462.8 | 88 |
| **Jameson 2$^{nd}$ & 4$^{th}$ order** | 302.1 | 58 |

Table 6. CPU-times needed to perform 100 Euler steps with grid #1 using the different FPM solution schemes. NACA wing-body, M=0.9 and $\alpha=4^o$.

## 5.5 Concluding remarks

The performance of the FPM methodology has been investigated in this chapter on the basis of a practical transonic test case. The analysis focused on accuracy, computational cost and parallel scalability of the meshless method, and was conducted in comparison with an equivalent and widely accepted Finite Element-based technique (PUMI code). This provided a standard to assess the characteristics of the meshless solver as well as its suitability to address practical applications. In this regard, a simplification of the basic FPM technique was proposed to improve its efficiency and reduce the performance gap with respect to conventional solution approaches. The most relevant findings of the comparative assessment are summarized below.

Regarding the observed accuracy of the FPM scheme, the grid convergence studies showed a convergence rate of about 1.4, which is within the expected range for the type of problem and characteristics of the solution approach employed. In addition, no significant differences were found with respect to the FEM scheme. The discretization error estimates in the FPM were lower for a similar grid spacing and this fact may point to a less diffusive solution. On the other hand, the number of iterations needed to achieve convergence is increased. The net effect is that the both methods are comparable with regards to the runtime performance.

The computational cost has been analyzed from the point of view of the construction of the approximation and related data structures, and the flow computation stage. Regarding the approximation, the FPM has proved more expensive than FEM (as expected due to its higher complexity). Nevertheless, the method shows good scaling when running in parallel, making the overall cost quite affordable. As far as the FPM flow computation stage is concerned, the proposed simplified evaluation of the diffusive fluxes (using a reduced stencil of points with an edge-based data structure) has proved effective in reducing the computational cost without affecting solution accuracy and robustness. Consequently, the measured CPU-times obtained in the FPM were comparable to those in FEM for the serial runs and a satisfactory parallel behavior was also observed. This gives the possibility to combine meshless advantages (e.g. regarding discretization) with a competitive computational cost, thus achieving efficient implementations. Moreover, the application of reduced-fidelity models has demonstrated that the computational cost can be still reduced significantly while preserving accuracy to a large extent. This can be of benefit in many rutinary computations performed in practice.

It cannot be denied that some additional improvements are still required in view of practical applications (e.g. more efficient convergence acceleration techniques, improved data structures and cloud construction procedures). Nonetheless, the performance of the FPM core algorithm has shown to be competitive with respect to its FEM counterpart. This is important because meshless techniques are often considered impractical due to the poor efficiency of the initial implementations.

In the next chapters, different applications and extensions of the basic FPM methodology here studied are presented with the aim to show its capabilities in specific areas where meshless procedures can make interesting contributions.

# 6 ADAPTIVITY AND ERROR ESTIMATES

Automatic grid adaptation (*h*-adaptation) techniques are of great interest in most of the fields of numerical computation. These techniques reduce the effort needed to obtain a proper discretization for numerical analysis as regards man-hours, CPU-time and memory requirements. In addition, adaptive procedures make the accurate computation of the smaller scales of the flow easier, especially when we do not have *a priori* information concerning the solution, or it changes in time during the simulation (e.g. unsteady flows). Another field of study, which is closely related to *h*-adaptivity, involves the design of error estimates and flow indicators aimed at guiding the automatic adaptative procedures. This topic has received much attention in computational mechanics, not only in relation to grid adaptation but also in the identification and quantification of numerical error.

The fact that meshless techniques do not require a conforming grid, subject to various topology and quality conditions, reduces the discretization requirements significantly. This characteristic provides to the meshless approach some advantages over conventional mesh-based techniques, especially for addressing problems involving automatic grid adaptivity, moving or deforming boundaries and changing domains, among other features that can be very demanding in terms of the spatial discretization. In order to start exploring the meshless capabilities, procedures for automatic grid adaptation and error identification in the context of compressible flow problems are developed in this chapter.

Next, an *h*-adaptive procedure is proposed in Section 6.1 and several numerical applications are presented in Section 6.2 to illustrate the performance of the algorithm. In the test cases presented, a simple feature-based indicator is used to guide the adaptive procedure. Then, focusing on this particular aspect of the methodology, error identification and evaluation techniques are discussed in Section 6.3 and a truncation-error based estimate is proposed with the aim to achieve a more effective improvement of the solution (note that this kind of indicator is more closely related to the discretization error). The behavior of the truncation-error based indicator is studied in relation with reliable solution error estimates, and comparisons with the simpler feature-based indicator are also provided.

## 6.1 The *h*-adaptive procedure

The proposed meshless adaptive procedure can be explained in three main steps: the identification of areas in which either local refinement or coarsening of the discretization is required, the removal and insertion of points, and the update of the related data structures. The latter involves the construction of the data associated to the new points and the re-construction of the data associated to clouds of points affected during the adaptive procedure. These basic steps are described below.

### 6.1.1 Refinement/coarsening indicator

Local clouds in the analysis domain where either refinement or coarsening is required can be identified according to several criteria. Basically, the local indicators can be based on detectors of particular flow features in the solution field or on numerical error estimates (these measurements may be related). A solution feature-based indicator is presented in this section.

The feature-based indicator proposed measures in an approximate manner the curvature of the solution and is defined at each star point $\boldsymbol{x}_\mathrm{i}$ by

$$r_i = \sum\nolimits_e \left| \boldsymbol{l}_{ij} \cdot \left( \nabla \rho_j - \nabla \rho_i \right) \right| \tag{6.1}$$

where $e$ indicates the edges in the first layer of nearest neighbors of $\boldsymbol{x}_i$ (obtained in the local cloud construction stage), $\boldsymbol{l}_\mathrm{ij} = \boldsymbol{x}_\mathrm{j}$-$\boldsymbol{x}_\mathrm{i}$ and ρ is the density of the fluid (note that the solution gradients are nodally available in the FPM). Naturally, another flow variable or a combination of flow variables can be adopted for calculating the refinement indicator. In this work the density field is employed as it is sensitive to all types of basic flow discontinuities.

The indicator (6.1) is not directly used to guide the adaptive process. To this end, the following smoothed dimensionless indicator is employed

$$\varphi_i = \log\left( \frac{1}{\rho_i} \frac{1}{n_e} \sum_{j=1}^{n_{nei}} r_j \right) \tag{6.2}$$

where $n_e$ is the number of edges and the logarithm is taken in order to compress the scale of the distribution. Then, the mean ($\varphi_\mathrm{m}$) and standard deviation ($s_\varphi$) of the φ's distribution are computed and local clouds in which refinement or coarsening is required are determined as follows:

f. the cloud is tagged for refinement when $\varphi_i > \varphi_\mathrm{m} + n_r s_\varphi$

g. the point $\boldsymbol{x}_\mathrm{i}$ is marked to be removed if $\varphi_i < \varphi_\mathrm{m} - n_c s_\varphi$

being $n_r$ and $n_c$ threshold parameters that must be set according to the problem under study. The sensitivity of the solution to these values is not high in practice and satisfactory results are generally obtained with $n_r \approx 1$ and $n_c = 0$.

Notice that the feature-based indicator (6.2) only provides detection of relatively high-gradient areas where it is assumed that the solution error will decrease as the grid is refined, but the error is not actually measured.

### 6.1.2 Removal of points

The adaptive stage begins with the removal of points tagged for deletion (this is essential for treating non-stationary problems). The coarsening of surface grids is performed by collapsing edges with marked nodes. Interior volume points are simply deleted from the vertices list. After the removal of points the data structure is updated. It is important to note that the removal of points is restricted only to the existing points that have been inserted in prior refinement levels. In other words, the initial set of points (original coarse discretization) is conserved through the calculation, although the spatial position of these points could change due to smoothing. This criterion avoids additional verifications to guarantee a minimum geometrical support for the calculation.

### 6.1.3 Surface refinement

The surface grids are refined after the coarsening stage. To this end, surface elements having tagged all their nodes are selected and a new point is inserted at its centroid (and the element is subdivided) if the distance to any other point in the discretization is greater than a minimum cut-off distance $h_{min}$. The latter determines the allowable level of refinement and is computed for the original discretization as an user-defined percentage of the distance between each point and the nearest neighbor in its cloud. For new points added during refinement stages, $h_{min}$ is inherited from the original points distribution. Once new boundary points are inserted, their positions (originally coincident with the centroid of the underlying element) are slightly improved by interpolation; see (Karbacher, Seeger & Häusler, 2001). This avoids excessive faceting of the surfaces in successive element subdivisions. Finally, edge swapping is applied on the affected boundary elements to improve their connectivity, thus facilitating the insertion and removal of points in future adaptive passes.

### 6.1.4 Volume refinement

Points in the interior of the fluid tagged for refinement are processed after surface refinement is completed. To this end, at the local cloud level, the Voronoi vertices surrounding the star point $\boldsymbol{x}_i$ are computed by means of its

Delaunay grid of nearest neighbors. Next, a new point is added at the location of each Voronoi vertex if the distance from the latter to any existing point is greater than $h_{min}$. In addition, boundary constraints are checked to verify that any ray from the star point to the new point does not pierce any surface (see Section 3.6.2). A dynamically updated bins structure is employed to perform the many spatial search operations required with efficiency.

### 6.1.5 Update

A few steps of a Laplacian smoothing are carried out on the affected area when the coarsening/refinement stage finishes. Then, the clouds of points and the local approximation are constructed for the new points, and the data related to existing clouds affected by deletion, insertion of new points or smoothing is re-constructed. Finally, the flow and other problem variables at the new points are calculated simply as an average of the variables at their previously existing layer of first nearest neighbors.

## 6.2 Application examples

Some numerical examples are presented next to illustrate the basic behavior of the proposed adaptive procedure. The applications focus mainly on showing detection capabilities and the convergence of the adaptive solutions in stationary problems. Additional test cases are presented later in this chapter to investigate the accuracy of the methodology in more detail.

### 6.2.1 Supersonic flow past a double-wedge airfoil

A double-wedge airfoil with unit chord c = 1 and wedge angle $\beta = 20^{\circ}$ is solved here for an upstream Mach number $M_{\infty} = 2$ and zero incidence. The initial coarse discretization is composed by an unstructured distribution of 1279 points and quadratic approximations are built on clouds with 15-20 points. Third-order MUSCL extrapolation (with van Albada limiter) and a 4-stage time marching scheme are adopted. The number of refinement levels is set arbitrarily large in this simulation (and $h_{min}$ small) in order to verify the convergence of the numerical solution. The first refinement pass is triggered when the density residual (L2 norm) drops below 1.0E-5. Then, consecutive refinement levels are carried out every 200 time steps. The final adapted discretization consists of 51907 points (70 refinement levels). The initial and final adapted discretizations are shown in Figure 52.

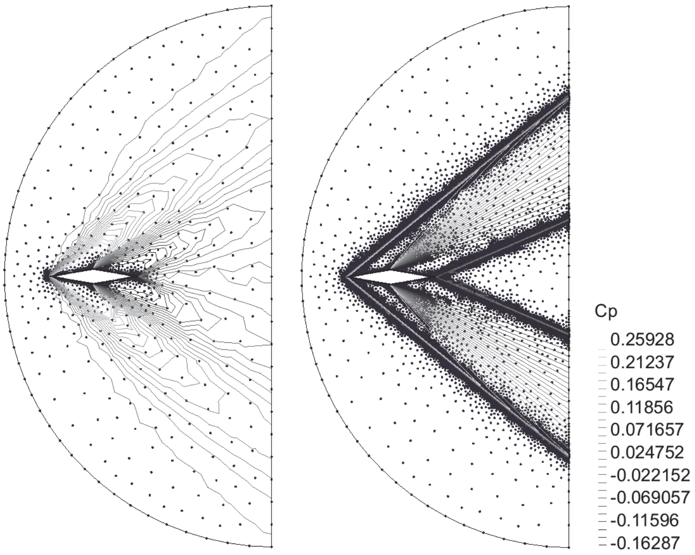

Figure 52. Original (left) and final adapted (right) discretizations showing Cp isolines. Double-wedge airfoil, $M_\infty$=2.0 and α=0º.

A comparison between the analytical solution of the problem (Cp) and the numerical solution calculated at some refinement levels is presented in Figure 53. The results are obtained along an x-cut in the domain, located 0.1c above the airfoil chord-line. It is possible to observe the convergence of the numerical solution as the grid is refined. Finally, the convergence history of the problem is shown in Figure 54. Note that the peaks correspond to each refinement level performed during the computation. The lower rate of convergence observed after refinement can be caused by the continuous action of limiters in the areas of finer discretization.

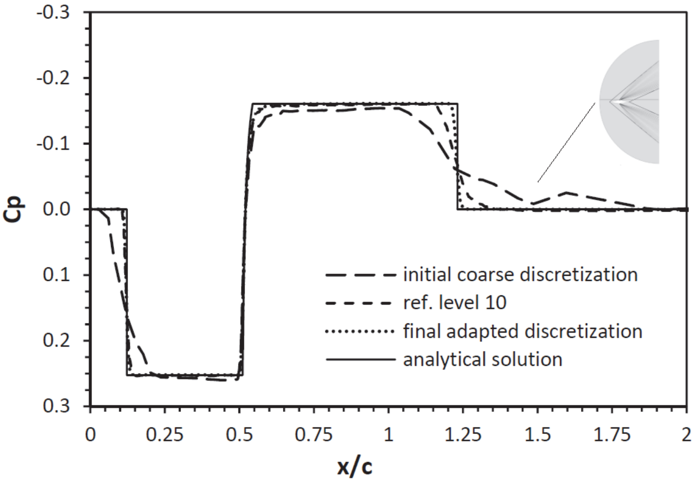

Figure 53. Comparison of analytical and numerical Cp distributions along an x-cut on the domain at different refinement levels. The cut is located at y/c=0.1 and the airfoil leading edge coincides with (x,y) = (0,0). Double-wedge airfoil, $M_\infty$=2.0 and α=0º.

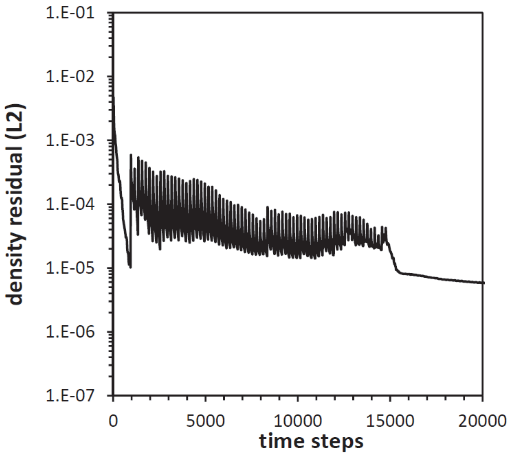

Figure 54. Convergence history (70 refinement levels). Double-wedge airfoil, $M_\infty$=2.0 and $\alpha$=0.0º.

### 6.2.2 The shock tube problem

The shock tube problem has been already described in the context of verification test cases (see Section 4.6). In the present adaptive application it is solved in a two-dimensional domain $\Omega$=(1x0.5) for a pressure ratio 10. The initial coarse discretization consists of a homogeneous distribution of 968 points and quadratic approximation bases in clouds having 15-20 points are adopted. Third-order MUSCL extrapolation with Van Albada limiter and a 4-stage time marching scheme are employed in the simulation. The successive refinements passes are performed at a fixed number of time steps. The simulation time run in this example is t=0.2 seconds. Figure 55 presents some snapshots of the point discretization taken at different times from the rupture of the diaphragm. The final discretization reaches a total of 2847 points.

Numerical solutions obtained for density at times t = 0.14 and 0.20 seconds are compared with the analytical solution in Figure 56. As it can be observed, the results seem to be a bit diffusive and small inaccuracies can be also seen in the shock position. This behavior may be caused by geometrical and allowable $h_{min}$ restrictions. These limit the quantity of new points inserted at a given refinement level, thus making the discretization unable to adapt instantaneously to a fast varying flow solution in a proper manner. Furthermore, the simple interpolation procedure adopted and the Laplacian smoothing can also play a role. Nevertheless, given the characteristics of the problem, the results are deemed satisfactory. A more efficient approach to deal with problems involving moving discontinuities will be proposed in the next chapter.

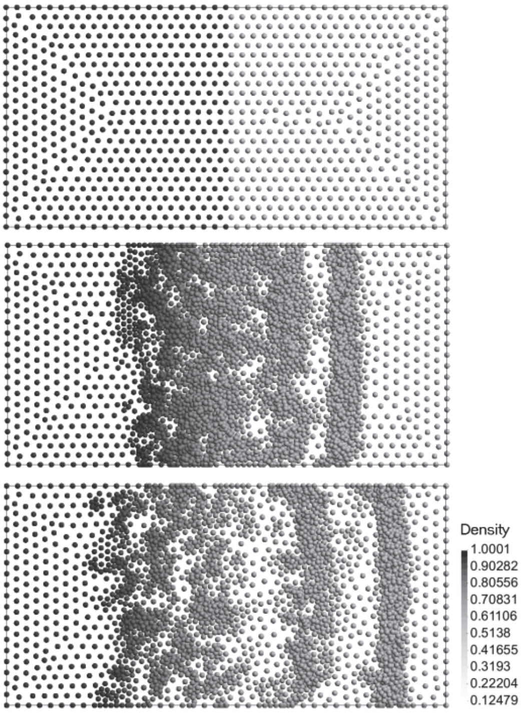

Figure 55. Adapted discretizations at different times from the rupture of the diaphragm (Top: initial discretization; Middle: 0.14s.; Bottom: 0.20s.). Shock-tube problem, $p_L/p_R$=10.

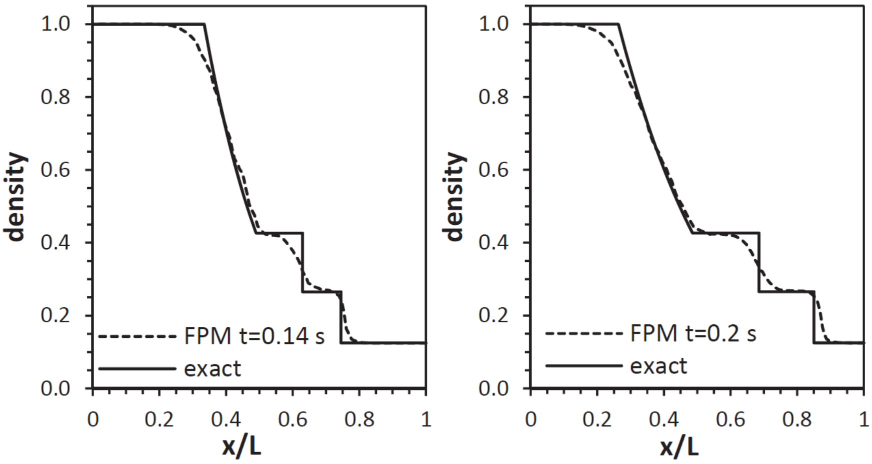

Figure 56. Numerical and analytical solutions for the density along the centerline of the shock tube at two different times from the rupture of the diaphragm ($p_L/p_R$=10).

### 6.2.3 Transonic flow around a NACA 0012 airfoil

This example concerns the computation of a transonic inviscid flow around a NACA 0012 airfoil. The freestream Mach number is $M_\infty$ = 0.8 and the incidence angle is $\alpha$ = 1.25º. The initial discretization consists of an unstructured distribution of 976 points, and quadratic approximation bases are used in clouds with 15-21 points. Third-order MUSCL extrapolation with Van

Albada limiter and a 4-stage time marching scheme are employed in the numerical simulation. The initial and final adapted point distributions are shown in Figure 57. The finest discretization consists of 4938 points and is achieved after 15 refinement levels. It is possible to note that the strong shock wave on the upper side of the airfoil, the weaker shock on its lower side, and the leading and trailing edge regions are properly captured by the refinement procedure.

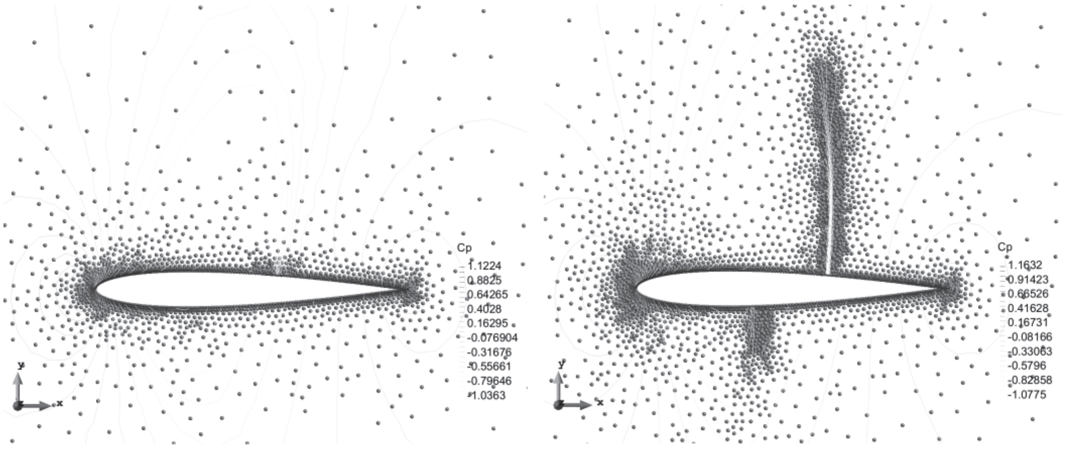

Figure 57. Original coarse and finest adapted discretization in the proximity of the airfoil. NACA 0012, $M_\infty$=0.8 and $\alpha$=1.25º.

The airfoil Cp distribution calculated on the finest discretization is displayed in Figure 58. A satisfactory agreement with the AGARD numerical results presented in (Pulliam & Barton, 1985) is achieved. Finally, the time convergence history of the problem is presented in Figure 59.

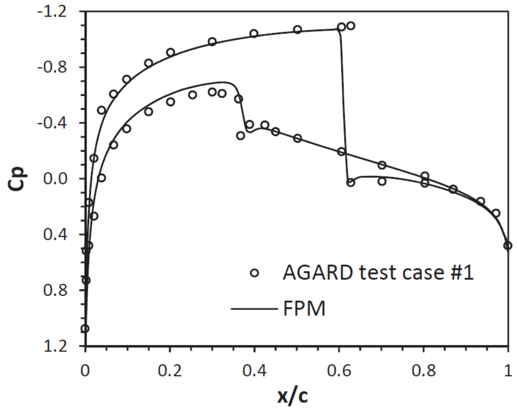

Figure 58. Comparison of computed Cp distribution (final adapted discretization) and AGARD numerical reference results. NACA 0012, $M_\infty$=0.80 and $\alpha$=1.25º.

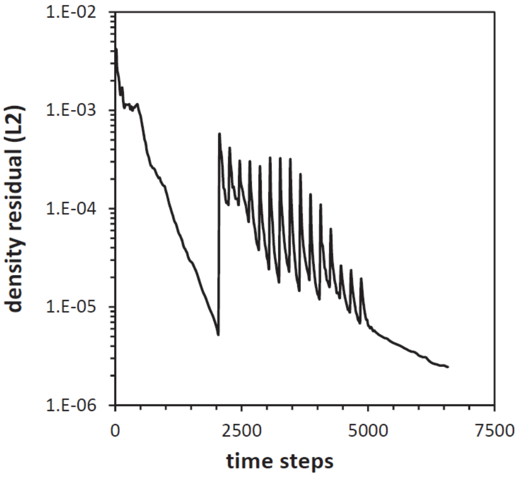

Figure 59. Convergence history (15 refinement levels). NACA 0012, $M_\infty$=0.80 and $\alpha$=1.25º.

### 6.2.4 ONERA M6 wing

The flow around the ONERA M6 wing is solved in this example for the freestream conditions given in Section 4.6.3. The initial coarse discretization consists of an unstructured distribution of 66864 points and quadratic approximation bases are employed in clouds with 30-45 points. The solution scheme uses third-order MUSCL reconstruction with Van Albada limiter and a 4-stage time marching scheme. In this simulation, the adapted discretization reaches a total of 102592 points after 35 refinement levels. Figure 60 shows a view of the original and final discretization of the wing (upper side). The original surface model has 14221 points and 28314 triangle elements and the final discretization achieves 15537 points and 30942 triangles.

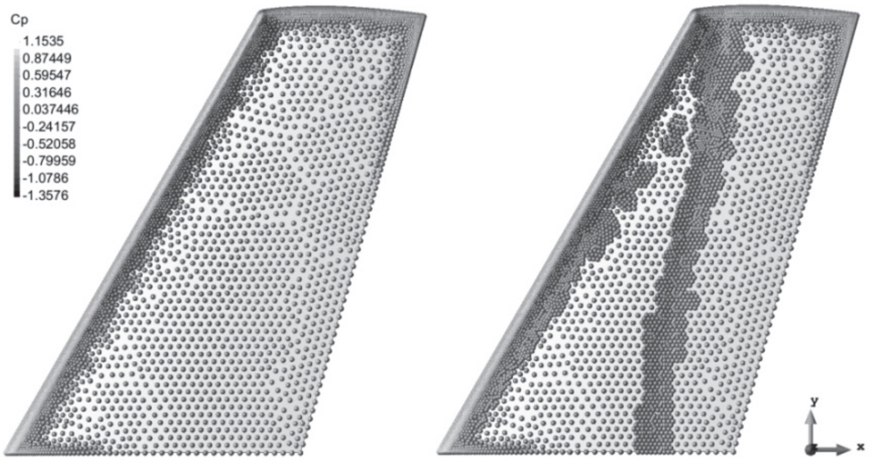

Figure 60. Original coarse and final adapted discretizations (35 refinement levels). ONERA M6 wing, $M_\infty$=0.84 and $\alpha$=3.06º.

Pressure distributions obtained for the original and finest discretization are compared along two wing sections in Figure 61. There, a view of the finest point discretization along the same spanwise stations is presented. The convergence of the density residual is shown in Figure 62.

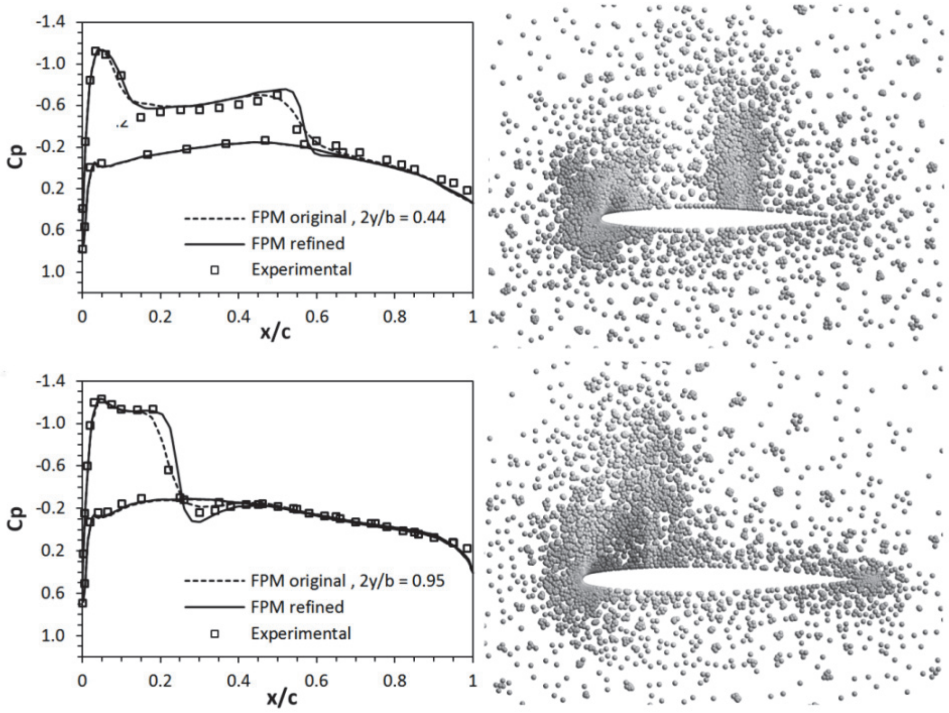

Figure 61. Cp distributions along sections η=0.44 (top) and η=0.95 (bottom) calculated with the original and the final discretizations. Cuts x-z of the finest domain passing through the same stations are also shown (right). ONERA M6 wing, $M_\infty$=0.84 and α=3.06º.

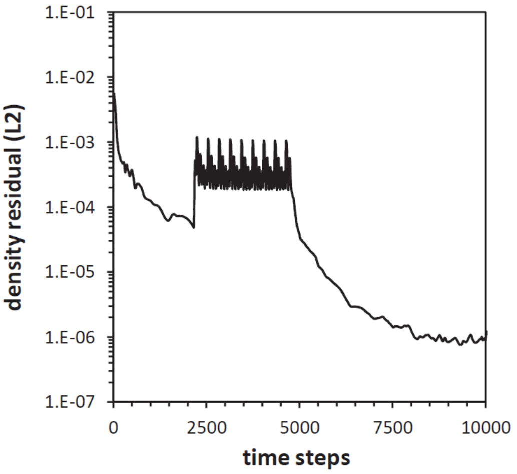

Figure 62. Convergence history (35 refinement levels). ONERA M6, $M_\infty$=0.84 and α=3.06º.

The computational cost of a single refinement pass in this example has proved to be approximately equal to the time required in the update stage. As the number of clouds to be processed is generally small compared with the total number of clouds in the domain, the CPU-time involved is usually low. Some timings will be given later in Section 7.4.

## 6.3 Error estimation

The identification and quantification of numerical errors is a subject which has been largely studied in computational mechanics, not only with the aim to perform local or global solution improvements (e.g. through grid adaptation), but also to determine the accuracy and reliability of the computational code and the numerical results. Error taxonomies and analysis methodologies developed with these aims in the framework of verification and validation (V&V) are becoming standard practices in numerical computation; see for instance (Oberkampf & Trucano, 2002; Roache, 1997; Roy, 2005; Thacker, Doebling, Hemez, Anderson, Pepin & Rodriguez, 2004) for an overview.

In the context of meshless methods, error estimation methodologies have followed the general lines of the approaches developed for conventional grid-based methods, but aiming at exploiting meshless advantages regarding problem data structures and topological requirements. In this sense, meshless approaches have a considerable potential to deal with general unstructured discretizations and, for example, this makes it easier to address higher-order approximations and reconstruction processes needed in error estimation techniques. Among the first meshless applications, error estimates based on residuals (Duarte & Oden, 1996a) and wavelets (Liu, Jun, Sihling, Chen & Hao, 1997) can be found. More recently, enhanced, recovered or higher-order solution fields have been employed to construct error estimates and indicators (Angulo, Pérez Pozo & Perazzo, 2009; Gavete, Cuesta & Ruiz, 2002; Lee & Zhou, 2004; Li & Lee, 2006; Rabczuk & Belytschko, 2005). Also, error functionals steaming from the construction of the meshless approximation have been derived, see for instance (Afshar & Lashckarbolok, 2008; Perazzo, Löhner & Perez-Pozo, 2007). These methodologies have been successfully applied to locally improve the numerical solution (reducing discretization errors) through *p* or *h*-adaptation.

In this section, following the ideas adopted in discrete error transport equations (DET) methods, an estimate of the numerical solution error is developed with basis on the truncation terms of the governing equations. The proposed estimate is mainly intended to guide the adaptive procedure (error identification) and is expected to be more effective than feature-based indicators (such as proposed in Section 6.1) in the reduction of local and global solution errors. The basic aspects of DET methods and the methodology proposed to compute the meshless indicator are presented below. Examples of application in compressible flows are also provided.

### 6.3.1 Discrete error transport equations

Methods based on discrete error transport equations directly model the generation and transport of error in the computational domain, see for instance (Hay & Visonneau, 2006; Shih & Qin, 2007; Zhang, Trépanier & Camarero, 2000). These equations are obtained from the definition of the discretization error, using linearized forms of the problem governing equations in which the truncation error, i.e. the difference between the original governing equation and its discrete approximation, acts as a source term. Due to the fact that explicit forms of the truncation error are difficult to obtain in general problems, approximated expressions have to be derived. Typical approximations have a close connection with differential residuals[60], see for example (Roy, 2009). It is important to note that the key role truncation terms play in DET equations makes them suitable candidates to evaluate discretization errors. An application in which a normalized form of the truncation error is adopted as an approximate solution of DET equations can be found in (Muzaferija & Gosman, 1996). Moreover, the proportionality between truncation and local and global discretization errors (and thus solution accuracy) is demonstrated in (Blottner & Lopez, 1998) over a considerable range of problems and simple discretization schemes. It should be noticed that this kind of estimators are also efficient to drive mesh adaptation processes because the truncation error terms account for both, mesh quality and flow features contributions to the discretization error; some applications can be found in (Aftosmis & Berger, 2002; Baker, 1997; Berger & Jameson, 1985; Roy, 2009).

[60] Methods based on differential residuals were initially developed in the FEM framework; see (Babuska & Miller, 1984; Oden, Wu & Ainsworth, 1993). Although typically these residuals are obtained by inserting an approximate solution into a modified functional form of the problem equations, these can be also computed by using an enhanced solution field with the same set of discrete equations. Non-zero local (and global) residual values can be related to a lack of enforcement (or balance) in the governing equations and this, in turn, to errors in the conserved quantities affecting the solution and output functionals (although the relationships are not straightforward). Applications using residual imbalances (e.g. in kinetic energy, momentum, etc.) have proven to be useful to guide grid adaptation and also to perform as surrogate error indicators in output quantities; see for instance (Chang & Haworth, 1997; Oden & Prudhomme, 1999) and more recently (Oñate, Arteaga, García & Flores, 2006), where power forms of mass and momentum imbalances are used to guide mesh adaptivity in incompressible flow problems.

### 6.3.2 A truncation-error meshless estimate

Following DET ideas, an estimate of the discretization error is developed in this section. This error, which usually is the largest affecting the numerical solution, depends on the underlying discretization and the characteristics of the numerical scheme, and can be defined as

$$e_h = \phi - \phi_h \tag{6.3}$$

where $\phi$ is the exact solution of the continuous problem, $L(\phi)=0$, and $\phi_h$ is the exact solution of its discrete counterpart, $L_h(\phi_h)=0$, computed on a grid with characteristic size *h* (iterative and round-off errors are considered negligible). Assuming that the discrete problem operator $L_h$ is linear (or has a linearized form) and applying it to Eq. (6.3), the following equation for the discretization error can be obtained

$$L_h(e_h) = L_h(\phi) - L_h(\phi_h) = L_h(\phi) - 0 = L_h(\phi) \tag{6.4}$$

which can be solved with the same numerical method employed to obtain the discrete solution, but with an extra source term which is responsible for the local generation and destruction of error in the problem domain. The equivalence between the source term $L_h(\cdot)$ (discrete differential residual) and the problem truncation error ($\tau_h$) can be demonstrated by using the following relation between the differential operators

$$L(\cdot) = L_h(\cdot) + \tau_h(\cdot) \tag{6.5}$$

where $(\cdot)$ denotes an arbitrary problem solution (Roy, 2009). Then, introducing $\phi$ in Eq. (6.5) the term $L(\phi)$ vanishes, and substituting the result into Eq. (6.4) it is possible to obtain

$$L_h(e_h) = -\,\tau_h(\phi) \tag{6.6}$$

which is the basis of methods using DET equations. Due to the fact that the truncation error and the problem solution are usually unknown in practical problems, approximate approaches are employed to solve Eq. (6.6).

Given that the main focus in this work is on error identification and adaptivity, an accurate solution of Eq. (6.6) is not required. Thus, a low-cost surrogate measure of $e_h$ is proposed. That measure is based on the problem truncation error and exploits the fact that the latter acts in Eq. (6.6) as a source term which behaves proportionally to the true discretization error (see proofs in (Blottner & Lopez, 1998)). Hence, neglecting transport phenomena, it is assumed that locally $e_h \propto \tau_h$ and a surrogate error measure is obtained from an approximate form of $\tau_h$ via Eq. (6.5), i.e.

$$\tau_h(\phi) = -\,L_h(\phi) = -\,R_h(\phi) \tag{6.7}$$

where R$_h$(·) is the discrete differential residual of the problem and the unknown exact solution ϕ is replaced by an accurate estimate.

