

Bericht des Instituts für Aerodynamik und Strömungstechnik
Report of the Institute of Aerodynamics and Flow Technology

IB 124-2008/1

**Discontinuous Galerkin Methods for inviscid low Mach
number flows**

A. Nigro

Herausgeber:

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, Germany, 38108 Braunschweig

ISSN 1614-7790

Stufe der Zugänglichkeit: 1
Braunschweig, im Januar 2008

Institutsdirektor:

Prof. Dr.-Ing. habil. C.-C. Rossow

Verfasser:

Dr.-Ing. A. Nigro

Abteilung: C²A²S²E

Abteilungsleiter:

Prof. Dr.-Ing. N. Kroll

Der Bericht enthält:

97 Seiten

32 Bilder

9 Tabellen

56 Literaturstellen

Abstract

In this work we present two preconditioning techniques for inviscid low Mach number flows. The space discretization used is a high-order Discontinuous Galerkin finite element method. The time discretizations analyzed are explicit and implicit schemes. The convective physical flux is replaced by a flux difference splitting scheme. Computations were performed on triangular and quadrangular grids to analyze the influence of the spatial discretization. For the preconditioning of the explicit Euler equations we propose to apply the fully preconditioning approach: a formulation that modifies both the instationary term of the governing equations and the dissipative term of the numerical flux function. For the preconditioning of the implicit Euler equations we propose to apply the flux preconditioning approach: a formulation that modifies only the dissipative term of the numerical flux function. Both these formulations permit to overcome the stiffness of the governing equations and the loss of accuracy of the solution that arise when the Mach number tends to zero. Finally, we present a splitting technique, a proper manipulation of the flow variables that permits to minimize the cancellation error that occurs as an accumulation effect of round-off errors as the Mach number tends to zero.

Note

This is the Ph.D. thesis of Eng. Alessandra Nigro. This research has been performed at the Department of Mechanical Engineering of the University of Calabria and during a 9 month research visit of Alessandra Nigro at the Institute of Aerodynamics and Flow Technology at the DLR, Braunschweig. This work has been supervised by Ing. Carmine De Bartolo (University of Calabria), by Prof. Francesco Bassi (University of Bergamo), and by Dr. Ralf Hartmann (DLR Braunschweig).

This work has been submitted to the Department of Mechanical Engineering of the University of Calabria on Nov. 23, 2007, in partial fulfilment of the requirements for the Ph.D. of Mechanical Engineering.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Background	4
1.2.1	High-Order Methods	4
1.2.2	Discontinuous Galerkin Methods	5
1.2.3	Preconditioning Techniques	5
1.3	Outline of Thesis	6
2	Physical model	7
2.1	Conservation Laws	8
2.1.1	Description	8
2.1.2	The compressible Euler equations	8
2.2	The preconditioned compressible Euler equations	10
2.3	Non-dimensionalization	11
3	Discontinuous Galerkin Formulation	13
3.1	Discretization of the Euler equations	14
3.2	Numerical Flux function	15
3.2.1	Roe-average numerical flux	15
3.2.2	Low Mach behaviour	16
3.3	Boundary Treatment	21
3.3.1	Boundary Conditions	21
3.3.2	Geometry Representation: Curved Boundaries	22
4	Explicit scheme:	
	Fully Preconditioning technique for the Euler equations	24
4.1	Preconditioning matrix	25
4.2	Time discretization scheme	26
4.3	Local time stepping	27

4.4	Preconditioned Roe's Numerical Flux	27
4.5	Boundary conditions	28
4.5.1	Preconditioned far-field	28
4.5.2	Slip wall	29
4.6	Results	30
4.6.1	Convergence	33
4.6.2	Accuracy	38
5	Implicit Scheme:	
	Flux Preconditioning technique for the Euler equations	54
5.1	Overview of the Implicit scheme	55
5.2	Overview of Flux preconditioning technique	55
5.3	Time discretization scheme	56
5.4	Jacobian of the preconditioned numerical flux function	57
5.5	Boundary Conditions	57
5.6	Results	58
5.6.1	Convergence	58
5.6.2	Accuracy	62
6	Cancellation problem	69
6.1	Behaviour of governing equations at low Mach numbers	70
6.2	Round-off error and relative treatment of the variables	70
6.3	Results	71
6.3.1	Convergence	72
6.3.2	Accuracy	78
A	Primitive variables	89
B	Roe Numerical Flux	91

1 Introduction

The use of numerical methods to simulate complex physical phenomena has become an invaluable part of engineering and modern science. Among them, Computational Fluid Dynamics (CFD) has matured significantly in past decades, in terms of time and computational resources, even if large aerodynamic simulations of aerospace vehicles are still very expensive.

Almost all discretizations of the compressible Euler and/or Navier-Stokes equations currently used in aerodynamic applications are based on Finite Volume Methods (FVM). The evolution of these methods, including the incorporation of the upwinding mechanisms [1, 2, 3, 4, 5] and advances in solution techniques for viscous flows [6, 7, 8, 9], have made the simulation of complex problem possible. However, these standard algorithms remain at best second-order accurate, meaning that the error decrease as $O(h^2)$ as the grid spacing h tends to zero. Moreover, while these methods are used heavily in aerospace design today, the time required to obtain reliably accurate solutions has hindered the realization of the full potential of CFD in the design process. In fact, it is unclear if the accuracy of current second-order finite volume methods is sufficient for engineering purposes. The results of the two AIAA Drag Prediction Workshop (DPW) [10, 11] suggest that the CFD technology currently in use may not produce sufficiently accurate results on meshes with typical grid sizes that are used in an industrial environment.

This problem could be alleviated by the development of high-order CFD algorithms. Traditional finite volume methods rely on extended stencils to achieve high-order accuracy. This may lead to difficulties in achieving stable iterative algorithms and higher-order algorithms on unstructured meshes. In fact, higher order ENO and WENO reconstruction methods on unstructured meshes are not used for industrial applications. To overcome these problems, significant research effort has been devoted to the development of new high-order accurate methods, among them the Discontinuous Galerkin Finite Element Methods (DGFEM, DG methods for short). It can be observed that the DG methods have experienced a resurgence of interest in multi-various disciplines of numerical mathematics including compressible flows and aerodynamics among many others, and that these methods are now applied to problems which traditionally were solved using the Finite Volume Methods [12]. The reason for this trend can be identified in several advantages of the discontinuous Galerkin methods over finite volume methods. In fact, DG methods allow higher order computations on unstructured meshes, they allow considerable flexibility in

the choice of the mesh design including hanging nodes, non-matching grids and *hp*-refinement and they can easily be parallelized. This potential of DG methods has attracted the attention to explore the benefits of this approach in the low Mach number limit.

Algorithms "traditionally" used, like Finite Volume Methods, encounter some difficulties to solve low Mach number flows. One of the difficulties consists in a degradation of the computational performance: convergence slows down and/or fails and often the numerical accuracy decreases as the Mach number tends to zero [13, 14]. To overcome the lack of numerical accuracy a very high mesh resolution is required with ulterior convergence rate reductions. Therefore the low Mach number flow simulations performed with numerical schemes currently in use are very expensive.

Another difficulty in computing flows at very low Mach number arises from the increasing of cancellation errors when the Mach number tends to zero. At low Mach number the changes of thermodynamic flow variables become small with respect to their stagnation values. The accuracy of the numerical solution is lost, as the round-off errors in computing the thermodynamic gradients accumulate and result in large cancellation error.

These are problems well-known and widely examined in literature and many different strategies have been studied trying to overcome these difficulties. However, most of these publications are devoted to the analysis and application of these strategies to FVM, while, to the knowledge of the author, only few are based on the DG method solving flows in the low Mach number regime [15, 16].

Motivated by the potential of the DG method and by the necessity to obtain converged and accurate solutions for low Mach number flows, this thesis gives at first a clearer understanding of the performance of the DG method in the low Mach number limit and secondly contributes to the development of a higher-order CFD algorithm which is able to overcome the lacks of convergence and accuracy exhibit by the compressible flow algorithms in the incompressible limit.

Remark: We note that the numerical methods described in this thesis have been implemented and tested based on two separate Discontinuous Galerkin flow solvers: In the flow solver `MIGALE` of Prof. Bassi [17] based on non-parametric elements and in the DG flow solver `PADGE` [18] which is based on the `deal.II` library [19, 20] implementing parametric elements. The results which have been obtained on quadrangular elements based on these two codes have been cross-checked and found to be very similar which significantly increased the confidence in the results obtained and the numerical effects encountered. However, most of the numerical results which are finally printed in this thesis have been produced based on the DG flow solver of Prof. Bassi as it – in contrast to `deal.II` – allows the use of both, quadrangular and triangular meshes.

1.1 Motivation

There is an ever-increasing need of computing compressible low Mach number flows or locally incompressible flows. Typical examples of compressible low speed flows can be found in natural convection flows in gas or liquid phase, subsonic combustion in heat engines or burners, heat transfer in heat exchangers and others. Additionally, many problems contain some regions with very low Mach numbers while other regions are decidedly compressible. Thus compressibility can not be neglected and numerical procedures for the solution of these problems must be capable of simultaneously treating both high and low speed flow regimes. Some examples include rocket motor flows in which the Mach number is zero at the closed end and supersonic at the divergent nozzle exit, high speed flows with large embedded recirculation zones, multi-phase flows in which the Mach number changes drastically

through the phase boundaries, and flow over a wing at high angle of attack [21].

However, it is very difficult or impossible to solve low speed flows with a conventional compressible algorithm. Algorithms used for compressible flows, usually denoted as density-based, as the continuity equation rules the time evolutions of density, suffers from slow convergence and lack of accuracy to solve low Mach number flows in which the density is almost constant. To overcome these problems, different approaches such as pressure correction, pseudo-compressibility methods and different preconditioning techniques have been developed. Up to now, most of the research effort devoted to the efficient computation of low speed flows has been concentrated on low-order methods like the FVM.

In this work we investigate the behaviour in the low Mach number limit and the effect of the preconditioning technique using a high-order method like the DG method.

1.2 Background

1.2.1 High-Order Methods

With the expression high-order method we refer to the order of accuracy of the method. The order of accuracy of a numerical method is the exponent of the first term in the Taylor series expansion of the difference between the analytical solution and the approximate solution. For efficiency of the method we refer to the time necessary to achieve a prescribed accuracy level. Since there is always a tradeoff between accuracy and computing time of any numerical method, the most desirable methods are the efficient ones, i.e. those for which the running time increases slowly as the acceptable error decreases. Higher-order methods are of interest because they have this potential: the potential to provide significant reductions in the time required to obtain accurate solutions.

The first high-order accurate numerical methods were spectral methods [22, 23], where the solution of a differential equation is approximated using a high-order expansion. By choosing the expansion functions properly, an arbitrarily high-order accuracy can be achieved. However, because of the global nature of the expansion functions, spectral methods are limited to very simple domains with simple boundary conditions. Motivated by the prospect of obtaining the rapid convergence rates of spectral methods with the greater geometric versatility provided by finite element methods, in the early 1980s the researchers introduced the so called p -type finite element method. In the p -type finite element method, the grid spacing, h , is fixed, while the polynomial degree, p , is increased to decrease the error. In 1981, Babuska *et al.* [24] applied this method to elasticity problems. They concluded that based on degrees of freedom, the rate of convergence of the p -type method cannot be slower than that of the h -type and that, in cases with singularities present at vertices, the convergence rate of the p -type is twice as fast. Starting from these first studies significant research effort has been aimed at developing high-order accurate methods, among others the Discontinuous Galerkin Finite Element Method.

1.2.2 Discontinuous Galerkin Methods

The Discontinuous Galerkin Method was originally developed by Reed and Hill [25] in 1973 for neutron transport problems and first analyzed by Le Saint and Raviart in 1975. Since that time, development of the method has proceeded rapidly. Cockburn *et al.* present an extensive overview of the history of DG methods in [26].

The DG methods combine ideas from the finite element and the finite volume methods: the accuracy obtained by high-order polynomial approximations within elements and the physics of wave propagation expressed by Riemann problems. In contrast to standard finite element methods the DG methods are based on discontinuous basis functions. Given a mesh of the computational domain, the DG methods approximate the solution within each element by a function from a low-dimensional vector space of functions, e.g. as a linear combination of basis functions like polynomials. For a pair of adjacent mesh elements, the approximate solution computed in the interior of the elements does not have to agree on the element interface.

The DG method has several desirable properties that have made it popular:

- It can sharply capture solution discontinuities relative to a computational mesh.
- It simplifies adaption since inter-element continuity is neither required for mesh refinement and coarsening, nor for p -adaptivity.
- It conserves the appropriate physical quantities (e.g. mass, momentum and energy) on an element-by-element basis.
- It can handle problems in complex geometries to high order.
- Regardless of order, it has a simple communication pattern to elements sharing a common face that simplifies parallel computation.