In the context of compressible flow problems governed by the discrete equations (4.15), the following form of the surrogate error is proposed

$$\boldsymbol{e}_{\tau_i} = \left|\boldsymbol{R}_h^i\left(\mathbf{U}^*\right)\right| = \left|\frac{\Delta\mathbf{U}_i^*}{\Delta \mathrm{t}} + \sum\nolimits_j b_{ij}^k \mathbf{F}^k\left(\mathbf{U}_j^*\right)\right| \qquad \forall j \in \Omega_i \tag{6.8}$$

where **U*** is an accurate estimate of the problem solution, $\mathbf{F}^k$ is the *k*-component of the convective flux vector, and $\boldsymbol{b}_{ij}$ are the metric approximation coefficients for the cloud $\Omega_i$. In addition, a global measure of the numerical error over the whole domain is obtained from the local error vector (6.8) according to the following L2 norm

$$\left\|\boldsymbol{e}_\tau\right\| = \left(\frac{1}{n}\sum_i \left(\boldsymbol{e}_\tau^2\right)_i\right)^{1/2} \tag{6.9}$$

in which *n* is the total number of points in the computational domain. Note that additional dissipation terms coming from the upwind flux discretization are not accounted for in Eq. (6.8). This approach reduces the smoothing of the error estimate around discontinuities (enhancing detection capabilities) as well as the computational cost.

### 6.3.2.1 Computing accurate solution estimates

The accurate estimates of the exact solution needed in Eq. (6.8) are computed in this work in terms of the underlying FPM approximation and the discrete problem solution. To this end, Eq. (3.11) is recast in a more suitable form by exploiting the partition of unity property of the approximation and scaling by a half the stencil of points surrounding $\boldsymbol{x}_i$. This leads to the following estimate of the unknown exact solution

$$\phi_i \approx \phi_i^* = \hat{\phi}_i + 2\sum_{i \neq j} a_{ij}\left(\phi_{ij} - \hat{\phi}_i\right) \tag{6.10}$$

where $\phi_{ij}$ are samples of a higher-order reconstructed solution field taken at the midpoint of the edges connecting the star point $\boldsymbol{x}_i$ with any point $\boldsymbol{x}_j$ in the local cloud (see Figure 12 in Chapter 4) and $\hat{\phi}$ are local values of the discrete (actual) FPM solution (Eq. (3.11)). The objective of the sampled values $\phi_{ij}$ is to introduce locally improved information into the FPM approximation process in order to obtain a more accurate approximation to the exact solution. Note that Eq. (6.10) can be seen as a projection operator transferring information from an unknown finer grid solution into the actual discretization.

The methodology adopted in this work to compute the reconstructed solution samples $\phi_{ij}$ is inspired on the slope-limited geometric methods discussed in Section 4.2.2[61]. Typically, these techniques reconstruct a solution between two points $\boldsymbol{x}_\mathrm{i}$ and $\boldsymbol{x}_\mathrm{j}$ by fitting high-order interpolants on an enlarged spatial support. In addition, non-linear limiters should be introduced in the reconstruction process to avoid numerical oscillations across discontinuities, and this step can contaminate the recovered solution field (see for instance (Berger & Aftosmis, 2005)). Although slope-limited reconstruction has proven effective in the present context, the numerical experiments performed have shown an undesirable dependence of the level of accuracy obtained on the limiter chosen. Aimed at circumventing this drawback, a more robust reconstruction technique is described in the next section.

### 6.3.2.1.1 ENO polynomial reconstruction

Essential non-oscillatory schemes (ENO) (Harten, Engquist, Osher & Chakravarthy, 1987) are a family of reconstruction methods in which the solution interpolant is chosen between the smoothest of piecewise polynomials computed for the same sampling location on different local supports (stencils). As the selection procedure adopted avoids interpolating across discontinuities, limiting is not necessary. ENO schemes have also interesting properties, such as conservation of the mean, and guarantee total variations bounded by the truncation error of the reconstruction. Improvements to the basic ENO technique have been also proposed to solve certain deficiencies concerning the need of large stencils (difficult to obtain in unstructured discretizations) and the lack of accuracy in smooth data reconstruction problems; see for instance Weighted-ENO (WENO) (Liu, Osher & Chen, 1994) and Hermite-WENO (HWENO) (Qiu & Shu, 2003).

In this work a standard ENO technique is employed to compute the $\phi_{ij}$ samples required in Eq. (6.10). Thus, an enhanced solution field is

[61] Notice that the estimates $\phi_i^*$ could also be obtained by using higher-order accuracy schemes. Even though this methodology is less efficient from the computational point of view, meshless-based implementations offer some attractive alternatives (Barth, 1991; Cueto-Felgueroso, Colominas, Nogueira, Navarrina & Casteleiro, 2007; Zhang & Naga, 2005). Hence, a preliminary investigation with higher-order FPM approximations was performed in this work. The results obtained, though satisfactory, evidenced a considerably dependence on the geometrical characteristics of the cloud and the problem setting (see comments in Section 3.4.2). In addition, some lack of robustness was also detected due to the large spatial support required for constructing the approximants (especially in 3D problems). Although these facts make geometric extrapolation techniques more attractive in the present context, further investigation on the employment of higher-order approximations should be required.

reconstructed in the interval $\mathbf{I}_i=[\boldsymbol{x}_{i-1/2},\boldsymbol{x}_{i+1/2}]$ (in the direction of the edge $\boldsymbol{l}_{ij} = \boldsymbol{x}_j-\boldsymbol{x}_i$) by one of the following central, left and right-biased quadratic interpolation polynomials (Serna & Marquina, 2004),

$$\begin{aligned} p_{i-1}(x') &= \hat{\phi}_i - \frac{D_{i-1}}{24} + x'\left[d_{i-1/2} + \frac{D_{i-1}}{2}\left(x'+1\right)\right] \\ p_i(x') &= \hat{\phi}_i - \frac{D_i}{24} + x'\left[d_i + \frac{D_i}{2}x'\right] \\ p_{i-1}(x') &= \hat{\phi}_i - \frac{D_j}{24} + x'\left[d_{i+1/2} + \frac{D_{i+1}}{2}\left(x'-1\right)\right] \end{aligned} \tag{6.11}$$

where $\hat{\phi}_i$ is the discrete (actual) problem solution at $\boldsymbol{x}_i$, $0 \le \boldsymbol{x'} \le 1$ a non-dimensional length parameter along the edge $\boldsymbol{l}_{ij}$ and

$$d_{i+1/2} = \hat{\phi}_{i+1} - \hat{\phi}_i \quad , \quad D_i = d_{i+1/2} - d_{i-1/2} \quad , \quad d_i = (d_{i+1/2} + d_{i-1/2})/2 \tag{6.12}$$

In order to evaluate first and second differences in Eq. (6.12), the discrete solution at fictitious points $\boldsymbol{x}_{j+1}$, $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_{i-2}$ in the direction of $\boldsymbol{l}_{ij}$ is required (see Figure 63). These values can be obtained by means of the following central and 2nd-order backward finite differences approximations

$$\begin{aligned} \hat{\phi}_{i-1} &\approx \hat{\phi}_j - 2\boldsymbol{l}_{ij}\cdot\nabla\hat{\phi}_i \\ \hat{\phi}_{j+1} &\approx \hat{\phi}_i + 2\boldsymbol{l}_{ij}\cdot\nabla\hat{\phi}_j \\ \hat{\phi}_{i-2} &\approx 4\hat{\phi}_{i-1} - 3\hat{\phi}_i + 2\boldsymbol{l}_{ij}\cdot\nabla\hat{\phi}_i \end{aligned} \tag{6.13}$$

where the nodal gradients are directly computed from the actual FPM solution by Eq. (3.12).

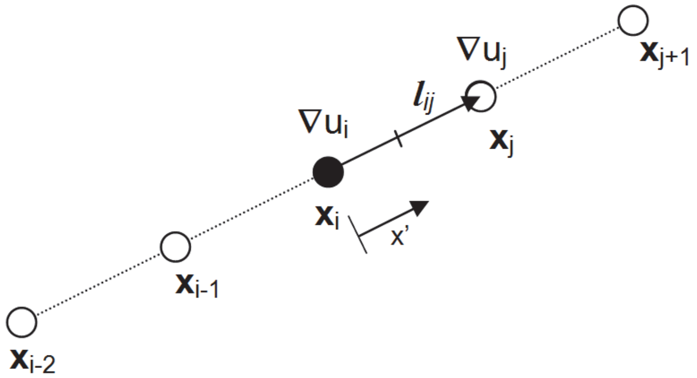

Figure 63. Stencil of points employed for cubic ENO reconstruction around $\boldsymbol{x}_i$ in the direction of the edge $\boldsymbol{l}_{ij}$.

The interpolation polynomial employed to reconstruct the enhanced solution field in the interval [$\boldsymbol{x}_{i-1/2}$,$\boldsymbol{x}_{i+1/2}$] along $\boldsymbol{l}_{ij}$ is chosen according to the standard ENO selection procedure

```
if $|d_{i-1/2}| \le |d_{i+1/2}|$ then
    if $|D_{i-1}| \le |D_i|$ then ; $p_{i-1}(x')$ ; else ; $p_i(x')$ ; endif
else                                                              (6.14)
    if $|D_i| \le |D_{i+1}|$ then ; $p_i(x')$ ; else ; $p_{i+1}(x')$ ; endif
endif
```

and the edges' midpoint samples required in Eq. (6.10) are obtained by

$$\phi_{ij} = p_k\left(x'=1/2\right) \tag{6.15}$$

being $k = i\text{-}1$, $i$ or $i+1$ according to the polynomial chosen in (6.14). Note that the polynomials (6.11) allow a continuous reconstruction of the unknown function in the interval [$\boldsymbol{x}_{i-1/2}$,$\boldsymbol{x}_{i+1/2}$] and this fact enables to use a different cloud scaling in Eq. (6.10). The tests performed showed no significant influence of the sampling location on the resultant accuracy of $\phi_i^*$.

### 6.3.2.1.2 Numerical tests

The accuracy and convergence of the accurate solution estimates (6.10) is investigated in this section by studying the behavior of the effectivity index in several numerical approximation examples. This parameter is defined as (Zienkiewicz & Taylor, 2000)

$$\theta = {}^{\bar{e}}\!\!\diagup\!_{e} \tag{6.16}$$

where $\bar{e}$ = $\|\phi^* - \hat{\phi}\|$ is the error norm of the actual FPM approximation (Eq. (3.11)) referenced to the accurate estimate (6.10) and e = $\|\phi - \hat{\phi}\|$ is the exact error norm. The discrete models employed in the analysis have regular point discretizations and the refinement applied is uniform. Moreover, a similar setting of the FPM approximation is adopted to minimize the effects of the geometrical characteristics of the clouds and the weighting functions settings on the convergence of the approximation (see Sections 3.3 and 3.4).

The first example involves the approximation of a smooth Gaussian function defined in a one-dimensional domain [0,1] according to

$$f(x) = ae^{-(x-b)^2/2c^2} \tag{6.17}$$

where a=4/πδ , b=0.5, $c^2$=2δ and δ=0.005. The problem is initially discretized by a regular distribution of points with point spacing $h$ = 0.125. Then, successively refined discretizations are generated by halving $h$ with the objective to study the convergence of the effectivity index. The errors in Eq. (6.16) are computed in L2 norm and complete quadratic polynomial bases are employed in clouds with 5 points. Both, the standard FPM approximation (original) and the more accurate solution estimate (improved), are displayed in Figure 64 along with the evolution of the effectivity index (6.16). The sampled solutions show that a noticeably enhanced approximation is obtained with the estimate (6.10). A good convergence of the latter to the exact solution is also observed through the evolution of the effectivity index.

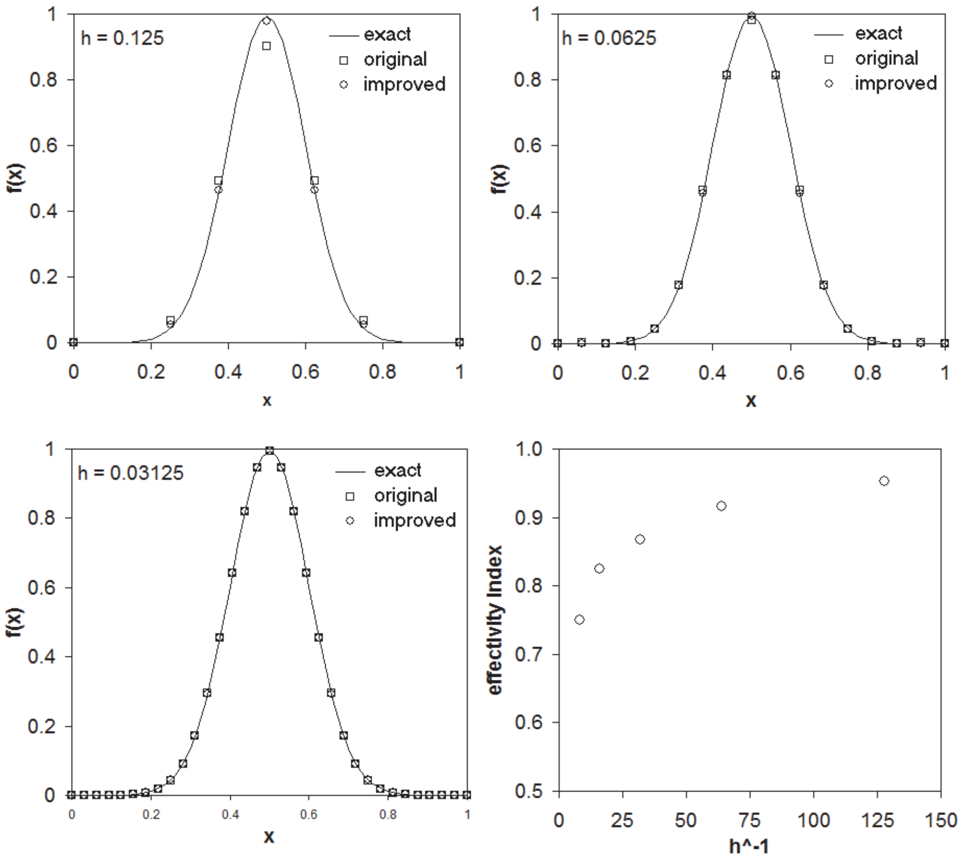

Figure 64. Approximation of a smooth Gaussian function in a one-dimensional domain.

A similar problem setting is employed in the second example, but this time a sign discontinuity is introduced in the Gaussian test function. The results obtained show a satisfactory performance of the accurate estimate (see Figure 65), but no asymptotic convergence of the effectivity index is achieved. This behavior can be explained due to the fact that the approximation near the discontinuity reverts to about O(h), thus hampering the convergence of the global error estimate.

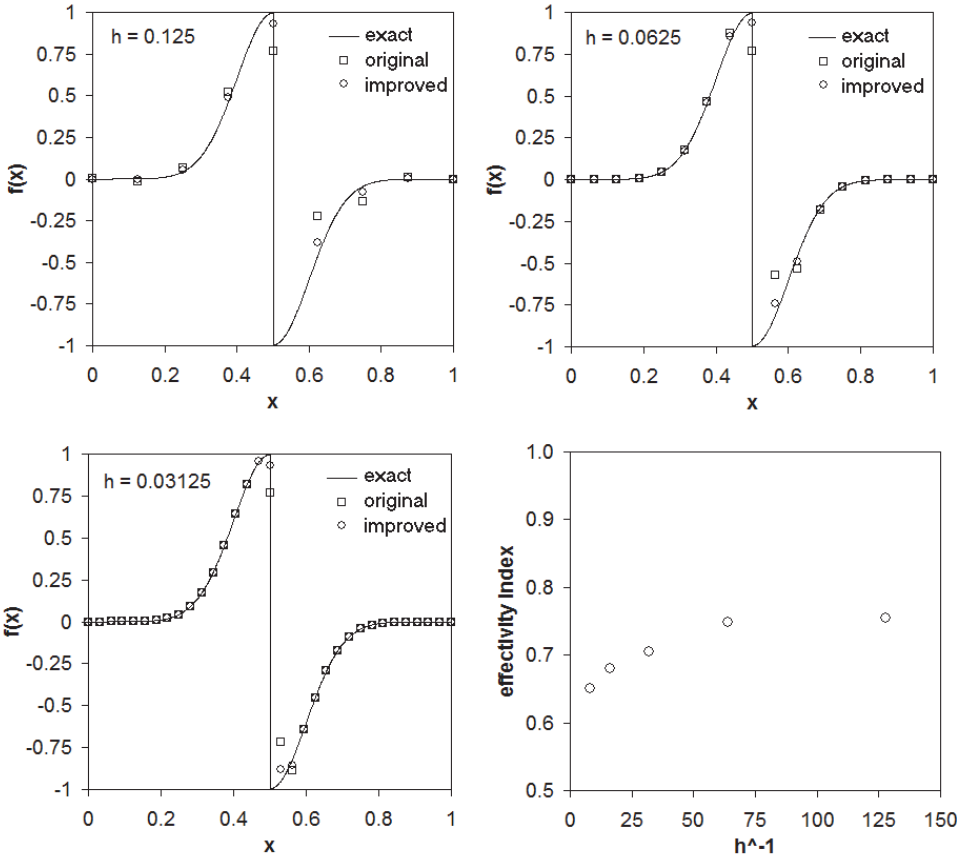

Figure 65. Approximation of a one-dimensional Gaussian function with a sign discontinuity.

The last two examples are intended to investigate the performance of the proposed accurate solution estimate in three-dimensional approximation problems. To this end, the following test function is adopted in a cubic analysis domain $(0,1)^3$

$$f(x,y,z) = 64x(1-x)y(1-y)z(1-z) \tag{6.18}$$

The initial discretization consists of an unstructured distribution of 1527 points (h≈0.1) and successive refined discretizations are obtained by halving the point spacing. Complete quadratic polynomial bases in clouds with 30 points are employed. As observed in Figure 66, a satisfactory convergence of the accurate estimate is also achieved in this example.

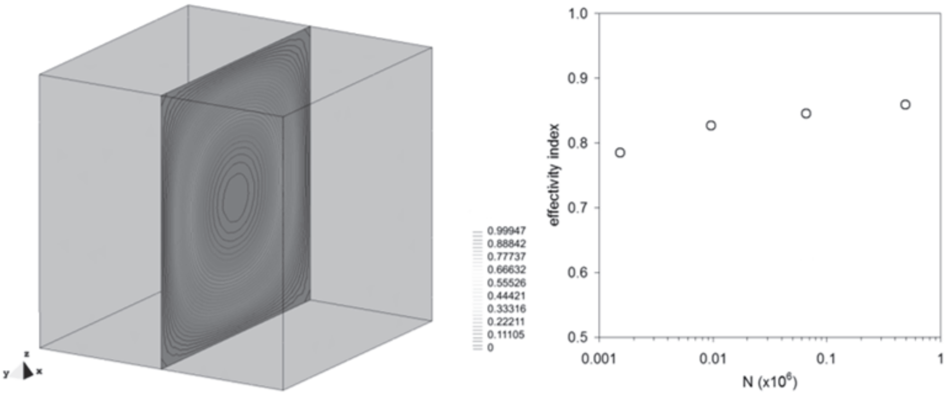

Figure 66. Behavior of the approximation error for a smooth three-dimensional function.

Finally, using a similar problem setting, a sign discontinuity at x=0.5 is introduced in the test function (6.18). The results presented in Figure 67 show a satisfactory accuracy of the solution estimate but, like the one-dimensional case (Figure 65), the discontinuity affects both, the global approximation error and its convergence rate. Overall, the tests performed indicate the suitability of the proposed methodology to obtain the exact solution estimates required to compute the surrogate local error (6.8).

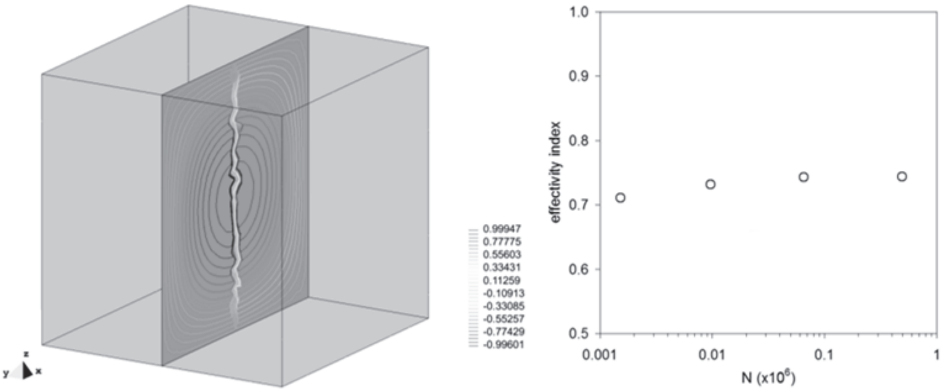

Figure 67. Behavior of the approximation error for a three-dimensional function with a sign discontinuity.

### 6.3.3 Applications to adaptive compressible flow problems

The performance of the truncation-error based indicator is investigated here by means of two additional adaptive test cases involving subsonic and transonic flows around an airfoil and a wing-body configuration. The focus in the numerical examples is on the relation of the truncation-error based estimate with the solution error (mainly convergence rate) and its suitability to drive efficiently adaptive procedures. Comparisons with the feature-based error indicator presented in Section 6.1.1 are also provided.

### 6.3.3.1 Transonic flow around a NACA 0012 airfoil

The first example involves the flow solution around a NACA 0012 airfoil presented in Section 6.2.3. In order to study the performance of the proposed error indicator, four discretizations with characteristic point spacings $h_1<h_2<h_3<h_4$ are generated from a coarse model ($h_4$) by scaling the grid sizes uniformly. In this way, discretizations ranging from 2487 to 5836 nodes are obtained with an averaged refinement ratio $r = h_{k+1}/h_k \approx (n_{k+1}/n_k)^{-1/2} = 1.15$. The simulation conditions adopted involve different freestream Mach numbers and angles of attack, namely: M = 0.3 at $\alpha$ = 1.25º and 4º and M = 0.8 at $\alpha$ = 0º, 1.25º and 4º. For each simulation test case, the observed order of accuracy of the scheme is computed by Richardson's extrapolation using the three finest discretizations (see a description of the procedure in Section 5.3.3). Values ranging between 1.54 and 1.87, with mean p = 1.74, are obtained. This averaged convergence rate is assumed to be representative of the true discretization error of the problem.

#### 6.3.3.1.1 Convergence assessment

The evolution of the density ($\mathbf{e}_{\tau1}$), momentum ($\mathbf{e}_{\tau2,3}$) and energy ($\mathbf{e}_{\tau4}$) components of the surrogate error estimate (6.8) with the grid refinement are presented in Figure 68. There, convergence rates close to that predicted by Richardson's method are obtained. This verifies both, that the current indicator is proportional to the discretization error, and that the truncation terms are satisfactorily approximated through the methodology presented in Section 6.3.2.

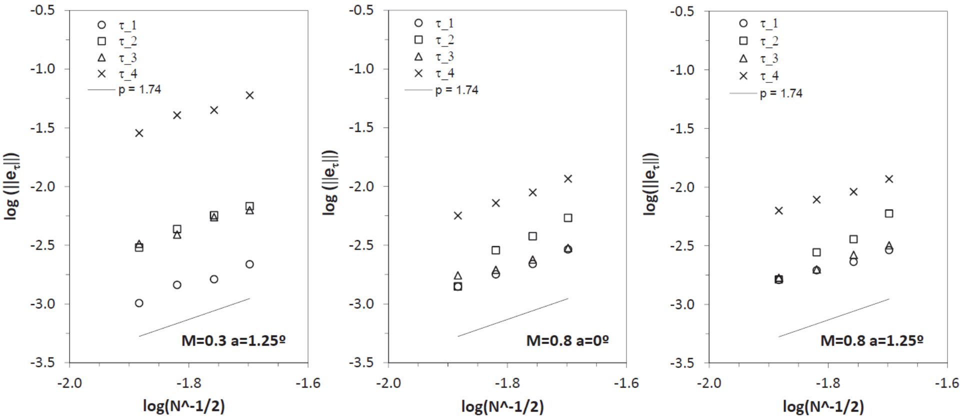

Figure 68. Convergence of the components of the global truncation error-based estimate computed for the NACA 0012 airfoil problem (the line p = 1.74 is only indicative of the slope).

The iterative convergence of Eq. (6.9) is displayed in Figure 69 for subsonic and transonic test cases. It is observed that the global error estimate does not vary significantly during the overall computation and converges after a short initial transitory, typically faster than the flow solution field. This result may indicate that the contribution of the underlying discretization (resolution and quality) to the truncation error prevails over the contribution of the flow solution itself. Thus, it can be advantageous in the sense that surrogate error measures based on the current estimate could be obtained quickly, regardless of the fact that full solution convergence is achieved or not. However, a reduced sensitivity of the estimate to the solution features may also cause a lack of correlation with actual solution errors, hindering an extrapolation of the results to variables of practical interest.

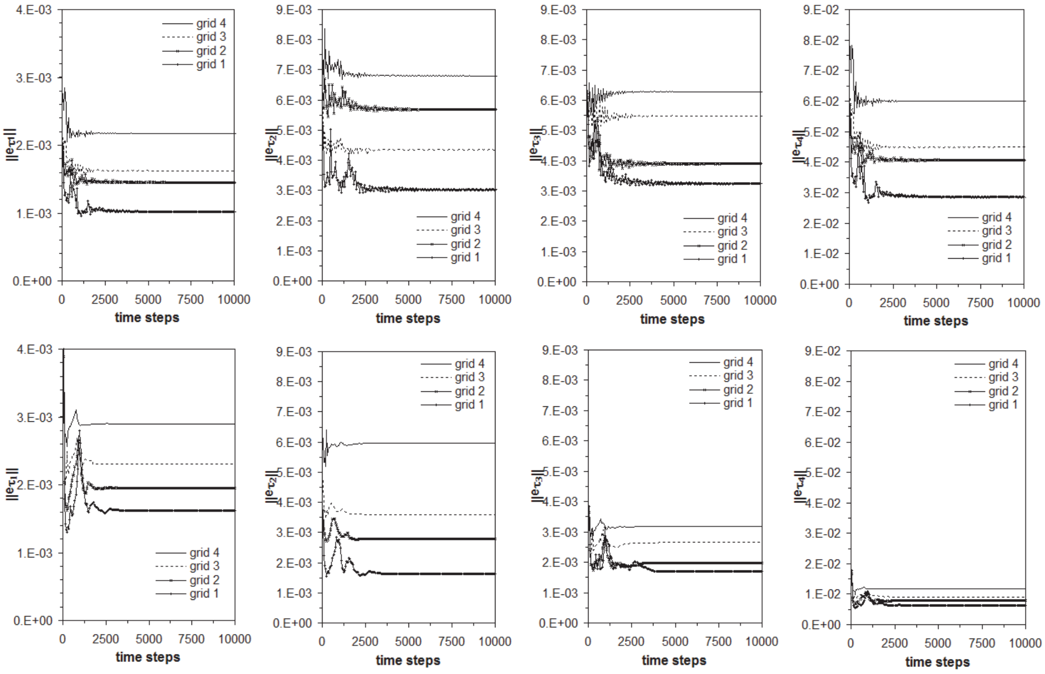

Figure 69. Iterative convergence of the components of the global truncation error-based estimate. NACA 0012 airfoil at α = 1.25º; top: M = 0.3, bottom: M = 0.8.

## 6.3.3.1.2 Automatic grid adaptation

The suitability of the local truncation error-based estimate (6.8) to drive grid adaptive procedures is investigated next by comparison with the feature-based indicator given by Eq. (6.1). As mentioned before, both error indicators are based on the flow density field. The adaptive test case involves a transonic inviscid flow with $M_\infty = 0.8$ and α = 1.25º. The initial domain discretization consists of an unstructured distribution of 2674 points. Four refinement levels are performed during the adaptive simulation; the first one is triggered after achieving solution convergence on the initial coarse grid and additional refinement passes are performed subsequently, ensuring the convergence of the intermediate problem solutions (a threshold parameter $n_r = 0.55$ is used).

A close view of the initial and final distributions of points around the airfoil is shown in Figure 70. The truncation error-driven procedure leads to a refined discretization with 6165 points, and 7105 points are obtained by using the feature-based indicator. It can be observed that the truncation error-based indicator not only concentrates new points around sharper solution features but also near the airfoil contour and the wake; two zones which are not markedly refined by the curvature-based indicator.

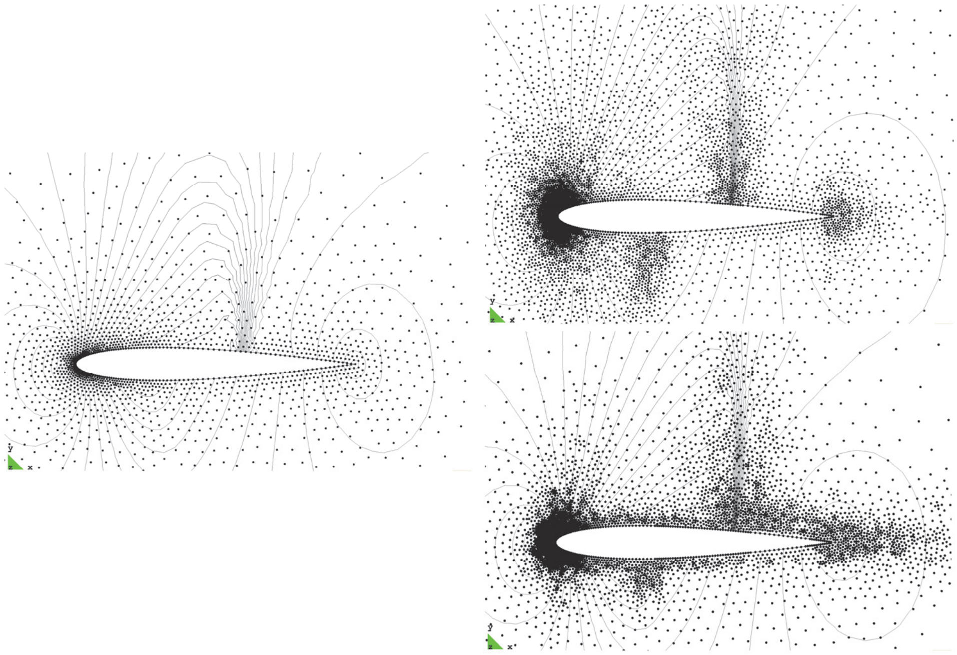

Figure 70. Initial coarse (left) and final adapted (right) discretizations. Curvature-based adaptivity (top) and truncation error-based adaptivity (bottom). NACA 0012 airfoil, $M_\infty$=0.8 and $\alpha$=1.25º.

Figure 71 displays the convergence of the normal force error ($C_{Nh}$) and the global estimate (6.9) with the grid refinement. The aerodynamic force error is computed in relation to a grid-independent result ($C_{Nf}$) which is obtained from a very fine uniform discretization (n=15021). The results show that a lower error in the resultant aerodynamic force of the problem is achieved with the truncation error-based indicator (see Figure 71 left), although the convergence rates observed in both methods are quite similar. As regards the global error estimate (see Figure 71 right), its magnitude decreases with the grid size at a rate close to the order of accuracy of the scheme, but the convergence speeds drop slightly after the second refinement stage. This effect is more notorious when the curvature-based indicator is employed (probably as a consequence of a more localized refinement).

Pressure distributions computed for the original coarse and final adapted discretizations are compared in Figure 72. It can be seen that truncation error-driven adaptivity resolves the upper shock better than the feature-based procedure (possibly due to a higher and more uniform density of points added near the airfoil) but the shock on the lower side of the airfoil is not well resolved. As mentioned before, this could reflect a reduced sensitivity of the proposed indicator to the solution flow features (particularly the weaker). Regarding leading and trailing edge airfoil zones, both indicators achieve similar results and the computed solutions coincide with that corresponding to the reference fine discretization.

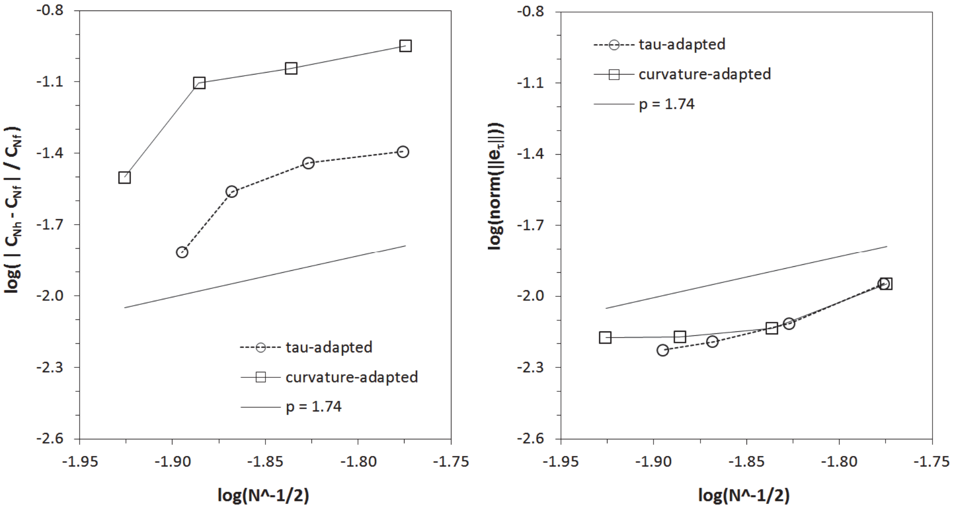

Figure 71. Behavior of the normal force error (left) and the numerical error estimate (right) with the grid refinement; norm($||e_{\tau}||$) denotes the norm of the global error vector (6.9). NACA 0012 airfoil, M=0.8 and α=1.25º.

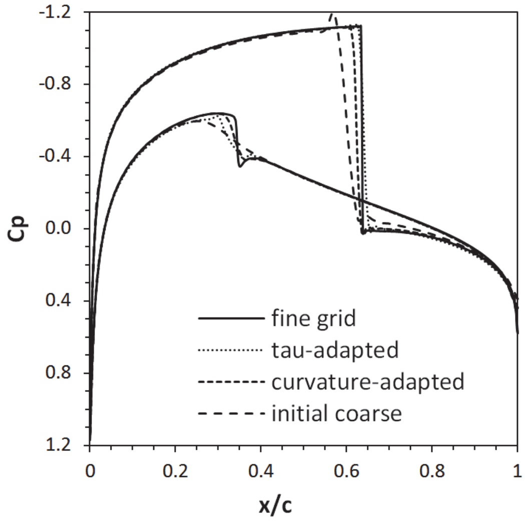

Figure 72. Pressure distributions obtained from the adapted flow solutions and the fine reference discretization. NACA 0012 airfoil, M=0.8 and α=1.25º.

### 6.3.3.2 Transonic flow around a NACA wing-body

The proposed methodology is applied next to the transonic solution of the NACA wing-body presented in Section 4.6.3 and the results obtained are compared with error estimates based on Richardson's extrapolation. The discretizations employed in the analyses are similar to those in Section 5.3, which range from 427900 to 2860318 nodes with 83162 to 694482 boundary triangles respectively. The resultant averaged refinement ratio is $r = h_{k+1}/h_k = (N_{k+1}/N_k)^{-1/3} = 1.23$. The adopted simulation conditions involve a freestream Mach number $M_\infty = 0.9$ with angles of attack $\alpha=1$ and $4^o$. Similar to the previous example, reference solution errors are computed using higher-order estimates of the problem aerodynamic forces. The observed order of accuracy of the scheme is also computed and values of $p = 1.69$ and $p = 1.41$ are obtained for $\alpha = 1^o$ and $4^o$ respectively.

### 6.3.3.2.1 Convergence assessment

The convergence rate of the global error estimate (6.9) is examined next (Figure 73). It can be observed that the components of the truncation error-based estimate converge at a rate near the order of accuracy of the scheme found by Richardson's extrapolation. Like in the previous test case, a fast convergence of the truncation error-based estimate is also obtained in this example (but results are not reported here).

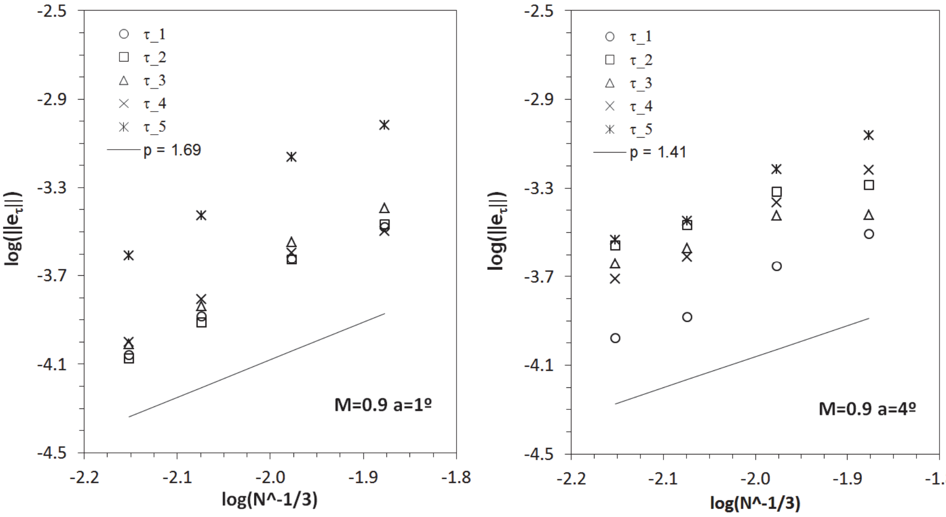

Figure 73. Convergence of the truncation-error based estimate for the NACA wing-body test case.

### 6.3.3.2.2 Automatic grid adaptation

The error estimate (6.8) is employed to drive *h*-adaptation and the results are compared with those corresponding to the conventional feature-based indicator (6.1). The flow freestream Mach number is $M_\infty = 0.9$ and the body angle of incidence is $\alpha=4^o$. The initial domain discretization consists of an unstructured distribution of 606756 points and 174812 boundary triangles. Four refinement levels are performed during the adaptive simulation and the settings of the adaptive procedure are similar to those used in the previous example. The level of refinement achieved with both error indicators results very similar. The truncation error-driven procedure achieves 1001624 points and 198822 boundary triangles, while 1041624 points and 205320 triangles are obtained with the curvature-based indicator. Next, the evolution of both, normal force error (computed with respect to the extrapolated value $C_{Ne}$) and the global error estimate (6.9) during the adaptive computation is shown in Figure 74. As in the previous test case, the truncation error-based indicator leads to a lower aerodynamic force error. The convergence rates and the reduction of the surrogate error estimate (6.9) achieved are quite similar for both indicators.

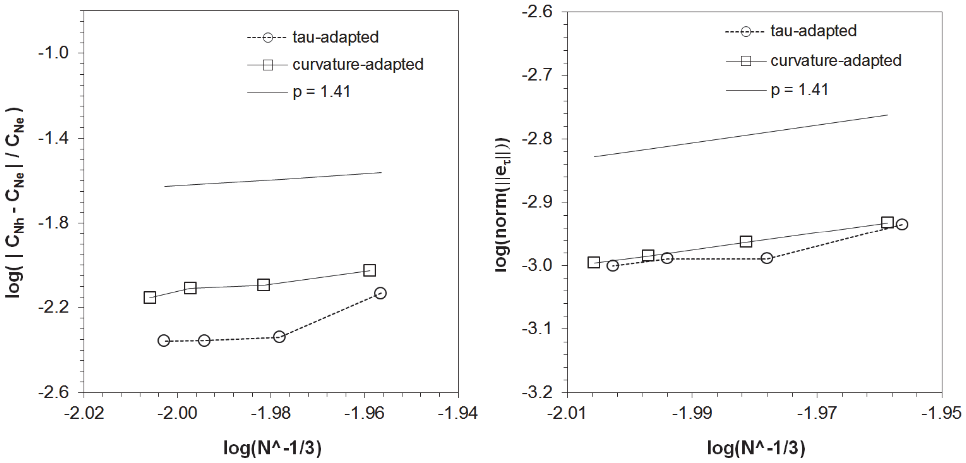

Figure 74. Behavior of the normal force error (left) and the numerical error estimate (right) with the grid refinement; norm($||e_\tau||$) denotes the norm of the global error vector (6.9). NACA wing-body, M=0.9 and $\alpha$=4º.

Pressure distributions computed along two wing spanwise stations 2y/b = 0.40 and 0.8 are presented in Figure 75 along with experimental and reference numerical results obtained on the finest grid model. It can be seen that both error indicators lead to similar results around the main flow features. A view of the final adapted discretizations is shown in Figure 76. As observed in the previous test case, feature-based adaptivity shows more effective around weaker flow discontinuities (see for example the shock on the fuselage) but truncation error adaptivity also refines in areas where the flow is relatively smoother (trailing edge and wing tip), probably contributing to a more effective reduction of the solution errors (see Figure 74).

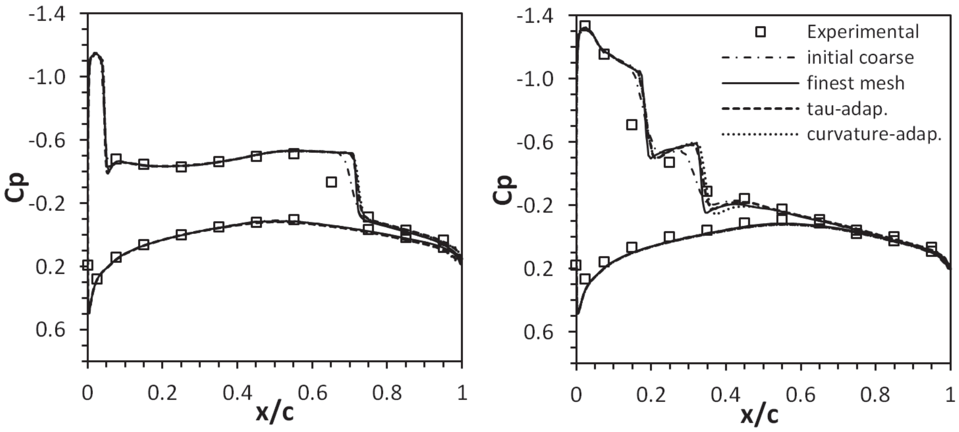

Figure 75. Comparison of pressure distributions at two spanwise stations 2y/b = 0.4 (left) and 0.8 (right). NACA wing-body, M=0.9 and α=4º.

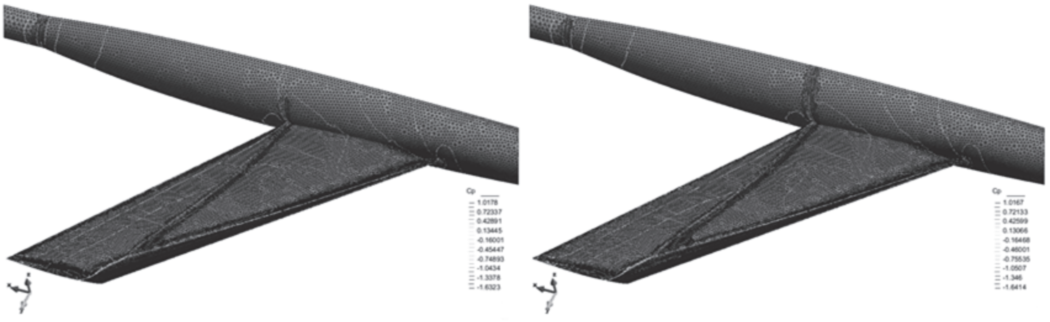

Figure 76. Final adapted boundary discretizations obtained with the truncation error-based indicator (left) and the curvature-based indicator (right). NACA wing-body, M=0.9 and α=4º.

## 6.4 Concluding remarks

Automatic grid adaptation is one of the areas in which meshless approaches offer attractive alternatives to conventional mesh-based techniques. With the purpose to exploit these possibilities, an *h*-adaptive meshless procedure has been developed in this chapter. The proposed algorithm uses a simple solution feature-based indicator and the refinement/coarsening scheme works entirely at the local cloud level. Several test cases involving stationary and non-stationary flows have been presented with the purpose of illustrating the performance of the meshless adaptive technique. The procedure has shown a satisfactory accuracy and robustness in most of the test situations, although the resolving capability, particularly in unsteady flows, could be further improved. To address such problems more efficiently, an improved solution approach will be proposed in the next chapter.

With the aim of achieving a more effective reduction of the numerical error in adaptive procedures, a more suitable error estimate based on the truncation terms of the problem equations has been also developed. Following standard approaches in error analysis, this error estimate is obtained from the residual of the differential governing equations. The latter are calculated in a nodal manner by reconstructing an improved solution field on a modified stencil of points, but employing the same underlying approximation used to compute the actual problem solution. The meshless reconstruction technique adopted in this work, which is based on ENO schemes, has demonstrated satisfactory accuracy and robustness levels as well as a low computational cost.

The numerical applications have shown that the truncation-error based estimate has a good correlation with the problem discretization error, with convergence rates closer to the observed accuracy of the scheme. A relatively faster convergence of the global error estimate was also obtained and this may suggest a higher sensitivity of the indicator to discretization related aspects. This can be advantageous because fast surrogate error measures can be obtained in this way, but a reduced sensitivity to the flow solution can make it difficult to obtain reliable correlations with the error in quantities of practical interest. Regarding the local performance, the truncation error-based estimate has shown to be an effective indicator to drive grid adaptivity. This was reflected not only in reduced aerodynamic force errors but also in the convergence of the surrogate error estimate (proportional to the discretization error). Moreover, the adapted point distributions seem to be more uniform if obtained with the truncation error-based indicator (rather than the feature-based indicator).

Overall, the performance of the adaptive procedure and the proposed local error indicator were found to be satisfactory. The correlation between the truncation error-based indicator and the problem discretization error enables further enhancements and extensions of the methodology, for example in the line of the methods that use residual imbalances to obtain surrogate measures of the error in quantities of practical interest.

# 7 MOVING BOUNDARY PROBLEMS

Problems involving moving or deforming boundaries play a key role in many fields of engineering. Since the beginning of the last century, the study of such problems has been extensively addressed through experimental and simplified numerical approaches, and numerous techniques involving different levels of complexity have been proposed. Nowadays, the design requirements frequently go beyond the scope of classical methodologies and more complex fluid-structure coupled approaches based Euler or Navier-Stokes equations must be considered[62]. This kind of simulations often involves difficulties regarding mesh motion and deformation; and adaptative techniques can be also needed to achieve enough mesh resolution to capture local behavior and/or keep acceptable mesh topology in highly deformed zones of the domain. These facts make the problem solution more complex and increase the computational cost, making sometimes the numerical approach unable to meet the practical needs. In this context, meshless approaches can offer some advantages over conventional techniques.

Meshless procedures for dealing with unsteady problems accounting for body motion have been successfully applied in the literature; see for instance some approaches using small perturbation boundary conditions and domain deformation techniques in (Anandhanarayanan, 2010; Kirshman & Liu, 2004; Wang, Chen & Periaux, 2009). In this chapter, a general solution approach is proposed with basis on the *h*-adaptive methodology previously presented in Chapter 6. The procedure draws on *h*-adaptivity to properly resolve evolving discontinuities and regenerate highly distorted local zones in the analysis domain, thus increasing the robustness and quality of the numerical solution.

This chapter is organized as follows. Section 7.1 and 7.2 present the governing equations and the modifications introduced in the basic flow solver presented in Chapter 4 for its solution. The domain deformation technique adopted, based on the spring network analogy, is outlined in Section 7.3. Aimed at assessing the performance of the meshless approach, several numerical applications are presented in Section 7.4. The computational cost

[62] An interesting and updated discussion on the challenges and trends in the field of airplane aeroelasticity can be found in (Livne, 2003).

and multi-core performance of the method is also discussed through the examples provided.

## 7.1 ALE-form of the fluid equations

In order to facilitate the treatment of moving/deforming bodies in the computational domain, the Euler equations (4.9) are expressed in a more suitable form by adopting an arbitrary Lagrangian-Eulerian (ALE) frame of reference. There, the equations can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}^k}{\partial x_k} + \mathbf{S} = \mathbf{0} \tag{7.1}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_i \\ \rho e_t \end{bmatrix} \quad , \quad \mathbf{F}^k = \begin{bmatrix} \rho(u_k - w_k) \\ \rho u_i (u_k - w_k) + \delta_{ik} p \\ \rho e_t (u_k - w_k) + u_k p \end{bmatrix} \quad , \quad \mathbf{S} = \mathbf{U} \frac{\partial w_k}{\partial x_k} \tag{7.2}$$

being $\rho$, $p$ and $e_t$ the fluid density, pressure and total energy, respectively. The Cartesian components of the fluid velocity are $u_i$, and $w_i$ denotes the components of the discrete points (kinematic) velocity. Note that velocities relative to the moving boundaries are now considered in the flux vectors and an additional source term accounting for grid deformation (**S**) is included. Like in the Eulerian case, the state relations for a perfect gas are adopted. It should be noted that all the variables in this chapter are considered dimensionless (see Eqs. (4.8)).

## 7.2 Flow solution approach

The solution approach adopted for solving the Eqs. (7.1) follows the general lines presented in Chapter 4, albeit with some modifications, particularly with regard to time discretization, which are introduced to solve the unsteady problem more efficiently. The basic aspects of the solution scheme are described in the succeeding text.

### 7.2.1 Equations discretization

Following the developments of Section 4.2, the semi-discrete form of the Eqs. (7.1) is written as

$$\frac{\partial \hat{\mathbf{U}}_i}{\partial t} = -2\sum_{j\neq i} b_{ij}^k \left[ \mathbf{F}_{ij}^k - \mathbf{F}_i^k \right] - \hat{\mathbf{U}}_i \sum_j b_{ij}^k w_j^k \qquad \forall j \in \Omega_i \tag{7.3}$$

where all the terms are computed similarly, but accounting for the relative velocities in the convective flux vectors. The discretization adopted for the source term follows from Eq. (3.12), and $\boldsymbol{w}_i$ denotes the kinematic point velocity vector.

### 7.2.2 Time integration

Explicit time integration approaches as presented in Section 4.3 are highly penalized in problems where the characteristic times involved are far larger than that permitted for the stability of the numerical scheme. This is a typical situation when dealing with moving/deformable unsteady problems. Therefore, it is customary to adopt more efficient solution algorithms; implicit schemes are typically preferred in these cases.

A solution approach which takes both, the advantages of implicit schemes regarding allowable time steps and the simplicity of the explicit solution, is the Jameson's dual-time steeping method (Jameson, 1991). This procedure allows solving implicitly each physical time increment by means of inner explicit iterations in a fictitious time. To this end, a second-order backward difference operator is applied to the time derivative in Eqs. (7.3) leading to

$$\frac{3\mathbf{U}_i^{n+1} - 4\mathbf{U}_i^n + \mathbf{U}_i^{n-1}}{2\Delta \mathrm{t}} = -\mathbf{R}(\mathbf{U}_i^{n+1}) \tag{7.4}$$

where all the right-hand side discrete terms are collected in vector $\mathbf{R}(\cdot)$. Next, for a specified increment in physical time Δt, a modified (unsteady) residual is defined from Eq. (7.4) as

$$\mathbf{R}_i^*(\mathbf{U}_i^*, \mathbf{U}_i^n, \mathbf{U}_i^{n-1}, \Delta \mathrm{t}) = \frac{1}{2\Delta \mathrm{t}}\left(3\mathbf{U}_i^* - 4\mathbf{U}_i^n + \mathbf{U}_i^{n-1}\right) + \mathbf{R}_i(\mathbf{U}_i^*) = \mathbf{0} \tag{7.5}$$

verifying that $\mathbf{U}_\mathrm{i}^*$ approaches $\mathbf{U}_\mathrm{i}^{\mathrm{n}+1}$ as $\mathbf{R}_\mathrm{i}^* \to \mathbf{0}$. In order to solve Eq. (7.4) in an explicit manner, a temporal derivative with respect to a fictitious time $\mathrm{t}^*$ is introduced. This leads to

$$\frac{\partial \mathbf{U}_i^*}{\partial \mathrm{t}^*} + \mathbf{R}_i^*(\mathbf{U}_i^*, \mathbf{U}_i^n, \mathbf{U}_i^{n-1}, \Delta \mathrm{t}) = \mathbf{0} \tag{7.6}$$

and this system is driven to the steady state by means of the multi-stage scheme defined in Section 4.3. Accordingly, the intermediate solution $\mathbf{U}_\mathrm{i}^*$ is advanced from a fictitious time level *m* to a level *m+1* by

$$\hat{\mathbf{U}}_i^{*(m+j)} = \hat{\mathbf{U}}_i^{*m} - \alpha_j \Delta \mathrm{t}_\mathrm{i}^* \mathbf{R}^* \left( \mathbf{U}_i^{*(m+j-1)}, \mathbf{U}_i^n, \mathbf{U}_i^{n\text{-}1}, \Delta \mathrm{t} \right) \tag{7.7}$$

being $\alpha_\mathrm{j}$ integration coefficients depending on the number of stages employed and $\Delta \mathrm{t}_\mathrm{i}^*$ a fictitious local time step (subject to stability requirements). In addition to local time steeping, implicit residual smoothing is applied to accelerate the convergence of the system (7.7) in fictitious time. A system of linear equations must be also solved for recovering the internal nodal values from the approximate solution at each integration stage (see Eq. (4.60)). Different tests performed with arbitrary grid motions (in the range of the problems here addressed) have shown no problems related with geometric conservation[63] and, therefore, no particular treatment is applied in this regard.

It should be notice that the explicit treatment of the term ($3\mathbf{U}_\mathrm{i}^*/2\Delta\mathrm{t}$) could lead to numerical instabilities if the physical time step is small (Melson, Sanetrik & Atkins, 1993). Fortunately, this problem can be easily overcome by treating implicitly that term in Eq. (7.7), see (Venkatakrishnan & Mavriplis, 1995). This is the procedure adopted in this work.

### 7.2.3 Boundary conditions

The boundary conditions applied in ALE problems follows the same lines described in Section 4.5. However, relative velocities should be accounted for when the kinematic point velocity is different than zero. In this way, the slip-wall condition on solid moving boundaries results

$$\left( \boldsymbol{u} - \boldsymbol{w} \right) \cdot \hat{\boldsymbol{n}} = 0 \quad \forall \boldsymbol{x}_i \in \Gamma_w \tag{7.8}$$

where $\hat{\boldsymbol{n}}$ is the unit boundary normal vector at point $\boldsymbol{x}_i$. Typically, the spatial position of the discrete points, their kinematic velocities and the boundary normal vectors are obtained from prescribed or computed body movements.

## 7.3 Points movement strategy

Problems involving deforming or moving bodies require the domain discretization to conform continuously to the instantaneous body shape. An overview of typical solution strategies to deal with domain deformation can

[63] Geometric Conservation (Thomas & Lombard, 1979) can be seen as a consistency condition which requires the algorithm to reproduce a constant solution on a moving grid, independently of the velocity and the level of distortion of the mesh. In spite of the fact that this condition can be derived from the governing equations and enforced explicitly in the numerical scheme (or satisfied by construction, e.g. (Mavriplis & Yang, 2006)), these procedures are often omitted in practice. The effects on the numerical solution are usually negligible for moderate time steps and grid velocities.

be found in (Löhner R., 2002). Moreover, more recent promising techniques based on Delaunay mapping are presented in (Liu, 2006) (see a 2D meshless application in (Wang, Chen & Periaux, 2009)). With the purpose of maintaining the meshless character of the solution methodology, a classical spring network approach is adopted in this work; see (Batina, 1989; Blom, 2000). Therefore, the displacement of any interior point $\boldsymbol{x}_i$ in response to instantaneous displacements of body points are obtained by enforcing the static equilibrium of the forces exerted by all the points $\boldsymbol{x}_j$ connected through the $\boldsymbol{x}_i$'s layer of nearest neighbors in the local cloud (outer boundary points are considered to be fixed). This leads to a system of equations in terms of displacements which is solved by Jacobi iterations as follows

$$\boldsymbol{\delta}_i^{m+1} = \frac{\sum_j k_{ij}\, \boldsymbol{\delta}_j^m}{\sum_j k_{ij}} \tag{7.9}$$

being $m$ the iteration counter, $\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$ displacement vectors of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ respectively and $k_{ij} = \| \boldsymbol{x}_j - \boldsymbol{x}_i \|^{-1}$ a link stiffness which prevents the clouds near the moving boundary from excessive distortion. After some iterations (10-50 iterations showed enough to propagate satisfactorily the body displacements) the position of the discrete points is updated according to $\boldsymbol{x}_i^{new} = \boldsymbol{x}_i^{old} + \boldsymbol{\delta}_i$ and their velocities are simply estimated by $\boldsymbol{w}_i = \boldsymbol{\delta}_i / \Delta\mathrm{t}$ being $\Delta\mathrm{t}$ the physical time increment employed in the simulation.

## 7.4 Application examples

Four numerical examples are presented in order to assess the performance of the methodology. The first three examples involve typical validation and verification test cases for unsteady transonic flows. The adaptive solution approach is employed in these computations and an analysis is performed from the point of view of the accuracy and computational cost. The last example, which is a coupled fluid-structure interaction problem involving static aeroelasticity, is intended to give an idea of the potential of the meshless technique to deal with practical problems.

### 7.4.1 NACA wing subject to pitching oscillations

This example is a typical AGARD benchmark for unsteady flow which involves a NACA 0012 airfoil subject to prescribed pitching oscillations (Landon, 1982). The problem is solved in 3D using a wing with unit chord, and a span which is half of the chord. The analysis domain comprises two symmetry lateral planes at the wing tips to force two-dimensional flow, and the outer boundary, where Riemann freestream boundary conditions are prescribed, is located 10 chords away from the wing. The discrete model consists of an unstructured distribution of 33409 points. The freestream Mach number is set to $M_\infty = 0.755$ and the instantaneous angle of attack of the wing (in degrees) is varied about the c/4 chord point according to

$$\alpha(t) = 0.016 + 2.51\sin(\omega_\alpha t) \tag{7.10}$$

with a reduced frequency $\kappa = \omega_\alpha c / 2U_\infty = 0.0814$. The converged steady solution of the problem is employed to initialize the unsteady simulations. At each time increment, the unsteady residual is reduced at least 4 orders of magnitude (500 iterations were enough in our tests).

Simulation runs with 4, 8, 12, 16 and 32 physical time steps per oscillation period (T) are performed first to assess the accuracy of the time integration scheme. The time evolution of the lift presented in Figure 77 shows the convergence of the computed solutions as the physical time step is reduced. The accuracy is evaluated by defining a temporal error as the magnitude of the difference between the lift coefficient computed at time $t = 115$ for a given solution, and that obtained at the same instant time with the smaller time step computation ($\Delta t$=T/32). A convergence rate close to the design order of accuracy of the scheme can be observed.

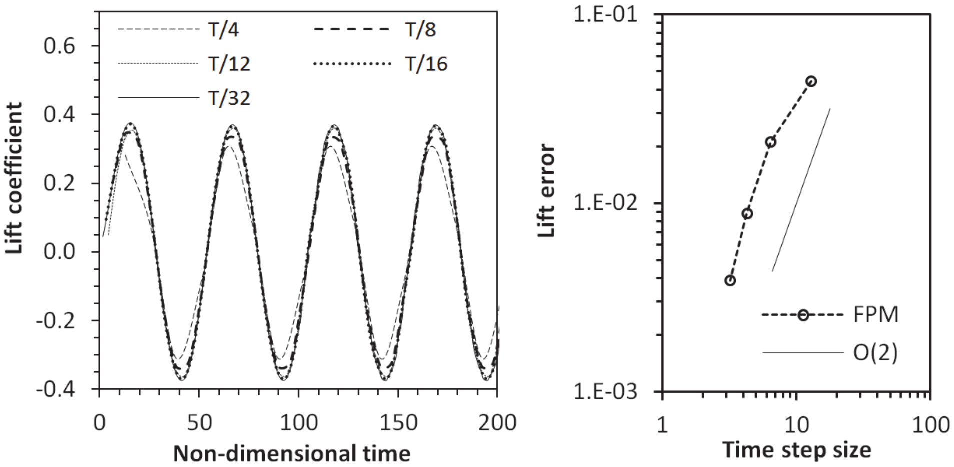

Figure 77. Time evolution of the lift coefficient for the NACA 0012 wing computed with different time step sizes (left) and convergence of the temporal error (right).

The run having 16 physical time steps per oscillation cycle is repeated, but adding 2 *h*-adaptivity passes per physical time step. On average, no more than 7000 points are added during the adaptive steps while the number of removed points does not exceed 3000, achieving this maximum when the shock wave moves from the upper to the lower side of the wing and vice versa. The coefficient of pressure calculated along the mid-span of the wing at two instant times (phase angles φ = 67.8 and 253.8 degrees) are compared in Figure 78 with those obtained for the original coarse discretization and with experimental results (Landon, 1982). The variation of lift and pitching moment with the instantaneous angle of attack for the adapted solution is also compared with experimental results in Figure 79. As the difference found between adapted and non-adapted solutions was very small, the latter is not included in this figure (possibly the original coarse discretization was fine enough to be in the range of converged grid solutions). A good agreement between numerical and experimental results can be observed. Figure 80 shows refined points distributions in a cut plane along the mid-span of the wing at two different instant positions.

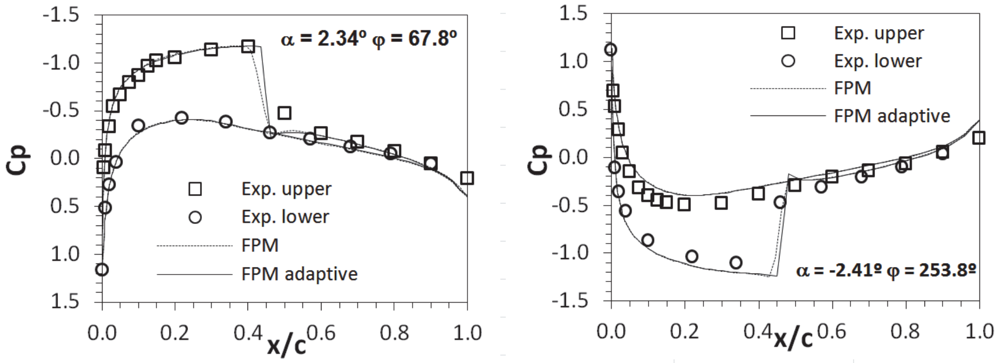

Figure 78. Surface pressure distributions along the NACA 0012 wing for two different time instants during the oscillation cycle.

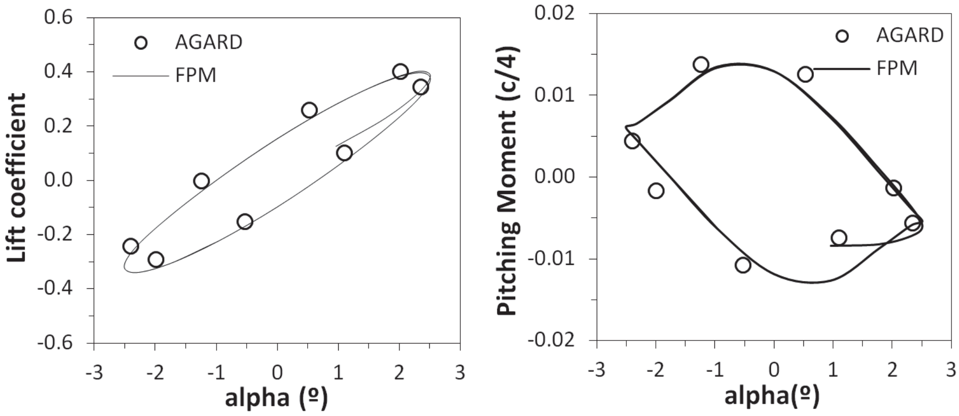

Figure 79. Comparison of computed and experimental force and moment variation with the instant angle of attack for the NACA 0012 wing (adapted solution, Δt=T/16).

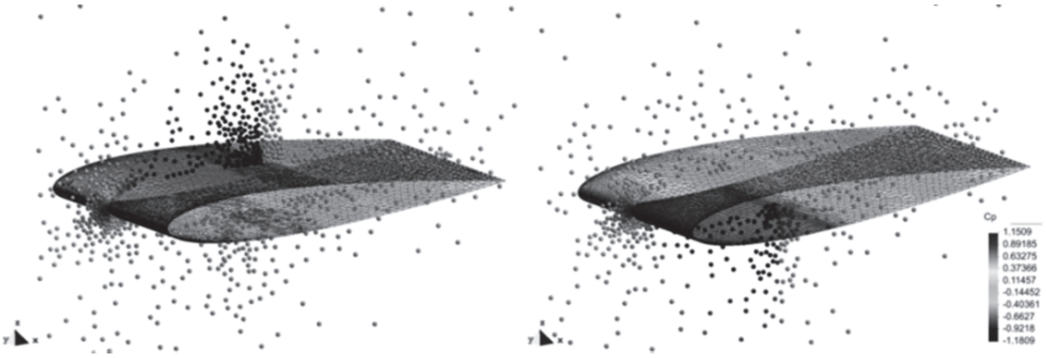

Figure 80. Refined point distributions along a plane passing through the wing mid-span. Left α=2.34° and φ=67.8; Right: α=-2.41° and φ = 253.8°.

### 7.4.2 Wing with oscillating flap

An adaptive calculation of a wing having an oscillating flap is solved in this second example to assess the capability of the meshless method to deal with problems involving relative body motion. The section of the wing corresponds to the William airfoil (configuration B) (Williams, 1971), and the flow conditions are defined as in (Dubuc, Cantariti, Woodgate, Gribben, Badcock & Richards, 2000): the freestream Mach number is set to $M_\infty = 0.58$ and the instantaneous flap angle (in degrees) is

$$\delta_f(t) = 7.0 + 7.0\sin(\omega_\delta t) \tag{7.11}$$

with reduced frequency $\kappa = 0.0814$. The flap hinge position is located at coordinates (x,z) = (0.98, -0.07), which are non-dimensional with the chord of the main wing section. The rotation of the flap is measured with respect to its original position in the Williams airfoil.

The analysis domain is similar to the previous example and discretized by an unstructured distribution of 20713 points. The numerical simulation employs 12 physical time steps per flap oscillation cycle and 2 *h*-adaptive stages are performed in each time step. The evolution of the normal force and pitching moment with the instantaneous flap angle is compared in Figure 81 with the numerical results presented in (Dubuc, Cantariti, Woodgate, Gribben, Badcock & Richards, 2000). Finally, Figure 82 shows two plane cuts of the adapted discretization taken at different instant times during the simulation. It is possible to observe the displacement of the shock wake along the main wing and the appearance of a shock on the flap for its maximum incidence angle.

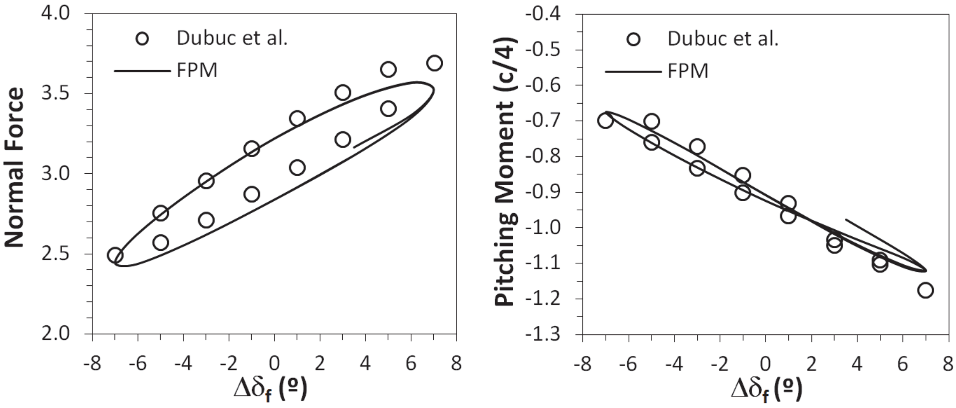

Figure 81. Normal force and pitching moment coefficient (wing+flap) of the Williams wing as a function of the flap instant rotation (Δt=T/12).

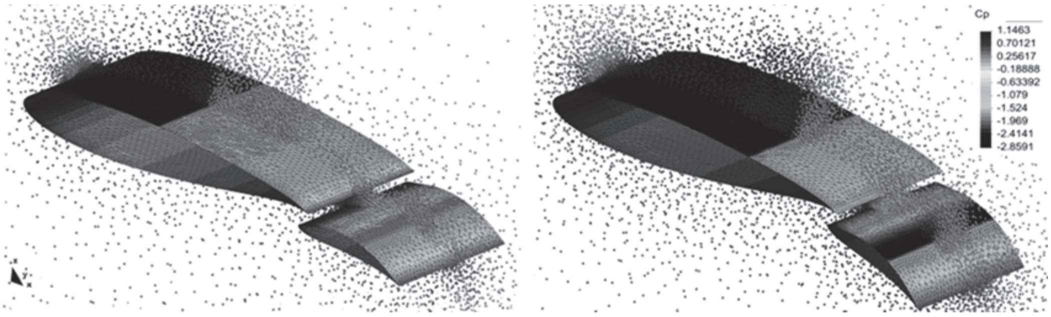

Figure 82. Refined point distributions along a plane passing through the mid-span of the Williams wing. Left: $\delta_f$=0º and φ=270º; Right: $\delta_f$ =14.0º and φ = 90º.

### 7.4.3 Twist deformation of an ONERA M6 wing

The third example concerns the calculation of an ONERA M6 wing subject to twisting deformation about a line passing through the tip quarter chord point, see (Yang & Mavriplis, 2005). The freestream Mach number is $M_\infty = 0.84$ and the wing incidence is α = 3.06º. The twist deformation is achieved by forcing the tip section to pitch with

$$\alpha_{tip}(t) = 2.51\sin(\omega_\alpha t) \quad \text{(in degrees)} \qquad (7.12)$$

while keeping the root section fixed. The instant angle of attack of the intermediate sections is varied linearly along the wing semi-span and the reduced frequency is $\kappa$ = 0.1628. The analysis domain includes a symmetry plane and a hemispherical outer freestream boundary.

In order to investigate the impact of *h*-adaptivity from the point of view of accuracy and computational cost, two different unstructured domain discretizations are constructed: a fine discretization having 258919 points and a coarse one with 107847 points (uniform scaling of the grid spacing is adopted). In addition, an adapted discretization is obtained from the coarse model by performing a single *h*-adaptive pass per physical time step. A view of these discretizations is presented in Figure 83.

Several twist cycles are simulated with the discretizations defined above running 8 physical time steps per cycle. The number of fictitious time iterations is fixed to 500 and the resultant unsteady residual is reduced approximately 4 orders of magnitude. Figure 84 displays the time evolution of the normal force coefficient. Some instantaneous Cp distributions along the wing are compared in Figure 85. Even though the number of new points added during the *h*-adaptivity stages represents as much as the 12% of the number of original points, it can be seen that the adapted solution improves noticeably over the non-adapted coarse simulation.

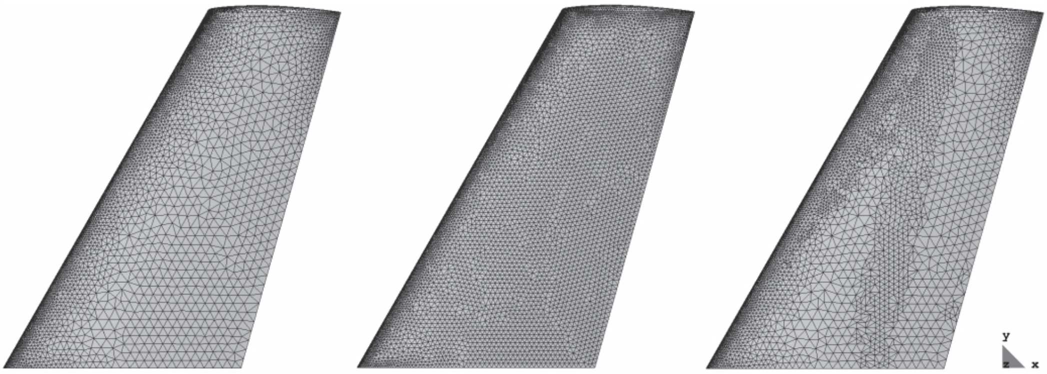

Figure 83. Top view of the ONERA M6 wing surface grids. Left: coarse; Center: fine and Right: adapted-coarse at φ=0º (the adapted discretization does not change substantially during the twist cycle).