On the other hand, with a discontinuous basis, the DG methods include more unknowns for a given order of accuracy on a given mesh than traditional finite element or finite volume methods, which may lead to some inefficiency.

1.2.3 Preconditioning Techniques

It is well-known that convergence and accuracy slow down solving low Mach number flows [14]. The reason of the bad convergence is the large disparity between acoustic and convective wave speeds that causes the ill-conditioning (stiffness problem) of the governing equations. The decreasing accuracy results from a lack of artificial dissipation for small Mach number, as observed in related work carried out by Turkel *et al.* [27]. In particular, for the upwind schemes Guillard and Viozat [28] show that the dissipative terms of the numerical flux become negligible with respect to the centered ones as the Mach number tends to zero. In order to accurately and efficiently solve nearly incompressible inviscid flows these difficulties must be overcome, which is the goal of the preconditioning techniques.

The preconditioning technique artificially modifies the acoustic wave speeds by modifying the time derivative terms of the governing equations. These pseudo-acoustic wave speeds can be chosen of the same order as the local velocity to drastically reduce the condition number. As a consequence, the convergence of the time-stepping or iterative solution process is significantly enhanced. Furthermore, the accuracy of the discretization can be improved by preconditioning if the numerical dissipation term is modified accordingly. In particular, the preconditioned governing equations preserve the accuracy for nearly incompressible flows. This is obtained by balancing appropriately the artificial viscosity term with the inviscid flux term [27, 28, 29]. Some of the most recognized local preconditioners for inviscid and viscous flows were proposed by Choi and Merkle [30], Turkel [31, 32], Lee and van Leer [33] and Weiss and Smith [34], respectively. As the preconditioning destroys the time accuracy, it is applicable to steady-state simulations, only. To overcome this limitation, dual time-stepping techniques may be employed [34]. Numerous studies have been carried out on these topics in the past; a complete review of the preconditioning techniques is given in [31, 32, 33, 35].

1.3 Outline of Thesis

This thesis deals with a high-order accurate discontinuous finite element method for the numerical solution of the compressible Euler equations on triangular and quadrangular unstructured grids in the low Mach number limit.

The outline of this work present Thesis is as follows:

- In Chapter 2 we present the physical model obtained premultiplying the time derivative of the governing Euler equations by the preconditioned matrix.
- In Chapter 3 we describe the Discontinuous Galerkin discretization of the preconditioned Euler equations.
- In Chapter 4 we present the fully preconditioning technique employed for explicit schemes. This technique modifies both the instationary terms of the governing equations and the dissipative terms of the numerical flux.
- In Chapter 5 we present the flux preconditioning technique employed for implicit schemes. This technique modifies the dissipative terms of the numerical flux, only, while the instationary terms of the governing equations remain unchanged.
- In Chapter 6 we present a splitting technique to minimize cancellation errors that occur when computing flows at very low Mach number.

Finally, we give a conclusion at the end of this work.

2 Physical model

This chapter is devoted to the introduction of the physical model used to investigate the behaviour of the DG method in the low Mach number limit.

After a short description of the well-known conservation laws, we focus our attention on the compressible Euler equations that describe the pure convection of flow quantities in an inviscid fluid. The Euler equations are commonly written in conservative variables but, since the density is a constant in the incompressible limit, a different choice of set of variables could be more appropriate; furthermore, for low Mach number, the system of Euler equations become stiff when marching in time. This chapter is then devoted to explain the reasons that induce to apply the preconditioning techniques and the mathematical meaning of preconditioned physical model.

2.1 Conservation Laws

2.1.1 Description

A conservation law is a mathematical statement concerning the conservation of one or more quantities. In physical applications an example might be the statement that mass, momentum and energy should be conserved with respect to a specified control volume. The conservation of these flow quantities means that their total variation inside an arbitrary volume can be expressed as the net effect of the amount of the quantity being transported across the boundary, any internal forces and sources, and external forces acting on the volume.

In two space dimensions a system of conservation laws is given by

$$\frac{\partial}{\partial t} \mathbf{w}(\mathbf{x}, t) + \frac{\partial}{\partial x_1} \mathbf{f}(\mathbf{w}(\mathbf{x}, t)) + \frac{\partial}{\partial x_2} \mathbf{g}(\mathbf{w}(\mathbf{x}, t)) = 0, \quad (2.1)$$

where $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$, $\mathbf{w} : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^m$ is an m -dimensional vector of conserved quantities, or state variables and $\mathbf{f}(\mathbf{w})$ and $\mathbf{g}(\mathbf{w}) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ are the convective flux functions for the system of conservation laws. The equation (2.1) must be augmented by some initial conditions and also possibly boundary conditions on a bounded spatial domain. The simplest problem is the pure initial value problem, or Cauchy problem, in which (2.1) holds for $-\infty < \mathbf{x} < \infty$ and $t \geq 0$.

In this case we must specify initial conditions only

$$\mathbf{w}(\mathbf{x}, 0) = \mathbf{w}_0(\mathbf{x}) \quad -\infty < \mathbf{x} < \infty$$

For brevity, partial derivatives will be denoted by subscripts in the following, and the flux functions will be grouped in the flux vector $\mathbf{F} = \mathbf{F}(\mathbf{f}, \mathbf{g})$. Using this notation, equation (2.1) is given by

$$\mathbf{w}_t + \nabla \cdot \mathbf{F} = 0.$$

The conservation laws considered in this work are the compressible Euler equations that constitute an hyperbolic system of partial differential equations. We assume that the System (2.1) is hyperbolic if any real combination of $\alpha \mathbf{f}' + \beta \mathbf{g}'$ of the flux Jacobians is diagonalizable with real eigenvalues.

2.1.2 The compressible Euler equations

The compressible Euler equations describe the pure convection of flow quantities in an inviscid fluid. In two space dimension they are given in strong and conservative form as follows

$$\mathbf{w}_t + \nabla \cdot \mathbf{F} = 0, \quad (2.2)$$

where \mathbf{w} is the state vector of conservative variables given by

$$\mathbf{w} = \left(\rho, \quad \rho u, \quad \rho v, \quad \rho E \right)^T,$$

and $\mathbf{F} = \mathbf{F}(\mathbf{f}, \mathbf{g})$ is the inviscid flux vector, with the inviscid flux functions

$$\mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u H \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v H \end{pmatrix}.$$

Here, ρ is the fluid density, u and v are velocity components, p is the pressure and E is the total internal energy per unit mass. The total enthalpy per unit mass, H , is given by $H = E + p/\rho$, and, assuming the fluid satisfies the equation of state of a perfect gas, the pressure is given by $p = (\gamma - 1) \rho [E - (u^2 + v^2)/2]$, where γ is the ratio of specific heats of the fluid, given by $\gamma = c_p/c_v$.

The conservative variables are commonly used in compressible flow computations. Nevertheless, as the Mach number tends to zero, the density becomes constant and cannot be used as a variable in the incompressible limit. Thereby, the set of conservative variables cannot be employed for these flow conditions. A different set of variables may be used, but it must be carefully chosen as each set possesses unique properties that influence the performance of the numerical scheme[36].

The advantage of the primitive variables over other sets is that they are more appropriate for incompressible flow. Since the density is a constant in the incompressible limit, the choice of pressure p as a fundamental variable proved to be more adequate. Moreover, for viscous flows, temperature gradients have to be computed for the thermal diffusion terms, so it is also more convenient to work with temperature.

For these reasons, the choice of primitive variables is a "natural" choice. Furthermore, in [36] it has been shown that the conservative incompressible formulation is well defined only for entropy variables and primitive variables including pressure. It is also shown that these two sets of variables possess the most attributes for practical problem solving, with the primitive variables being more accurate than the entropy variables for low speed and incompressible flows computations. For these reasons the primitive variables are often preferred to perform low Mach number flow computations and they can be used to derive numerical schemes that are suitable for compressible and incompressible flows.

For these reasons, the starting point to obtain compressible Euler equations that are more adequate for the low Mach number limit, is to transform Equations (2.2) in terms of primitive variables, obtaining

$$\mathbf{\Gamma} \mathbf{q}_t + \nabla \cdot \mathbf{F} = 0.$$

Here, \mathbf{q} is the set of primitive variables given by

$$\mathbf{q} = (p, u, v, T)^T,$$

and $\mathbf{\Gamma}$ is the transformation matrix from conservative to primitive variables

$$\mathbf{\Gamma} = \frac{\partial \mathbf{w}}{\partial \mathbf{q}} = \begin{pmatrix} \rho_p & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & \rho_T u \\ \rho_p v & 0 & \rho & \rho_T v \\ \rho_p H - 1 & \rho u & \rho v & \rho_T H + \rho c_p \end{pmatrix},$$

where ρ_p and ρ_T are given by

$$\rho_p = \left. \frac{\partial \rho}{\partial p} \right|_{T=\text{const.}}, \quad \rho_T = \left. \frac{\partial \rho}{\partial T} \right|_{p=\text{const.}}.$$

For an ideal gas we have

$$\rho_p = 1/T, \quad \rho_T = -\rho/T.$$

2.2 The preconditioned compressible Euler equations

For low Mach numbers the system of Equations (2.2) becomes stiff. The stiffness of the governing equations, when marching in time, is determined by the condition number. The condition number of a general matrix \mathbf{A} based on the L_p norm is

$$K_p(\mathbf{A}) = \|\mathbf{A}\|_p \|\mathbf{A}^{-1}\|_p.$$

For the 2D Euler equations, the respective matrices to be considered are linear combinations of the flux Jacobians which have a complete set of eigenvalues and eigenvectors. Thereby we have

$$K_2(\mathbf{A}) = \frac{|\lambda|_{\max}}{|\lambda|_{\min}},$$

where $|\lambda|_{\max}$ and $|\lambda|_{\min}$ are largest and smallest absolute wave speeds. The wave speeds of the 2D Euler equations are $(u_n, u_n, u_n + c, u_n - c)$ where $u_n = \mathbf{v} \cdot \mathbf{n}$ is the component of the velocity vector $\mathbf{v} = (u, v)^T$ along the unit normal \mathbf{n} , also called contravariant velocity, and c is the acoustic velocity (speed of sound).

Thereby, the condition number is given by

$$K_2(\mathbf{A}) = \frac{|u_n| + c}{|u_n|}. \quad (2.3)$$

In order to explain why the characteristic condition number determines the stiffness of the system of equations when marching in time, we have to recall the concept of explicit local time-stepping. The local time-stepping must satisfy the CFL condition:

$$\Delta t \leq \frac{h}{|\lambda|_{\max}} \leq \frac{h}{|u_n| + c},$$

where h is some representative mesh width. We see, that the allowable local time step is limited by the fastest moving wave $|u_n| + c$ but we also see that during such a time step the slowest wave moves only over a fraction of the cell width:

$$|\lambda|_{\min} \Delta t \leq \frac{|\lambda|_{\min}}{|\lambda|_{\max}} h \leq \frac{h}{K_2(\mathbf{A})}.$$

Thus a large condition number reduces the efficiency of wave propagation, needed for convergence.

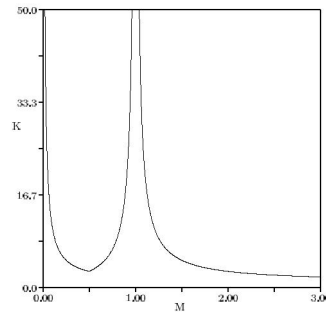


Figure 2.1: Condition Number

Figure (2.1) shows the condition number for different flow regimes, indicating that the stiffness of the original Euler equations increases beyond bound as the Mach number approaches 0 or 1. This implies that, in order to reduce the stiffness, preconditioning should focus on the incompressible and transonic flow regions.

The central idea of preconditioning is the pre-multiplication of the unsteady terms in Equations (2.2) with a matrix, which changes the eigenvalues such that they get closer together. Altering the speed of the $u_n + c$ and $u_n - c$ waves such that they are comparable with the u_n waves, the condition number becomes closer to 1 overcoming the time-stepping restriction and the solution is obtained in much fewer iterations or time steps. Unfortunately the introduction of the preconditioning matrix leads to a formulation that is not consistent in time and thus is applicable to steady flows, only. To overcome this limitation, dual time-stepping techniques may be employed.

The preconditioning technique consists of replacing Γ by another matrix $\bar{\Gamma}$, which we leave unspecified at the moment. The preconditioned Euler equations, still in terms of primitive variables, are then given by

$$\bar{\Gamma} \mathbf{q}_t + \nabla \cdot \mathbf{F} = 0.$$

2.3 Non-dimensionalization

Mathematical problem formulations based on dimensional and non-dimensionalized variables are essentially equivalent and do not per se alter their solutions. Since the Euler equations are homogeneous, it is preferable to solve them in a non-dimensionalized form. These results can then be applied to any problem with the same relative geometric dimensions. Furthermore, non-dimensionalized variables can be used to extract useful information about relative scales in equations and/or boundary conditions that can guide the preconditioned formulation. For these reasons, the following reference values denoted by a subscript r are used: length l_r , density ρ_r , pressure p_r , constant gas R_r .