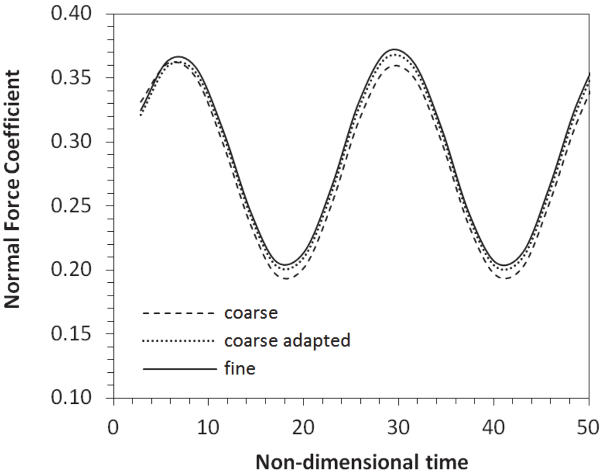

Figure 84. Time evolution of the normal force computed for the twisting ONERA M6 wing with the coarse, fine and adapted-coarse discretizations (Δt=T/8).

The CPU-time required to advance the simulation one physical time step (Δt=T/8 and 500 fictitious time iterations) is compared next for the coarse, fine and coarse-adapted discretizations. The simulations are run in a desktop computer with Intel Core2 Quad Processor Q9550 @ 2.83 GHz. According to Table 7, the cost in the adapted simulation is slightly higher than in the coarse discretization, but low if compared with the fine discretization run. At the same time, the adapted solution improves considerably over the solution obtained with the coarse discretization and matches closely that obtained with the finest model. Based on these results, meshless adaptivity shows effective.

The parallel speed-up observed for the adaptive simulation is depicted in Figure 86. The results obtained are satisfactory and in line with those previously presented in Section 5.

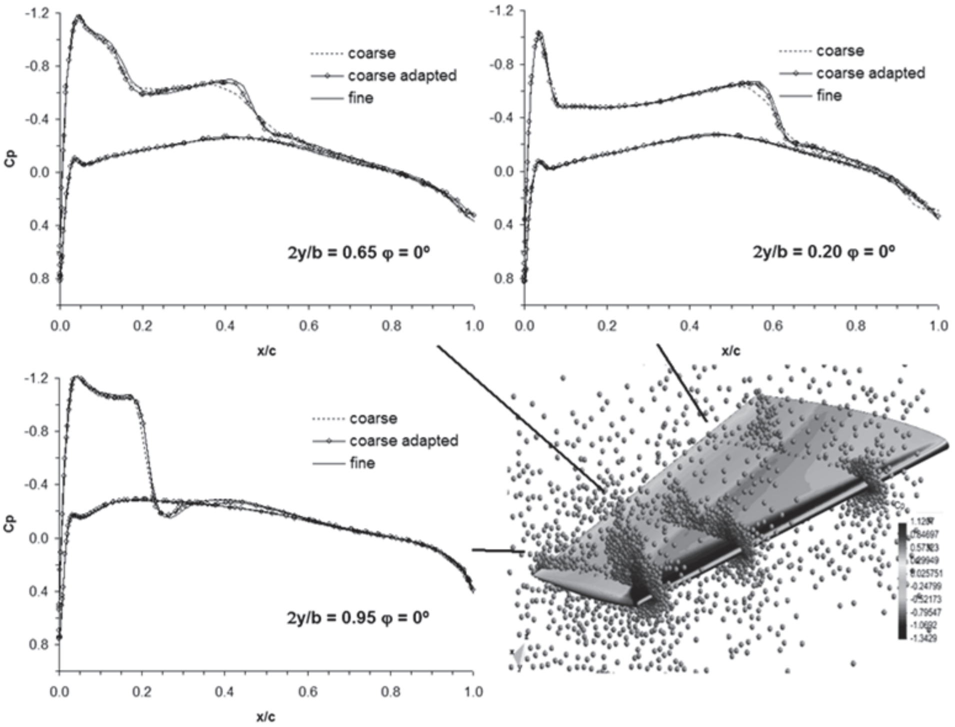

Figure 85. Comparison of instantaneous Cp distributions along 3 sections of the ONERA M6 wing computed for the coarse, fine and adapted discretizations (a view of the latter is displayed on the right bottom).

| grid # | CPU-time (secs.) | Relative Cost |
|---|---|---|
| **fine (reference)** | 318.6 | 1.00 |
| **coarse-adapted** | 192.6 | 0.60 |
| **coarse** | 170.8 | 0.54 |

Table 7. Comparison of CPU-time needed to advance the solution of the twisting ONERA M6 wing a single physical time step (Δt=T/8 and 500 fictitious time iterations) using 4 running cores.

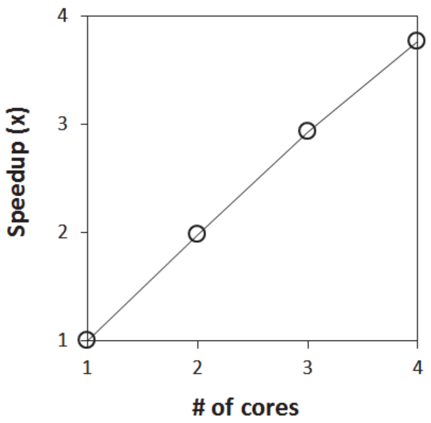

Figure 86. Observed speed-up in the ONERA M6 wing adapted simulation (Δt=T/8 and 500 iterations in fictitious time loop).

### 7.4.4 Coupled analysis of the HiReTT wind tunnel model

Static aeroelastic deformation of a wing-body configuration is solved in this last example to assess the meshless technique in a realistic application scenario. The geometry of analysis corresponds to one of the cryogenic wind tunnel models employed in the European research project HiReTT (High Reynolds Number Tools and Techniques for Civil Aircraft Design) to investigate scale effects on aircraft performance; see an overview in (Rolston, 2001). A half-span symmetric version of the N44 test model (Wright, 2000) with a clean F7-1 wing configuration (b/2=790mm) is employed.

#### 7.4.4.1 Computational models

In order to accomplish the coupled aeroelastic solution of the analysis configuration, aerodynamic and structural models are constructed. The analysis domain for the aerodynamic problem (CFD-model) includes a symmetry vertical plane along the fuselage centerline and a hemispherical outer freestream boundary located 15 half-spans away from the body. The resultant discrete model has 338886 boundary triangles and a total of 3100159 points generated with the technique described in Section 3.6.1. A view of the model boundary discretization is shown in Figure 87. As large flow separation areas are not expected in the range of simulation conditions to be adopted here, the flow is considered to be inviscid and the boundary conditions applied are similar to those employed in the previous examples.

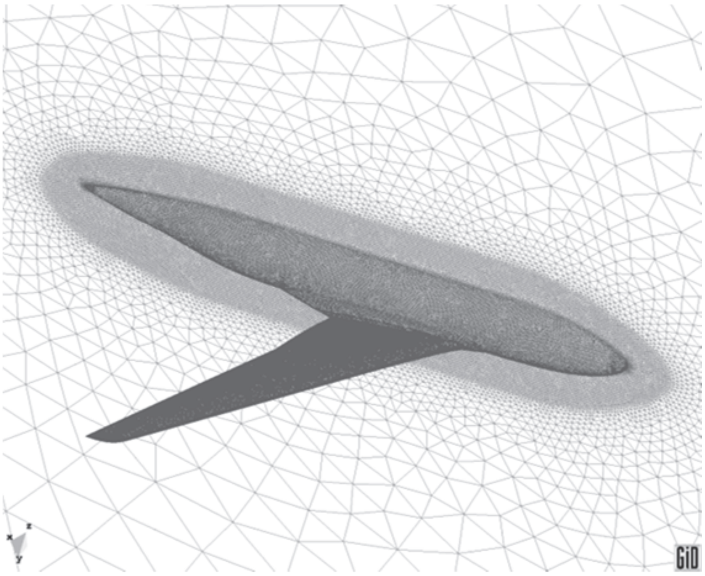

Figure 87. Boundary grid of the half-span model employed for flow computations (approx. 340K triangles).

The deformation of the model under aerodynamic loading is studied by means of a finite-element structural model (FE-model) which has been developed assuming that the fuselage and the support system (single rear-sting) flexibility can be neglected. This assumption is intended to reduce the model complexity and can be explained by the fact that the expected body deformations are negligible if compared to those of the wing. Furthermore, the experimental results in (Wright, 2000) already account for the deflection of the supporting mechanism. As a further simplification, the internal wing cavities needed to install and route measurement instrumentation are also neglected in the FE-model and the wing is considered to be solid (the internal cavities are designed to have a minimum influence on the model stiffness).

Regarding the modeling of the wing-fuselage junction, the wing is considered to be clamped at the interface. A detailed representation of this junction is considered unnecessary in this study because wing deformation is only relevant for outboard wing stations. The structural model of the wing developed is discretized by means of 19456 20-node hexahedral elements and 94749 nodes. The material adopted is a high-strength maraging steel with Young's modulus E = 196 GPa and Poisson's ratio $\upsilon = 0.30$. A view of the FE-model is presented in Figure 88. Note that in both, CFD and FE models, the wing geometry corresponds to the manufacturing or jig-shape experimental test model.

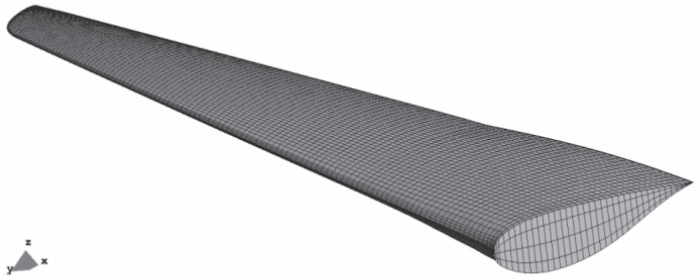

Figure 88. Structural model of the HiReTT N44 F7-1 wing (approx. 19.5K 20-node hexahedra).

### 7.4.4.2 Coupled solution strategy

The coupled solution methodology is based on the time-marching procedure presented in Section 7.2.2. At each physical time step of the CFD-model, aerodynamic pressure loads are transferred to the FE-model, which is solved in order to obtain the displacements (the computations are performed using an in-house linear elastic solid solver). Next, the computed deformation is mapped back onto the CFD-model boundaries, the volume discretization is deformed according to the procedure described in Section 7.3 and the flow solution is converged in fictitious time. Then, a new interaction cycle is started by transferring the resultant aerodynamic pressures to the FE-model. The iterations continue until a given displacement tolerance is achieved. Due to the fact that only the steady solution is of interest in this example, the problem is solved using large physical time steps, and load under-relaxation is also applied to accelerate the convergence of the coupled solution.

The mapping of variables between the CFD and FE grids is performed as follows. For each FE node on the wing surface, the CFD boundary triangle where it lays on is identified and the aerodynamic pressure acting on that node is interpolated by using the shape functions of the underlying CFD triangle. Then, averaged pressures at the FE facets are computed and converted into surface tractions which are passed to the structural solver. Once the structure is solved, averaged (barycentre) displacements are obtained at each CFD triangle by using the displacements of the structural nodes falling on it, and a smoothing procedure is accomplished to obtain the nodal values. This procedure guarantees a smooth deformation of the CFD boundary grid which matches very closely that computed in the FE-model.

The correspondence between surface FE nodes and the CFD elements needed to transfer variables between the models is obtained by using a spatial search algorithm based on bins. Given that both, structural and fluid boundary grids deform in a similar way, the spatial search and the construction of the data structure required to transfer the variables are performed only once at the beginning of the computations, and stored to be used later.

#### 7.4.4.3 Numerical results

The aeroelastic computations presented next adopt the freestream flow conditions defined for the wind tunnel run #204 (Wright, 2000). This test involves a freestream Mach number $M_\infty = 0.85$ and a Reynolds number Re = 32.5M. The analysis here is performed for angles of attack ranging from -2.62 to 3.69 degrees, and the fluid is considered to be inviscid. The solution obtained for the rigid model is used to initialize the coupled simulations.

Pressure distributions computed for the rigid and elastic models at three wing sections and angles of attack $\alpha = -2.625^o$, $0.0613^o$ and $2.234^o$ are presented in Figure 89. It is possible to observe that in the case of $\alpha = -2.625^o$ a poorly loaded wing (model lift is near zero) results only in a small model deformation. The results obtained for the rigid and elastic models are very similar for the inboard wing sections, but more marked differences are observed near the wing tip. There, the rotation of the sections (where stiffness is lower) has an evident effect on the shock position, which is well captured by the aeroelastic solution. For the runs at positive angles of attack, the aeroelastic effects become more relevant and the negative twist modifies notoriously the wing suction area and the shock wave position, particularly at outboard spanwise sections (the impact of bending is usually small). As observed in Figure 90, this effect reduces the resultant model lift in the elastic computation. The results presented show that the aeroelastic solution improves considerably over the conventional rigid model analysis. It should be noticed that the inviscid flow assumption adopted can introduce some discrepancies with measured data for runs at higher angles of attack, where flow separation effects become more relevant (the lift curve measured in the wind tunnel test loses its linearity at about $\alpha = 3^{o}$). This effect can be seen in the evolution of the difference between computed and experimental lift coefficients shown in Figure 90.

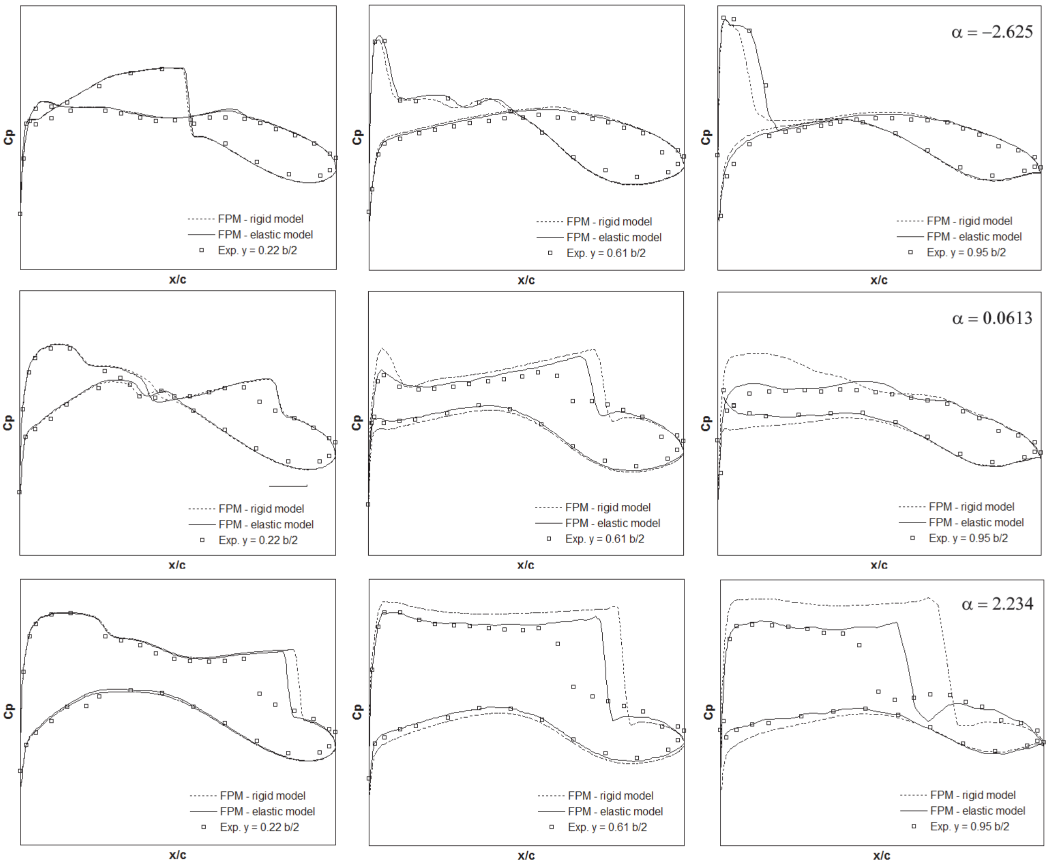

Figure 89. Effects of wing deformation on pressure distributions computed at different spanwise stations. HiReTT N44 model, $M_{\infty} = 0.85$ $\alpha = -2.625^{o}$ (top), $\alpha = 0.0613^{o}$ (middle) and $\alpha = 2.234^{o}$ (bottom).

Twist and bend deformation of a section near the wing tip are depicted in Figure 91 for the complete alpha sweep; the trailing edge deflection ($\Delta z_{TE}$) is measured from the original (wind-off) position and the twist angle is evaluated as $\tan^{-1}[(z_{LE}-z_{TE})/(x_{LE}-x_{TE})]$. Figure 92 shows bending and twisting effects along the wing span for $\alpha=2.234^{o}$. A view of the original and deformed model configuration is displayed in Figure 93.

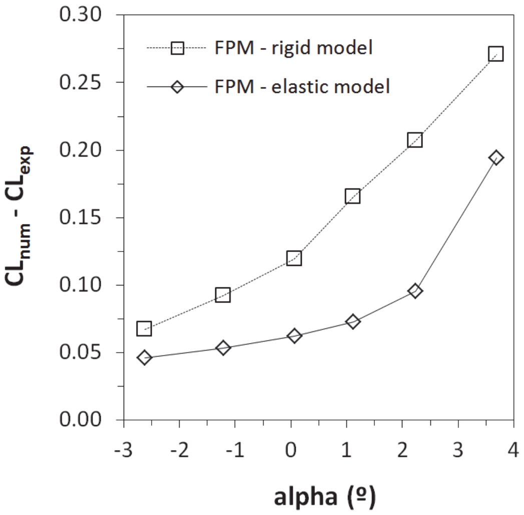

Figure 90. Differences between numerical and experimental lift coefficients computed for the rigid and elastic HiReTT N44 model at $M_\infty = 0.85$.

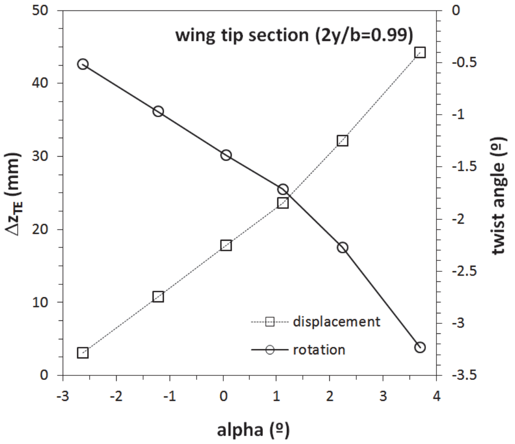

Figure 91. Rotation and vertical displacement of a wing tip section computed for the HiReTT N44 model at different angles of attack at $M_\infty = 0.85$ (b/2=790mm).

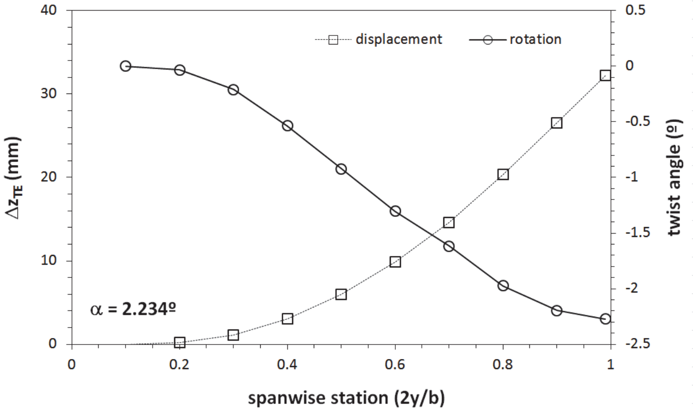

Figure 92. Spanwise wing deformation of the HiReTT N44 model (b/2=790mm), $M_\infty = 0.85$ and $\alpha = 2.234$º.

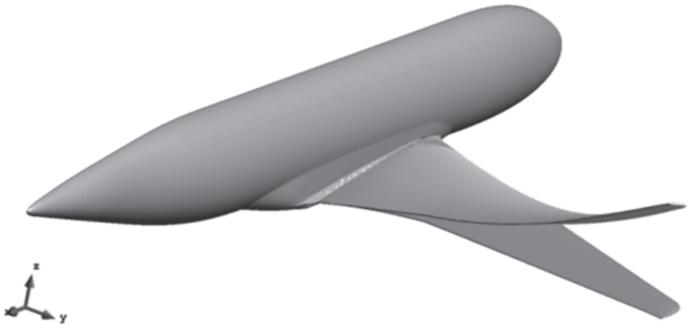

Figure 93. Original (wind-off) and deformed model configurations (5x magnification factor). HiReTT N44, $M_\infty = 0.85$ and $\alpha = 2.234°$.

The convergence history of the coupled simulation for $\alpha = 2.234°$ is presented in Figure 94. At each physical time step a maximum number of 3000 iterations in fictitious time or a density residual < 1.0e-6 is set as exit condition. A load relaxation factor of 0.5 is adopted. Starting from the rigid model solution, a few coupled iterations are required to achieve the static deformed shape of the model.

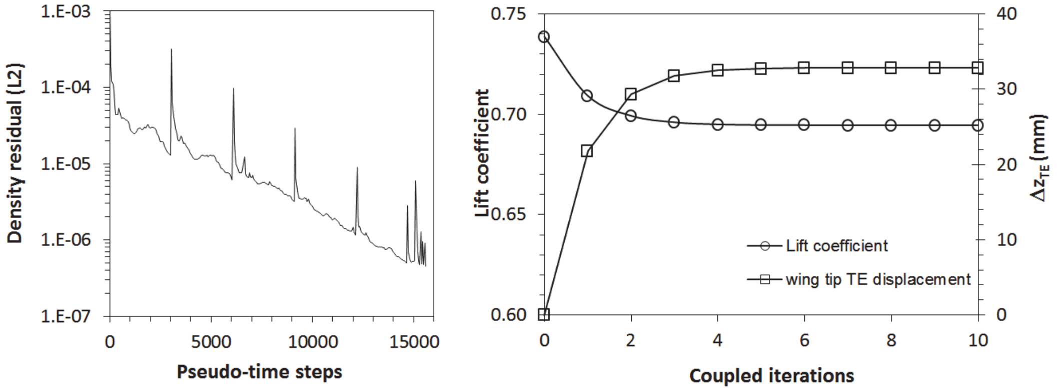

Figure 94. Convergence history. HiReTT N44 model at $M_\infty = 0.85$ and $\alpha = 2.234°$.

In spite of the fact that the convergence speed achieved in the present test case is deemed satisfactory, it is important to say that more efficient convergence acceleration techniques would be required in view of larger application problems. Some interesting implicit and meshless multigrid approaches can be found in (Katz & Jameson, 2009b; Kennet, Timme, Angulo & Badcock, 2012; Singh, Ramesh & Balakrishnan, 2010).

CPU-times and multi-core performance measurements are presented next to give an idea of the computational cost involved in the coupled simulation. Like in the analysis presented in Section 5.3.4, the meshless flow solution is divided into two parts, a pre-process stage which includes the construction of the numerical approximation and its related data structures (clouds of points, shape functions and derivatives), and a flow computation stage which refers to the inner Euler steps in the pseudo-time integration procedure. The structural solution stage is also evaluated. The tasks performed at each structural step involve the transfer of loads and displacements between the fluid and structural models, the solution for the displacements (FE), the deformation of the interior (CFD) volume points according to the computed model position, and the update of the CFD problem data structures, mainly cloud shape functions and derivatives (cloud connectivity is assumed fixed along the simulation). The runs are performed on a cluster node having 2 Intel Xeon E5645 processors @ 2.4 GHz with 12 Mb L2 cache.

CPU-times for the pre-process, flow solution and structural computation stages as well as multi-core performances are displayed in Figure 95. As far as pre-process tasks are concerned, the inherent complexity of the operations involved in the meshless approximation makes it costlier than in conventional mesh-based methods (cf. Section 5.3.4). However, as these tasks are performed only once at the beginning of the computations, the cost does not have a significant impact on the entire simulation time. Moreover, in cases in which the data structures must be updated (for instance due to excessive domain deformation or adaptive refinement/coarsening), this is performed in a local manner with a low computational cost (see results presented in Section 7.4.3). On the other hand, as previously shown in Section 5.3.4, the cost involved in the flow solution (which takes most of the total simulation time) does not differ substantially from what could be expected in a typical mesh-based technique, making the FPM technique quite competitive.

Regarding multi-core performance the results observed are in line with those in Section 5.3.4. It is possible to see in Figure 95 that the speed-up degrades noticeably when more than 4 cores are employed. This effect is more accentuated in the pre-process stage (presumably due to the complexity of the operations involved) than in the flow computation, where a higher data locality and totally independent computations allow for the scaling to continue, albeit at a reduced rate. The loss of performance observed when the number of running cores is augmented has been previously analyzed (Section 5.3.4). In this respect, a better scalability should be expected in higher-performance multi-processors hardware platforms.

Concerning the structural computation stage, the CPU-time presented in Figure 95 corresponds to a serial run because this part of the code is not fully parallelized. It should be noticed that most of the computational efforts at this stage are devoted to the solution of the linear system involved in the structural problem.

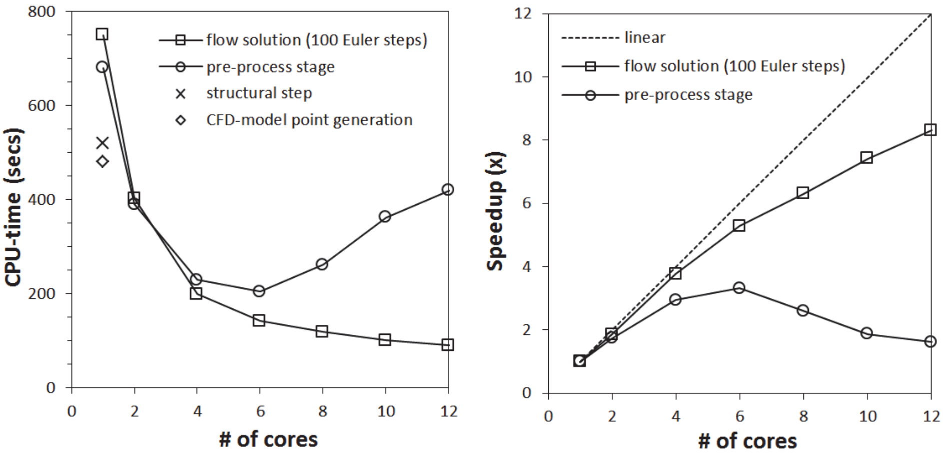

Figure 95. CPU-times and parallel performance of the coupled simulation (3.1M fluid points and 95K structural nodes). HiReTT N44 model at M = 0.85 and α = 2.234º.

Finally, it is interesting to note that the time involved in the discretization of the aerodynamic model (also displayed in Figure 95) shows that the efficiency of the point generator could make it possible to regenerate the complete CFD model discretization with a low cost, if compared with the total simulation time (the CPU-time needed is comparable to that in a single structural step). This could be an attractive solution approach when very large body movement or deformations must be faced. The point discretization in this example was generated using a desktop computer with AMD Opteron 246 processor @ 1.99 GHz running on a Windows 64-bit system.

## 7.5 Concluding remarks

A modification of the basic FPM solution scheme is introduced in this chapter with the objective to solve compressible flow problems involving moving/deforming boundaries. The *h*-adaptive procedure was also employed to facilitate the proper solution of evolving discontinuities, thus increasing the accuracy and robustness of the solution but keeping the computational cost low. The overall methodology was designed following well-established numerical techniques, but aiming at the full exploitation of meshless features.

The results obtained are satisfactory and illustrative of the possibilities of the present FPM technique. In spite of the fact that there is room for further improvements, the methodology shows considerable robustness and accuracy, as well as capabilities to address practical computations. Among other achievements, it is worth of mention the results obtained in the case of the HiRReT aeroelastic model, where it was observed that a complete regeneration of the model discretization could be an affordable solution strategy, even in large simulation problems. This result presents new possibilities for the meshless analysis of moving/deforming domain problems.

# 8 HIGH-REYNOLDS NUMBER VISCOUS FLOWS

The numerical solution of high-Reynolds viscous flows entails a number of difficulties, mainly related to the discretization of the analysis domain. In such problems, the development of very thin boundary layers, free shear layers and wakes, generate rapid localized changes in the flow (normal to the streamwise direction) which need a high spatial grid resolution to be properly resolved. Typical solution approaches use highly stretched grids in the flow direction to obtain the necessary spatial resolution across the viscous layers in a cost effective manner. These anisotropic grids can be generated through structured or hybrid approaches, and diverse techniques are being employed with mesh-based methods; see an overview in (Garimella & Shepard, 2000; Löhner, 2001). While effective, these procedures often need a certain degree of manual input to obtain suitable grids in general problems (e.g. block definition for structured grids or cleanup around complex geometric features, such as intersections of several surfaces, in the case of hybrid grids). Furthermore, as highly-distorted cells can be created during the generation process, some inspection of the grid quality may be still required to avoid degrading the accuracy of the flow solver. In this context, meshless approaches can facilitate both, the generation of the discrete models and the construction of a proper numerical approximation and solution scheme.

The application of meshless methods for solving viscous flows is not new and many successful approaches, mostly focused on low-Reynolds number problems, can be found in the literature, e.g. (Batina, 1992; Katz, 2009; Oñate & Idelsohn, 1998; Oñate, Sacco & Idelsohn, 2000). Regarding high-Reynolds flows, the applications are relatively recent in the literature and the procedures applied do not seem to exploit the full meshless potential yet. For example, the trends followed on domain discretization are in general quite similar to those employed in conventional methods, or the meshless technique is often used only to simplify the treatment of near-body and overset areas in hybrid Cartesian and multi-block structured mesh-based discretizations; see for instance (Kennet, Timme, Angulo & Badcock, 2012) and (Jahangirian & Hashemi, 2012; Munikrishna & Balakrishnan, 2011). This under-exploitation of the meshless potential can be explained to some extent. Despite the fact that there is certain flexibility in the generation of a meshless discretization, additional problems arise in relation to the accuracy and robustness of the approximation (and flow solution schemes) in highly stretched clouds of points. These difficulties impose restrictions on the topological characteristics of the discretization that hamper the development of the meshless potential. Thus, the construction of more robust numerical approaches in distorted distributions of points is essential to exploit meshless capabilities.

The aim of this chapter is to explore high-Reynolds number problems focusing on two main aspects: the easiness and automation of the meshless discretization of viscous layers, and the construction of a robust numerical approximation in the highly stretched clouds of points resulting in such domain areas.

This chapter is organized as follows. The fluid governing equations are presented in Section 8.1 and the basic aspects of the solution approach are described in Section 8.2. A methodology for constructing meshless discretizations (with a focus on boundary layers) is presented in Section 8.3, and a procedure to compute the numerical approximation in stretched cloud of points is proposed in Section 8.4. Finally, several numerical examples involving attached boundary layer flows are presented in Section 8.5 to illustrate the basic performance and suitability of the meshless procedure.

## 8.1 Navier-Stokes equations

The compressible Navier-Stokes equations can be written in non-dimensional conservative form as

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}^k}{\partial x_k}=\frac{\mathrm{M}_\infty}{\mathrm{Re}_\infty}\frac{\partial \mathbf{G}^k}{\partial x_k} \tag{8.1}$$

where $\mathrm{M}_\infty$ and $\mathrm{Re}_\infty$ are freestream Mach and Reynolds numbers, $\mathbf{U}$ is the vector of conserved quantities (mass, momentum and energy) and $\mathbf{F}^k$ is the advective flux vector in the coordinate direction $x_\mathrm{k}$ (see Eqs. (4.10)). The dimensionless viscous fluxes $\mathbf{G}^k$ are given by

$$\mathbf{G}^k=\begin{bmatrix} 0 \\ \tau_{ik} \\ u_j\tau_{jk}-q_k \end{bmatrix} \tag{8.2}$$

where the fluid viscous shear stress (assuming Stoke's hypothesis) is

$$\tau_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{8.3}$$

and the components of the heat flux vector result

$$q_k = -\frac{\mu}{\Pr(\gamma - 1)} \frac{\partial T}{\partial x_k} \tag{8.4}$$

being γ the specific heat ratio and Pr the fluid Prandtl number. The dimensionless viscosity ($\mu_{\mathrm{dim}}/\mu_\infty$) is calculated in terms of the fluid temperature by the Sutherland's law

$$\mu = T^{3/2} \frac{1 + S_0 / T_0}{T + S_0 / T_0} \tag{8.5}$$

where $S_0$ = 110.4 K and $T_0$ = 273.15 K for air[64]. The Prandtl number can be assumed to be constant for gases at moderate temperature (Pr≈0.71 for air up to 600 K) and the state equations (4.11) are adopted. Note that dimensionless variables shall be assumed in this chapter unless otherwise stated.

## 8.2 Flow solution approach

The discrete solution approach is based on the inviscid scheme presented in Chapter 4, but supplemented by a central-type discretization of the viscous terms. In addition, a Reynolds-Averaged Navier-Stokes (RANS) approach is followed to account for flow turbulence effects using the algebraic model of Baldwin and Lomax (Baldwin & Lomax, 1978). The complete flow solution procedure and boundary conditions are described next.

### 8.2.1 Equations discretization

Following the standard discretization approach in the FPM, the semi-discrete system of equations (8.1) is written at a sampling point $\boldsymbol{x}_i$ by

$$\frac{\partial \hat{\mathbf{U}}_i}{\partial t} = -\frac{\partial \mathbf{F}_i^k}{\partial x_k} + \frac{\partial \mathbf{G}_i^k}{\partial x_k} = \mathcal{F}_i + \mathcal{G}_i \tag{8.6}$$

where the tilde indicates approximated variables. As in Section 4.2, the convective term is recasted as

$$\mathcal{F}_i = -\frac{\partial \mathbf{F}_i^k}{\partial x_k} = -2\sum_{j \neq i} b_{ij}^k \left[ \mathbf{F}_{ij}^k - \mathbf{F}_i^k \right] \qquad \forall j \in \Omega_i \tag{8.7}$$

[64] Note that the employment of Sutherland's law requires defining an additional reference condition regarding the freestream temperature.

where $\boldsymbol{b}_{ij}$ are the cloud metric coefficients, $\mathbf{F}^{k}{}_{i} = \mathbf{F}^{k}(\mathbf{U}_{i})$ is the $k$-component of the convective flux vector and $\mathbf{F}^{k}{}_{ij}$ is a numerical flux introduced for stabilization purposes. This can be computed according to Eq. (4.28) (upwind approach) or by means of the artificial diffusion models described in Section 4.2.3. The viscous terms in Eq. (8.6) are discretized at each point $\boldsymbol{x}_{i}$ by

$$\mathcal{G}_{i} = \frac{\partial \mathbf{G}_{i}^{k}}{\partial x_{k}} = \sum_{j} b_{ij}^{k} \mathbf{G}_{j}^{k} \qquad \forall j \in \Omega_{i} \tag{8.8}$$

being $\mathbf{G}^{k} = \mathbf{G}^{k}(\mathbf{U}_{j})$ the $k$-component of the nodal viscous fluxes. Note that Eq. (8.8) is equivalent to a central-type approximation of the diffusive terms[65].

With the aim of simplifying the computation of the viscous fluxes, two simplifications are adopted. Firstly, the terms involving spatial derivatives in Eq. (8.8) (i.e. the viscous stresses and the heat flux vector) are computed previously to assemble the diffusive flux by applying Eq. (3.12). This procedure avoids the need to store second-derivative coefficients, and although some smoothing is introduced in the calculations, it has not demonstrated any negative impact on the numerical solution. The second simplification adopted consists on assuming that the approximated and internal nodal parameters at a point $\boldsymbol{x}_{i}$ are coincident when computing the viscous fluxes, i.e. $\hat{\boldsymbol{\tau}} \cong \boldsymbol{\tau}$ and $\hat{\boldsymbol{q}} \cong \boldsymbol{q}$. In this way, recovery of the nodal parameters is avoided and the computational cost reduced. The error introduced by this assumption is expected to be low because well-behaved FPM approximations tend to interpolate point data. In addition, as it will be seen later, this assumption is particularly true for clouds in boundary and shear layers due to the discretization procedure adopted in such domain zones.

[65] In order to discretize the viscous flux terms, different meshless approaches can be found in the literature. For example, when quadratic approximation bases are used, the viscous flux derivatives can be expanded into their components (first and second derivatives of velocity and temperature) and computed in a direct manner (e.g. (Munikrishna & Balakrishnan, 2011; Oñate, Idelsohn, Zienkiewicz, Taylor & Sacco, 1996b)). This approach is direct but requires a considerable enlargement of the data structure for storing the second derivative coefficients, which can be disadvantageous in view of practical applications. Other authors propose, for instance, to use discretization approaches for the viscous fluxes in the line of Eq. (8.7), but using modified gradients for each edge to avoid odd-even instabilities and increase the robustness of the discretization, e.g. (Katz, 2009; Kennet, Timme, Angulo & Badcock, 2012). In this respect, it should be noted that no stability issues related to the discretization of the viscous fluxes were observed in this work. However, it presumably can depend on the local cloud discretization, hence the proposal of the aforementioned authors.

### 8.2.2 Time integration

The time integration of the semi-discrete scheme (8.6) is performed according to the explicit time-marching procedure described in Section 4.3. However, the allowable time step in Eq. (4.61) is modified to account for viscous diffusivity. In non-dimensional form, the time step size is computed by

$$\Delta \mathrm{t}_i = \mathcal{C} \min\left( \frac{\left\| \boldsymbol{l}_{ji} \right\|}{\left| u_i^k \hat{n}_{ij}^k \right| + c_i}, \frac{\mathrm{Re}_\infty}{2\mathrm{M}_\infty} \frac{l_{ji}^k l_{ji}^k}{\upsilon_i} \right) \qquad \forall j \in \Omega_i \tag{8.9}$$

where $\hat{\boldsymbol{n}}$ is a unit vector in the direction of $\boldsymbol{l}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$, $c$ is the speed of the sound, $\upsilon = \mu/\rho$ is the kinematic viscosity and $\mathcal{C}$ is the Courant number. Local time-stepping and residual smoothing are applied to accelerate the convergence in steady-state problems.

### 8.2.3 Turbulence modeling

Flow turbulence is accounted for through averaged RANS equations using the algebraic model of Baldwin and Lomax (B&L) (Baldwin & Lomax, 1978). In this way, the turbulent stresses and heat fluxes are made proportional to the laminar ones and obtained by adding a turbulent viscosity $\mu_T$ and Prandtl number $\mathrm{Pr}_T$ to the molecular values, i.e. the terms $\mu$ and $\mu/\mathrm{Pr}$ in Eqs. (8.4) are replaced by $(\mu+\mu_T)$ and $(\mu/\mathrm{Pr}+\mu_T/\mathrm{Pr}_T)$, respectively. In general, the turbulent Prandtl number can be considered to be constant (0.9 in boundary layer flows and 0.5 for shear layers in air) and the turbulent (eddy) viscosity is computed as a non-local function of the flow variables. In the B&L model a two-layer approach is adopted, i.e.

$$\mu_T = \begin{cases} \mu_{T_I} \;\; y \le y_c & \text{inner region} \\ \mu_{T_O} \;\; y > y_c & \text{outer region} \end{cases} \tag{8.10}$$

where $y_c$ is the smallest wall distance at which $\mu_{Ti} = \mu_{To.}$

The inner-layer viscosity is based on the mixing length formulation and can be expressed in non-dimensional form by

$$\mu_{T_i} = \frac{\mathrm{Re}_\infty}{\mathrm{M}_\infty} \rho l^2 \left\| \Omega \right\| \tag{8.11}$$

where $\rho$ is the fluid density, $\boldsymbol{\Omega}$ is the vorticity vector and $l$ is a damped mixing length (van Driest approach) obtained by

$$l = k y \left( 1 - e^{-y+/A+} \right) \tag{8.12}$$

The variable $y$ denotes the normal distance from the wall (or wake centerline) and the dimensionless distance parameter $y^+$ is given by

$$y^+ = \frac{y u_\tau}{\upsilon_w} \tag{8.13}$$

where $u_\tau$ is the friction velocity

$$u_\tau = \left( {}^{\tau_w}\!/_{\rho_w} \right)^{1/2} \tag{8.14}$$

and $\tau_w$, $\rho_w$ and $\upsilon_w$ are dimensional wall shear stress, density and kinematic molecular viscosity, respectively. The wall shear stress can be obtained by

$$\tau_w = \mu_w \left. \frac{\partial \mathrm{u_t}}{\partial y} \right|_{y=0} \tag{8.15}$$

where $\mathrm{u_t}$ is the component of the velocity tangent to the wall and $y$ is the normal wall distance. Eqs. (8.13) and (8.14) can be recasted in terms of dimensionless variables by using the relationships given in Section 4.1.2.

The outer eddy viscosity is computed by

$$\mu_{T_O} = \frac{\mathrm{Re}_\infty}{\mathrm{M}_\infty} \rho \alpha C_{cp} F_{wake} F_{Kleb} \tag{8.16}$$

where $\mathrm{F_{Kleb}}$ (Klebanoff function) accounts for laminar-turbulent intermittency

$$F_{Kleb} = \left( 1 + 5.5 \left( y \frac{C_{Kleb}}{y_{\max}} \right)^6 \right)^{-1} \tag{8.17}$$

and $\mathrm{y_{max}}$ is the distance from the wall at which the function

$$F(y) = y \left\| \Omega \right\| \left( 1 - e^{-y+/A+} \right) \tag{8.18}$$

achieves its maximum ($\mathrm{F_{max}}$). This value is computed for $\mathrm{y} \geq 0$ along each viscous station. Note that the function F(y) can exhibit different peaks in flows with mixed boundary and shear layers; therefore, further verifications can be required to avoid wrong peaks affecting the computed length scales. Usually, the innermost peak occurring in the boundary layer region tends to be used, see (Mavriplis, 1990). The function $\mathrm{F_{wake}}$ is given by

$$F_{wake} = \min \left( y_{\max} F_{\max} \ , C_{wk} y_{\max} \frac{u_{dif}^2}{F_{\max}} \right) \tag{8.19}$$

being $u_{dif}$ the difference between the maximum and minimum magnitude of the velocity tangent to the wall ($u_t$) along the viscous station profile.

The standard model closure coefficients are A+=26, k=0.40, $\alpha$=0.0168, $C_{cp}$=1.6, $C_{Kleb}$=0.3 and $C_{wk}$=0.25, but these can be adjusted according to the problem under consideration, cf. (Granville, 1987; Wilcox, 1994). In addition, a transition location $x_{tr}$ should be specified, and the turbulent viscosity is set to zero for upwind stations $x < x_{tr}$.

#### 8.2.3.1 Scope and implementation issues

The B&L model is robust and simple and gives accurate results in attached flow computations. However, the results obtained in boundary layers subject to elevated adverse gradients, almost or fully separated, should be used with care (the mixing-length model is no longer valid in those situations). Some modifications intended to improve separated flow computations can be found in the literature; see for instance (Rostand, 1989). Since the applications in this work involve only attached flows, these are within the scope of the B&L turbulence model and thus, no modification or particular setting of the scheme is applied.

From the point of view of the numerical implementation, the B&L model is best suited for structured grid settings, though unstructured extensions are also possible, e.g. by using background meshes (Mavriplis, 1990; Rostand, 1989). In this work, as it will be seen later, the near-wall discretization has a structured distribution which facilitates the definition of the viscous stations; hence the computation of the eddy viscosity profiles. The domain points outside the near-wall area (which usually belong to an unstructured-type isotropic discretization) are associated with a given viscous station according to a minimum normal wall distance criterion. These outer points can be somewhat misaligned with the viscous station assigned and this can introduce some inaccuracies; however, this approach is considered acceptable for the analysis of attached boundary layer flows since viscous effects are mostly confined in the near-wall area. For more complex flows involving separated flow areas, multiple shear layers and/or solid bodies, improved approaches can be necessary. It should be noticed though, that the B&L model is not well suited for such kinds of computations.

### 8.2.4 Boundary conditions

The boundary conditions applied in this work for the solution of viscous flows are far-field and no-slip solid wall conditions. The former follows the same lines described in Section 4.5 and the latter can be expressed by

$$\boldsymbol{u} = \boldsymbol{0} \quad \forall \boldsymbol{x}_i \in \Gamma_w \tag{8.20}$$

and enforced by cancelling the velocity components at the corresponding boundary points. Additionally, since the viscous terms account for heat transfer effects, a temperature or heat flux condition should be also specified on solid walls. A fixed temperature condition can be enforced in the conservative variables vector through the total energy by setting

$$\overline{e}_t = \frac{1}{\gamma(\gamma-1)} T_w \quad \forall \boldsymbol{x}_i \in \Gamma_{\mathrm{Tfix}} \tag{8.21}$$

where $T_w$ is the specified dimensionless wall temperature and the overbar indicates corrected values. On the other hand, heat fluxes can be directly specified when computing the viscous fluxes (8.2) by setting

$$\overline{\boldsymbol{q}}_i = \boldsymbol{q}_{fix} \quad \forall \boldsymbol{x}_i \in \Gamma_{q_{fix}} \tag{8.22}$$

being $\boldsymbol{q}_{\mathrm{fix}}$ a specified heat flux vector. A typical assumption in aerodynamic flows is to consider adiabatic walls. In such cases, the heat flux vector across the boundaries must be set to zero. This is accomplished by enforcing

$$\overline{\boldsymbol{q}}_i = \boldsymbol{q}_i - \left(\boldsymbol{q}_i \cdot \hat{\boldsymbol{n}}_i\right)\hat{\boldsymbol{n}}_i \quad \forall \boldsymbol{x}_i \in \Gamma_{q_n=0} \tag{8.23}$$

where $\hat{\boldsymbol{n}}_i$ and $\boldsymbol{q}_i$ are the boundary normal and heat flux vectors, respectively.

### 8.2.5 Limitations for high-Reynolds number flows

The discrete scheme described above can be used for solving general compressible viscous flows. However, in the case of problems involving high-Reynolds numbers, the required highly-stretched point discretizations can cause problems, mainly related to the ill-conditioning of the cloud matrices and the instability of the collocation method[66], reducing the robustness and accuracy of the solution process. Although numerous alternative procedures have been proposed to this end (cf. Section 3.6), fully effective solutions are still hardly achieved if the anisotropy of the cloud is elevated (the extra cost involved can be also considerable). Consequently, more effective approaches are necessary to solve high-Reynolds number problems.

In order to address these issues, a methodology to generate viscous discretizations for high-Reynolds number boundary layer flows and a simple procedure to build local FPM approximations in the resulting anisotropic clouds have been developed in the context of this work. These are presented in the next section.

[66] Additional problems affecting the performance of the flow solution schemes may also arise in highly stretched discretizations; see an analysis in (Xinrong, Satoru & Kazuhiro, 2013).

## 8.3 Near-wall discretization

The methodology adopted to construct near-wall viscous discretizations is inspired in the advanced layers methods used to generate hybrid conventional meshes (Pirzadeh, 1994). Broadly speaking, the procedure proposed involves the following three main steps:

a. Given a user-supplied isotropic discretization (valid for instance for inviscid analysis) a set of unit surface normal vectors (marching directions) with root at the vertices of the boundary mesh is computed. Additionally, a Laplacian smoothing is applied for the normal vectors obtained.

b. At each surface vertex (viscous station), a one-dimensional (1D) distribution of points is obtained along the marching direction for a given number of points and minimum wall distance. The point spacing can follow geometric or exponential variations (cf. (Garimella & Shepard, 2000)) calculated according to a specified total thickness of the boundary layer, or in terms of the local spacing in the original (isotropic) discretization. The latter leads to a smoother size transition between the anisotropic and isotropic discretization areas and is the main choice used in this work.

c. Once the positions of the new nodes at each viscous station are known, the original isotropic discretization is deformed to accommodate these new points by using the spring network technique described in Section 7.3 (Batina, 1989). The updated location of the original points is obtained by a few Jacobi iterations with a low computational cost. It should be noticed that the vertices on the body surface are duplicated and one of them is displaced with the original discretization (these points are called interface points).

This approach is robust, cost effective and truly meshless because only information from the local cloud is needed. Moreover, the manual input required is very low and different problems, requiring different geometrical characteristics of the near-body discretization, can be solved in an automatic manner without modification of the base geometry. Figure 96 shows a discretization generated in this way for a RAE 2822 airfoil.

Note that according to the characteristics of the problem to be solved, the proposed discretization procedure may require further improvements. For instance, geometry features such as sharp edges may need the definition of double marching directions at each boundary vertex to obtain a more uniform distribution of points near the body. Also, problems involving multiple bodies may need additional tests to avoid the overlapping of the viscous stations and/or boundary layer growths outside the analysis domain. As shown by the extensive work done in the field of hybrid mesh generation, all these issues can be handled with minor modifications and computational effort (see examples in (Garimella & Shepard, 2000; Ito & Nakahashi, 2002; Löhner, 2001); and always profiting from the advantages that the topology and the characteristics of the final discretization are not major issues for the meshless technique. Extensions of these methods can be also implemented to discretize wakes, an aspect which is not accounted for in this work.

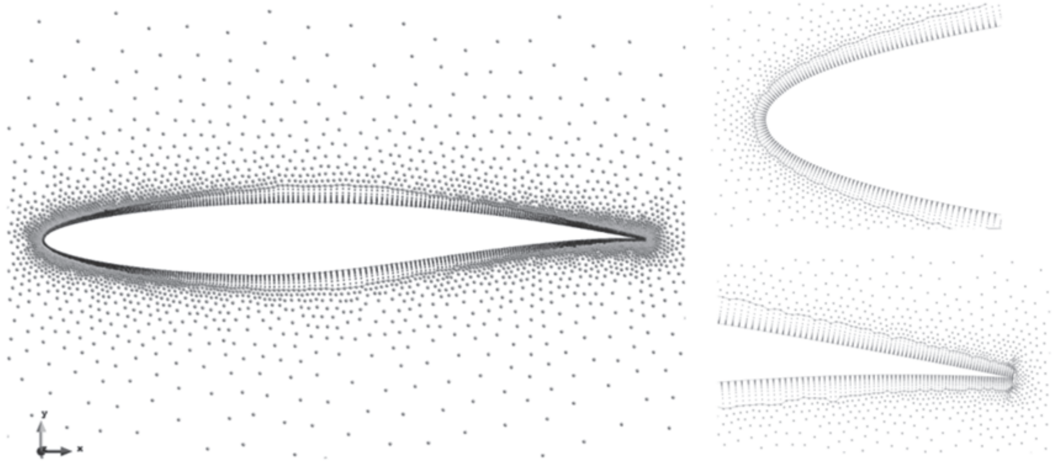

Figure 96. Discretization of a RAE 2822 airfoil using 20 viscous layers with minimum point spacing 1.0e-5 (the airfoil has unit chord and blunt trailing edge).

## 8.4 Local approximation in near-wall areas

The near-wall area in this work is resolved by considering local coordinate systems (η,ξ) defined at each viscous station with origin at the root boundary vertices (the axis η points along the marching (normal) direction and ξ is tangent). The construction of the local cloud of points and the numerical approximation at each point in the near-wall area are described below.

### 8.4.1 Local cloud of points

The local cloud of points in the anisotropic near-wall areas are easily obtained due to the structured nature of the discretization. In this work, point stencils which are symmetric with respect to the star point are generally employed, except at the boundary vertices, where asymmetric point distributions are required (see Figure 97).

In order to construct the clouds corresponding to the interface points (originally located on the boundaries of the isotropic discretization), the procedure described in Section 3.6 is applied, but without accounting for the anisotropic (new) points and using the updated point positions corresponding to the final discretization. Once the first neighbors layer is computed for a given interface cloud, the nearest new points corresponding to the anisotropic discretization (inherited points) are added as shown in Figure 97 (right). This provides the necessary overlapping between isotropic and anisotropic domain areas. Note that the inherited points for each isotropic cloud are automatically identified during the generation process, thus avoiding extra cost. Finally, further points are added to the interface clouds (according to increasing distances from the star point) to complete the cloud and/or improve the quality of the local approximation (cf. Section 3.5).

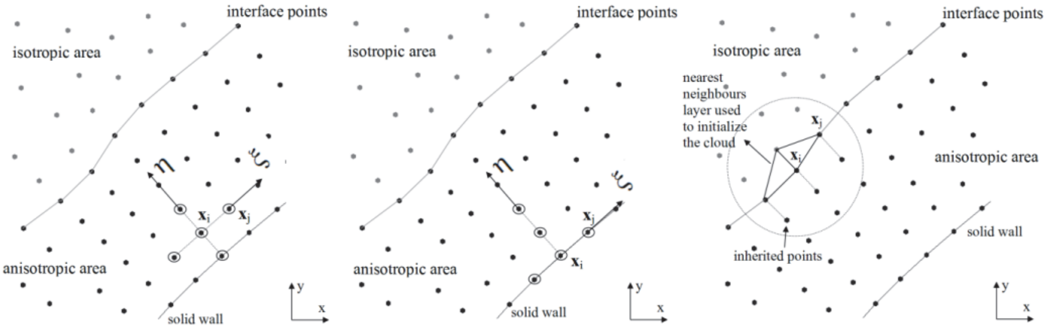

Figure 97. Examples of local clouds in the anisotropic and interface domain areas.

## 8.4.2 Numerical approximation

The numerical approximation at a point $\boldsymbol{x}_i$ in the anisotropic area is obtained by applying the methodology described in Section 3.1 along each local direction (η,ξ). This reduces the ill-conditioning of the cloud matrices and eliminates to a large extent robustness and accuracy issues. The 1D quadratic approximation bases employed are, for instance in the η direction

$$\boldsymbol{p}^T(\eta) = \left[1\,\eta\,\eta^2\right] \tag{8.24}$$

with local scaled coordinates defined by

$$\eta = (\boldsymbol{x}_j - \boldsymbol{x}_i)\cdot\hat{\eta} / \Delta\eta_{\max} \tag{8.25}$$

being $\hat{\eta}$ the unit marching direction vector and $\Delta\eta_{\max}$ the distance between the star point and the furthest point in that direction. The definitions are similar for the approximation bases in the ξ direction. There, the employment of projections onto the local direction axis accounts for some misalignment that may arise due to the body curvature and slight differences in the height of neighbor viscous layers (the number of points at each station is the same).

Regarding the number of points used to solve the minimization problem, three-point local supports are considered along each local direction to keep the clouds as compact as possible (this results in clouds with 5 points). This choice has additional advantages. Among them, the approximation functions interpolate point data, the minimization problem[67] can be solved explicitly and, in addition, weighting is not required.

Finally, once the approximations to the derivatives in the local system (η,ξ) are obtained, these are mapped back to the global coordinates (x,y) by applying the chain rule. The observed order of accuracy obtained for the gradients in the anisotropic area is second-order for symmetric clouds of points but tends to first-order with the asymmetry of the cloud (Fischer, 1996). It should be noticed that no significant effects of the body curvature on the accuracy of the approximation were observed for the range of problems addressed in this work.

## 8.5 Application examples

In order to evaluate the performance of the proposed technique (mainly accuracy and convergence), several numerical examples are presented in this section. The tests cases chosen are typical verification and validation benchmarks involving laminar and turbulent flows, ranging from low subsonic to supersonic freestream Mach numbers.

### 8.5.1 Flat-plate boundary layer flows

The test cases provided in this section involve subsonic, transonic and supersonic boundary layer flows along a flat plate. The original isotropic FPM model consist of a unstructured distribution of 3099 points, with 135 boundary points along the plate which are used to generate the anisotropic boundary layer discretization. The number of layers at each viscous station varies according to the problem between 15 and 20. Figure 98 shows the discrete analysis domain and the boundary conditions employed. A quadratic FPM approximation is computed in local clouds with 15-20 points in the isotropic domain area and three-point supports are used in each local direction in the anisotropic area (5 points per cloud). The flow solver uses 3rd-order MUSCL extrapolation with the van Albada limiter. It should be noticed that the eigenvalues in the Roe solver are limited to a very low value (1.0e-4 to 1.0e-3 times the spectral radius) to allow the numerical dissipation scales with the Mach number as the surface is approached (see Eqs. (4.37)). This reduces the contamination of the physical viscous dissipation near the wall. A density

[67] The spatial derivatives in such kind of stencils can be also obtained by using polynomial finite differences along each direction, see for instance (Hoffmann & Chiang, 2000). However, this would place a restriction on the number of supporting points which reduces the flexibility of the meshless technique.

residual drop of at least four orders of magnitude is considered to achieve solution convergence in the present simulations.

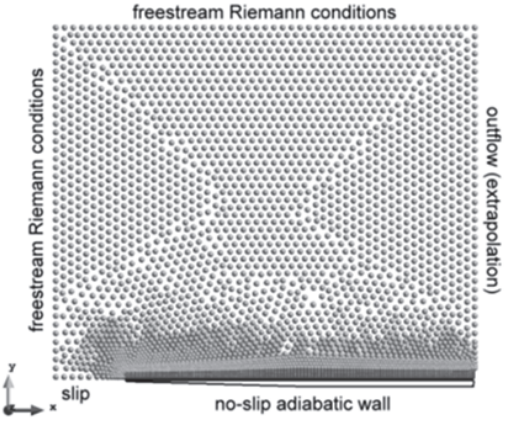

Figure 98. Problem discretization and boundary conditions used for the flat-plate boundary layer problems (3099 original isotropic points).

The first three examples presented below involves laminar flows with a low Reynolds number $Re_L = 5K$ based on the length of the plate (L=1). At each viscous station, 15 points are generated using an exponential variation of the normal point spacing with a minimum height 1.0e-3. Figure 99 shows non-dimensional velocity profiles in the boundary layer computed for a freestream Mach number $M_\infty = 0.1$ at a streamwise station x/L = 0.8 ($Re_x = 4K$). The skin friction coefficient along the plate is also given. The results exhibit a good agreement with the exact Blasius solution.

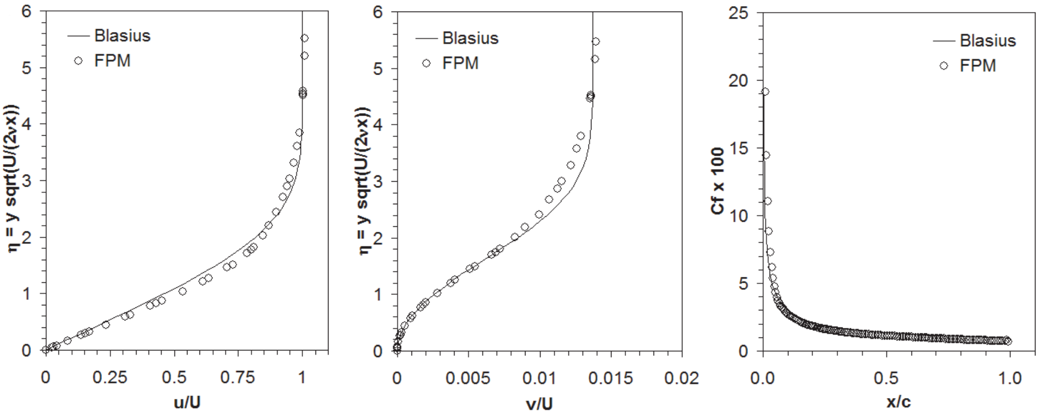

Figure 99. Comparison of velocity profiles (x/L=0.8) and skin friction along the plate. $M_\infty = 0.1$ and $Re_L = 5K$.

In the next example the freestream Mach number is set to $M_\infty = 0.8$. The numerical results for a streamwise station x/L = 0.6 ($Re_x = 3K$) are compared with the analytical solution presented in (Mavriplis, Jameson & Martinelli, 1989). There, the plate normal distance is scaled with the local density through the boundary layer according to

$$Y = \int_0^\infty \rho / \rho_\infty dy \tag{8.26}$$

which corresponds to the Howarth-Dorodnitsyn transformation of the incompressible Blasius solution; see (Stewartson, 1964). A good correlation between numerical and exact solutions can be observed in Figure 100.

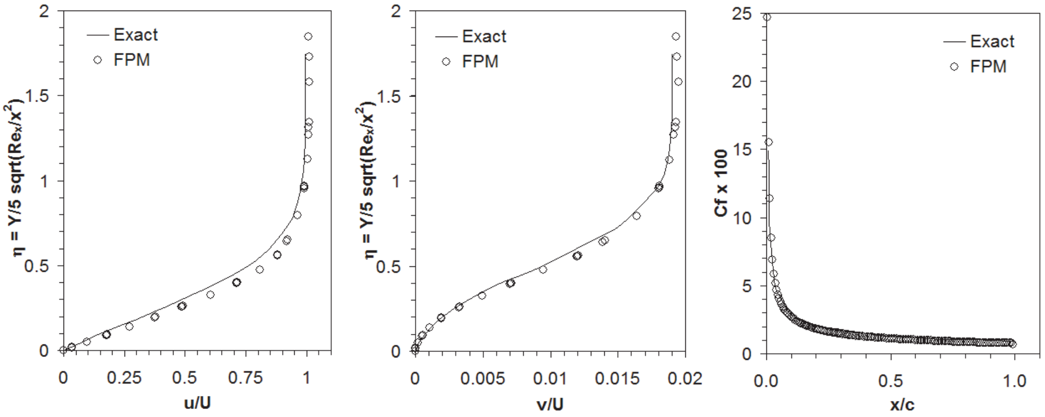

Figure 100. Computed and exact velocity profiles (x/L=0.6) and skin friction. $M_\infty$ = 0.8 and $Re_L$ = 5K.

A problem involving shock-boundary layer interaction at $M_\infty$ = 4 is presented next for $Re_L$ = 5K, Pr = 0.75 and $\gamma$ = 1.4. In this example, all the flow variables are prescribed at freestream boundaries (see Figure 98) and the flow outlet is let free. Both adiabatic and fixed temperature ($T_w/T_\infty$=1) wall conditions are adopted for the simulations. The results in Figure 101 demonstrate a satisfactory agreement with the analytical Crocco-Busemann's solution; see (Van Driest, 1952; White, 1991).

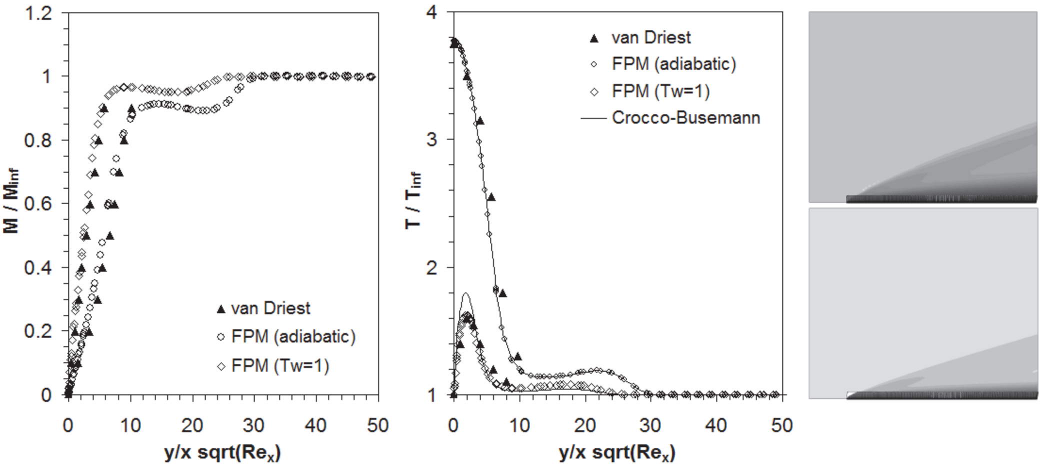

Figure 101. Comparison of calculated Mach and temperature profiles (x/L=0.8) with the exact solutions. Mach number contours are shown on the right. $M_\infty$ = 4.0 and $Re_L$ = 5K (Pr = 0.75).

The next example involves a low-subsonic laminar flow with $M_\infty$ = 0.1 but at a higher Reynolds number $Re_L$ = 100K. The minimum distance from the wall is set to 5.0e-4 (y+ is of order unity) and 20 points are used along each viscous station. The calculated boundary layer profiles (x/L=0.8) and skin friction compare satisfactorily with the incompressible Blasius solution (see

Figure 102). The results are slightly better than those previously obtained, presumably due to the finer discretization of the boundary layer.

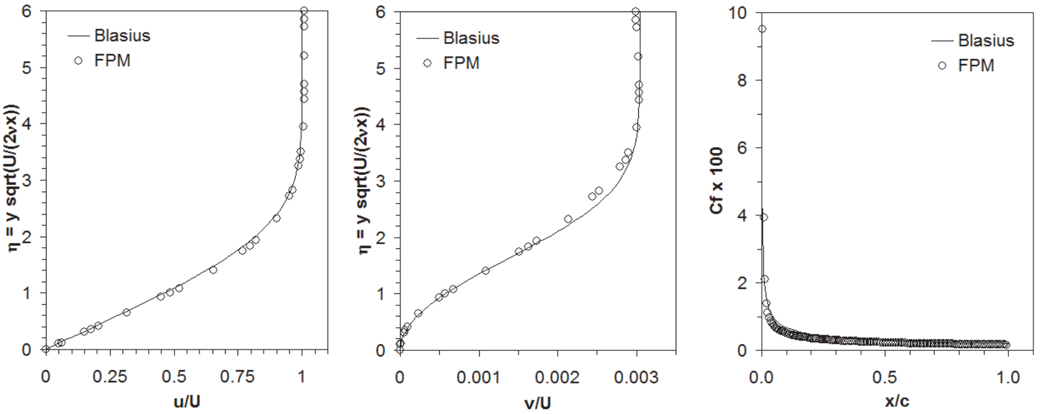

Figure 102. Comparison of velocity profiles (x/L=0.8) and skin friction along the plate. $M_\infty$=0.1 and $Re_L$ = 100K.

Finally, a high-Reynolds turbulent flow is solved for $M_\infty$ = 0.1 and $Re_L$ = 11M. The B&L turbulence model is adopted in the simulation and the transition is triggered immediately after the leading edge of the plate. The minimum distance from the wall is set to 2.5e-6 (minimum y+ about 1) and 20 points are used along each viscous station. The maximum cloud aspect ratio in this example is 1:3000. Dimensionless velocity profiles are presented in Figure 103 for different streamwise stations along the plate. The agreement of these results with the experimental data from Wieghardt and Tillman (Wieghardt & Tillman, 1951)[68] is satisfactory.

### 8.5.2 RAE 2822 airfoil

High-subsonic and transonic turbulent flows around a RAE 2822 airfoil are computed in this section according to the experimental AGARD Test Cases 3 and 6 presented in (Cook, Mcdonald & Firmin, 1979). The model employed has unit chord (c=1) and the unstructured (original) discretization consists of 4598 points. This is employed to construct an anisotropic boundary layer discretization which reaches 11781 points (561 points along the airfoil and 20 viscous layers at each station). The minimum height from the surface is set to 1.0e-5 (minimum y+ about 1) and a blunt trailing edge is considered. The maximum cloud aspect ratio in this model is about 1:1000. Some views of the discrete model have been presented in Figure 96.

[68] The data points employed in the present comparisons are those tabulated in (Coles & Hirst, 1969) and available online at (NASA (2012)).

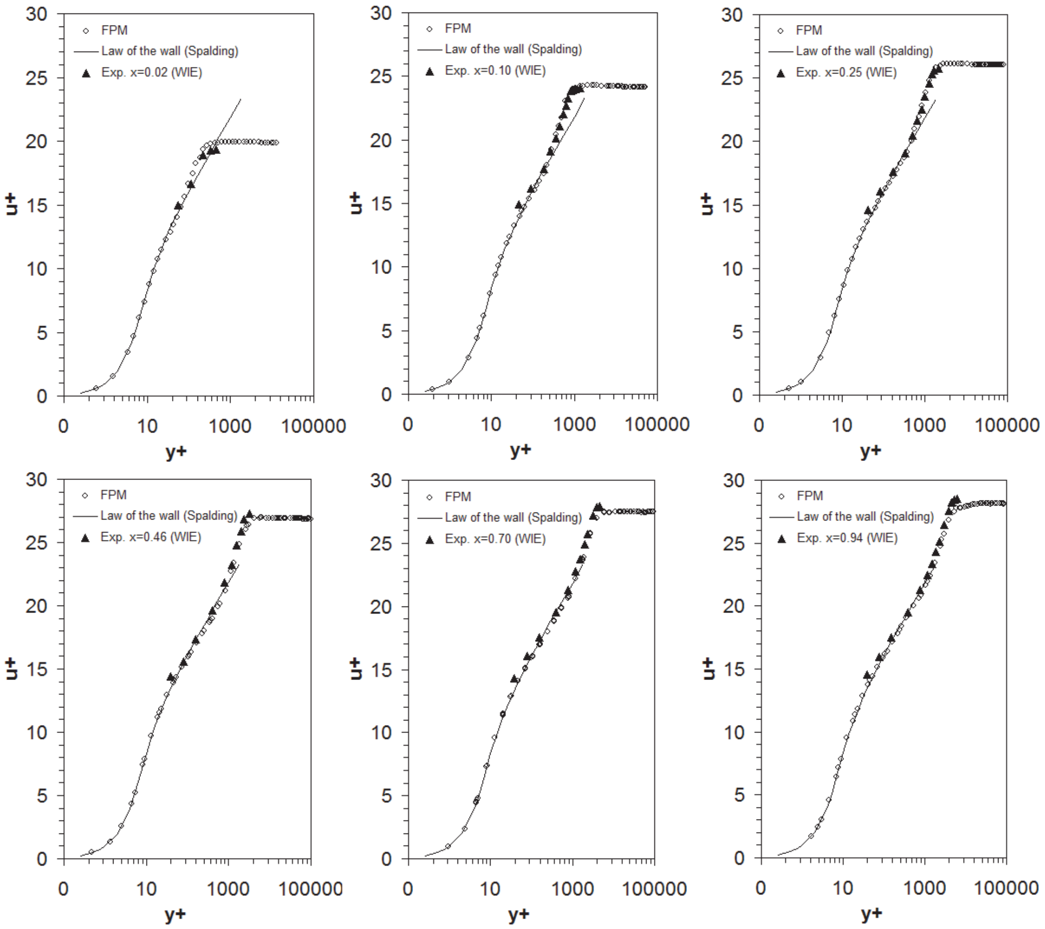

Figure 103. Computed and experimental velocity profiles at different streamwise stations along the flat plate. $M_\infty = 0.1$ and $Re_L = 11M$.

The FPM approximation is set similarly to the previous examples; the clouds in the isotropic domain area range between 15-20 points and three-point approximation supports are used in each local direction within the boundary layer (5 points per cloud). The Baldwin and Lomax turbulence model is used. According to the location of the transition trip in the wind tunnel model, the transition is triggered at x/c=0.03. The boundary conditions adopted are freestream Riemann conditions for outer domain boundaries and no-slip adiabatic wall along the airfoil contour.

The first test case involves a freestream Mach number $M_\infty$=0.6 with $Re_c$=6.3M (Case 3) and an angle of attack of $\alpha$ = 2.44º (corrected by wind tunnel wall interference). The second example is a transonic test case with $M_\infty$ = 0.725, $Re_c$ = 6.5M and $\alpha$ = 2.31º (Case 6), which is solved using the same discrete model and problem setting. In this case the relatively low intensity of the shock wave on the upper side of the airfoil allows the flow to remain attached.