Reference values for the other quantities are derived from these by functional relationships.

With this choice of non-dimensionalized variables, all the equations given previ-

ously remain unchanged, except that the variables are now understood to be non-dimensionalized.

3 Discontinuous Galerkin Formulation

In this chapter we introduce a high-order accurate discretization of the compressible Euler equations. The formulation given in this chapter is valid for both versions of the conservative system equations considered in this work: the standard and the preconditioned Euler equations, both expressed in terms of primitive variables. The introduction of the preconditioned matrix modifies the characteristics of the system equations. As a consequence the preconditioned system needs different formulations of the numerical flux and of the boundary conditions, which both are discussed in this chapter. In particular, we perform an analysis on the numerical accuracy of the standard and preconditioned Roe's approximate Riemann solver in the low Mach number limit.

3.1 Discretization of the Euler equations

The preconditioned Euler equations in strong and conservative form are given by

$$\bar{\Gamma} \mathbf{q}_t + \nabla \cdot \mathbf{F} = 0, \quad (3.1)$$

where $\bar{\Gamma}$ represent the preconditioned matrix. In absence of preconditioning $\bar{\Gamma}$ reduces to the transformation matrix Γ from conservative to primitive variables and the discretization given below refers to the non-preconditioned Euler equations expressed in terms of primitive variables.

Multiplying Equations (3.1) by a vector-valued test function \mathbf{v} and integrating by parts, we obtain the weak formulation:

$$\int_{\Omega} \mathbf{v}^T \bar{\Gamma} \mathbf{q}_t d\mathbf{x} - \int_{\Omega} \nabla \mathbf{v}^T \cdot \mathbf{F} d\mathbf{x} + \int_{\partial\Omega} \mathbf{v}^T \mathbf{F} \cdot \mathbf{n} ds = 0 \quad \forall \mathbf{v} \in H^1(\Omega)$$

where Ω is the domain, $\partial\Omega$ is its boundary, and \mathbf{n} is the outward pointing unit normal. To discretize in space, we define V_h^p to be the space of discontinuous vector-valued polynomials of degree p on a subdivision T_h of the domain into non-overlapping elements such that $\Omega = \bigcup_{\kappa \in T_h} \kappa$. Thus, the solution and test function space is defined by

$$V_h^p = \{ \mathbf{v} \in L^2(\Omega) : \mathbf{v}|_{\kappa} \in P^p, \forall \kappa \in T_h \},$$

where P^p is the space of polynomial functions of degree at most p . The discrete problem then takes the following form: find $\mathbf{q}_h \in V_h^p$ such that

$$\sum_{\kappa \in T_h} \left\{ \int_{\kappa} \mathbf{v}_h^T \bar{\Gamma} (\mathbf{q}_h)_t d\mathbf{x} - \int_{\kappa} \nabla \mathbf{v}_h^T \cdot \mathbf{F} d\mathbf{x} + \int_{\partial\kappa \setminus \partial\Omega} \mathbf{v}_h^{+T} H_i(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) ds + \int_{\partial\kappa \cap \partial\Omega} \mathbf{v}_h^{+T} H_b(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n}) ds \right\} = 0 \quad (3.2)$$

for all $\mathbf{v}_h \in V_h^p$, where $H_i(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n})$ and $H_b(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n})$ are numerical flux functions defined on interior and boundary faces, respectively.

In this work the Roe-averaged flux is used for the inviscid numerical flux. This flux difference splitting scheme is based on the characteristics of the governing equations. This means that two different formulations of the Roe-averaged flux have to be used:

- The standard Roe numerical flux for the non-preconditioned system.
In this case the transformation matrix $\mathbf{\Gamma}$ doesn't change the characteristics of the system of equations.
- The preconditioned Roe numerical flux for the preconditioned system.
In this case the preconditioned matrix $\bar{\mathbf{\Gamma}}$ is introduced in order to change the characteristics of the governing equations in such a manner that they get closer together. This means that to be compatible with the preconditioned system, the flux difference splitting scheme is adapted according to the new characteristic values.

The boundary conditions are imposed weakly by constructing an exterior boundary state, \mathbf{q}_h^b , which is a function of the interior state and known boundary data. In this work wall and far-field boundary conditions are used. The far-field boundary conditions are based on the characteristic variables. This means that, like for the numerical flux, two different formulations of the far-field boundary conditions have to be used:

- The standard far-field boundary conditions for the non-preconditioned system.
- The preconditioned far-field boundary conditions for the preconditioned system.

3.2 Numerical Flux function

3.2.1 Roe-average numerical flux

The numerical flux functions used on interior and boundary faces in Equation (3.2) could be any kind of upwind numerical flux. Since the fluxes are normal to the element interface and discontinuities are allowed across the interface, a local Riemann problem can be solved based on the interior and the exterior states \mathbf{q}_h^+ and \mathbf{q}_h^- . Therefore, like in Finite Volume methods, various Riemann solver can be used to compute the numerical flux.

In this work we employ the Roe's approximate Riemann solver. This is a linearised solver which means that the governing equations of the Riemann problem have been approximated. Obviously this implies that the solution of the Riemann problem will not be exact anymore, but Roe's approach has shown that despite the approximations good results can be obtained.

In order to clarify how the numerical flux is modified for the preconditioned system to preserve the accuracy of the solution in the low Mach number limit, we begin by

introducing the standard Roe's approximate Riemann solver that is the one used for the non-preconditioned simulations,

$$\mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \frac{1}{2} \left((\mathbf{F}^+ + \mathbf{F}^-) - \tilde{\mathbf{F}}(\mathbf{w}^+, \mathbf{w}^-) \right) \cdot \mathbf{n} \quad (3.3)$$

where \mathbf{F}^+ and \mathbf{F}^- are fluxes computed using the solution vectors \mathbf{w}^+ and \mathbf{w}^- on each (the interior and the exterior) side of the face, $\tilde{\mathbf{F}} = |\tilde{\mathbf{A}}|\Delta\mathbf{w}$, and $\Delta\mathbf{w} = \mathbf{w}^- - \mathbf{w}^+$. The matrix $|\tilde{\mathbf{A}}|$ denotes the so-called Roe matrix or dissipation matrix and is equal to the Jacobian $\partial\mathbf{F}/\partial\mathbf{w}$. The symbol \sim denotes that the matrix is calculated using the so-called Roe-averaged variables (see Appendix B for details). The dissipation matrix $|\tilde{\mathbf{A}}|$ is defined by

$$|\tilde{\mathbf{A}}| = \tilde{\mathbf{T}}|\tilde{\mathbf{\Lambda}}|\tilde{\mathbf{T}}^{-1}$$

where $|\tilde{\mathbf{\Lambda}}| = \text{diag}(\tilde{u}_n, \tilde{u}_n, \tilde{u}_n + \tilde{c}, \tilde{u}_n - \tilde{c})$ is the matrix of eigenvalues evaluated using Roe's averaging, as well as the matrix of left, $\tilde{\mathbf{T}}^{-1}$, and right, $\tilde{\mathbf{T}}$, eigenvectors, remembering that $\tilde{\mathbf{T}}$ is the modal matrix that diagonalizes the matrix $\tilde{\mathbf{A}}$.

For the preconditioned system we change the eigenvalues and the eigenvectors of the system used in the definition of $|\tilde{\mathbf{A}}|$. For this reason we rewrite the second term on the right hand side of Equation (3.3), the so-called dissipation term $|\tilde{\mathbf{A}}|\Delta\mathbf{w}$ of the Roe's numerical flux, using following relation

$$\begin{aligned} \tilde{\mathbf{A}}\Delta\mathbf{w} &= \bar{\Gamma}\bar{\Gamma}^{-1}\frac{\partial\mathbf{F}}{\partial\mathbf{w}}\Delta\mathbf{w} \\ &= \bar{\Gamma}\left(\bar{\Gamma}^{-1}\frac{\partial\mathbf{F}}{\partial\mathbf{q}}\right)\Delta\mathbf{q} \\ &= \bar{\Gamma}\tilde{\mathbf{A}}_{\bar{\Gamma}}\Delta\mathbf{q} \end{aligned}$$

where $\Delta\mathbf{q} = \mathbf{q}^- - \mathbf{q}^+$. Thus, $\Delta\mathbf{w}$ is replaced by $\Delta\mathbf{q}$, and $\tilde{\mathbf{A}}$ by $\bar{\Gamma}\tilde{\mathbf{A}}_{\bar{\Gamma}}$, where $\tilde{\mathbf{A}}_{\bar{\Gamma}}$ is defined in terms of the preconditioned eigenvalues and eigenvectors by

$$\tilde{\mathbf{A}}_{\bar{\Gamma}} = \tilde{\mathbf{T}}_{\bar{\Gamma}}\tilde{\mathbf{\Lambda}}_{\bar{\Gamma}}\tilde{\mathbf{T}}_{\bar{\Gamma}}^{-1}.$$

Here the subscript $\bar{\Gamma}$ denotes that the diagonal matrix of eigenvalues and the modal matrix are derived from the preconditioned system, where $\tilde{\mathbf{\Lambda}}_{\bar{\Gamma}}$ is the diagonal matrix of the preconditioned eigenvalues, and $\tilde{\mathbf{T}}_{\bar{\Gamma}}$ diagonalizes the matrix $(\bar{\Gamma}^{-1}\partial\mathbf{F}/\partial\mathbf{q})$. Similarly we replace $|\tilde{\mathbf{A}}|\Delta\mathbf{w}$ by $\bar{\Gamma}|\tilde{\mathbf{A}}_{\bar{\Gamma}}|\Delta\mathbf{q}$.

3.2.2 Low Mach behaviour

In this section we want to analyse the behaviour of the standard and the preconditioned flux difference splitting approximation in the low Mach number limit.

For clarity, here we report the Roe's numerical flux formulation in the case of the non-preconditioned and the preconditioned scheme:

- Roe non-preconditioned

$$\mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \frac{1}{2} \left((\mathbf{F}^+ + \mathbf{F}^-) - \tilde{\mathbf{F}}(\mathbf{w}^+, \mathbf{w}^-) \right) \cdot \mathbf{n}$$

- Roe preconditioned

$$\mathbf{H}(\mathbf{q}^+, \mathbf{q}^-, \mathbf{n}) = \frac{1}{2} \left((\mathbf{F}^+ + \mathbf{F}^-) - \tilde{\mathbf{F}}_{\bar{\Gamma}}(\mathbf{q}^+, \mathbf{q}^-) \right) \cdot \mathbf{n}$$

As seen in the previous subsection, the dissipation term of the Roe preconditioned $\tilde{\mathbf{F}}_{\bar{\Gamma}}$ is computed as

$$\bar{\Gamma} \tilde{\mathbf{T}}_{\bar{\Gamma}} |\tilde{\Lambda}_{\bar{\Gamma}}| \tilde{\mathbf{T}}_{\bar{\Gamma}}^{-1} \Delta \mathbf{q}. \quad (3.4)$$

For brevity the analysis is performed for the one-dimensional case; the extension to the multi-dimensional case is immediate.

The preconditioned matrix $\bar{\Gamma}$ used in the present work is the local preconditioning matrix of Weiss and Smith [34] written in the one-dimensional case as follows

$$\bar{\Gamma} = \begin{pmatrix} \Theta & 0 & \rho_T \\ \Theta u & \rho & \rho_T u \\ \Theta H - 1 & \rho v & \rho_T H + \rho c_p \end{pmatrix},$$

where Θ is given by

$$\Theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho c_p} \right) = \left(\frac{1}{U_r^2} + \frac{1}{T c_p} \right). \quad (3.5)$$

Here, U_r is a reference velocity and, for an ideal gas, is defined as

$$U_r = \begin{cases} \varepsilon c & \text{if } |\mathbf{v}| < \varepsilon c, \\ |\mathbf{v}| & \text{if } \varepsilon c < |\mathbf{v}| < c, \\ c & \text{if } |\mathbf{v}| > c, \end{cases} \quad (3.6)$$

where c is the acoustic speed and ε is a small number included to prevent singularities at stagnation points.