The computed FPM solutions (presented in Figure 104) correlate well with the experimental data and particularly good results are obtained for the pressure distribution around the airfoil. Note that although the shock position in the transonic test case may be quite sensitive to the turbulence model employed, it is well captured in the numerical results. Regarding skin friction, the numerical solutions seem to overestimate the wall viscous stresses, particularly towards the leading edge of the airfoil (a better agreement is obtained near the trailing edge). Several experiments performed have not revealed significant grid dependence issues in these examples, but only small effects due to the parameters of the Roe-MUSCL solver and the turbulence model. As regards the discrepancies found in skin friction, it is important to note that a similar trend can also be seen for these test cases in other published results; for example (Spalart & Allmaras, 1992). To illustrate this, some data points taken from the aforementioned reference are included in Figure 104 for the transonic test case. These points, indicated as BL(SA), were computed with a similar turbulence model to that employed in this work. As can be seen, the agreement with the FPM results is fairly good. It should be noticed that certain effects due to uncertainties in the experimental conditions and data reduction procedures, as well as the suitability of the turbulence model adopted, may cause the differences observed with respect to the experimental results. This behavior deserves further investigation.

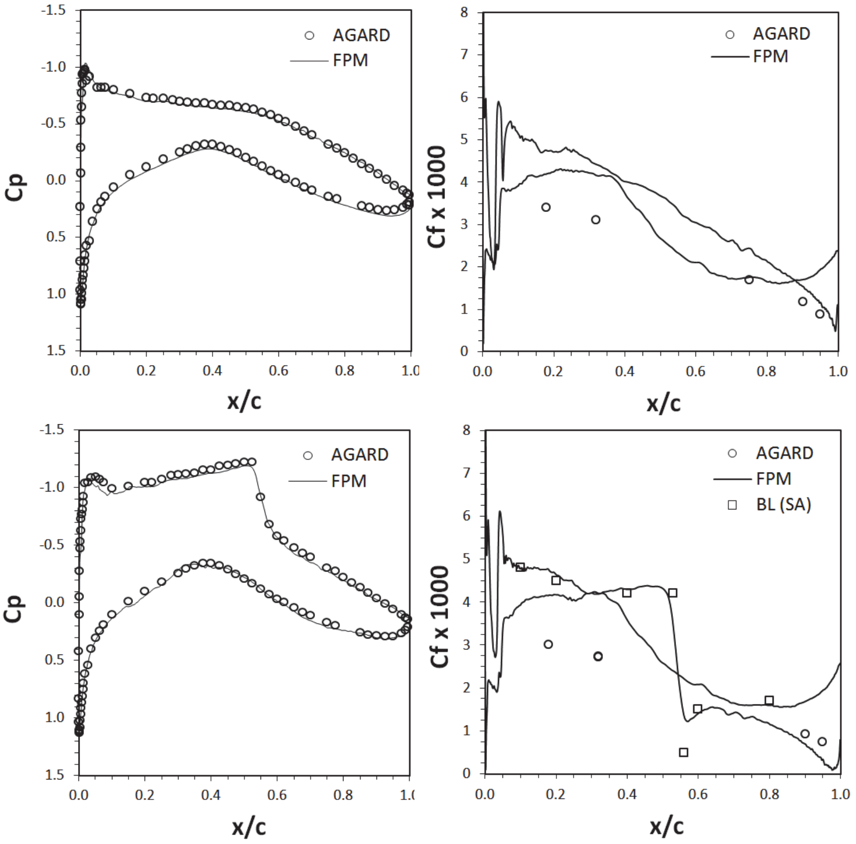

Figure 104. Pressure and skin friction distributions for the RAE airfoil. Top: $M_\infty = 0.6$ and $Re_L = 6.3M$ (Case 3). Bottom: $M_\infty = 0.725$ and $Re_L = 6.5M$ (Case 6).

The iterative convergence of the aerodynamic forces is presented in Figure 105 for Cases 3 and 6. The computed normal force coefficients show a good correlation with the experimental measurements (these depend mainly on the pressure distribution) but a higher drag coefficient is obtained. The latter could be the result of the overestimation of the skin viscous stresses, but the poor resolution of the airfoil wake (not accounted for in this work) and the known low performance of the B&L model in this area may have some impact. As regards the speed of convergence, it should be noticed that the small point spacing within the boundary layer imposes significant restrictions to the permissible time step sizes, making the solution convergence slow. Hence, more specific solution approaches, based on implicit schemes or improved convergence acceleration procedures (like multigrid), are to be required to solve this kind of problems with efficiency; see interesting meshless implementations in (Katz & Jameson, 2009b; Kennet, Timme, Angulo & Badcock, 2012).

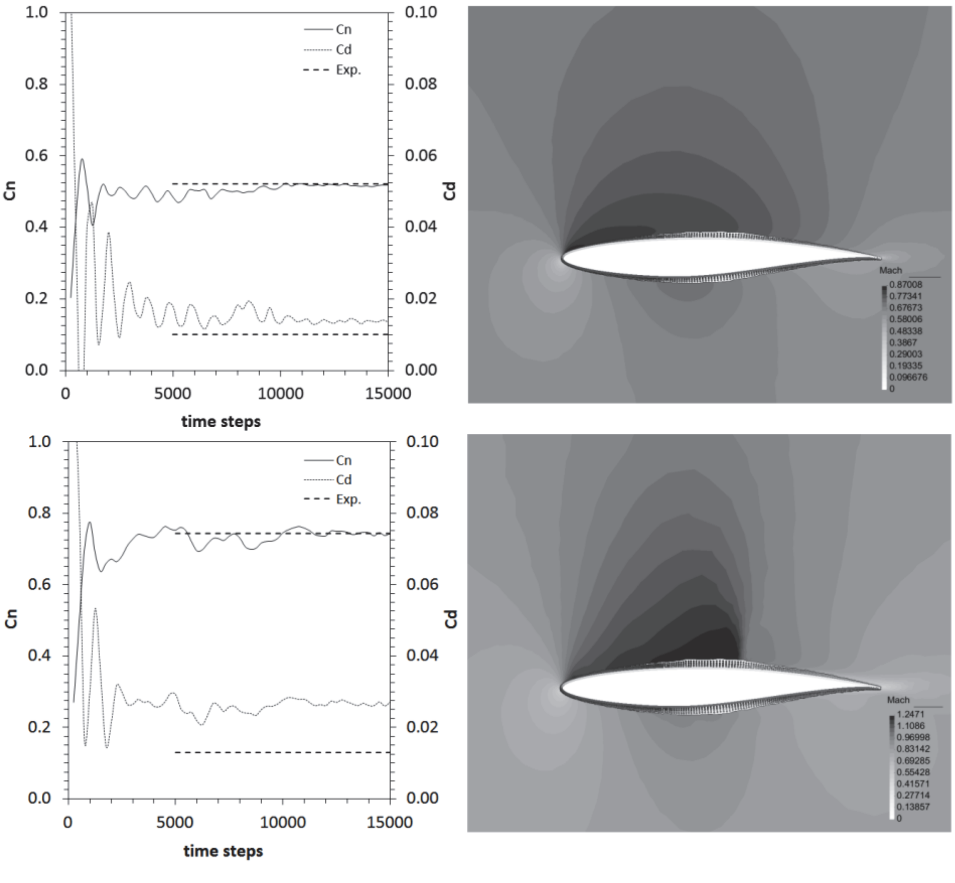

Figure 105. Aerodynamic forces convergence and Mach number contours for the RAE airfoil. Top: $M_\infty = 0.6$ and $Re_L = 6.3M$ (Case 3). Bottom: $M_\infty = 0.725$ and $Re_L = 6.5M$ (Case 6).

## 8.6 Concluding remarks

A meshless approach for solving high-Reynolds viscous flows has been presented in this chapter. The developments focused mainly on exploiting meshless capabilities on two main aspects of the methodology: simplicity and automation of the viscous discretization, and the construction of a robust and computationally efficient numerical approximation in highly-anisotropic clouds of points. As far as the flow solution scheme is concerned, the approach adopted is quite standard and robust. The solver is based on the one proposed for the inviscid Euler equations, but extended to account for viscous effects. An algebraic turbulence model has been also implemented.

Regarding the viscous (boundary layer) discretization, a simple and versatile technique based on advancing layers was proposed. The discretization requirements are lower in the meshless context and this allows generation procedures to be simplified in relation to similar mesh-based implementations. Another additional advantage of this approach is that the original problem discretization, valid for instance for inviscid analysis, ensures to some extent the validity of the resulting viscous grid. Moreover, once the original (inviscid) model is generated for a given problem, this technique allows testing different viscous discretizations, adapted to particular flow conditions, with minimum turnaround times.

Concerning the construction of the FPM approximation in highly-anisotropic clouds, a simple and essentially one-dimensional procedure was proposed. This approach, which builds on the structured-like distribution of points in the boundary layer, has proven to be fairly accurate and robust. Furthermore, the reduced spatial support required (5 points per cloud) improves the locality of the computations (with a minimum smearing of the flow features) and reduces the computational cost significantly. The time savings in the viscous layers are very important because more than half of the total domain points in practical problems are usually located in such areas.

The numerical applications presented (involving attached boundary layer flows) have demonstrated satisfactory accuracy, comparable to that observed in conventional analyses, but taking advantage of the meshless procedures. This result encourages to extend the proposed FPM methodology in order to address three-dimensional problems.

# 9 RELATED APPLICATIONS: SHALLOW WATER EQUATIONS

Shallow water equations (SWEs), which are a particular case of the general Navier-Stokes equations, model a wide range of geophysical flow problems which have very important social, economic and environmental implications. As in other fields of numerical computation, there are particular applications of the SWEs where meshless approaches could make significant contributions. Thus, taking advantage of the similarity between the solution procedures typically used for SWEs and that developed in this work, the capability of the FPM to address shallow water problems is investigated.

This chapter is organized as follows. Section 9.1 outlines the problem governing equations, and the meshless solution approach is described in Section 9.2. Typical validation and verification test cases are presented in Section 9.3 to demonstrate the performance of the proposed methodology. In addition, with the aim of exploring the possibilities the FPM has for solving realistic problems, a numerical simulation of the 2004 Indian Ocean tsunami is also presented in this section. The problem is mainly focused on the propagation and effects of the tsunami waves on the southern coast of Thailand. The adaptive methodology developed in the preceding chapters is applied for the exploitation of meshless advantages.

## 9.1 The shallow water equations

Shallow water flows are characterized by the fact that the vertical length scale of the problem (depth) is considerably smaller than the characteristic wavelength of the disturbances traveling within the analysis domain. Since the flow movement in these problems is predominantly horizontal, some assumptions can be made in order to simplify the analysis. Thus, assuming that the acceleration in the depth direction is negligible and the pressure is mostly hydrostatic, the incompressible Navier-Stokes equations are integrated in depth. This procedure leads to a set of hyperbolic equations in a horizontal plane (two-dimensions) where the flow variables are all considered as depth-averages; see for instance (Zienkiewicz, Taylor & Nithiarasu, 2005). These equations, known as shallow water equations, can be written in conservative (dimensional) differential form as

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}^k}{\partial x_k}=\frac{\partial \mathbf{G}^k}{\partial x_k}+\mathbf{Q} \tag{9.1}$$

where **U** is the vector of conservative variables, $\mathbf{F}^k$ and $\mathbf{G}^k$ are the advective and the diffusive fluxes in the spatial direction $x_k$ and **Q** is a source term. The vectors **U**, $\mathbf{F}^k$ and $\mathbf{G}^k$ are given by

$$\mathbf{U}=\begin{bmatrix} H \\ Hu_1 \\ Hu_2 \end{bmatrix} \;,\; \mathbf{F}^k=\begin{bmatrix} Hu_k \\ Hu_1u_k+\delta_{1k}g(H^2-h^2)/2 \\ Hu_2u_k+\delta_{2k}g(H^2-h^2)/2 \end{bmatrix} \;,\; \mathbf{G}^k=\begin{bmatrix} 0 \\ \bar{\tau}_{1k} \\ \bar{\tau}_{2k} \end{bmatrix} \tag{9.2}$$

where $H = h + \eta$ is the total fluid depth, defined as the sum of the resting depth $h$ and the free-surface height $\eta$. The $k$-component of the mean velocity vector is $u_k$, $g$ is the gravity acceleration and $\delta_{ik}$ is the Kronecker's delta. The depth-averaged viscous shear stresses $\bar{\tau}_{ik}$ are given by

$$\bar{\tau}_{ik}=\upsilon H\left(\frac{\partial u_i}{\partial x_k}+\frac{\partial u_k}{\partial x_i}\right) \tag{9.3}$$

in which $\upsilon$ is the kinematic viscosity of the fluid. If the effects due to free-surface stresses (e.g. wind) and variations in the atmospheric pressure are omitted, the source term **Q** can be written as (Wang & Liu, 2005)

$$\mathbf{Q}=\begin{bmatrix} 0 \\ g\eta S_{o_1}-gHS_{f_1}+HC_fu_2 \\ g\eta S_{o_2}-gHS_{f_2}-HC_fu_1 \end{bmatrix} \tag{9.4}$$

being $C_f$ the Coriolis parameter and $S_{f_k}$ bottom friction terms. The latter can be defined as (Glaister, 1993)

$$S_{f_k}=\left(\frac{n}{\kappa}\right)^2\frac{u_k\left\|\boldsymbol{u}\right\|}{H^{4/3}} \tag{9.5}$$

where $n$ is the Gauckler-Manning coefficient and $\kappa$ is a dimensional constant equal to 1.0 for SI units or 1.486 for US customary units. Finally, the bed slope terms in Eq. (9.4) can be obtained by

$$S_{o_k}={\partial h}/{\partial x_k} \tag{9.6}$$

and the resting depth profile h($\boldsymbol{x}$) (bathymetry) should be specified for each particular problem under analysis.

## 9.2 Flow solution approach

Since the SWEs are obtained as a particular case of the general Navier-Stokes equations, the numerical solution approaches are generally based on similar techniques[69]. In this work, the basic upwind scheme developed in Section 4 and Section 8 is adopted. The solution methodology is described below. It should be noticed that dimensional forms of the equations are employed.

### 9.2.1 Equations discretization

The semi-discrete system of SWEs can be written at a sampling point $\boldsymbol{x}_i$ by

$$\frac{\partial \hat{\mathbf{U}}_i}{\partial t} = -\frac{\partial \mathbf{F}_i^k}{\partial x_k} + \frac{\partial \mathbf{G}_i^k}{\partial x_k} + \hat{\mathbf{Q}}_i = \mathcal{F}_i + \mathcal{G}_i + \mathcal{Q}_i \tag{9.7}$$

where the tilde indicates approximated variables. The convective term $\mathcal{F}_i$ is discretized as in Section 4.2, by recasting the derivative as

$$\mathcal{F}_i = -\frac{\partial \mathbf{F}_i^k}{\partial x_k} = -2\sum_{j\neq i} b_{ij}^k \left[\mathbf{F}_{ij}^k - \mathbf{F}_i^k\right] \qquad \forall j \in \Omega_i \tag{9.8}$$

where $\mathbf{F}_{ij}^k$ is the $k$-component of an unknown numerical flux computed at the midpoint of the edge connecting the star point $\boldsymbol{x}_i$ to another point $\boldsymbol{x}_j$ in the local cloud and $\mathbf{F}_i^k = \mathbf{F}^k(\mathbf{U}_i)$. The numerical flux, which is introduced for stabilization purposes, can be obtained by solving an approximate Riemann problem at each edge (upwind approach) or by means of the artificial diffusion models described in Section 4.2.3. Here the first procedure is adopted. Thus, the numerical flux is defined according to Eq. (4.28) by

$$\mathbf{F}_{ij}^k = \tfrac{1}{2}\left(\mathbf{F}_j^k + \mathbf{F}_i^k\right) - \tfrac{1}{2}\left|\mathbf{A}_{\hat{n}}(\mathbf{U}_i, \mathbf{U}_j)\right|\left(\mathbf{U}_j - \mathbf{U}_i\right)\hat{n}_{ij}^k \tag{9.9}$$

[69] Examples of some typical solution techniques are the Eulerian method due to Ata and Soulaïmani (Ata & Soulaïmani, 2005), the Lagrangian scheme presented by Hon et al. (Hon, Cheung, Mao & Kansa, 1999) and the mixed FV-meshless application proposed by Cueto-Felgueroso et al. with the purpose of achieving higher-order discretizations on unstructured grids (Cueto-Felgueroso, Colominas, Fe, Navarrina & Casteleiro, 2000). Moreover, upwind-type solution schemes can be found in (Ambrosi, 1995; Glaister, 1988; Wang & Liu, 2005).

where $\hat{\boldsymbol{n}}_{ij}$ is a unit vector in the direction of $\boldsymbol{l}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ and $|\mathbf{A}_{\hat{n}}(\mathbf{U}_i, \mathbf{U}_j)|$ is the positive Roe matrix calculated in the same direction. Details of the computation of this matrix for SWEs can be found in (Wang & Liu, 2005).

The introduction of the upwind flux (9.9) guarantees a non-oscillatory solution, but the resulting scheme is only first-order accurate. Like in Section 4.2, the spatial accuracy is increased by limited MUSCL extrapolation. The procedure is the same as the one used with Euler equations, but here the conservative variables are referenced to the resting fluid depth in the extrapolation procedure. This avoids unwanted effects due to the problem bathymetric data.

The discretization of the diffusive fluxes and source terms in Eq. (9.7) are performed using standard FPM approximations, i.e.

$$\begin{aligned} \mathcal{G}_i &= -\frac{\partial \mathbf{G}_i^k}{\partial x_k} = -\sum_j b_{ij}^k \mathbf{G}_j^k \\ \mathcal{Q}_i &= \mathbf{Q}_i = \sum_j a_{ij} \tilde{\mathbf{Q}}_j \end{aligned} \tag{9.10}$$

where $\mathbf{G}_j^k = \mathbf{G}^k(\mathbf{U}_j)$ and $\mathbf{Q}_j^k = \mathbf{Q}^k(\mathbf{U}_j)$ and $a_{ij}$ and $\boldsymbol{b}_{ij}$ are the cloud metric coefficients. Note that the approximation obtained for the viscous fluxes is equivalent to a central-type difference.

As in Section 8.2.1, the computation of the viscous fluxes and source terms is simplified by assuming that the approximated and nodal parameters of the derivatives at a given point are similar (velocity gradients in the depth-averaged viscous stresses and the $h$ gradient in the bed slope terms). Regarding the treatment of the source term, numerical instabilities may arise in problems presenting, for example, highly-stepped topography or important bed shear stresses. In such cases, improved treatments of the source term may be required; see for instance (Garcia-Navarro & Vazquez-Cendon, 2000). In the present work, no misbehavior related to the source terms was observed in the range of problems addressed.

### 9.2.2 Time integration

The time derivative in the semi-discrete equations (8.7) is approximated according to the time marching scheme described in Section 4.3 (the procedure developed in Section 7.2.2 can be also applied). The only difference is the computation of the allowable time step. This is obtained in the case of SWEs by

$$\Delta \mathrm{t}_i = \mathcal{C} \min\left( \frac{\left\| \boldsymbol{l}_{ji} \right\|}{\left| \boldsymbol{u}_i \cdot \hat{\boldsymbol{n}} \right| + c_i}, \frac{l_{ji}^k l_{ji}^k}{2\upsilon_i} \right) \qquad \forall j \in \Omega_i \qquad (9.11)$$

where $\mathcal{C}$ denotes the Courant number, $c_i = \sqrt{gH_i}$ is the wave speed, and the rest of the variables are interpreted as they have been defined before in Section 4.3 and Section 7.2.2.

### 9.2.3 Boundary conditions

The boundary conditions adopted in shallow water problems are similar to those previously described in Sections 4.5 and 7.2.3. These are discussed in the context of particular applications in the next section.

## 9.3 Application examples

The numerical examples presented in this section involve typical verification and validation test cases and are mainly intended to illustrate the basic performance of the meshless technique. In addition, the methodology is also applied with the adaptive scheme of Chapter 6 to the simulation of the tsunami which hit the Indonesian coast on 26th December 2004. This test case aims at demonstrating the applicability of the present technique to a more realistic shallow water problem.

### 9.3.1 Dam-break flood problem

A straight channel initially divided by a dam is considered in this example. The water depth at the left side of the dam (*$H_L$)* is higher than the depth at the right of the dam ($H_R$), the fluid is at rest at both sides and viscous effects are neglected. The simulation starts when the dam is suddenly removed. This creates a bore wave moving from left to right and a depression wave propagating towards the left. The computational domain is set to 1.0 m long and 0.5 m width and the dam is located at $x$ = 0.5 m. The problem is discretized by a structured distribution of 100×30 points and quadratic approximations are built in clouds having 12-17 points. The initial conditions are $H_L$ = 1 m, $H_R$ = 0.1 and $\boldsymbol{u}_L = \boldsymbol{u}_R = 0$ and the boundaries are modeled as solid slip-walls. A three-stage scheme is adopted to perform time integration, and 3rd-order MUSCL extrapolation with van Albada limiter is used.

Water depth and velocity variations along the channel are calculated at times t = 0, 0.02, 0.05 and 0.08 seconds from the breaking of the dam. These results are compared with the analytical solutions due to Wu et al. (Wu, Huang & Zheng, 1999) in Figure 106, where good agreement is observed. Some snapshots of the fluid surface obtained at the same simulation times are displayed in Figure 107.

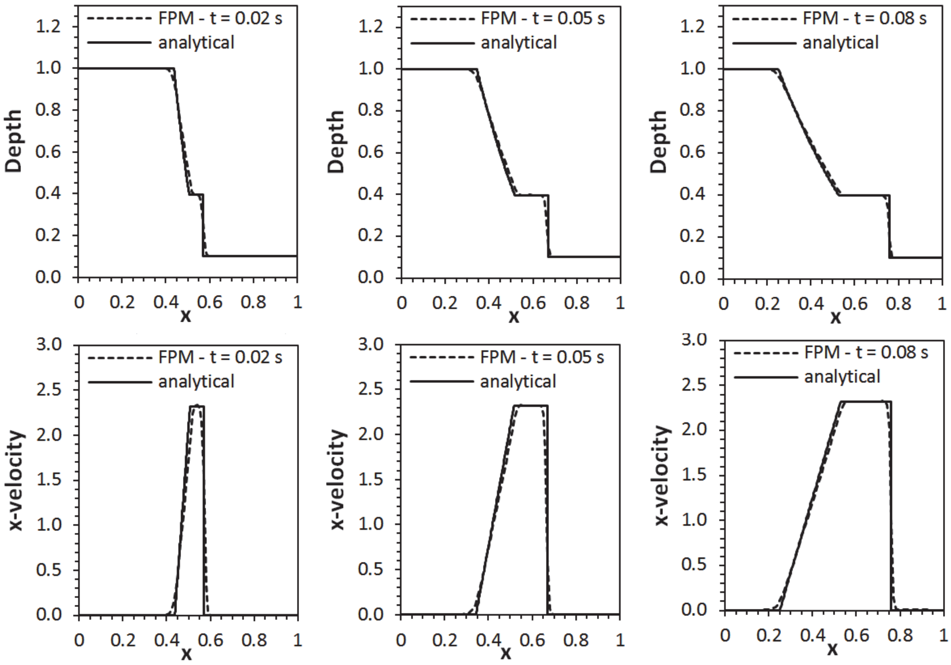

Figure 106. Dam-break problem $H_L/H_R$ = 10. Comparisons between computed and analytical results.

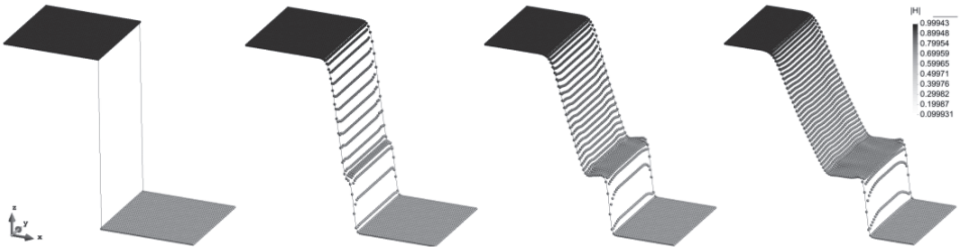

Figure 107. Water surface profiles computed at several times from the breaking of the dam ($H_L/H_R$ = 10). From left to right: 0.0, 0.02, 0.05 and 0.08 secs.

### 9.3.2 Tide-driven flow in a long straight channel

The development of tidal waves along a channel is solved in this example with the purpose to investigate the observed accuracy of the scheme. The problem domain consists of a constant rectangular section channel having 8.8x$10^5$ m long and a resting water depth $h$ = 10 m. The fluid is assumed to be inviscid and the channel walls are modeled as slip boundaries, with exception of the upstream entry, where the following tidal function is enforced

$$\eta(t) = \eta_0 \sin(\omega t) \quad \text{with} \quad \begin{cases} \eta_0 = 0.1\,\text{m} \\ \omega = 2\pi / T \\ T = 12.4\,\text{hr.} \end{cases} \tag{9.12}$$

In order to introduce linear bottom dissipation, the source term given by Eq. (9.4) is defined in this example as

$$\mathbf{Q} = -\tau H \left[0, u_1, u_2\right]^{\mathrm{T}} \tag{9.13}$$

with a friction coefficient $\tau = 5.0\text{E-}5\ \mathrm{s}^{-1}$.

The grid convergence analysis is performed by using four homogeneous distributions of 153, 554, 2161 and 8396 points (with approximately 25, 50, 100, and 200 uniform spaced points along the channel length). The approximation and solver settings are similar to those employed in the preceding example. The steady-state periodic solution of the problem is reached after running a few oscillation cycles. Then, the global solution error is estimated by the following L2 error norm

$$e = \left( \frac{\sum_i (\eta_i^n - \eta_i^e)^2}{\sum_i (\eta_i^e)^2} \right)^{1/2} \tag{9.14}$$

where $\eta_i$ is the free-surface height obtained for a phase angle of 270º along the centerline of the channel, and the subscripts *n* and *e* indicate calculated and exact reference values. The latter corresponds to the analytical linearized solution of the problem obtained in (Lynch & Gray, 1978).

The convergence of the error as a function of the point spacing is depicted in Figure 108. There, the observed order of accuracy of the scheme is about *p* = 1.3. Although slightly lower in this case, the accuracy obtained does not differ much from that observed for the Euler equations. Figure 109 shows comparisons of FPM and exact free-surface levels at different phase angles during one-period oscillation (n=2161 points). Again, the results show a good agreement with the analytical solution for both, amplitude and wave location.

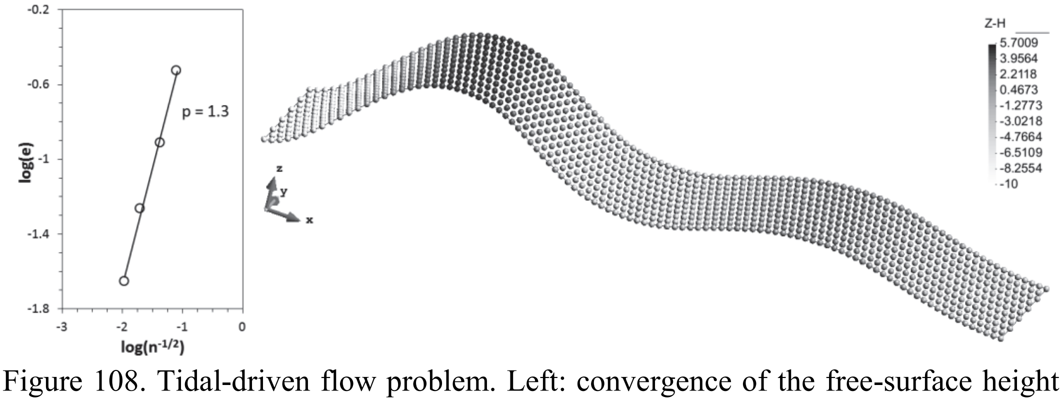

Figure 108. Tidal-driven flow problem. Left: convergence of the free-surface height. Right: surface elevation computed for a phase angle of 270º (n=2161 points). The vertical scale is magnified in order to highlight the displacements.

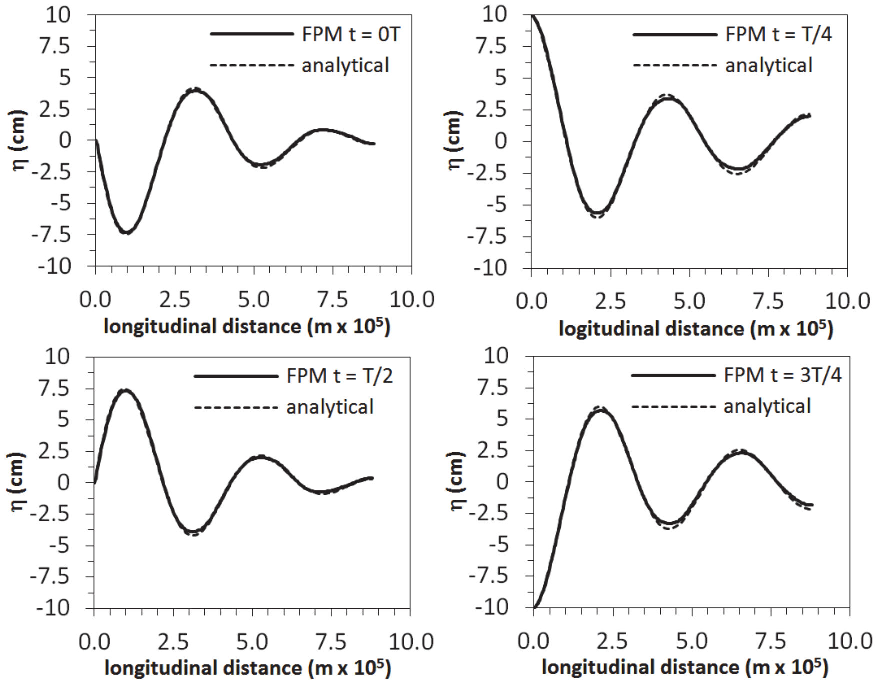

Figure 109. Development of tidal waves along a channel. Comparisons between numerical and analytical results at different times during one-period oscillation.

### 9.3.3 Flow past a channel with a backward step

This example involves a flow through a channel with a backward step and is intended to assess the viscous behavior of the scheme. The step, located at a distance of 12 m from the upstream flow entry, has dimension 1 m and the downstream flow outlet is located 13 m ahead the step (see Figure 110). The computational domain is discretized by a non-structured distribution of 1577 points and the problem settings are identical to the previous examples. The velocity at the inflow boundary and the fluid viscosity are set in such a way that the Reynolds number, based on the dimension of the step, is Re = 73 ($u$=0.5 m/s and υ=0.00685 m$^2$/s). A still water level of $H$ = 1 m is prescribed at the outflow boundary and no-slip walls are considered along the channel.

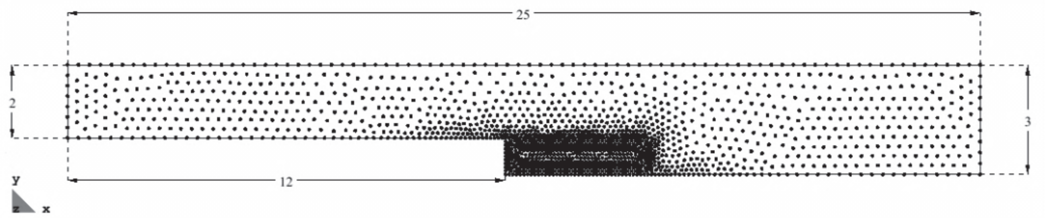

Figure 110. Top-view of the channel with a backward step displaying point discretization (n=1577 points).

The flow reattachment after the step is investigated and the results are compared which those presented in (Wang & Liu, 2005). Figure 111 shows that the computed reattachment length is approximately 3.97 m and this result is very close to the 3.95 m reported by the aforementioned authors. According to the applied boundary conditions, a still water level of $H = 1$ m is obtained throughout the channel.

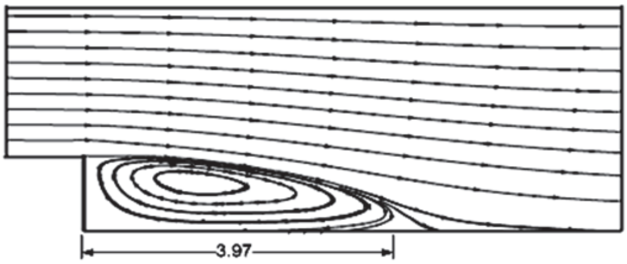

Figure 111. Top-view of the flow past a backward step showing the reattachment length.

## 9.3.4 Flow in a channel with varying width and bed slope

An inviscid flow in a channel with a smooth constriction and variable bed slope is solved here to assess the behavior of bed slope terms. The channel is $1.0 \text{x} 10^4$ m long and its width varies according to a cosine function, from 1000 m at the inflow boundary (left-side) to 500 m at the throat, increasing afterwards to 1000 m at the outflow boundary. The bed slope is set to 1:100 (downward) along the channel excepting for the zone between $4500 \leq x \leq 5500$ m where the bed slope is 1:50 (see Figure 112). The computational domain is discretized by a homogeneous distribution of 5584 points and the problem settings are similar to those used in the previous examples. A volumetric flow rate $Q = 2000 \text{ m}^3\text{s}^{-1}$ is prescribed at the channel entry; whereas the depth is extrapolated from the fluid side at the outflow boundary. Slip conditions are enforced on the walls along the channel.

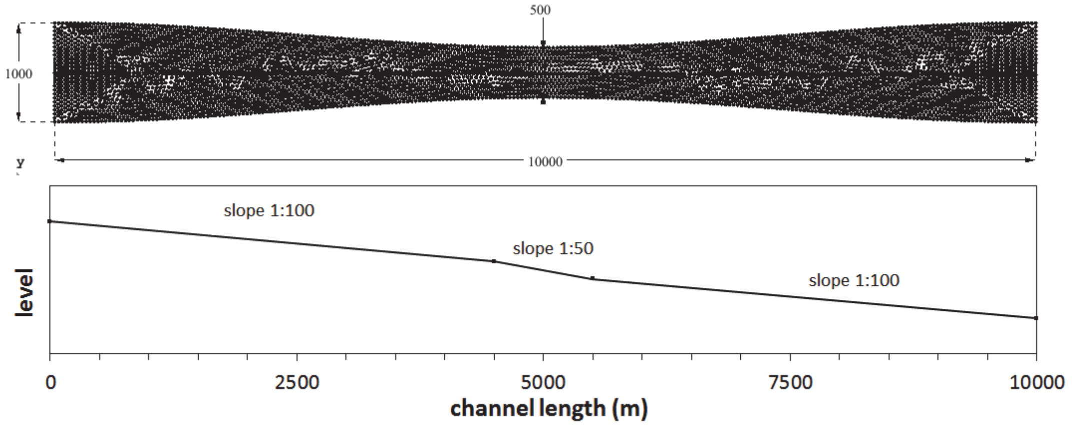

Figure 112. Top-view of the channel showing point discretization (5584 points) and bed slope variation.

Open channel problems generally lack exact solutions; thus these have been extensively studied numerically and several reference results are available in the literature. In the present case, the FPM computation is compared with the numerical solution obtained in (Glaister, 1993). This is computed by using a one-dimensional finite-differences flux-splitting scheme on a domain having 300 uniform cells. Comparisons of water depth, Froude number and volumetric flow along the centerline of the channel are presented in Figure 113. A good agreement between the FPM results and the reference Glaister's solution is obtained.