The resulting eigenvalues of the preconditioned one-dimensional Euler equations are given by

$$\lambda \left(\bar{\Gamma}^{-1} \frac{\partial \mathbf{F}}{\partial \mathbf{q}} \right) = (\lambda_1, \lambda_2, \lambda_3)^T = (u_n, u'_n + c', u'_n - c')^T,$$

where

$$\begin{aligned} u'_n &= u_n(1 - \alpha), \\ c' &= \sqrt{\alpha^2 u_n^2 + U_r^2}, \\ \alpha &= \frac{1 - \beta U_r^2}{2}, \\ \beta &= \left(\rho_p + \frac{\rho_T}{\rho c_p} \right). \end{aligned} \quad (3.7)$$

Then, the matrices $\tilde{\Lambda}_{\bar{\Gamma}}$, $\tilde{\mathbf{T}}_{\bar{\Gamma}}$, and $\tilde{\mathbf{T}}_{\bar{\Gamma}}^{-1}$ used in (3.4) are given by

$$\tilde{\mathbf{\Lambda}}_{\bar{\Gamma}} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix},$$

$$\tilde{\mathbf{T}}_{\bar{\Gamma}} = \begin{pmatrix} \frac{U_r^2}{cpT} & 0 & -\frac{\rho U_r^2}{T} \\ \frac{r}{t} & \frac{\rho U_r^2}{t} & 0 \\ -\frac{s}{t} & -\frac{\rho U_r^2}{t} & 0 \end{pmatrix}, \quad \tilde{\mathbf{T}}_{\bar{\Gamma}}^{-1} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & -\frac{s}{\rho U_r^2} & -\frac{r}{\rho U_r^2} \\ \frac{T}{\rho U_r^2} & \frac{1}{\rho c_p} & \frac{1}{\rho c_p} \end{pmatrix}, \quad (3.8)$$

where

$$\begin{aligned} r &= \lambda_2 - \lambda_1, \\ s &= \lambda_3 - \lambda_1, \\ t &= \lambda_2 - \lambda_3. \end{aligned}$$

For the non-preconditioned system $\tilde{\mathbf{T}}_{\bar{\Gamma}}$ and $\tilde{\mathbf{T}}_{\bar{\Gamma}}^{-1}$ reduce to the left and right eigenvector matrices in primitive variables, respectively. Using (3.8) we obtain the entries of the preconditioned dissipation matrix $\tilde{\mathbf{\Gamma}}\tilde{\mathbf{T}}_{\bar{\Gamma}} | \tilde{\mathbf{\Lambda}}_{\bar{\Gamma}} | \tilde{\mathbf{T}}_{\bar{\Gamma}}^{-1}$ as follows

$$\begin{aligned} d_{\Gamma 11} &= \frac{|\lambda_1|}{c_p T} + c_1, & d_{\Gamma 12} &= \rho c_2, & d_{\Gamma 13} &= -\frac{\rho |\lambda_1|}{T}, \\ d_{\Gamma 21} &= \frac{u_n |\lambda_1|}{c_p T} + c_3, & d_{\Gamma 22} &= \rho c_4, & d_{\Gamma 23} &= -\frac{\rho u_n |\lambda_1|}{T}, \\ d_{\Gamma 31} &= \frac{u_n^2 |\lambda_1|}{c_p T} + c_5, & d_{\Gamma 32} &= \rho c_6, & d_{\Gamma 33} &= -\frac{\rho u_n^2 |\lambda_1|}{T}, \end{aligned} \quad (3.9)$$

where

$$\begin{aligned} c_1 &= \frac{|\lambda_2| r - |\lambda_3| s}{t U_r^2}, \\ c_2 &= \frac{|\lambda_2| - |\lambda_3|}{t}, \\ c_3 &= \frac{|\lambda_2| r (u_n - s) - |\lambda_3| s (u_n - r)}{t U_r^2}, \\ c_4 &= \frac{|\lambda_2| (u_n - r) - |\lambda_3| (u_n - s)}{t}, \\ c_5 &= \frac{|\lambda_2| r (H - u_n s) - |\lambda_3| s (H - u_n r)}{t U_r^2}, \\ c_6 &= \frac{|\lambda_2| (H - u_n s) - |\lambda_3| (H - u_n r)}{t}. \end{aligned}$$

If the absolute values are computed by assuming that $|\lambda_1| = \lambda_1$, $|\lambda_2| = \lambda_2$, $|\lambda_3| = -\lambda_3$ and the quantities r, s, t and the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ are written in terms of u'_n, α, c' and U_r , see Equations (3.6) and (3.7), we obtain

$$\begin{aligned}
c_1 &= \frac{c'^2 - u_n^2(1 - \alpha)\alpha}{c'U_r^2}, \\
c_2 &= \frac{u_n(1 - \alpha)}{c'}, \\
c_3 &= \frac{u_n c'^2(2 - \alpha) - u_n^2 \alpha(1 - \alpha^2)}{c'U_r^2}, \\
c_4 &= \frac{c'^2 + u_n^2(1 - \alpha^2)}{c'}, \\
c_5 &= \frac{c'^2 [H + u_n^2(1 - \alpha)] - u_n^2 \alpha(1 - \alpha)(H + u_n^2 \alpha)}{c'U_r^2}, \\
c_6 &= \frac{u_n c'^2 + u_n(1 - \alpha)(H + u_n^2 \alpha)}{c'}.
\end{aligned} \tag{3.10}$$

At low Mach number we have $\alpha \cong 1/2$, $u'_n \pm c' = 1/2 u_n (1 \pm \sqrt{5})$ and resulting terms are summarized in Table 3.1. In the second column of this table the terms of Equations (3.10) are given in absence of preconditioning ($\alpha = 0$, $u'_n = u_n$, $c' = c = U_r$). We thus obtain the corresponding terms of the Roe non-preconditioned dissipation matrix, $\mathbf{\Gamma} |\mathbf{\Gamma}^{-1} \partial \mathbf{F} / \partial \mathbf{q}|$, where $\mathbf{\Gamma}$ is the transformation matrix between conservative and primitive variables.

	$\alpha \cong 1/2, (M \simeq 0)$	$\alpha = 0$
c_1	$\frac{1}{c'}$	$\frac{1}{c}$
c_2	$\frac{u_n}{2c'}$	$\frac{u_n}{c}$
c_3	$\frac{2c'}{3u_n}$	$\frac{c}{u_n}$
c_4	$\frac{4c'^2 + 3u_n^2}{4c'}$	$\frac{c^2 + u_n^2}{c}$
c_5	$\frac{H + u_n^2/2}{c'}$	$\frac{H + u_n^2}{c}$
c_6	$\frac{u_n [2c'^2 + (H + u_n^2/2)]}{2c'}$	$\frac{u_n (H + c^2)}{c}$

Table 3.1: Terms occurring in the dissipation matrix of the preconditioned Roe scheme at low speed (first column) and in the dissipation matrix of the non-preconditioned Roe scheme (second column).

Table 3.2 presents the order of magnitude of variables occurring in the preconditioned dissipation matrix, $\bar{\mathbf{\Gamma}} |\bar{\mathbf{\Gamma}}^{-1} \partial \mathbf{F} / \partial \mathbf{q}|$, and in the non-preconditioned dissipation matrix $\mathbf{\Gamma} |\mathbf{\Gamma}^{-1} \partial \mathbf{F} / \partial \mathbf{q}|$.

We now substitute the terms of Table 3.1 in Equations (3.9), use the order of magnitude of the variables given in Table 3.2, and simplify by neglecting all terms except of the lowest-order terms in M .

λ_1, u_n	$\lambda_2, \lambda_3, r, s, t, U_r$	H, ρ, T, c
$O(M)$	$O(M)$ if $\alpha \neq 0$ $O(1)$ if $\alpha = 0$	$O(1)$

Table 3.2: Order of magnitude of variables occurring in the dissipation matrices.

For the non-preconditioned Roe scheme at low Mach number ($\alpha = 0$, $M \simeq 0$) we obtain

$$\Gamma |\Gamma^{-1} \partial \mathbf{F} / \partial \mathbf{q}| = \begin{pmatrix} \frac{1}{c} & \rho \frac{u_n}{c} & -\rho \frac{u_n}{T} \\ \frac{u_n}{c} & \rho c & -\rho \frac{u_n^2}{T} \\ \frac{H}{c} & \frac{\rho u_n (H + c^2)}{c} & -\rho \frac{u_n^3}{2T} \end{pmatrix} = \begin{pmatrix} O(1) & O(M) & O(M) \\ O(M) & O(1) & O(M^2) \\ O(1) & O(M) & O(M^3) \end{pmatrix}.$$

For the preconditioned Roe scheme at low Mach number ($\alpha \cong 1/2$, $M \simeq 0$) we obtain

$$\bar{\Gamma} |\bar{\Gamma}^{-1} \partial \mathbf{F} / \partial \mathbf{q}| = \begin{pmatrix} \frac{2}{u_n \sqrt{5}} & \frac{\rho}{\sqrt{5}} & -\rho \frac{u_n}{T} \\ \frac{3}{\sqrt{5}} & \frac{4\rho u_n}{\sqrt{5}} & -\rho \frac{u_n^2}{T} \\ \frac{2H}{u_n \sqrt{5}} & \frac{\rho H}{\sqrt{5}} & -\rho \frac{u_n^3}{2T} \end{pmatrix} = \begin{pmatrix} O(M^{-1}) & O(1) & O(M) \\ O(1) & O(M) & O(M^2) \\ O(M^{-1}) & O(1) & O(M^3) \end{pmatrix}.$$

The order of magnitude of the variation of all thermodynamic variables is $O(M^2)$, whereas the order of magnitude of the variation of the flow velocity is $O(M)$. Thus $\Delta \mathbf{q} = (\Delta p, \Delta u, \Delta T)^T = (O(M^2), O(M), O(M^2))^T$. Now we multiply the preconditioned and standard Roe dissipation matrices by $\Delta \mathbf{q}$, to obtain the corresponding dissipation vectors to compare with the centred terms of the Roe's approximate Riemann solver. For the non-preconditioned Roe scheme in the low Mach number limit we obtain

$$\begin{aligned} \Gamma |\Gamma^{-1} \partial \mathbf{F} / \partial \mathbf{q}| \Delta \mathbf{q} &= \begin{pmatrix} O(1) & O(M) & O(M) \\ O(M) & O(1) & O(M^2) \\ O(1) & O(M) & O(M^3) \end{pmatrix} \begin{pmatrix} O(M^2) \\ O(M) \\ O(M^2) \end{pmatrix} \\ &= \begin{pmatrix} O(M^2) \\ O(M) \\ O(M^2) \end{pmatrix}. \end{aligned} \tag{3.11}$$

Considering that the order of magnitude of the centred terms in the Roe approximation are

$$\frac{1}{2}(\mathbf{F}^+ + \mathbf{F}^-) = \begin{pmatrix} O(M) \\ O(1) + O(M^2) \\ O(M) \end{pmatrix}, \quad (3.12)$$

it is evident that the dissipative terms of the non-preconditioned Roe scheme do not scale properly with the convective terms. In particular the comparison of the centred, Equation (3.12), and dissipative terms, Equation (3.11), of the non-preconditioned Roe scheme in the low Mach number limit shows that there is a lack of numerical dissipation of order of $O(M^{-1})$ in the continuity and energy equations, whereas an excess of numerical viscosity, of order of $O(M^{-1})$, results in the momentum equations.

On the contrary for the preconditioned Roe scheme in the limit of low Mach number we obtain

$$\begin{aligned} \bar{\Gamma} |\bar{\Gamma}^{-1} \partial \mathbf{F} / \partial \mathbf{q}| \Delta \mathbf{q} &= \begin{pmatrix} O(M^{-1}) & O(1) & O(M) \\ O(1) & O(M) & O(M^2) \\ O(M^{-1}) & O(1) & O(M^3) \end{pmatrix} \begin{pmatrix} O(M^2) \\ O(M) \\ O(M^2) \end{pmatrix} \\ &= \begin{pmatrix} O(M) \\ O(M^2) \\ O(M) \end{pmatrix}. \end{aligned} \quad (3.13)$$

Therefore the dissipative terms of the preconditioned Roe scheme in (3.13) scale properly with the convective terms in (3.12). In fact, the preconditioning increases the numerical dissipation term associated to the continuity and energy equations by a factor of $1/M$ [28], but reduces the numerical viscosity associated to the momentum equation by a factor of M .

3.3 Boundary Treatment

3.3.1 Boundary Conditions

Numerical flow simulations consider only a certain part of the physical domain. The truncation of the computational domain creates artificial boundaries, where values of the physical quantities have to be specified. Furthermore, walls which are exposed to the flow represent natural boundaries of the physical domain. The correct imposition of boundary conditions is a crucial part of every flow solver. Furthermore, subsonic flow problems are particular sensitive to the boundary conditions. An inadequate imposition can lead to a significant slow down of convergence to the steady state and the accuracy of the solution may be negatively influenced. In particular, the far-field boundary conditions have proven to be decisive for the accuracy and the convergence of steady flows at low Mach numbers. In fact, if the fast acoustic waves may be reflected at a boundary, very quickly corrupting the interior flow field and

thereby impairing accuracy and convergence, respectively. Various methodologies were developed which are capable of absorbing the outgoing waves at the artificial boundary [37, 38]. A review of different non-reflecting boundary conditions can be found in [39, 40].