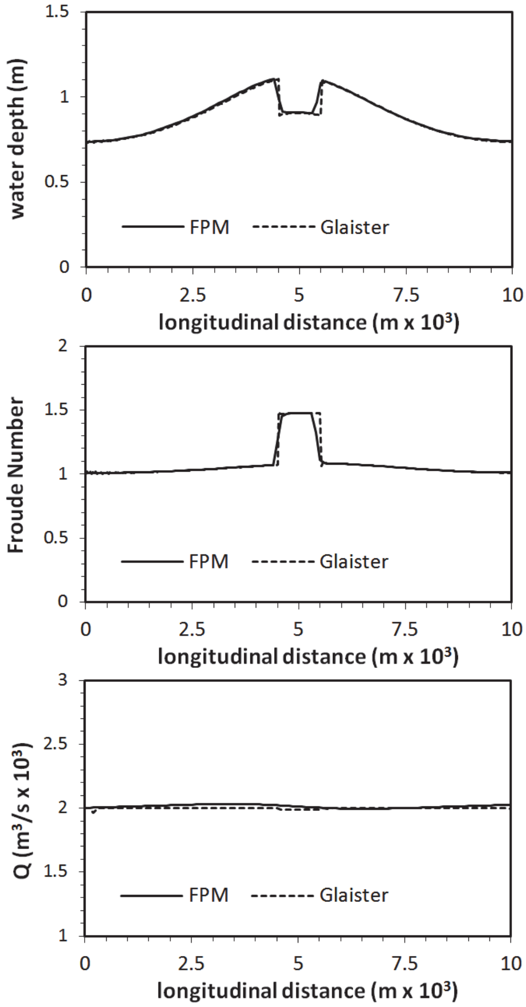

Figure 113. Comparison between FPM and the reference numerical solutions for a channel with varying width and slope.

The FPM water surface profile is shown in Figure 114. There, the hydraulic jumps induced by the changes in bed slope at the central part of the channel can be observed.

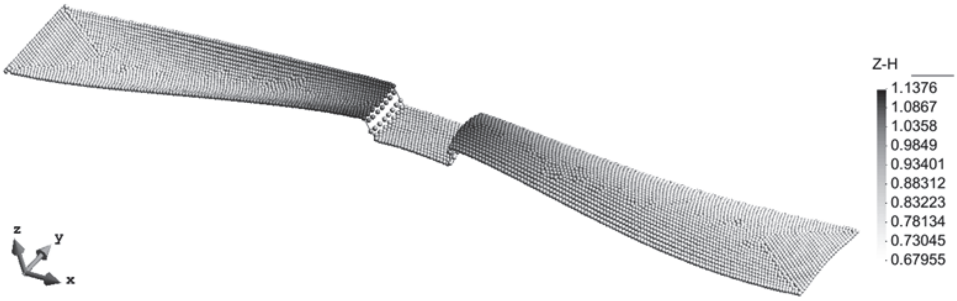

Figure 114. Computed FPM water surface profile along the channel (the vertical scale is augmented in order to highlight surface elevation)

### 9.3.5 Adaptive simulation of the 2004 Indian Ocean tsunami

The Indian Ocean earthquake, which struck the northern coast of Sumatra on 26th December 2004, generated an unprecedented tsunami with devastating consequences in terms of loss of human lives, infrastructure and environmental damage affecting mainly Indonesia, Sri Lanka, India and Thailand. Such a natural disaster led many research groups around the world to focus on developing improved early alert systems, in which numerical models play an essential role (particularly those based on SWEs); see for instance (Annunziato & Best, 2005), (Bernard, Mofjeld, Titov, Synolakis & González, 2006) and (Annunziato, 2007). Although significant advances have been made in forecasting technology in recent years and mature methodologies are available, current trends pursue even more accurate models with improved capabilities and higher computational efficiency. Following these lines, a preliminary exploration of the possibilities of the FPM in this field is carried out.

The numerical application presented in this section focuses on the propagation of the Indian Ocean tsunami waves towards the southern coast of Thailand, particularly on Phuket Island[70]. The adaptive methodology developed in Section 6 is used in the simulation in order to exploit the advantages of the meshless approach. The simulation parameters, computational domain and boundary conditions adopted in this example, as well as the numerical results obtained, are discussed below.

#### 9.3.5.1 Problem set-up

The inviscid SWEs (9.7) are solved taking into account Coriolis and bottom stresses in the source terms (9.4). The initial domain discretization consists of

[70] The data employed in this example was provided by Prof. Worsak Kanok-Nukulchai from the Asian Institute of Technology and Dr. Pruettha Nanakorn from the Sirindhorn International Institute of Technology.

an unstructured distribution of 8240 points and quadratic approximations are built in local clouds with 15-25 points. Third-order MUSCL extrapolation is employed with the van Albada limiter and a three-stage scheme is used to perform time integration. As regards the refinement strategy, the feature-based indicator of Section 6.1.1 is employed using the free surface height (η) as sensing variable. The total simulation time run is approximately 150 minutes counting from the tsunami generation event. The refinement passes are performed at intervals of approximately 2.5 minutes (real simulation time) and a maximum resolution of 2.5 km is set for the adapted discretizations. It is important to note that the successive retreat and rise cycles occurring after the tsunami hits the coastline are not simulated because the topography employed is not reliable enough (resolution) to reproduce this phenomenon with accuracy.

#### 9.3.5.2 Topographical and bathymetric data

The analysis domain and the bathymetric data in the present simulation are those used in (Kanok-Nukulchai & Nanakorn, 2007), where the original data extracted from the ETOPO2 database (NGDC, 2006) was mapped into a FEM mesh to perform the computations. It should be noted that in that work the computational domain was simplified by modeling all the small islands (e.g. Andaman Islands), including Phuket, as a part of the ocean with a water depth of five meters. Such procedure was justified because no local features around the coasts were sought and only the global wave propagation was of interest.

In the present simulation the same bathymetric data is mapped into the point discretization obtained at each refinement level achieving both, bathymetry and topography of the computational model to be improved simultaneously (up to the resolution of the original data). In order to map bathymetric data for each target point in the computational domain, the centroids of the elements in the background bathymetry mesh are used for searching the element where the target point lies on by using bins. Then, the bathymetric data is interpolated to the target point using the standard shape functions of the underlying element. This procedure behaves efficiently even under large discretizations and a considerable speed-up is achieved by increasing the number of bins (notice that the bins data structure must be constructed just once, as the background bathymetry mesh is fixed). Figure 115 shows the geographical delimitation of the analysis domain and the bathymetric data computed for the initial coarse discretization (n=8240).

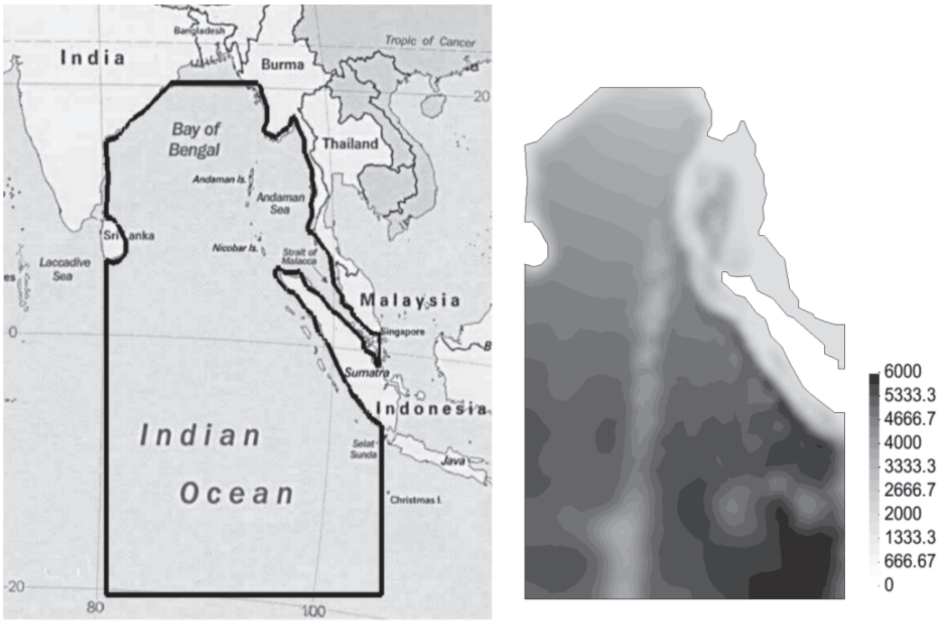

Figure 115. Geographical delimitation of the analysis domain (left) and initial coarse discretization showing bathymetric data in meters (right).

### 9.3.5.3 Initial and boundary conditions

The 26th December 2004 tsunami was triggered by an undersea earthquake located 160 km west of the north coast of Sumatra which reached a magnitude 9.0 on the Richter scale. The earthquake was generated by a subsidence fault of approximately 20 m width and 1000 km length in the confluence of the Eurasian and the Indo-Australian tectonic plates. After the incident, the location and extension of the fault as well as the vertical seabed displacements have been determined and several fault models have been proposed. In accordance with (Kanok-Nukulchai & Nanakorn, 2007), the test case presented here adopts the model given in (NORSAR, 2006).

Following this fault model, a sudden upward movement of the west side of the fault of 5 m and a downward movement of the east side of -3.5 m are considered. The water above the fault is supposed to move in a similar way (no-slip) and these displacements are applied as initial condition for the free-surface water height η. In addition, the initial velocity is considered to be zero everywhere in the analysis domain. Figure 116 depicts the initial free-surface height used in the simulation.

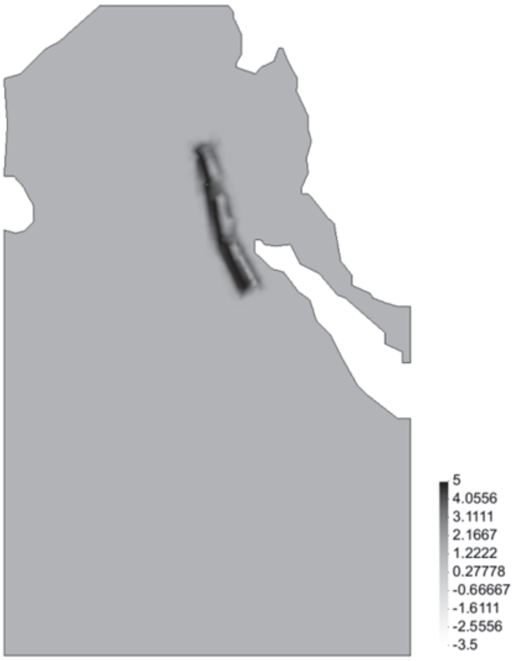

Figure 116. Initial condition for the free-surface height η (in meters).

As regards boundary modeling, slip-velocity and non-reflective conditions are employed. The former are applied on all the coastlines in the computational domain (see Figure 115) by cancelling, at each point, the normal component of the flux vector. Non-reflective conditions are implemented through variables extrapolation from the inside of the domain. These conditions are intended to simulate open-sea boundaries avoiding undesirable reflections.

### 9.3.5.4 Numerical results

The numerical results computed at different simulation times from the initial excitation triggering the tsunami are shown in Figure 117. There, west and east travelling waves, in agreement with the leading north-south orientation of the subsidence fault, can be observed. The west moving waves travel faster than the eastern ones and exhibit longer wavelength with smaller amplitude as the ocean depth is higher on this side of the fault. This fact explains the behavior of the adapted point discretizations displayed in Figure 117, for which only few new points are added on the west side area when the simulation time goes beyond 30 min from the initial excitation. As regards east travelling waves, it can be observed in Figure 117 that the tsunami is very close to the north-west coast of Sumatra (Indonesia) around 15 min after the initial excitation. The timeline elaborated in (Annunziato & Best, 2005) according to media reports indicates that the tsunami hit Banda Aceh (located in the northern coast of Sumatra) 17 min after the earthquake, while several numerical results given in the same reference also estimate around 20 min. The tsunami arrival time computed for Banda Aceh in the present example is approximately 19 min, which is consistent with the aforementioned results (the tsunami arrival time is considered to occur for a zero free-surface height, just before the main wave rises up). Approximately 75 min after the earthquake, the east travelling waves approach the southern coast of Thailand.

Focusing on the tsunami propagation toward Phuket Island, free-surface water levels are plotted in Figure 118 along a straight line (a-a) going from the excitation area to Phuket, covering a distance of approximately 600 km. As can be observed, the main tsunami wave decreases its wavelength and increases its amplitude as it approaches Phuket (ocean depth decreases). For a simulation time of 90 min, the wave crest is located around 70 km from Phuket and the computed free-surface level is around 5 m. All these results follow the same trend presented in (Kanok-Nukulchai & Nanakorn, 2007).

The time history of free-surface water levels computed at Banda Aceh and Phuket Island are shown in Figure 119. Regarding Banda Aceh, the results show a positive tsunami wave, preceded by an ahead running depression, which reaches approximately 11 m height (Figure 119 left). This value is within the range of measurements collected by the United States Geological Survey (USGS, 2005), which reports wave heights between 5 and 13 m at Banda Aceh, increasing up to 30 m at locations along the west coast of the island. The evolution of the sea free-surface level computed at Phuket location (Figure 119 right) also shows the ahead running depression wave causing the water level to recede between 75 and 100 min simulation time. Around 105 min after the initial excitation, a sudden increase of water level occurs due to the arrival of the positive tsunami wave and the water level rises for a period of 20 min, reaching its maximum height at around 125 min simulation time.

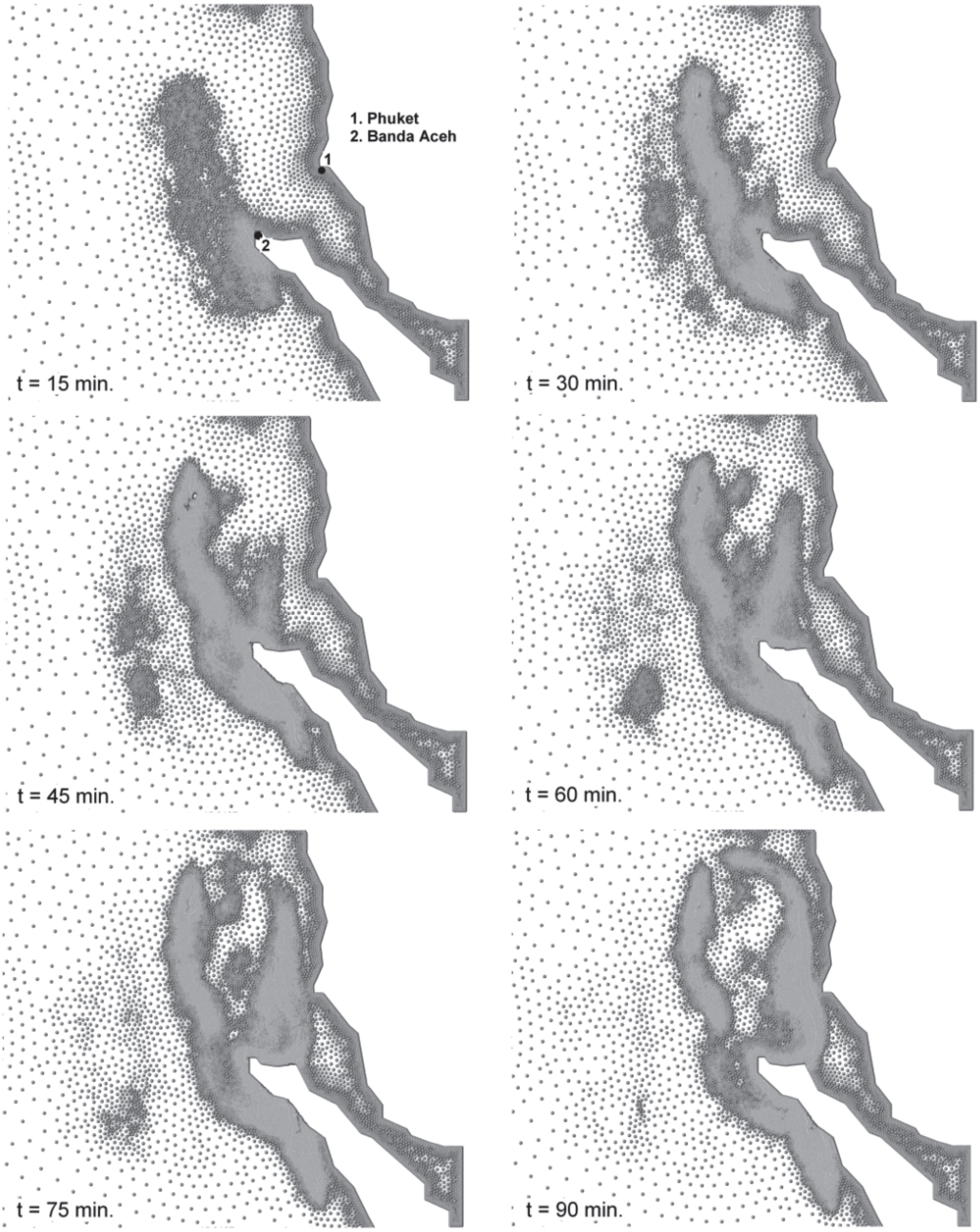

Figure 117. Adapted finite point discretizations computed at different simulation times from the tsunami triggering event.

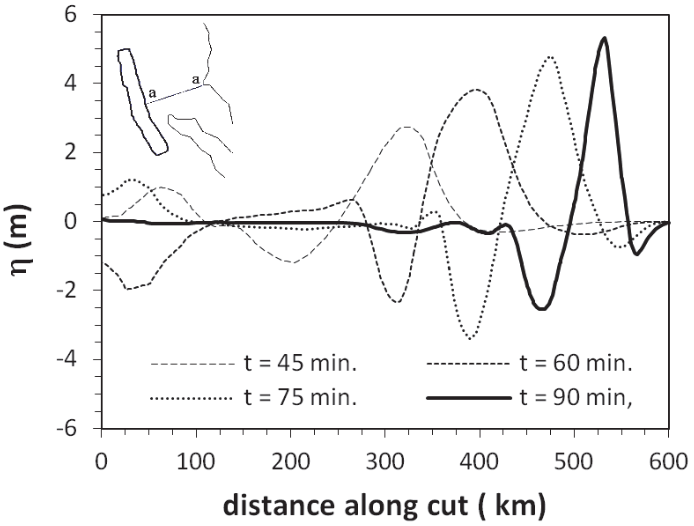

Figure 118. Free-surface water level along the path a-a computed at different times from the initial excitation.

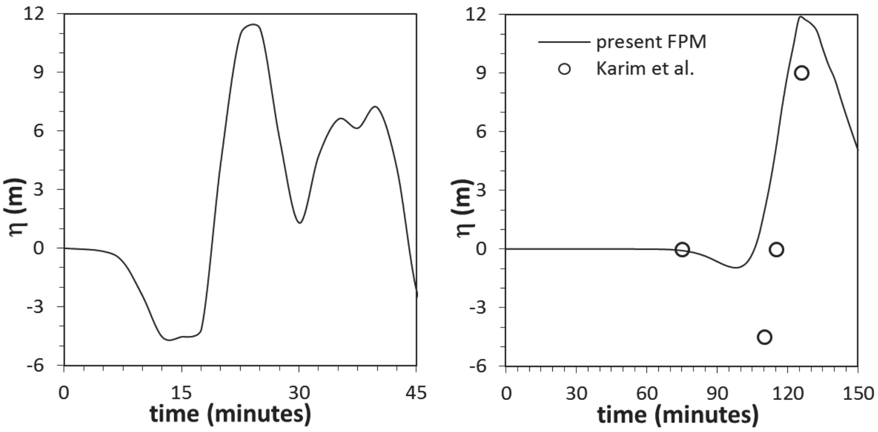

Figure 119. Free-surface water levels computed at Banda Aceh (left) and Phuket Island (right) starting from the initial excitation.

Figure 119 (right) also includes a few point data computed in (Karim, Roy, Ismail & Meah, 2009) for the north-west coast of Phuket. Even though the estimated time of events is pretty similar, the attained water levels exhibit some discrepancy. This fact could be expected because the local topography of Phuket Island is not resolved exactly in the present example. Regarding the maximum wave height, a value around 12 m is computed in this work, which is higher than that obtained in (Karim, Roy, Ismail & Meah, 2009), but not incongruous with the range of values between 1.5 and 12 m the same authors found in the island from South to North. As far as the negative wave is concerned, a larger discrepancy is observed in Figure 119 (right). This can be explained by differences in the initial conditions triggering the tsunami, mainly in the downward seabed displacements, although the approximate topography used in the present example may have also played a role.

The tsunami arrival time calculated at Phuket location is compared in Table 8 with other results published in the literature. In spite of the fact that slight differences are observed, in general the agreement is satisfactory. It should be noticed that small differences in the reported values could be attributed to methodological factors such as the arrival criteria, the model resolution, the initial conditions and the sampled locations adopted in each particular work.

| | | Arrival time (min) |
|---|---|---|
| **Observed data** | News reports (Annunziato & Best, 2005) | 105 |
| | (USGS, 2005) | 90-120 |
| **Computed data** | Present FPM | 105 |
| | (Annunziato & Best, 2005) | 90 |
| | (Karim, Roy, Ismail & Meah, 2009) | 90-110 |

Table 8. Tsunami arrival times for Phuket Island.

Finally, concerning the computational performance of the present tsunami simulation, adapted discretizations are generated with a very low time cost. The number of points in the domain does not surpass 50K at any time step; thus, the computational cost is kept low but not at the expense of the model resolution. This allows performing complete runs on a desktop computer in a few minutes and renders the adaptive finite point technique suitable for fast preliminary analyses. Additionally, the proposed methodology can benefit further from parallel computing strategies (not accounted for in this example), allowing increased resolution with low computational times.

## 9.4 Concluding remarks

An application of the FPM to the solution of shallow water problems has been presented in this chapter. The basic methodology developed for solving compressible aerodynamics problems has been extended in a simple way to deal with the SWEs. According to the results, a satisfactory performance was obtained from the point of view of the accuracy and robustness of the numerical solutions. Although specific tests relative to the computational cost were not performed here, the results obtained in the previous chapters apply (the algorithm is essentially the same). Therefore, it is possible to assume that competitive accuracy-cost ratios, comparable to those obtained with conventional techniques, can be achieved in the present implementation, but profiting from meshless advantages. In this regard, the FPM technique can make interesting contributions in problems that require a high spatial resolution, as well as low simulation and turn-around times.

# 10 CONCLUSIONS AND FUTURE WORK

The aim of this investigation was to contribute to the development of meshless approaches in the field of compressible aerodynamics problems. Based on a particular technique, the Finite Point Method (FPM), the work focused primarily on determining its potential to address practical applications, one of the most outstanding issues in meshless methods. Special attention has been given to those areas where meshless procedures can contribute to current numerical simulation.

As mentioned in the introduction, this work is divided into three main parts. After the introduction and literature review (Chapters 1 and 2), the first part deals with the construction and analysis of the FPM spatial approximation (Chapter 3). The second part is devoted to the development and assessment of the basic (inviscid) flow solution scheme (Chapters 4 and 5). Finally, practical applications and extensions of the basic solution methodology are carried out in the last part of this work (Chapter 6 to 9). The main objectives and achievements in each of these areas are summarized below.

The construction of the meshless spatial approximation followed the standard approaches in finite point techniques. With focus on three-dimensional general approximations, the studies presented aimed at characterizing the behavior of the FPM in order to detect the aspects of the methodology that can be improved to increase its robustness and accuracy; key aspects in view of practical applications. The results obtained allowed to establish ranges for setting the relevant approximation parameters as well as to determine the performance likely to be attained in practice. In this regard, it was found that quadratic polynomial bases lead to a good balance between accuracy and computational cost, while the gain expected from higher-order approximations was quite problem-dependent. Moreover, it was determined that the number of points in the local cloud and the parameter $w$, affecting the shape of the weighting function, are suitable variables to improve the local approximation. With this purpose, an automatic procedure including the adaptive adjustment of these parameters was proposed for constructing robust approximations in general application problems. This procedure simplified the application of the method, reducing problem- and user-dependence without affecting the flexibility of the meshless technique. Other aspects related to the construction of the local approximation, such as the global and local discretization, have also been discussed in this work. The Delaunay point generation technique employed (which was not a focus of this research) showed that suitable point discretizations can be generated with low computational cost and linear complexity. This technique, coupled with the robust local discretization procedure adopted (which largely benefits from parallel implementations), makes meshless discretizations a good alternative to alleviate mesh generation problems in particular applications.

Regarding the discretization of the compressible flow equations, this has followed well-established approaches in the field, but drawing on the meshless character of the methodology. In order to suit the requirements of practical applications, the procedures were designed and implemented with particular emphasis on robustness and efficiency. In this regard, a simplification of the basic FPM technique was proposed to make it competitive in relation to conventional solution techniques. The performance of the resultant flow solver has been studied in detail to evaluate the real capabilities of the meshless approach. The analyses focused on accuracy, computational cost and parallel scalability, and were conducted in comparison with an equivalent Finite Element technique (PUMI) which was taken as a standard for evaluating the meshless solver. Based on a typical inviscid transonic flow simulation, the results obtained demonstrated that the accuracy and computational cost of the basic meshless approach is comparable to those of similar mesh-based implementations. This is an important result that indicates that the meshless solution approach can be competitive, and allows exploiting meshless advantages with efficiency.

The practical applications in this work have been intended not only to show the basic capabilities of the meshless technique, but also to exploit meshless advantages in areas of research with good opportunities for such developments. Automatic *h*-adaptive procedures, moving domain and fluid-structure interaction problems are a sample of the topics explored. In addition, a preliminary meshless approach to solve viscous high-Reynolds number problems has been presented, as well as an application (closely related to compressible flows) that involves shallow water equations. In general, the results were satisfactorily accurate and competitive in terms of computational cost. Meshless advantages (*h*-adaptivity and domain deformation) have been also exploited with robustness and efficiency in the selected problems. This fact constitutes a good starting point towards more challenging applications.

All in all, the FPM methodology developed in this work has revealed a strong potential to deal with practical compressible flow problems. Thus, it offers a way forward to exploit meshless procedures further. It is true that some aspects of the present technique should still be improved in order to meet the objective to develop the FPM into a practical application tool. The implementation of efficient convergence acceleration procedures (e.g. multigrid) and the improvement of the cloud construction technique and the data structures are examples of the present needs. These advances should also be complemented by more extensive testing of the methodology on large-scale problems, using higher-performance hardware platforms. In addition, the solution of viscous three-dimensional high-Reynolds number flows is another key challenge to meet current requirements in computational fluid dynamics. In this regard, the extension of the viscous approach presented here can pave the way for future developments in this field. In brief, the work presented opens a door to promising meshless developments, which can be also extended to other areas of application in numerical simulation.

# Appendix: List of publications

The main publications arising from this research are listed below. Their contributions to this work are also indicated.

Ortega, E., Oñate, E. & Idelsohn, S. (2007). An improved finite point method for tridimensional potential flows. *Computational Mechanics, 40(6), 949-963*. (Chapter 3).

Ortega, E., Oñate, E. & Idelsohn, S. (2009). A finite point method for adaptive three-dimensional compressible flow calculations. *International Journal for Numerical Methods in Fluids, 60(9), 937-971*. (Chapter 4 and Chapter 6).

Ortega, E., Oñate, E., Idelsohn, S. & Buachart, C. (2011). An adaptive finite point method for the shallow water equations. *International Journal for Numerical Methods in Engineering, 88(2), 180–204*. (Chapter 9).

Ortega, E., Oñate, E., Idelsohn, S. & Flores, R. (2013). A meshless finite point method for three-dimensional analysis of compressible flow problems involving moving boundaries and adaptivity. *International Journal for Numerical Methods in Fluids, 73(4), 323-343*. (Chapter 6 and Chapter 7).

Ortega, E., Oñate, E., Idelsohn, S. & Flores, R. (2014). Comparative accuracy and performance assessment of the finite point method in compressible flow problems. *Computers & Fluids, 89, 53-65*. (Chapter 5).

Ortega, E., Oñate, E., Idelsohn, S. & Flores, R. (2014). Application of the finite point method to high-Reynolds number compressible flow problems. In press, *International Journal for Numerical Methods in Fluids. DOI: 10.1002/fld.3871*. (Chapter 8).

Ortega, E., Oñate, E., Idelsohn, S. & Flores, R. An a-posteriori error estimate for the finite point method with applications to adaptive compressible flos problems. Submitted to *International Journal of Computational Methods* (2013). (Chapter 6).

# BIBLIOGRAPHY

Afshar, M. H. & Lashckarbolok, M. (2008). Collocated discrete least-squares (CDLS) meshless method: Error estimate and adaptive refinement. *International Journal for Numerical Methods in Fluids, 56(10)*, 1909-1928.

Aftosmis, M. J. & Berger, M. (2002). Multilevel error estimation and adaptive h-refinement for cartesian meshes with embedded boundaries. *AIAA Paper 2002-0863*.

Akkiraju, N., Edelsbrunner, H., Facello, M., Fu, P., Mücke, E. & Varela, C. (1995). Alpha shapes: definition and software. *In Proceedings of the 1st International Computational Geometry Software Workshop*, 63-66.

Ambrosi, D. (1995). Approximation of shallow water equations by Roe's Riemann solver. *International Journal for Numerical Methods in Fluids, 20(2)*, 157-168.

Anandhanarayanan, K. (2010). Development of three-dimensional grid-free solver and its applications to multi-body aerospace vehicles. *Defence Science Journal, 60*(6), 653-662.

Anandhanarayanan, K. & Nagarathinam, M. (2005). *Parallelisation of a Gridfree Kinetic Upwind Solver. AIAA paper 2005-4628*. Paper presented at the AIAA 17th Computational Fluid Dynamics Conference, Toronto CA.

Angulo, A., Pérez Pozo, L. & Perazzo, F. (2009). A posteriori error estimator and an adaptive technique in meshless finite point method. *Engineering Analysis with Boundary Elements, 33*(11), 1322-1338.

Annunziato, A. (2007). The tsunami assessment modelling system by the Joint Research Centre. *Science of Tsunami Hazards, 26*, 70-92.

Annunziato, A. & Best, C. (2005). The tsunami event. Analyses and models. *Institute for the Protection and Security of the Citizen. Joint Research Centre, European Commission*.

Ata, R. & Soulaïmani, A. (2005). A stabilized SPH method for inviscid shallow water flows. *International Journal for Numerical Methods in Fluids, 47(2)*, 139-159.

Atluri, S. N., Kim, H. G. & Cho, J. Y. (1999). A critical assessment of the truly meshless local Petrov-Galerkin (MLPG), and local boundary integral equation (LBIE) methods. *Computational Mechanics, 24*(5), 348-372.

Atluri, S. N. & Zhu, T. (1998). A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics, 22*(2), 117-127.

Babuška, I. & Melenk, J. M. (1996). The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering, 139*(1), 289-314.

Babuška, I. & Melenk, J. M. (1997). The Partition of Unity Method. *International Journal for Numerical Methods in Engineering, 40*(4), 727-758.

Babuska, I. & Miller, A. (1984). Post-processing approach in the finite element method. Part 3: a-posteriori error estimates and adaptive mesh selection. *International Journal for Numerical Methods in Engineering, 20*(12), 2311-2324.

Baker, T. J. (1997). Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements in Analysis and Design, 25(3)*, 243-273.

Baldwin, B. S. & Lomax, H. (1978). Thin layer approximation and algebraic model for separated turbulent flows. *AIAA Paper 88-0520*.

Barba, L. A., Leonard, A. & Allen, C. B. (2005). Advances in viscous vortex methods—meshless spatial adaption based on radial basis function interpolation. *International Journal for Numerical Methods in Fluids, 47*(5), 387-421.

Barth, T. J. (1991). A 3D upwind euler solver for unstructured meshes. *AiAA Paper 1991-1548*.

Batina, J. T. (1989). Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aeroelastic analysis. *AIAA Paper 89-1189*.

Batina, J. T. (1992). A gridless Euler/Navier-Stokes solution algorithm for complex two-dimensional applications. *NASA-TM-107631*.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. & Krysl, P. (1996). Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering, 139(1)*, 3-47.

Belytschko T., Gu, L. & Yu, L. L. (1994). Fracture and crack grow by element-free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering, 2*(3a), 519-534.

Berger, M. & Aftosmis, M. J. (2005). Analysis of slope limiters on irregular grids. *AIAA Paper 2005-0490*.

Berger, M. J. & Jameson, A. (1985). Automatic adaptive grid refinement for the Euler equations. *AIAA journal, 23(4), 561-568*.

Bernard, E. N., Mofjeld, H. O., Titov, V., Synolakis, C. E. & González, F. I. (2006). Tsunami: scientific frontiers, mitigation, forecasting and policy implications. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 364(1485)*, 1989-2007.

Blom, F. J. (2000). Considerations on the spring analogy. *International Journal for Numerical Methods in Fluids, 32(6)*, 647-668.

Blottner, F. G. & Lopez, A. R. (1998). Determination of solution accuracy of numerical schemes as part of code and calculation verification. *SANDIA report SAND98-2222*.

Boroomand, B., Tabatabaei, A. A. & Oñate, E. (2005). Simple modifications for stabilization of the finite point method. *International Journal for Numerical Methods in Engineering, 63(3)*, 351-379.

Calvo, N. (2005). Generación de mallas tridimensionales por métodos duales. *PhD thesis, Universidad Nacional del Litoral*.

Coles, D. E. & Hirst, E. A. (Eds.). (1969). *Computation of Turbulent Boundary Layers (vol. 2)*. Proceedings of the *AFSOR-IFP Stanford Conference.*

Cook, P. H., McDonald, M. A. & Firmin, M. C. P. (1979). Airfoil RAE 2822 - Pressure distributions, and boundary layers and wake measurements. *AGARD Advisory Report No. 138 Experimental data base for computer program assessment.*

Cueto-Felgueroso, L., Colominas, I., Fe, J., Navarrina, F. & Casteleiro, M. (2006). High order finite volume schemes on unstructured grids using moving least squares reconstruction. Application to shallow water dynamics. *International Journal for Numerical Methods in Engineering, 65(3)*, 295-311.

Cueto-Felgueroso, L., Colominas, I., Nogueira, X., Navarrina, F. & Casteleiro, M. (2007). Finite volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering, 196*(45), 4712-4736.

Cueto, E., Sukumar, N., Calvo, B., Martínez, M. A., Cegonino, J. & Doblaré, M. (2003). Overview and recent advances in natural neighbour Galerkin methods. *Archives of Computational Methods in Engineering, 10*(4), 307-384.

Chang, S. & Haworth, D. C. (1997). Adaptive grid refinement using cell-level and global imbalances. *International Journal for Numerical Methods in Fluids, 24(4)*, 375-392.

Chen, J. S., Wu, C. T., Yoon, S. & You, Y. (2001). A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering, 50*(2), 435-466.

Cheng, M. & Liu, G. R. (1999). Finite Point Method for analysis of fluid flow. In C. M. Wang, K. H. Lee & K. K. Ang (Eds.), *4th Asia-Pacific Conference on Computational Mechanics*.

Cheng, R. & Cheng, Y. (2008). Error estimates for the finite point method. *Applied Numerical Mathematics, 58*(6), 884-898.

Chiu, E., Wang, Q. & Jameson, A. (2011). A conservative meshless scheme: general order formulation and application to Euler equations. *AIAA 2011-651 49th Aerospace Sciences Meeting, Orlando, Florida*.

Chorin, A. J. (1973). Numerical study of slightly viscous flow. *Journal of Fluid Mechanics, 57*(4), 785-796.

Chorin, A. J. & Bernard, P. S. (1973). Discretization of a vortex sheet, with an example of roll-up. *Journal of Computational Physics, 13*(3), 423-439.

Demmel, J. W. (1997). *Applied numerical linear algebra*: Society for Industrial and Applied Mathematics (SIAM).

Deshpande, S. M., Anandhanarayanan, K., Praveen, C. & Ramesh, V. (2002). Theory and application of 3-D LSKUM based on entropy variables. *International Journal for Numerical Methods in Fluids, 40*(1-2), 47-62.

Deshpande, S. M., Kulkarni, P. S. & Ghosh, A. K. (1998). New developments in kinetic schemes. *Computers and Mathematics with Applications, 35*(1), 75-93.

Duarte, C. A. (1995). A review of some meshless methods to solve partial differential equations. *Texas Institute for Computational and Applied Mathematics (TICAM), Report 95-05*.

Duarte, C. A. & Oden, J. T. (1996a). An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering, 139(1)*, 237-262.

Duarte, C. A. & Oden, J. T. (1996b). Hp clouds-an hp meshless method. *Numerical Methods for Partial Differential Equations, 12*(6), 673-706.

Dubal, M. R. (1994). Domain decomposition and local refinement for multiquadric approximations I: second-order equations in one-dimension. *Journal of Applied Science and Computation, 1*(1), 146-171.