In this work we consider the following types of boundary conditions:

- **Far-field**

The numerical imposition of the far-field boundary conditions has to fulfil two basic requirements:

- The truncation of the domain should have no notable effects on the flow solution as compared to the infinite domain.
- Any outgoing disturbances must not be reflected back in to the flowfield.

The far-field boundary conditions are based on characteristic variables. Thus, at inflow the incoming variables that correspond to negative eigenvalues are specified, and the outgoing variables that correspond to positive eigenvalues are extrapolated.

The standard far-field used in this work for the non-preconditioned system follow the approach of Whitfield and Janus [41]. This approach is based on the characteristic form of the one-dimensional Euler equations normal to the boundary.

We note that for the preconditioned system the characteristics of the system are changed although the signs of the eigenvalues remain unchanged. Hence also the far-field boundary conditions must be modified for the preconditioned system.

- **Slip wall**

In the case of inviscid flows, the fluid slips over the surface. Since there is no friction force, the velocity vector must be tangential to the surface. This is equivalent to the condition that there is no flow normal to the surface, i.e.,

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{at slip wall boundaries,}$$

where \mathbf{n} denotes the outward unit normal vector at each integration point.

This boundary condition is not based on the characteristics and thus can be employed without change for both systems of equations, the standard and the preconditioned one.

3.3.2 Geometry Representation: Curved Boundaries

As shown by Bassi and Rebay [42], high-order DG methods are highly sensitive to the geometry representation. Thus it is necessary to build a higher-order representation of the domain boundary. In this work, the geometry is represented using a nodal

Lagrange basis. Thus the mapping between the canonical triangle or square and the element in physical space is given by

$$\mathbf{x} = \sum_j \mathbf{x}^{(j)} \phi_j(\xi), \quad (3.14)$$

where ϕ_j is the j th basis function, ξ is the location in the reference space, and $\mathbf{x}^{(j)}$ is the location of the j th node in physical space. In general, the Jacobian of this mapping is not constant, meaning that triangles and quadrangles with curved edges are allowed. Thus by placing the non-interior, higher-order nodes on the real domain boundary, a higher order geometry representation is achieved.

4 Explicit scheme: Fully Preconditioning technique for the Euler equations

In this section we discuss implementational issues and numerical results concerning the DG method for both the standard and the preconditioned version of the explicit scheme. For the preconditioned explicit scheme, we propose to apply the Fully Preconditioning approach: a formulation that modifies both the instationary terms and the dissipative terms of the numerical convective fluxes. This formulation permits to overcome both the stiffness of the equations and the loss of accuracy of the solution that arises when the Mach number tends to zero. On the other hand, it is not consistent in time and thus applicable to steady flows only.

4.1 Preconditioning matrix

In the explicit schemes the preconditioning matrix $\bar{\Gamma}$ is introduced in the compressible Euler equations in order to overcome the stiffness problem that produces serious time-stepping restrictions. The stiffness problem, that we have already see in Section 2.2, is determined by the condition number and is due to the large discrepancy between the speed of sound and the fluid velocity.

For clarity here we recall the preconditioned Euler equations,

$$\bar{\Gamma} \mathbf{q}_t + \nabla \cdot \mathbf{F} = 0 \quad (4.1)$$

where $\mathbf{F} = (\mathbf{f}, \mathbf{g})$ and

$$\mathbf{q} = \begin{pmatrix} p \\ u \\ v \\ T \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u H \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho v H \end{pmatrix},$$

the transformation matrix from conservative to primitive variables Γ and the preconditioned matrix $\bar{\Gamma}$ [34], respectively given by:

$$\Gamma = \begin{pmatrix} \rho_p & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & \rho_T u \\ \rho_p v & 0 & \rho & \rho_T v \\ \rho_p H - 1 & \rho u & \rho v & \rho_T H + \rho c_p \end{pmatrix}, \quad \bar{\Gamma} = \begin{pmatrix} \theta & 0 & 0 & \rho_T \\ \theta u & \rho & 0 & \rho_T u \\ \theta v & 0 & \rho & \rho_T v \\ \theta H - 1 & \rho u & \rho v & \rho_T H + \rho c_p \end{pmatrix}. \quad (4.2)$$

Comparing the transformation matrix Γ with the preconditioned matrix $\bar{\Gamma}$, we notice that the only difference between these two matrices is due to the substitution of ρ_p by the θ parameter. The term ρ_p that multiplies the pressure time derivative in the continuity equation controls the speed of propagation of acoustic waves in the system. It is interesting to note that, for an ideal gas, $\rho_p = 1/RT = \gamma/c^2$, whereas for constant density flows $\rho_p = 0$, consistent with the notion of infinite pressure wave speeds in an incompressible fluid.

Thus, if we replace this term with one proportional to the inverse of the local velocity squared, we can control the eigenvalues of the system such that they are all of the same order. Keeping this in mind, we now proceed to analyse the choice of the θ parameter given by:

$$\theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho c_p} \right).$$

Here U_r is a reference velocity defined for an ideal gas as follows:

$$U_r = \begin{cases} \varepsilon c & \text{if } |\mathbf{v}| < \varepsilon c, \\ |\mathbf{v}| & \text{if } \varepsilon c < |\mathbf{v}| < c, \\ c & \text{if } |\mathbf{v}| > c, \end{cases} \quad (4.3)$$

where ε is a small number included to prevent singularities at stagnation points. We choose $\varepsilon = O(M)$ to ensure that the convective and acoustic wave speeds are of a

similar magnitude, proportional to the flow speed [27]. The resulting eigenvalues of the preconditioned system (4.1) are given by

$$\lambda = (u_n, u_n, u'_n + c', u'_n - c')^T,$$

where

$$\begin{aligned} u'_n &= u_n(1 - \alpha), \\ c' &= \sqrt{\alpha^2 u_n^2 + U_r^2}, \\ \alpha &= \frac{1 - \beta U_r^2}{2}, \\ \beta &= \left(\rho_p + \frac{\rho_T}{\rho c_p} \right). \end{aligned} \quad (4.4)$$

For an ideal gas $\beta = 1/c^2$.

We note that choosing the U_r parameter like in Equation (4.3), the preconditioned system is able to switch automatically from the preconditioned system to the non-preconditioned one. At low speed we have $U_r \rightarrow 0$, $\alpha \rightarrow 1/2$, and all the eigenvalues are of the same order as u_n .

For sonic speed and above, i.e. $|\mathbf{v}| > c$ we have $U_r = c$, $\alpha = 0$, $u'_n = u_n$, $c' = c$, and $\bar{\Gamma}$ reduces to the transformation matrix Γ between conservative and primitive variables. In this case Equation (4.1) reduces to the conservative formulation of the non-preconditioned Euler equations in terms of primitive variables.

4.2 Time discretization scheme

In this work we employ an explicit Runge-Kutta time discretization scheme. In Runge-Kutta schemes the solution is advanced in several stages [43] and the residual is evaluated at intermediate states. Coefficients are used to weight the residual at each stage. The coefficients can be optimized in order to expand the stability region and to improve the damping properties of the scheme and hence its convergence and robustness [43, 44, 45].

The Runge-Kutta scheme employed in this work is a s -stage SSP Runge-Kutta scheme. The solution of the preconditioned system is advanced from time t to time $t + \Delta t$ applying the following expression:

$$\begin{aligned} \mathbf{q}^0 &= \mathbf{q}^t, \\ \mathbf{q}^i &= \sum_{k=0}^{i-1} \alpha_{ik} \mathbf{q}^k + \beta_{ik} \Delta t (\bar{\Gamma} \mathbf{M})^{-1} \mathbf{R}(\mathbf{q}^k), \quad i = 1, 2, \dots, s, \\ \mathbf{q}^{t+\Delta t} &= \mathbf{q}^s, \end{aligned} \quad (4.5)$$

where i is the stage counter for the s -stage scheme and α_{ik} and β_{ik} , $k = 0, 1, \dots, i-1$, are the multistage coefficients for the i th-stage, $i = 1, 2, \dots, s$.

4.3 Local time stepping

The main disadvantage of explicit schemes is that the time step Δt is severely restricted by the so-called Courant-Friederichs-Lewy (CFL) condition [46]. On the other hand if we are interested in the steady-state solutions only, several convergence acceleration methodologies are known in literature. A very common technique is the so-called local time-stepping. The basic idea is to advance solutions in the temporal dimension using the maximum permissible time step for each cell. As a result, the convergence to the steady state is considerably accelerated, but the transient solution is no longer temporally accurate. We have to consider that the preconditioned Euler equations are not consistent in time so, for the preconditioned scheme the local time step Δt on each element κ is computed by considering the *CFL* stability condition:

$$\Delta t = CFL \cdot \frac{\Omega_\kappa}{\Lambda_c^x + \Lambda_c^y},$$

where the preconditioned convective spectral radii are defined as

$$\begin{aligned}\Lambda_c^x &= (|\bar{u}'_E| + \bar{c}'_x) \Delta S^x, \\ \Lambda_c^y &= (|\bar{v}'_E| + \bar{c}'_y) \Delta S^y.\end{aligned}$$

The variables ΔS^x and ΔS^y represent the projections of the elemental volume, Ω_κ , on the x and y axis, respectively, whereas \bar{u}'_E , \bar{c}'_x and \bar{v}'_E , \bar{c}'_y are obtained applying Equations (4.4) along the x and y directions and using the mean values of the flow quantities on each element κ .

4.4 Preconditioned Roe's Numerical Flux

The dissipation part of the preconditioned flux splitting scheme has been implemented in the following form:

$$\bar{\Gamma} |\tilde{\mathbf{A}}_{\bar{\Gamma}}| \Delta \mathbf{q} = |u_n| \begin{pmatrix} \Delta(\rho) \\ \Delta(\rho u) \\ \Delta(\rho v) \\ \Delta(\rho E) \end{pmatrix} \mathbf{n} + \delta u_n \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho H \end{pmatrix} \mathbf{n} + \delta p \begin{pmatrix} 0 \\ \mathbf{i} \\ \mathbf{j} \\ \mathbf{v} \end{pmatrix}, \quad (4.6)$$

where

$$\begin{aligned}\delta u_n &= M^* \Delta u_n + [c^* - (1 - 2\alpha) |u_n| - \alpha u_n M^*] \frac{\Delta p}{\rho U_r^2}, \\ \delta p &= M^* \Delta p + [c^* - |u_n| + \alpha u_n M^*] \rho \Delta u_n, \\ \Delta u_n &= \Delta \mathbf{v} \cdot \mathbf{n}, \\ c^* &= \frac{|u'_n + c'| + |u'_n - c'|}{2}, \\ M^* &= \frac{|u'_n + c'| - |u'_n - c'|}{2c'}.\end{aligned}$$

For the non preconditioned system ($\alpha = 0$, $u'_n = u_n$, $c' = U_r = c$) this reduces to the standard Roe's flux difference splitting when Roe-averaged values are used.

It is interesting to note that when the splitting is written in this form, rather than in the more common form factored in terms of u_n , $|u_n + c|$ and $|u_n - c|$ the physical significance of the various added dissipation terms becomes clear. The three terms in (4.6) represent the interpolation to the cell face of the convected variables, the flux velocity and the pressure, respectively. The first term $|u_n|$ has the effect of up-winding the convected variables. The second term δu_n is a modification to the convective velocity at the face. Here the term $M^* \Delta u_n$ appearing in δu_n causes the flux velocity to be up-winded when the normal velocity exceed the pseudoacoustic speed (since $M^* = \pm 1$ when $\pm u'_n > c'$). This occurs only for supersonic, compressible flows, since for low-speed and incompressible flows, M^* is always small. In addition, for low-speed flows, the $c^* \Delta p / \rho U_r^2$ term in δu_n is the added pressure dissipation that arises in simple artificial-compressibility implementations. Note that this augmented flux appears in all of the equations, not just the continuity equation. This term becomes less significant in high-speed flows where ρU_r^2 is much greater than local pressure differences. The third term δp is a modification to the pressure at the face. Here the $M^* \Delta p$ term in δp results in pressure up-winding when the normal velocity becomes supersonic. The entire δp term becomes small for low-speed flow.

4.5 Boundary conditions

4.5.1 Preconditioned far-field

A change in the time-dependent equations also changes the characteristics of the system (although the signs of the eigenvalues remain unchanged). Hence the far-field boundary conditions must be modified for the preconditioned system. In the present study, we have used the simplified preconditioned far-field boundary conditions suggested in [35]. In particular, at the inflow boundary the state \mathbf{q}^b has the same pressure as \mathbf{q}^+ whereas the vector velocity and the temperature are prescribed based on the free-stream values. Conversely, at the outflow boundary the state \mathbf{q}^b has the same temperature and velocity vector of \mathbf{q}^+ whereas the pressure is prescribed based on the free stream value.

Thereby:

$$\mathbf{q}^b = \begin{pmatrix} p^+ \\ u_\infty \\ v_\infty \\ T_\infty \end{pmatrix} \text{ at inflow,} \quad \mathbf{q}^b = \begin{pmatrix} p_\infty \\ u^+ \\ v^+ \\ T^+ \end{pmatrix} \text{ at outflow.} \quad (4.7)$$

4.5.2 Slip wall

The preconditioning of the Euler equations has no effect on the definition of the wall boundary conditions. This means that for the preconditioned scheme we can use exactly the same slip wall boundary conditions employed for the non-preconditioned DG scheme. In order to investigate the influence that the wall boundary conditions have on the accuracy of the solution with and without preconditioning technique, two different no-slip boundary conditions are used in this work: symmetry and local pressure.

- **Symmetry**

The state \mathbf{q}^b has the same pressure, temperature and tangential velocity component as \mathbf{q}^+ and the opposite normal velocity component, i.e.

$$\begin{aligned} p^b &= p^+, \\ (\mathbf{v} \cdot \mathbf{n})^b &= -(\mathbf{v} \cdot \mathbf{n})^+, \\ v_t^b &= v_t^+, \\ T^b &= T^+ \end{aligned} \tag{4.8}$$

where v_t is the tangential vector component of the velocity. In this way the mass flux computed by the Riemann solver is zero and the non-permeability condition is satisfy.

We note that this boundary condition is the same for the preconditioned and the non-preconditioned scheme, but that the Riemann solver used to determine the fluxes on the interior edges is also used on the wall boundary. This means that the fluxes on the wall boundary are computed with the Standard Roe for the non-preconditioned scheme and with the preconditioned Roe for the preconditioned scheme.

- **Local Pressure**

Here we set:

$$\begin{aligned} p^b &= p^+, \\ u^b &= u^+ - (\mathbf{v} \cdot \mathbf{n})^+ n_1, \\ v^b &= v^+ - (\mathbf{v} \cdot \mathbf{n})^+ n_2, \\ T^b &= T^+, \end{aligned} \tag{4.9}$$

where n_1 and n_2 are the components of the unit outward normal $\mathbf{n} = (n_1, n_2)^T$. In this case the conditions imposed on the velocity components of the right state ensure that the normal velocity component is zero on the boundary:

$$(\mathbf{v} \cdot \mathbf{n})^b = 0.$$

In this case the wall boundary fluxes are computed as follows:

$$(\mathbf{F} \cdot \mathbf{n})_{wall} = p^b \begin{pmatrix} 0 \\ n_1 \\ n_2 \\ 0 \end{pmatrix}.$$

This means that the fluxes on the wall boundary are computed in the same manner for both the preconditioned and the non-preconditioned DG schemes.

4.6 Results

The following computations are performed to highlight the potentiality of the DG scheme in the low Mach number limit and to investigate the effect on the performance of the method when using the preconditioning technique, for flows at very

low Mach number. We consider an inviscid flow around the NACA0012 airfoil with a zero angle of attack ($\alpha = 0$). This test case includes a stagnation region close to the leading edge and has been selected to investigate the robustness of the preconditioning method. Computations on different grids, for different low Mach numbers and different polynomial approximations are performed, in order to demonstrate the performance obtained in terms of accuracy and convergence.

We begin by giving a short summary of the simulations carried out:

- Different computational grids: quadrangular and triangular meshes.
Simulations for two different grid topologies are performed in order to investigate the behaviour of both standard and preconditioned DG method using different spatial discretizations.

- Different low Mach numbers: $M = 10^{-1}$, $M = 10^{-2}$ and $M = 10^{-3}$.

Different low free-stream Mach numbers are used to show the behaviour of the standard and preconditioned DG schemes as the Mach number tends to zero.

- Several polynomial approximations: P_1 , P_2 and P_3 elements.

Linear (P_1), quadratic (P_2) and cubic (P_3) elements are used to demonstrate the performance of both standard and preconditioned DG method in the low Mach number limit.

Quadrangular

Triangular

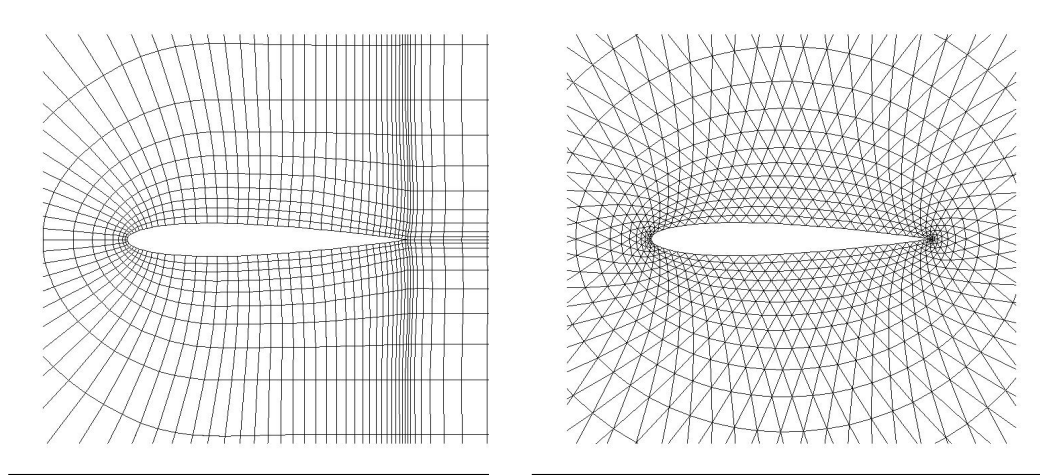


Figure 4.1: Computational Grids

In this work, we use a triangular and a quadrangular grid, both displayed in Figure 4.1. The quadrangular mesh is a C-grid with 1792 elements. The triangular mesh is a O-grid with 2048 elements. The far field boundary of both grids is located far away from the aerodynamic surface.

The discussion of the results obtained is split into two different sections, in order to highlight separately two different aspects, the convergence and the accuracy of the solutions.

- Convergence.

The residual histories versus iteration number were computed to evaluate the effect of the preconditioning technique on the rate of convergence of the solution process. The iteration history is plotted in terms of the L_2 norm of the residuals, that represents the change in the solution over an iteration averaged over all the grids points and equations.

The L_2 norm is computed as

$$L_2 = \sqrt{\frac{\sum_{i=1}^N \sum_{m=1}^M (\delta \bar{\mathbf{q}}_{i,m})^2}{M * N}},$$

where N is the total number of grid points and M is the number equations (4 for the 2D Euler Equations).

In all figures the residual values are normalized such that the first residual equals 1.

- Accuracy.

The accuracy of the numerical results is examined from a qualitative and a quantitative point of view.

The qualitative analysis is performed showing the normalized pressure p_{norm} on the NACA profile. The normalized pressure, p_{norm} , is defined as

$$p_{norm} = \frac{p - p_{min}}{p_{max} - p_{min}}.$$

The quantitative analysis is performed comparing the numerical drag value with the theoretical one (zero the subsonic inviscid flow).

All the computations refer to sufficiently converged solutions and were performed in double precision.

4.6.1 Convergence

Figure 4.2 shows the convergence histories for the quadrangular (left) and the triangular (right) grids at a Mach number of $M = 10^{-1}$, using linear (P_1), quadratic (P_2) and cubic (P_3) elements. Each plot shows the convergence history with and without preconditioning technique. Figures 4.3 and 4.4 show the corresponding convergence histories for the Mach numbers $M = 10^{-2}$ and $M = 10^{-3}$, respectively. The figures show that for both triangular and quadrangular grids the preconditioning technique leads to an acceleration of convergence in comparison to the non-preconditioned solution.

In particular, Figure 4.2 shows that on quadrangular grid the preconditioned computations reach the convergence after ~ 30.000 iterations, using linear elements, ~ 40.000 iterations with quadratic elements, and ~ 55.000 iterations using cubic elements, and are much faster than the corresponding non-preconditioned ones.

For both the preconditioned and the non-preconditioned simulations there are much more iterations required on the triangular grid than on the quadrangular grid. We note that in each plot the residual scale always ranges from 100 to 10^{-13} , however, the iterations scale goes from 0 to 100.000 and from 0 to 600.000 iterations for the plots referring to the quadrangular and triangular grid, respectively. The convergence of the residuals is faster on the quadrangular grid than on the triangular one not only due to the different number of the elements of the two grids (1792 elements for the quadrangular grid and 2048 for the triangular one), but also to the alignment of quadrangular elements with the principal direction of the flux in the case of the flow with zero angle of attack, $\alpha = 0$.

Furthermore, we see that the preconditioned residuals decrease less orders of magnitude than the corresponding non-preconditioned ones. As we will see in more

detail in Chapter 6, the round-off errors and resulting cancellation errors have a larger effect on the preconditioned scheme than on the non-preconditioned one.

Figure 4.3 shows that at $M = 10^{-2}$ the convergence without preconditioning, for a given polynomial approximation, is much slower than the corresponding one at $M = 10^{-1}$, while the number of iterations required to reach the convergence with preconditioning is approximately the same as the corresponding ones at $M = 10^{-1}$. The reason is that, whereas the condition number of the non-preconditioned governing equations, (2.3), increases as the Mach number tends to zero, the time-derivative preconditioning of the Euler equations allows to overcome the stiffness problem and yields a convergence rate that is independent of the Mach number.

The convergence rates of the preconditioned computations are shown to be independent of the Mach number up to $M = 10^{-3}$ in Figure 4.4. Furthermore, this figure shows that the ill-conditioning of the non-preconditioned Euler equations is highest at the lowest Mach number investigated. Finally, from Figures 4.2, 4.3 and 4.4, we see that for a given polynomial degree, the residuals decay reduce as the Mach number tends to zero. Here, round-off errors become increasingly important, as explained in Chapter 6 in more detail.

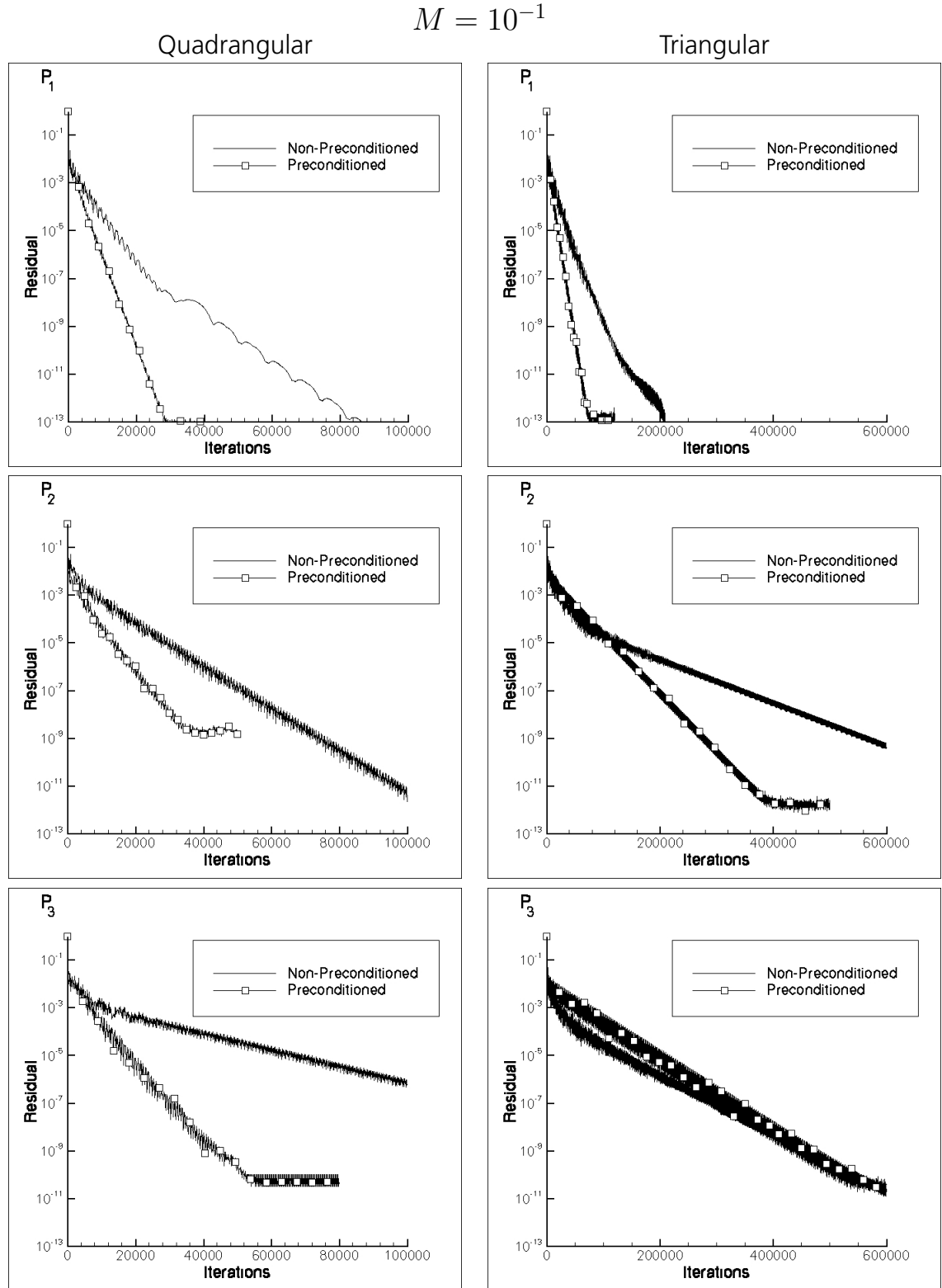


Figure 4.2: Residuals for $M = 10^{-1}$ (with and without preconditioning). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