Dubuc, L., Cantariti, F., Woodgate, B., Gribben, K., Badcock, K. J. & Richards, B. E. (2000). A grid deformation technique for unsteady flow computations. *International Journal for Numerical Methods in Fluids, 32(3)*, 285-311.

Duchon, J. (1976). Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *Rev. Francaise Autom. Inf. Rech. Oper., 10*(12), 5-12.

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In S. a. K. Zeller (Ed.), *Constructive Theory of Functions of Several Variables. Lecture Notes in Mathematics* (Vol. 571, pp. 85-100). Berlin: Springer-Verlag

Duchon, J. (1978). Sur l'erreur d'interpolation des fonctions de plusieurs variables par les Dm-splines. *Rev. Francaise Autom. Inf. Rech. Oper., 12*(4), 325-334.

Dyn, N. (1987). Interpolation of scattered data by radial functions. In L. Schumaker & F. Utreras (Eds.), *Topics in Multivariate Approximation* (pp. 47-61). New York: Academic Press

Dyn, N. (1989). Interpolation and approximation by radial and related functions. In L. Schumaker & J. Ward (Eds.), *Approximation Theory* (Vol. 6, pp. 211-234). New York: Academic Press.

Eigeldinger, J. J. (1988). Chopin: Pianist and Teacher: as seen by his pupils. In R. Howat (Ed.): Cambridge University Press.

Fasshauer, G. E. (1997). Solving partial differential equations by collocation with radial basis functions. In A. LeMehaute, C. Rabut & L. L.

Schumaker (Eds.), *Surface Fitting and Multiresolution Methods* (Vol. , pp. 131-138). Nashville TN: Vanderbilt University Press.

Fasshauer, G. E. (2006). *Meshfree methods. Handbook of theoretical and computational nanotechnology* (Vol. 2): American Scientific Publishers.

Fischer, T. (1996). *A contribution to adaptive numerical solution of compressible flow problems.* PhD Thesis, Universitat Politècnica de Catalunya.

Fischer, T., Onate, E. & Idelsohn, S. (1995). *A meshless technique for computer analysis of high speed flows*. Paper presented at the AGARD Symposium on Progress and Challenges in CFD Methods and Algorithms, Seville.

Flores, R., Ortega, E. & Oñate, E. (2011). PUMI: un código explícito no estructurado para resolver las ecuaciones de Euler. *Revista Internacional de Métodos Numéricos en la Ingeniería, 27(2)*, 129-146.

Franke, R. (1982). Scattered data interpolation: test of some methods. *Mathematics of Computation, 38*(157), 181-200.

Franke, R. & Neilson, G. (1980). Smooth interpolation of large sets of scattered data. *International Journal for Numerical Methods in Engineering, 15*(11), 1691-1704.

Fries, T. & Matthies, H. (2004). Classification and overview of meshfree methods. *Department of Mathematics and Computer Science, Technical University of Braunschweig. Inf. 2003-3*.

Garcia-Navarro, P. & Vazquez-Cendon, M. E. (2000). On numerical treatment of the source terms in the shallow water equations. *Computer and Fluids, 29(8)*, 951-979.

Garimella, R. V. & Shepard, M. S. (2000). Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering, 49(1)*, 193-218.

Gavete, L., Cuesta, J. L. & Ruiz, A. (2002). A numerical comparison of two different approximations of the error in a meshless method. *European Journal of Mechanics & Solids, 21(6)*, 1037-1054.

Ghosh, A. K. & Deshpande, S. M. (1995). Least squares kinetic upwind method for inviscid compressible flows. AIAA paper, 1735, 783-784.

GiD (2013). GiD. The personal pre and post processor (CIMNE). Retrieved August 10, 2013, from http://www.gidhome.com

Gingold, R. A. & Monaghan, J. J. (1977). Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society, 181*, 375-389.

Glaister, P. (1988). Approximate Riemann solutions of the shallow water equations. *Journal of Hydraulic Research, 26*, 293-306.

Glaister, P. (1993). Flux difference splitting for open-channel flows. *International Journal for Numerical Methods in Fluids, 16(7)*, 629-654.

Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik, 89*(3), 271-306.

Granville, P. S. (1987). Baldwin-Lomax factors for turbulent boundary layers in pressure gradients. *AIAA Journal, 25*(12), 1624-1627.

Gu, Y. T. (2005). Meshfree methods and their comparisons. *International Journal of Computational Methods, 2*(4), 477-515.

Günther, F. C. (1998). *A meshfree formulation for the numerical solution of the viscous compressible Navier-Stokes equations.* PhD Thesis, Northwestern University, Evaston.

Han, W. & Meng, X. (2001). Error analysis of the reproducing kernel particle method. *Computer Methods in Applied Mechanics and Engineering, 190(46)*, 6157-6181.

Harder, R. L. & Desmarais, R. N. (1972). Interpolation using surface splines. *Journal of Aircraft, 9*(2), 189-191.

Hardy, R. L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research, 76*(8), 1905-1915.

Harish, G. & Pavanakumar, M. (2006). *Store Separation Dynamics using Grid-free Euler Solver. AIAA paper 2006-3650.* Paper presented at the AIAA 24th Applied Aerodynamics Conference, San Francisco, CA.

Harten, A., Engquist, B., Osher, S. & Chakravarthy, S. R. (1987). Uniformly high-order accurate essential non-oscillatory schemes III. *Journal of Computational Physics, 71(2)*, 231-303.

Harten, A. & Hyman, J. M. (1983). Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics, 50*(2), 235-269.

Hay, A. & Visonneau, M. (2006). Error estimation using the error transport equation for finite-volume methods and arbitrary meshes.

*International Journal of Computational Fluid Dynamics, 20*(7), 463-479.

Hirsch, C. (1990). *Numerical computation of internal and external flows* (Vol. 2): John Wiley & Sons.

Hoffmann, K. A. & Chiang, S. T. (2000). *Computational fluid dynamics. Volume I*: Engineering Education System. Wichita, USA.

Hon, Y. C., Cheung, K. F., Mao, X. Z. & Kansa, E. J. (1999). Multiquadric solution for shallow water equations. *Journal of Hydraulic Engineering, 125*(5), 524-533.

Hon, Y. C., Lu, M. W., Xue, W. M. & Zhu, Y. M. (1997). Multiquadric method for the numerical solution of a biphasic mixture model. *Applied Mathematics and Computation, 88*(2), 153-175.

Hon, Y. C. & Mao, X. Z. (1998). An efficient numerical scheme for Burgers' equation. *Applied Mathematics and Computation, 95*(1), 37-50.

Idelsohn, S., Calvo N. & Oñate, E. (2003). Polyhedrization of an arbitrary 3D point set. *Computer Methods in Applied Mechanics and Engineering, 192(22)*, 2649-2667.

Idelsohn, S., Mier-Torrecilla, M. & Oñate, E. (2009). Multi-fluid flows with the particle finite element method. *Computer Methods in Applied Mechanics and Engineering, 198*(33), 2750-2767.

Idelsohn, S. & Oñate, E. (2006). To mesh or not to mesh. That is the question... *Computer Methods in Applied Mechanics and Engineering, 195*(37), 4681-4696.

Idelsohn, S., Storti, M. & Oñate, E. (2001). Lagrangian formulations to solve free surface incompressible inviscid fluid flows. *Computer Methods in Applied Mechanics and Engineering, 191*(6), 583-593.

Idelsohn, S., Marti, J., Limache, A. & Oñate, E. (2008). Unified Lagrangian formulation for elastic solids and incompressible fluids: application to fluid-structure interaction problems via the PFEM. *Computer Methods in Applied Mechanics and Engineering, 197*(19), 1762-1776.

Idelsohn, S., Oñate, E., Calvo, N. & Del Pin, F. (2003). The meshless finite element method. *International Journal for Numerical Methods in Engineering, 58*(6), 893-912.

Idelsohn, S., Oñate, E. & del Pin, F. D. (2004). The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering, 61*(7), 964-989.

Idelsohn, S., Oñate, E., del Pin, F. D. & Calvo, N. (2006). Fluid–structure interaction using the particle finite element method. *Computer Methods in Applied Mechanics and Engineering, 195*(17), 2100-2123.

Ito, Y. & Nakahashi, K. (2002). Unstructured mesh generation for viscous flow computations. *In Proceedings of the 11th International Meshing Roundtable (IMR). Ithaca,* 367-377.

Jahangirian, A. & Hashemi, M. Y. (2012). Adaptive cartesian grid with mesh-less zones for compressible flow calculations. *Computer and Fluids, 54*, 10-17.

Jameson, A. (1991). Time-dependent calculations using multigrid with applications to unsteady flows past airfoils and wings. *AIAA Paper 91-1596*.

Jameson, A. (1993). Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flows. *AIAA Paper 93-3359*.

Jameson, A. (1995). Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics, 4*(3-4), 171-218.

Jameson, A. & Baker, T. J. (1987). Improvements to the aircraft Euler method. *AIAA Paper 87-0452*.

Jameson, A., Schmidt, W. & Turkel, E. (1981). Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA Paper 81-1259*

Jin, X., Li, G. & Aluru, N. R. (2004). Positivity conditions in meshless collocation methods. *Computer Methods in Applied Mechanics and Engineering, 193*(12), 1171-1202.

Kanok-Nukulchai, W. & Nanakorn, P. (2007). Finite element modelling of tsunami propagation on the coast of Thailand. *Final project report submitted to the Office of the Education Council, Thailand*.

Kansa, E. J. (1986). *Application of Hardy's multiquadric interpolation to hydrodynamics*, Multiconference on Computer Simulation.

Kansa, E. J. (1990a). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - I surface approximations and partial derivative estimates. *Computer and Mathematics with applications, 18*(8-9), 127-145.

Kansa, E. J. (1990b). Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computer and Mathematics with applications, 19*(8-9), 147-161.

Kansa, E. J. & Carlson, R. E. (1992). Improved accuracy of multiquadric interpolation using variable shape parameters. *Computers and Mathematics with Applications, 24*(12), 99-120.

Kansa, E. J. & Hon, Y. C. (2000). Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Computers and Mathematics with Applications, 39*(7), 123-137.

Karbacher, S., Seeger, S. & Häusler, G. (2001). Refining triangle meshes by non-linear subdivision. *Chair of Optics, University of Erlangen, Germany*.

Karim, M. F., Roy, G. D., Ismail, A. I. M. & Meah, M. A. (2009). Numerical simulation of Indonesian tsunami 2004 along southern Thailand: a nested grid model. *International Journal of Mathematical, Physical and Engineering Sciences, 3*, 1.

Katz, A. (2009). Meshless methods for computational fluid dynamics. *PhD thesis, Standford University*.

Katz, A. & Jameson, A. (2009a). A comparison of various meshless schemes within a unified algorithm. *AIAA Paper 2009-596*.

Katz, A. & Jameson, A. (2009b). Multicloud: multigrid convergence with a meshless operator. *Journal of Computational Physics, 228(14)*, 5237-5250.

Kennet, D. J., Timme, S., Angulo, J. & Badcock, K. J. (2012). An implicit meshless method for application in computational fluid dynamics. *International Journal for Numerical Methods in Fluids, 71(8), 1007-1028*.

Kermani, M. J. & Plett, E. G. (2001). Modified Entropy Correction Formula for the Roe Scheme. *AIAA Paper 2001-0083*.

Kirshman, D. J. & Liu, F. (2004). Flutter prediction by a cartesian mesh Euler method with small perturbation gridless boundary conditions. *AIAA Paper 2004-0584*.

Kovačević, I., Poredoš, A. & Šarler, B. (2003). Solving the stefan problem with the radial basis function collocation mehod. *Numerical Heat Transfer: Part B: Fundamentals, 44*(6), 575-599.

Krige, D. G. (1951). *A statistical approach to some mine valuations and allied problems at the Witwatersrand.* Master Thesis, Witwatersrand University.

Krishnamurthy, T. (2005). *Comparison of response surface construction methods for derivative estimation using moving least squares, kriging and radial basis functions.* Paper presented at the 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, *AIAA 2005-1821* (pp. 18-21).

Lancaster, P. (1979). Moving weighted least-squares methods. *In Polynomial and Spline Approximation (Sahney B. Ed.), 49*, 103-120.

Lancaster, P. & Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Mathematics of Computation 37(155)*, 141-158.

Landon, R. H. (1982). NACA 0012. Oscillatory and transient pitching. *AGARD Report Nº 702 Compendium of unsteady aerodynamic measurements*.

Laney, C. B. (1988). *Computational gasdynamics*: Cambridge University Press.

Larese, A., Rossi, R., Oñate, E. & Idelsohn, S. (2008). Validation of the particle finite element method (PFEM) for simulation of free surface flows. *Engineering Computations, 25*(4), 385-425.

Larsson, E. & Fornberg, B. (2003). A numerical study of some radial basis function based solution methods for elliptic PDEs. *Computers and Mathematics with Applications, 46*(5), 891-902.

Lee, C. K., Liu, X. & Fan, S. C. (2003). Local multiquadric approximation for solving boundary value problems. *Computational Mechanics, 30*(5-6), 396-409.

Lee, C. K. & Zhou, C. E. (2004). On error estimation and adaptive refinement for element free Galerking method. Part I: stress recovery and a posteriori error estimation. *Computer and Structures, 82(4)*, 413-428.

Li, J. & Hon, Y. C. (2004). Domain decomposition for radial basis meshless methods. *Numerical Methods for Partial Differential Equations, 20*(3), 450-462.

Li, Q. & Lee, K. M. (2006). An adaptive meshless method for magnetic field computation. *IEEE Transactions on Magnetics, 42*(8), 1996-2003.

Li, S. & Liu, W. K. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews, 55(1)*, 1-34.

Lin, H. & Atluri, S. N. (2001). The meshless local Petrov-Galerkin (MLPG) method for solving incompressible Navier-Stokes equations. *Computer Modeling in Engineering and Sciences, 2*(2), 117-142.

Liszka, T., J., Duarte, C., A., M., & Tworzydlo, W., W. (1996). hp-Meshless cloud method. *Computer Methods in Applied Mechanics and Engineering 139(1)*, 263-288.

Liszka, T. & Orkisz, J. (1980). The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures, 11*(1), 83-95.

Liu, G. R. (2003). *Mesh Free Methods: moving beyond the Finite Element Method*: CRC Presss.

Liu, G. R. & Gu, Y. T. (2001a). A local point interpolation method for stress analysis on two-dimensional solids. *Structural Engineering and Mechanics, 11*(2), 221-336.

Liu, G. R. & Gu, Y. T. (2001b). A local radial point interpolation method (LR-PIM) for free vibration analyses of 2-D solids. *Journal of Sound and vibration, 246*(1), 29-46.

Liu, G. R. & Gu, Y. T. (2005). *An introduction to meshfree methods and their programming*: Springer.

Liu, J. L. & Su, S. J. (1996). *A Potentially Gridless Solution Method for the Compressible Euler-Navier-Stokes Equations. AIAA Paper 96-0526* presented at the AIAA 34th Aerospace Sciences Meeting and Exhibit, Reno, NV.

Liu, M. B. & Liu, G. R. (2006). Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics, 56*(1), 19-36.

Liu, M. B. & Liu, G. R. (2010). Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering, 17*(1), 25-76.

Liu, W. K., Jun, S., Sihling, D. T., Chen, Y. & Hao, W. (1997). Multiresolution reproducing kernel particle method for computational fluid dynamics. *International Journal for Numerical Methods in Fluids 24*(12), 1391-1415.

Liu, W. K., Jun, S. & Zhang, Y. F. (1995). Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids, 20*(8-9), 1081-1106.

Liu, W. K., Li, S. & Belytschko, T. (1997). Moving least-square reproducing kernel methods (I) methodology and convergence. *Computer Methods in Applied Mechanics and Engineering, 143*(1), 113-154.

Liu, X., Osher, S. & Chen, T. F. (1994). Weighted essentially non-oscillatory schemes. *Journal of Computational Physics, 115(1)*, 200-212.

Liu, X., Qin, N., Xia, H. (2006). Fast dynamic grid deformation based on Delaunay graph mapping. *Journal of Computational Physics, 211(2)*, 405-423.

Livne, E. (2003). Future of airplane aeroelasticiy. *Journal of Aircraft, 40*(6), 1066-1092.

Löhner, R. (2001). *Applied Computational Fluid Dynamics Techniques. An introduction based on Finite Element Methods*: John Wiley & Sons Ltd., Chichester.

Löhner, R. & Oñate, E. (1998). An advancing front point generation technique. *Communications in Numerical Methods in Engineering, 14*(12), 1097-1108.

Löhner, R., Sacco, C., Oñate, E. & Idelsohn, S. (2002). A Finite Point Method for compressible flow. *International Journal for Numerical Methods in Engineering, 53(8)*, 1765-1779.

Loving, D. & Estabrooks, B. (1951). Transonic-wing investigation in the Langley 8-foot high-speed tunnel at high subsonic Mach numbers and at a Mach number of 1.2. Analysis of pressure distribution of wing-fuselage configuration having a wing of 45º sweepback, aspect ratio 4, taper ratio 0.6, and NACA 65A006 airfoil section. *National Advisory Committee for Aeronautics. Research Memorandum NACA RM L51F07*.

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal, 82*(12), 1013-1024.

Lynch, D. R. & Gray, W. G. (1978). Analytic solutions for computer flow model testing. *ASCE Journal of the Hydraulics division, 104(10)*, 1409-1428.

Lyra, P. R. & Morgan, K. (2002). A review and comparative study of upwind biased schemes for compressible flow computation. Part III: Multidimensional extension on unstructured grids. *Archives in Computational Methods in Engineering, 9*(3), 207-256.

Madych, W. R. & Nelson, S. A. (1988). Multivariate interpolation and conditionally positive definite functions. *Approximation Theory and Applications, 4*(4), 77-89.

Madych, W. R. & Nelson, S. A. (1990). Multivariate interpolation and conditionally positive definite functions. II. *Mathematics of Computation, 54*(189), 211-230.

Mai-Duy, N. & Tran-Cong, T. (2001). Numerical solution of Navier-Stokes equations using multiquadric radial basis function networks. *International Journal for Numerical Methods in Fluids, 37*(1), 65-86.

Maisuradze, G. G. & Thompson, D. L. (2003). Interpolating moving least-squares methods for fitting potential energy surfaces: illustrative approaches and applications. *The Journal of Physical Chemistry, 107*(37), 7118-7124.

Matheron, G. (1969). *Le krigeage universel*: *École nationale supérieure des mines de Paris.*

Matheron, G. (1970). *La théorie des variables régionalisées et ses applications* (Vol. 5): Cah. Centre Morphol. Math.

Mavriplis, D. J. (1990). Algebraic turbulence modeling for unstructured and adaptive meshes. *ICASE Report 90-30*.

Mavriplis, D. J., Jameson, A. & Martinelli, L. (1989). Multigrid solution of the Navier-Stokes equations on triangular meshes. *ICASE Report 89-11*.

Mavriplis, D. J. & Yang, Z. (2006). Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. *Journal of Computational Physics, 213*(2), 557-573.

McLain, D. H. (1974). Drawing Contours from Arbitrary Data Points. *The Computer Journal, 17*(4), 318-324.

McLain, D. H. (1976). Two Dimensional Interpolation from Random Data. *The Computer Journal, 19*(2), 178-181.

Meinguet, J. (1979). Multivariate interpolation at arbitrary points made simple. *Zeitschrift für angewandte Mathematik und Physik ZAMP, 30(2)*, 292-304.

Meinguet, J. (1984). Surface Spline Interpolation: Basic Theory and Computational Aspects. In S. P. Singh (Ed.), *In Approximation Theory and Spline Functions,* 124-142.

Melson, N. D., Sanetrik, M. D. & Atkins, H. L. (1993). Time-accurate Navier-Stokes calculations with multigrid acceleration. *6th Copper Mountain conference on multigrid methods*, 423-439.

Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation, 2*(1), 11-22.

Monaghan, J. J. (1982). Why particle methods work. *SIAM Journal on Scientific and Statistical Computing, 3*(4), 422-433.

Munikrishna, N. & Balakrishnan, N. (2011). Turbulent flow computations on a hybrid cartesian point distribution using meshless solver LSFD-U. *Computer and Fluids, 40(1)*, 118-138.

Muzaferija, S. & Gosman, D. (1996). Finite-Volume CFD procedure and adaptive error control strategy for grids of arbitrary topology. *Journal of Computational Physics, 138(2)*, 766-787.

NASA (2013). NPARC Alliance CFD verification and validation web site. Retrieved May 14, 2013, from http://www.grc.nasa.gov/WWW/wind/valid/fpturb/fpturb.html

Nayroles, B., Touzot, G. & Villon, P. (1992). Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics, 10*(5), 307-318.

NGDC (2006). National Geophysical Data Center, 2-minute gridded global relief data (ETOPO 2v2). Retrieved June 16, 2009, from http://www.ngdc.noaa.gov/mgg/fliers/01mgg04.html

Nguyen, V. P., Rabczuk, T., Bordas, S. & Duflot, M. (2008). Meshless methods: a review and computer implementation aspects. *Mathematics and Computers in Simulation, 79*(3), 763-813.

Niedoba, P., Čermák, L. & Jícha, M. (2013). Meshfree methods for computational fluid dynamics. *EPJ Web of Conferences* (Vol. 45, pp. 6): EDP Sciences.

NORSAR (2006). Norwegian Seismic Array, Tsunami studies. Retrieved June 16, 2009, from http://www.norsar.no/seismology/Earthquakes/TsunamiStudies/

Oberkampf, W. L. & Trucano, T. G. (2002). Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences, 38(3)*, 209-272.

Oden, J. T. & Prudhomme, S. (1999). New approaches to error estimation and adaptivity for the Stokes and Oseen equations. *International Journal for Numerical Methods in Fluids, 31(1)*, 3-15.

Oden, J. T., Wu, W. & Ainsworth, M. (1993). An a-posteriori error estimate for finite element approximations of the Navier-Stokes equations.

*Computational Methods in Applied Mechanics and Engineering, 111(1)*, 185-202.

Oñate, E. (1996). On the stabilization of numerical solution of convective transport and fluid flow problems. Research report nº 81: International Center for Numerical Methods in Engineering.

Oñate, E. (1998). Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems. *Computer Methods in Applied Mechanics and Engineering, 151*(1), 233-265.

Oñate, E., Arteaga, J., García, J. & Flores, R. (2006). Error estimation and mesh adaptivity in incompressible viscous flows using a residual power approach. *Computer Methods in Applied Mechanics and Engineering, 195(4)*, 339-362.

Oñate, E. & Idelsohn, S. (1998). A mesh-free finite point method for advective-diffusive transport and fluid flow problems. *Computational Mechanics, 24*(4-5), 283-292.

Oñate, E., Idelsohn, S. & Zienkiewicz, O. C. (1995). Finite Point methods in computational mechanics. CIMNE publication Nº 74: International Center for Numerical Methods in Engineering.

Oñate, E., Idelsohn, S., Zienkiewicz, O. C. & Fisher, T. (1995). A Finite Point Method for analysis of fluid flow problems. *Proceedings of the 9th Int. Conference on Finite Elements Methods in Fluids, Venize, Italy*, 15-21.

Oñate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L. (1996a). A Finite Point Method for analysis of fluid mechanics problems. Applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering, 39(2)*, 3839-3866.

Oñate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L. & Sacco, C. (1996b). A stabilized Finite Point Method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering, 139(1)*, 315-346.

Oñate, E., Idelsohn, S., Celigueta, M. & Rossi, R. (2008). Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows. *Computer Methods in Applied Mechanics and Engineering, 197*(19), 1777-1800.

Oñate, E., Idelsohn, S., del Pin, F. & Aubry, R. (2004). The particle finite element method-an overview. *International Journal of Computational Methods, 1*(2), 267-307.

Oñate, E., Sacco, C. & Idelsohn, S. (2000). A finite point method for incompressible flow problems. *Computing and visualization in science, 3*(1-2), 67-75.

Ortega, E., Oñate, E. & Idelsohn, S. (2007). An improved finite point method for three-dimensional potential flows. *Computational Mechanics, 40(6)*, 949-963.

Ortega, E. & Sacco, C. (2003). Solución de las ecuaciones de flujo compresible mediante el método de puntos finitos. In M. C. Rosales, M. B. Cortínez & D. V. Bambill (Eds.), *Mecánica Computacional* (Vol. 23): Asociación de Mecánica Computacional Argentina.

Perazzo, F., Löhner, R. & Perez-Pozo, L. (2007). Adaptive methodology for meshless finite point method. *Advances in Engineering Software, 39(3)*, 156-166.

Perrone, N. & Kao, R. (1975). A general finite difference method for arbitrary meshes. *Computers and Structures, 5*(1), 45-57.

Pirzadeh, S. (1994). Unstructured viscous grid generation by the advancing-layers method. *AIAA Journal, 32*(8), 1735-1737.

Praveen, C. & Deshpande, S. M. (2003). A New Grid-Free Method for Conservation Laws. In S. W. Armfield, P. Morgan & K. Srinivas (Eds.), *Computational Fluid Dynamics 2002* (pp. 128-133): Springer Berlin Heidelberg.

Praveen, C. & Deshpande, S. M. (2007). Kinetic meshless method for compressible flows. *International Journal for Numerical Methods in Fluids, 55*(11), 1059-1089.

Praveen, C. A. (2004). A positive meshless method for hyperbolic equations. *Report 2004-FM-16, ARDB Centre of excellence for aerospace CFD, Indian Institute of Science*.

Pulliam, T. H. & Barton, J. T. (1985). *Euler computations of AGARD working group 07 airfoil test cases.* Paper presented at the AIAA 23rd Aerospace Summer Meeting, Reno NE.

Qiu, J. & Shu, C. (2003). Hermite WENO schemes and their application as limiters for the Runge-Kutta discontinuous Galerking method: one dimensional case. *Journal of Computational Physics, 193(1)*, 115-135.

Rabczuk, T. & Belytschko, T. (2005). Adaptivity for structured meshfree particle methods in 2D and 3D. *International Journal for Numerical Methods in Engineering, 63(11)*, 1559-1582.

Rizzi, A. & Viviand, H. E. (1981). Numerical methods for the computation of inviscid transonic flows with shock waves *Notes on numerical fluid mechanics* (Vol. 3). Braunschweig/Wiesbaden: Wieweg and Sohn.

Roache, P. J. (1994). Perspective: a method for uniform reporting of grid refinement studies. *Transactions American Society of Mechanical Engineering, Journal of Fluids Engineering, 116*, 405-413.

Roache, P. J. (1997). Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics, 29(1)*, 123-160.

Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics, 43(2)*, 357-372.

Rolston, S. (2001). High Reynolds number tools and techniques for civil aircraft design - an overview of the HiReTT programme. *Air & Space Europe, 3*(3/4).

Rostand, P. (1989). Algebraic turbulence models for the computation of two-dimensional high-speed flows using unstructured grids. *International Journal for Numerical Methods in Fluids, 9(9)*, 1121-1143.

Roy, C. J. (2003). Grid convergence error analysis for mixed-order numerical schemes. *AIAA Journal, 41(4)*, 595-604.

Roy, C. J. (2005). Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics, 205(1)*, 131-156.

Roy, C. J. (2009). Strategies for driving mesh adaption in CFD. *AIAA Paper 2009-1302*.

Sacco, C. (2002). Desarrollo del método de puntos finitos en mecánica de fluidos. *PhD Thesis, Universitat Politècnica de Catalunya.*

Salas, M. D. (2006). Some observations on grid convergence. *Computers and Fluids, 35(7),* 688-692.

Šarler, B., Perko, J. & Chen, C. S. (2004). Radial basis function collocation method solution of natural convection in porous media. *International Journal of Numerical Methods for Heat and Fluid Flow, 14*(2), 187-212.

Schaback, R. (1995). Creating surfaces from scattered data using radial basis functions. *Mathematical methods for curves and surfaces*, 477-496.

Schmitt, V. & Charpin, F. (1979). Pressure distributions on the ONERA-M6-Wing at transonic Mach numbers *Experimental data base for computer program assessment B1-1. AGARD AR-138*.

Serna, S. & Marquina, A. (2004). Power ENO methods: a fifth-order accurate Weighted Power ENO method. *Journal of Computational Physics, 194(2)*, 632-658.

Shepard, D. (1968). A two dimensional interpolation function for irregularly spaced data. *In Proceedings of the 1968 23rd ACM National Conference (pp. 517-524).*

Shih, T. & Qin, Y. (2007). A-posteriori method for estimating and correcting grid-induced errors in CFD solutions. Part 1: Theory and method. *AIAA Paper 2007-100*.

Shu, C., Ding, H., Chen, H. Q. & Wang, T. G. (2005). An upwind local RBF-DQ method for simulation of inviscid compressible flows. *Computer Methods in Applied Mechanics and Engineering, 194*(18), 2001-2017.

Shu, C., Ding, H. & Yeo, K. S. (2003). Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering, 192*(7), 941-95

Sibson, R. (1980). A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society, 87*(1), 151-155.

Singh, M. K., Ramesh, V. & Balakrishnan, N. (2010). Convergence acceleration of mesh-less Euler solver. *12th Annual CFD Symposium, Bangalore*.

Sod, G. A. (1978). A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics, 27*(1), 1-31.

Spalart, P. R. & Allmaras, S. R. (1992). A one-equation turbulence model for aerodynamic flows. *AIAA-92-0439, AIAA 30th Aerospace Sciences Meeting, Orlando, Florida*.

Sridar, D. & Balakrishnan, N. (2003). An upwind finite difference scheme for meshless solvers. *Journal of Computational Physics, 189(1)*, 1-29.

Srinarayana, N., Gonzalez, L. F., Whitney, E. J. & Srinivas, K. (2006). Aerodynamic Optimisation using a Robust Evolutionary Algorithm and Grid-free Flowsolver. AIAA paper 2006-52, AIAA 44th Aerospace Sciences Meeting and Exhibit, Reno, NV.

Stewartson, K. (1964). The theory of laminar boundary layers in compressible fluids: *Eds. G. Temple and I. James. Mathematical Monographs. Oxford University Press.*

Strouboulis, T., Babuška, I. & Copps, K. (2000). The design and analysis of the generalized finite element method. *Computer Methods in Applied Mechanics and Engineering, 171*(1), 43-69.

Sukumar, N., Moran, B. & Belytschko, T. (1998). The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering, 43*(5), 839-887.

Sulsky, D., Zhou, S. J. & Schreyer, H. L. (1995). Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications, 87*(1), 236-252.

Sunday, D. (2001). Intersection of rays and triangles (3D). Retrieved 8 August, 2013, from http://geomalgorithms.com/a06- intersect-2.html

Tarwater, A. E. (1985). Parameter study of Hardy's multiquadric method for scattered data interpolation. *UCRL-53670. Lawrence Livermore National Laboratory, CA, USA*.

Taylor, R. L., Zienkiewicz, O. C., Oñate, E. & Idelsohn, S. (1995). Moving Least Square approximations for solution of differential equations *CIMNE publication Nº 74* (pp. 31): International Center for Numerical Methods in Engineering.

Thacker, B. H., Doebling, S. W., Hemez, F. M., Anderson, M. C., Pepin, J. E. & Rodriguez, E. A. (2004). Concepts of model verification and validation. *Los Alamos National Laboratory, LA-14167-MS*

Thomas, P. D. & Lombard, C. K. (1979). Geometric conservation law and its application to flow computations on moving grids. *AIAA journal, 17*(10), 1030-1037.

Tongsuk, P. & Kanok-Nukulchai, W. (2004). Further investigation of Element-Free-Galerkin method using moving kriging interpolation. *International Journal of Computational Methods, 1*(2), 345-365

Turkel, E. (1988). Improving the accuracy of central difference schemes. *ICASE Report 88-53*, 586-591.

USGS (2005). U. S. Geological Survey (USGS), Sound Waves montly newsletter (March 2005). Retrieved 30 August, 2013, from http://soundwaves.usgs.gov/2005/03/

van Albada, G. D., van Leer, B. & Roberts Jr, W. W. (1982). A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics, 108*, 76-84.

Van Driest, E. R. (1952). Investigation of laminar boundary layer in compressible fluids using the Crocco method. *National Advisory Committee for Aeronautics. Technical Note NACA TN 2597*.

Van Leer, B. (1977). Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics, 23*(3), 276-299.

Van Leer, B. (1979). Towards the ultimate conservative difference scheme V. A second-order sequel to Godunov's method. *Journal of Computational Physics, 32*(1), 101-136.

Venkatakrishnan, V. & Mavriplis, D. J. (1995). Implicit method for the computation of unsteady flows on unstructured grids. *ICASE Report 95-60*.

Wang, J., Chen, H. Q. & Periaux, J. (2009). A study of gridless method with dynamic clouds of points for solving unsteady CFD problems in aerodynamics. *International Journal for Numerical Methods in Fluids, 64*(1), 98-118.

Wang, J. W. & Liu, R. X. (2005). Combined finite volume-finite element method for shallow water equations. *Computer and Fluids, 34(10)*, 1199-1222.

Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics, 4*(1), 389-398.

Wendland, H. (1999). Meshless Galerkin method using radial basis functions. *Mathematics of Computation, 68*(228), 1521-1533.

White, F. M. (1991). *Viscous fluid flow*: McGraw-Hill Inc.

Wieghardt, K. & Tillman, W. (1951). On the turbulent friction layer for rising pressure. *National Advisory Committee for Aeronautics. Technical Memorandum NACA TM-1314*.

Wilcox, D. C. (1994). *Turbulence modeling for CFD*: DCW Industries, Inc. La Cañada, California.

Williams, B. R. (1971). An exact test case for the plane potential flow about two adjacent lifting airfoils. *Aeronautics Research Council, Reports and Memoranda No. 3717*.

Wright, M. (2000). HiReTT Task 1.1.2: High speed wind tunnel tests on the full span model N44. *Test Report E 9011 TR019, European Transonic Windtunnel*.

Wu, C., Huang, G. & Zheng, Y. (1999). Theoretical solution of dam-break shock wave. *Journal of Hydraulic Engineering, 125(11)*, 1210-1215.

Wu, X. H., Tao, W. Q., Shen, S. P. & Zhu, X. W. (2010). A stabilized MLPG method for steady state incompressible fluid flow simulation. *Journal of Computational Physics, 229*(22), 8564-8577.

Wu, Z. (1995). Compactly supported positive definite radial functions. *Advances in Computational Mathematics, 4*(1), 283-292.

Xiaozhong, J., Gang, L. & Aluru, N., R. . (2004). Positivity conditions in meshless collocation methods. *Computer Methods in Applied Mechanics and Engineering 193(12)*, 1171-1202.

Xinrong, S., Satoru, Y. & Kazuhiro, N. (2013). Analysis of a meshless solver for high Reynolds number flow. *International Journal for Numerical Methods in Fluids*, 72(5), 505-527.

Yagawa, G. & Furukawa, T. (2000). Recent developments of free mesh method. *International Journal for Numerical Methods in Engineering, 47*(8), 1419-1443.

Yagawa, G. & Yamada, T. (1996). Free mesh method: a new meshless finite element method. *Computational Mechanics, 18*(5), 383-386.

Yang, Z. & Mavriplis, D. J. (2005). Unstructured dynamic meshes with high-order time integration schemes for the unsteady Navier-Stokes equations. *AIAA Paper 2005-1222*.

Zhang, X. D., Trépanier, J. Y. & Camarero, R. (2000). A posteriori error estimation for finite-volume solutions of hyperbolic conservation laws. *Computational Methods in Applied Mechanics and Engineering, 185(1)*, 1-19.

Zhang, Z. & Naga, A. (2005). A new finite element gradient recovery method: superconvergence property. *SIAM Journal of Scientific Computing, 26*(4), 1192-1213

Zienkiewicz, O. C. & Taylor, R. L. (2000). *The finite element method, Volume 1*: Butterworth-Heinemann.

Zienkiewicz, O. C., Taylor, R. L. & Nithiarasu, P. (2005). *The Finite Element Method for Fluid Dynamics* (6th ed.): Elsevier.

Zuppa, C. (2003). Good quality point sets and error estimates for moving least square approximations. *Applied Numerical Mathematics, 47*(3), 575-585.