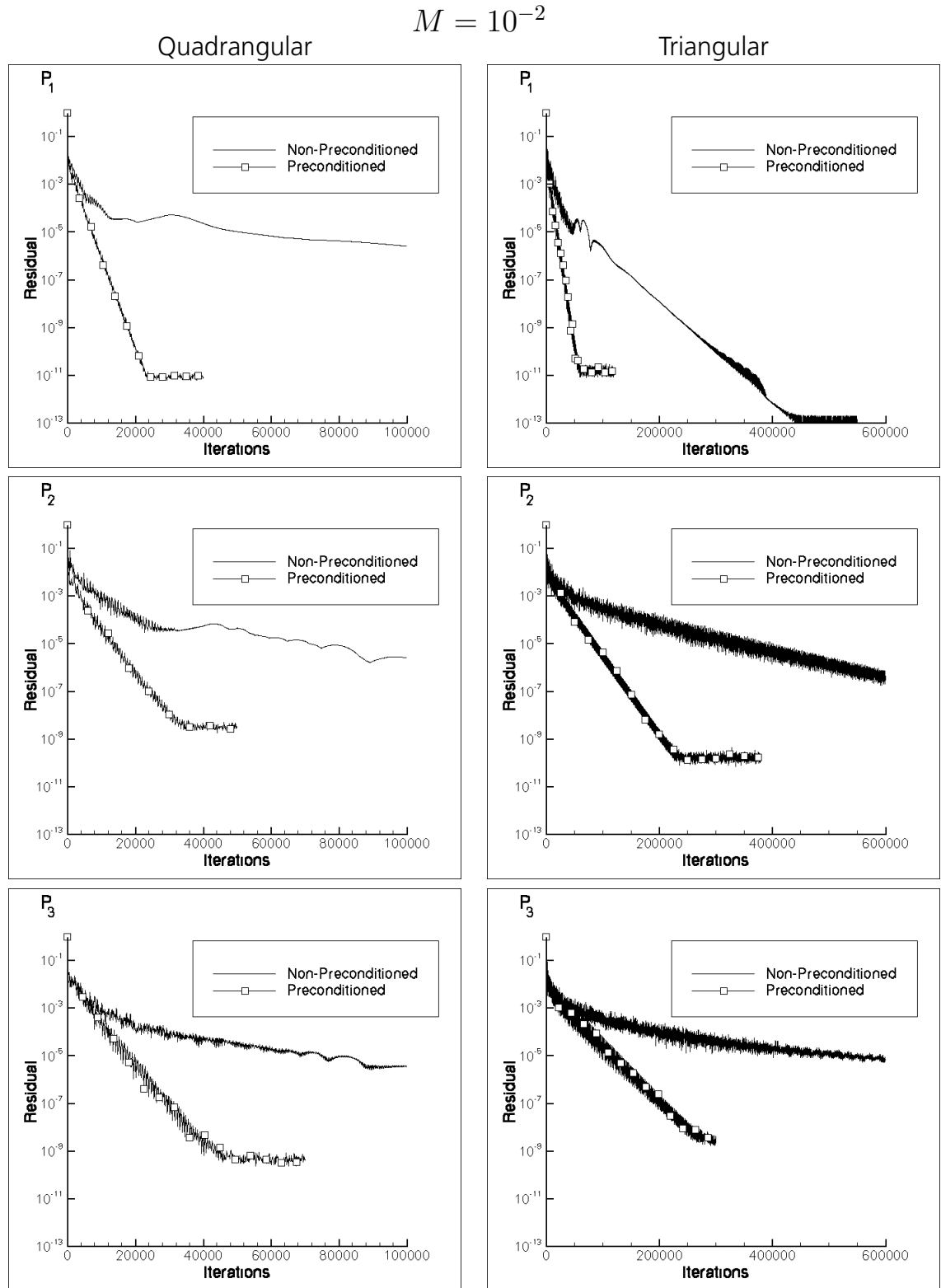


Figure 4.3: Residuals for $M = 10^{-2}$ (with and without preconditioning). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

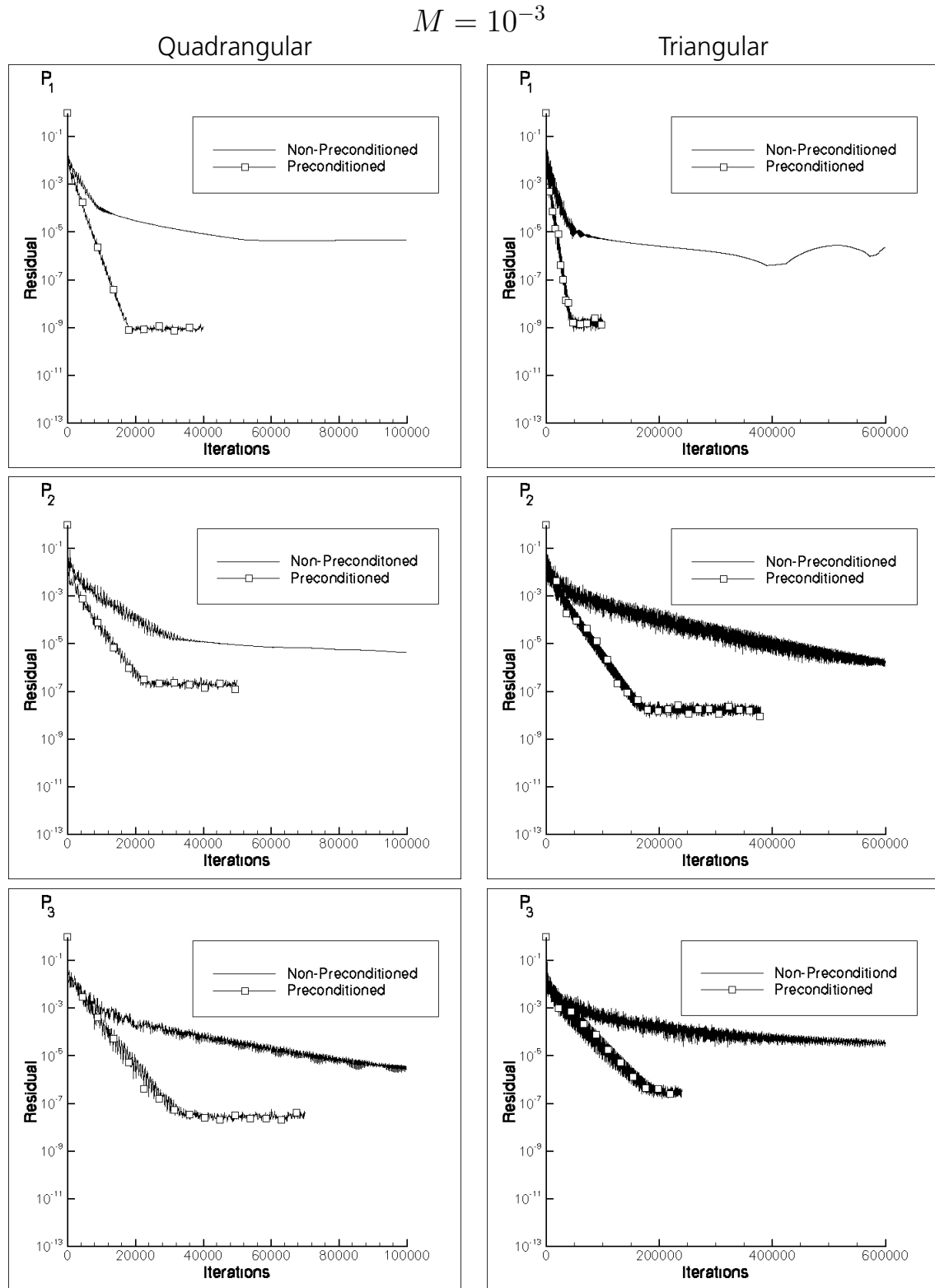


Figure 4.4: Residuals for $M = 10^{-3}$ (with and without preconditioning). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

Figure 4.5 collects the residual histories of each of the governing equations computed at different Mach numbers on the quadrangular grid to show the effect of the time-derivative preconditioning on the stiffness problem. The residual history of the thermodynamic variables and velocity components, as a function of iteration numbers, are presented for both non-preconditioned (left) and preconditioned (right) computations, using linear elements.

It appears that the delay in satisfying the convergence criterion for the non-preconditioned computations is due above all to the residual of velocity and this delay increases as the Mach number reduces. This is due to the fact that the acoustic waves travel M^{-1} times faster than the convective waves in the computational domain. On the contrary, the preconditioned residual histories have almost the same rate of convergence, independently of the Mach number, showing that the stiffness problem is strongly reduced by the preconditioning. Similar results are obtained for P_2 and P_3 approximation and for triangular discretization.

The residual histories based on two very different Mach numbers using P_1 , P_2 and P_3 elements on both quadrangular and triangular grids are shown in Figure 4.6 to quantify the acceleration obtained by the preconditioning.

The results shown in the figure at $M = 10^{-3}$ and $M = 0.4$ refer to the preconditioned and non-preconditioned scheme, respectively. In particular the non-preconditioned computations refer to a well conditioned problem and adopt the typical CFL condition of the Runge-Kutta TVDRK scheme

$$CFL = \frac{1}{(2p + 1)}$$

where p is the polynomial degree. We see that the convergence rates at $M = 10^{-3}$ with the preconditioning technique are equal or greater (P_1 elements for triangular grid and P_3 elements for quadrangular grid) than the corresponding ones at $M = 0.4$ without preconditioning.

4.6.2 Accuracy

In this section we analyze the accuracy of the solutions for different Mach numbers and different polynomial degree with and without preconditioning technique. The results refer to computations performed on quadrangular and triangular grids, both shown in Figure 4.1.

Figures 4.7, 4.8 and 4.9 refer to quadrangular grid. Figure 4.7 shows both the preconditioned and the non-preconditioned contours of normalized pressure at $M = 10^{-1}$ for linear (P_1), quadratic (P_2) and cubic (P_3) elements. Figures 4.8 and 4.9 show the corresponding contour plots of normalized pressure at $M = 10^{-2}$ and $M = 10^{-3}$ respectively. From Figures 4.7, 4.8 and 4.9, we see that the accuracy of the solution obtained using the preconditioning technique is much better than those obtained without preconditioning.

In particular at $M = 10^{-1}$, see Figure 4.7, the comparison between non-preconditioned (left) and preconditioned (right) P_1 solutions shows a clear loss of accuracy of the

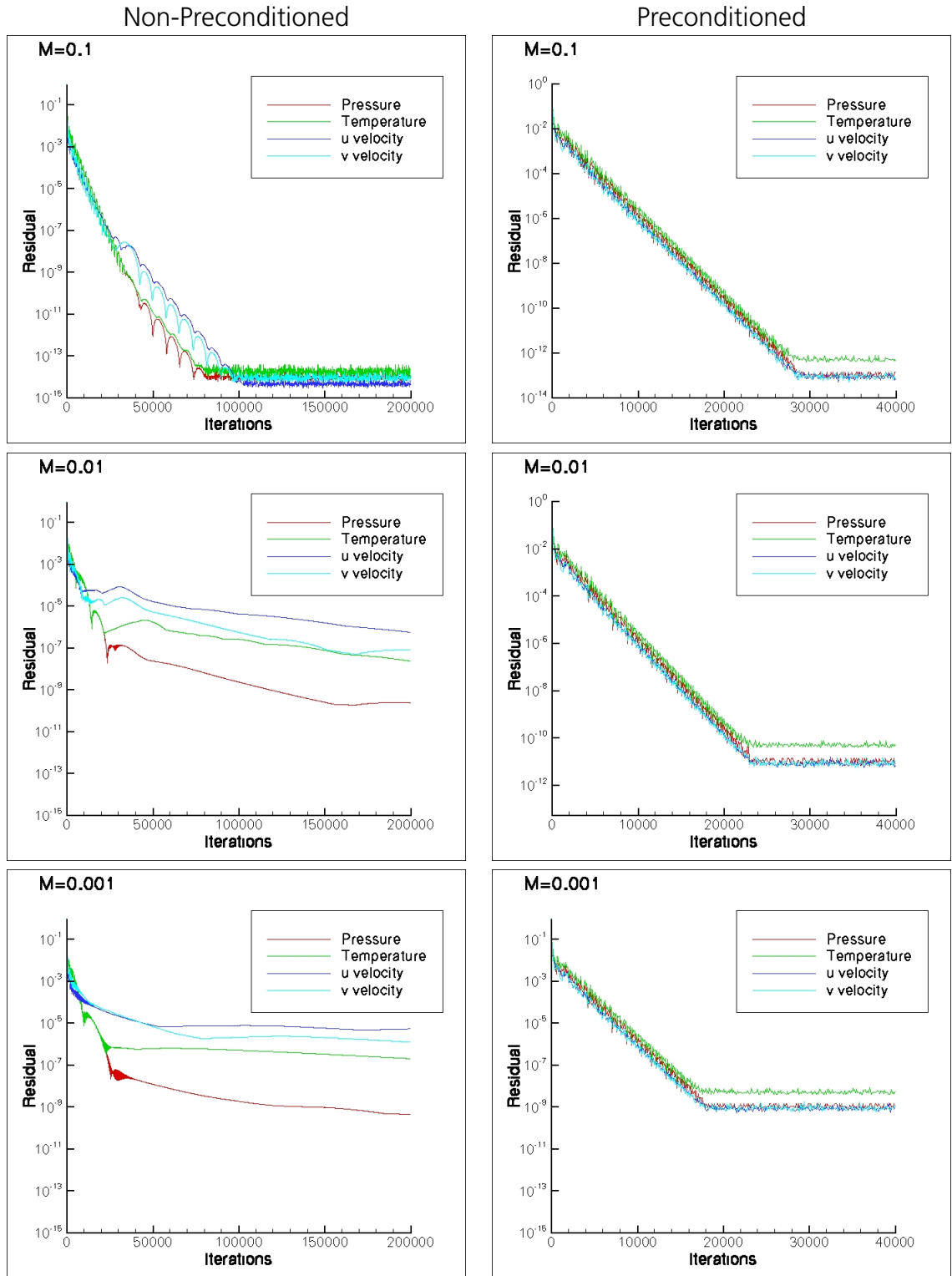
Quadrangular grid: P_1 elements

Figure 4.5: Quadrangular grid: residuals of linear elements (P_1). $M = 0.1$ (top), $M = 0.01$ (middle) and $M = 0.001$ (bottom); without (left) and with (right) preconditioning technique.

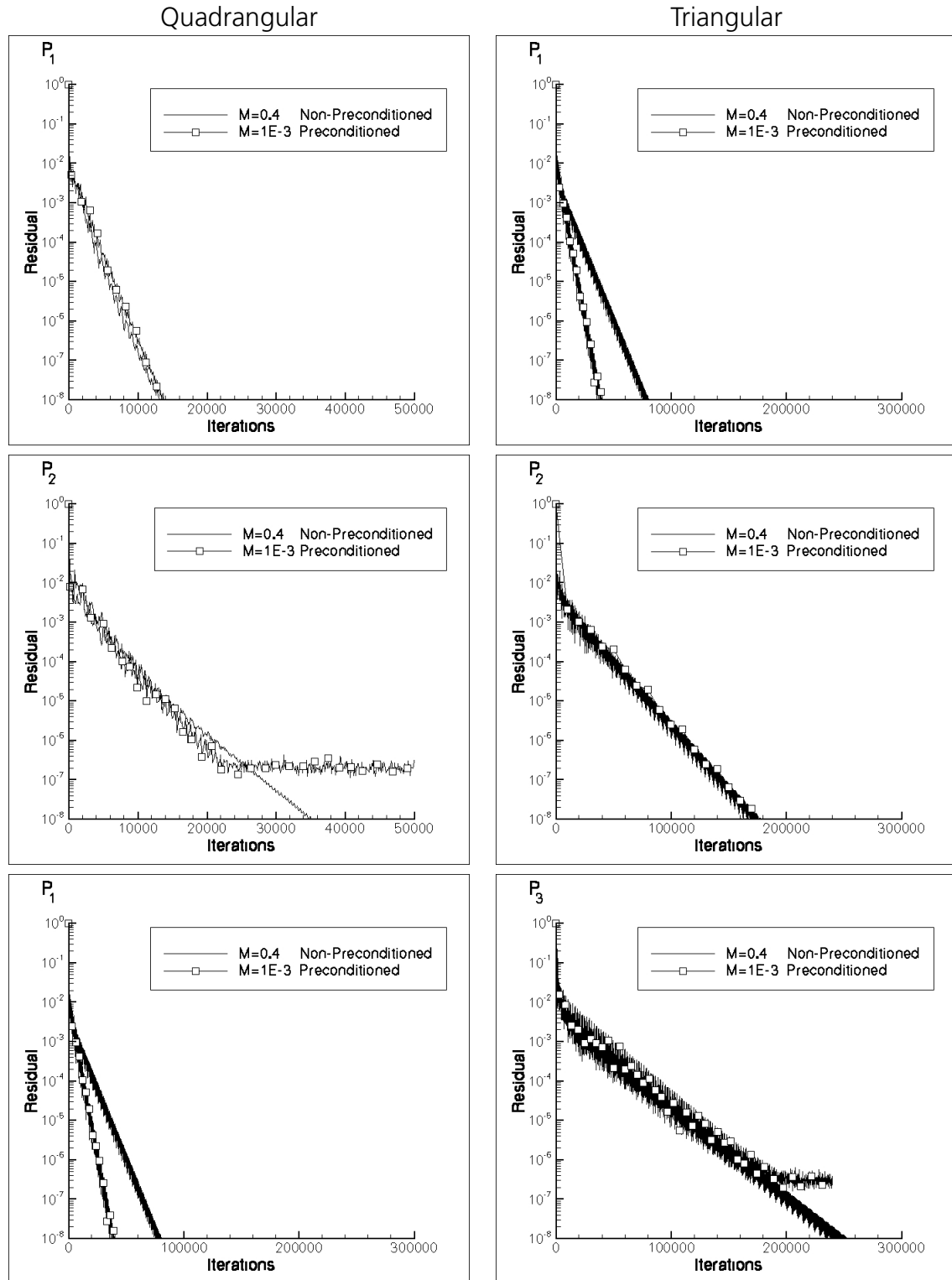


Figure 4.6: Residuals with preconditioning technique for $M = 10^{-3}$ and without preconditioning technique for $M = 0.4$. Quadrangular (left) and triangular (right) grid; linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

non-preconditioned solution in comparison to the preconditioned one. This loss of accuracy is less evident for P_2 elements (middle), whereas for P_3 element there are no visible differences in terms of pressure isolines. Finally, we see that for a given Mach number, the lower the polynomial degree, the higher is the difference between preconditioned and non-preconditioned solutions.

This behaviour is confirmed in Figures 4.8 and 4.9. Furthermore at $M = 10^{-2}$, see Figure 4.8, we see that at least P_3 elements are required to obtain an acceptable level of accuracy without preconditioning. In fact, the P_1 and P_2 non-preconditioned solutions suffer from a lack of accuracy in comparison to the corresponding non-preconditioned ones, with the P_2 solution being much better than the P_1 solution. Then, Figures 4.7 and 4.8 show that, for a given polynomial degree, the quality of the non-preconditioned solution becomes worse in comparison to the corresponding preconditioned one as the Mach number reduces.

The previous considerations are confirmed at $M = 10^{-3}$ in Figure 4.9. In particular, this figure shows that at the lowest Mach number investigated there is a clear difference between preconditioned and non-preconditioned solutions also if P_3 elements are used.

From Figures 4.7, 4.8 and 4.9, it appears also that the higher the polynomial degree, the higher the accuracy of the solution obtained using the Roe's preconditioned numerical flux. Furthermore, the accuracy of preconditioned solutions is independent of the Mach number: for a given polynomial degree, the preconditioned pressure isolines at $M = 10^{-3}$ are similar to those computed at $M = 10^{-2}$ and $M = 10^{-1}$.

Now we extend the analysis to triangular grid. Figure 4.10 shows the contours of normalized pressure at $M = 10^{-1}$ for linear (P_1), quadratic (P_2) and cubic (P_3) elements, comparing the results obtained with and without preconditioning technique. Figures 4.11 and 4.12 show the corresponding pressure isolines at $M = 10^{-2}$ and $M = 10^{-3}$, respectively. From these figures we see that the non-preconditioned computations are more accurate than the corresponding ones obtained using the quadrangular grid. In particular, for all polynomial degrees the preconditioned and non-preconditioned solutions are almost indistinguishable on the triangular grid.

In summary, the results in Figures 4.7-4.12 show the robustness of the Weiss and Smith preconditioner [34] consisting in its ability to accurately solve a stagnation flow test problem in the incompressible limit.

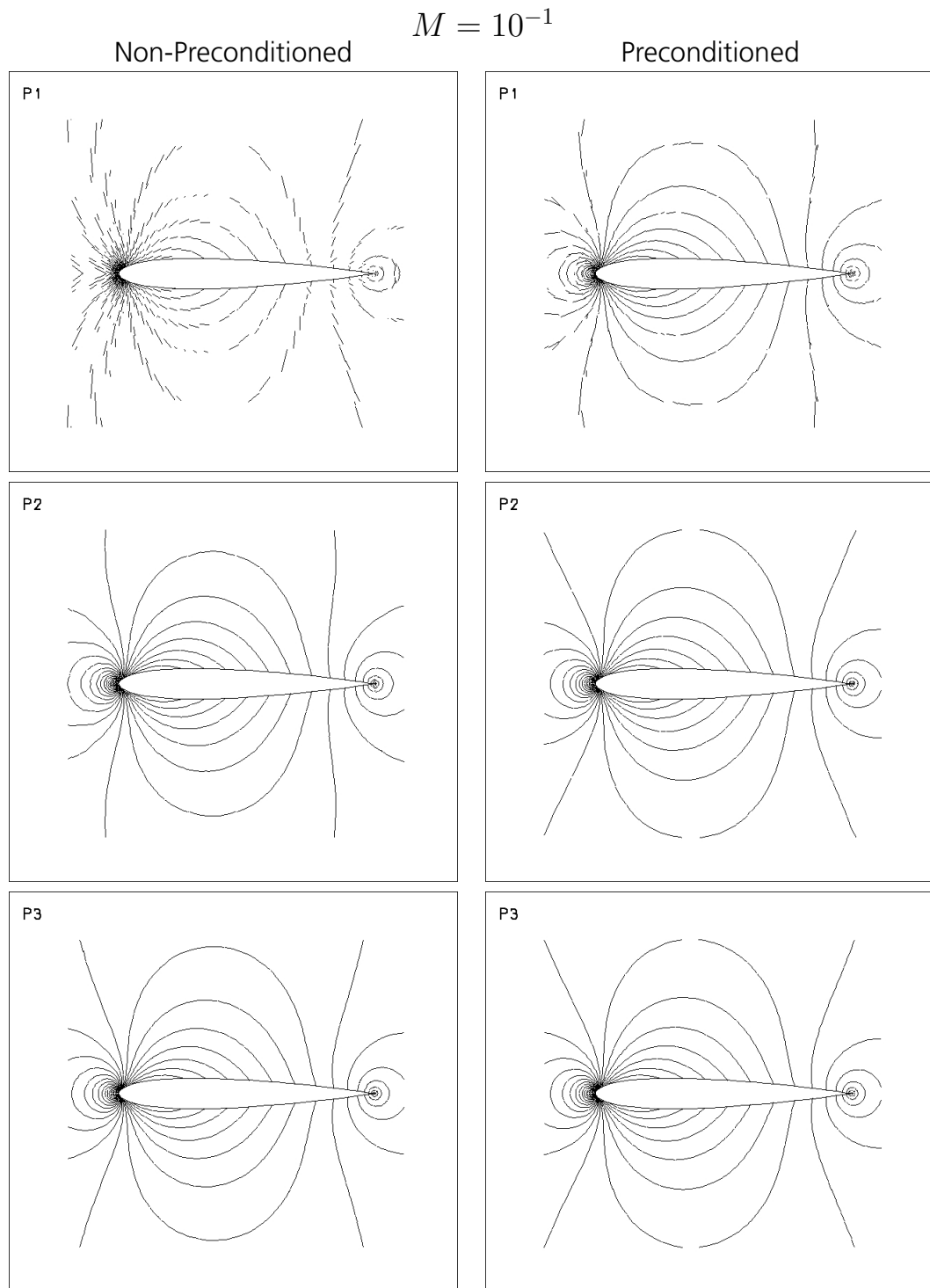


Figure 4.7: Quadrangular grid: contours of normalized pressure for $M = 10^{-1}$. Non-preconditioned (left) and preconditioned (right); linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

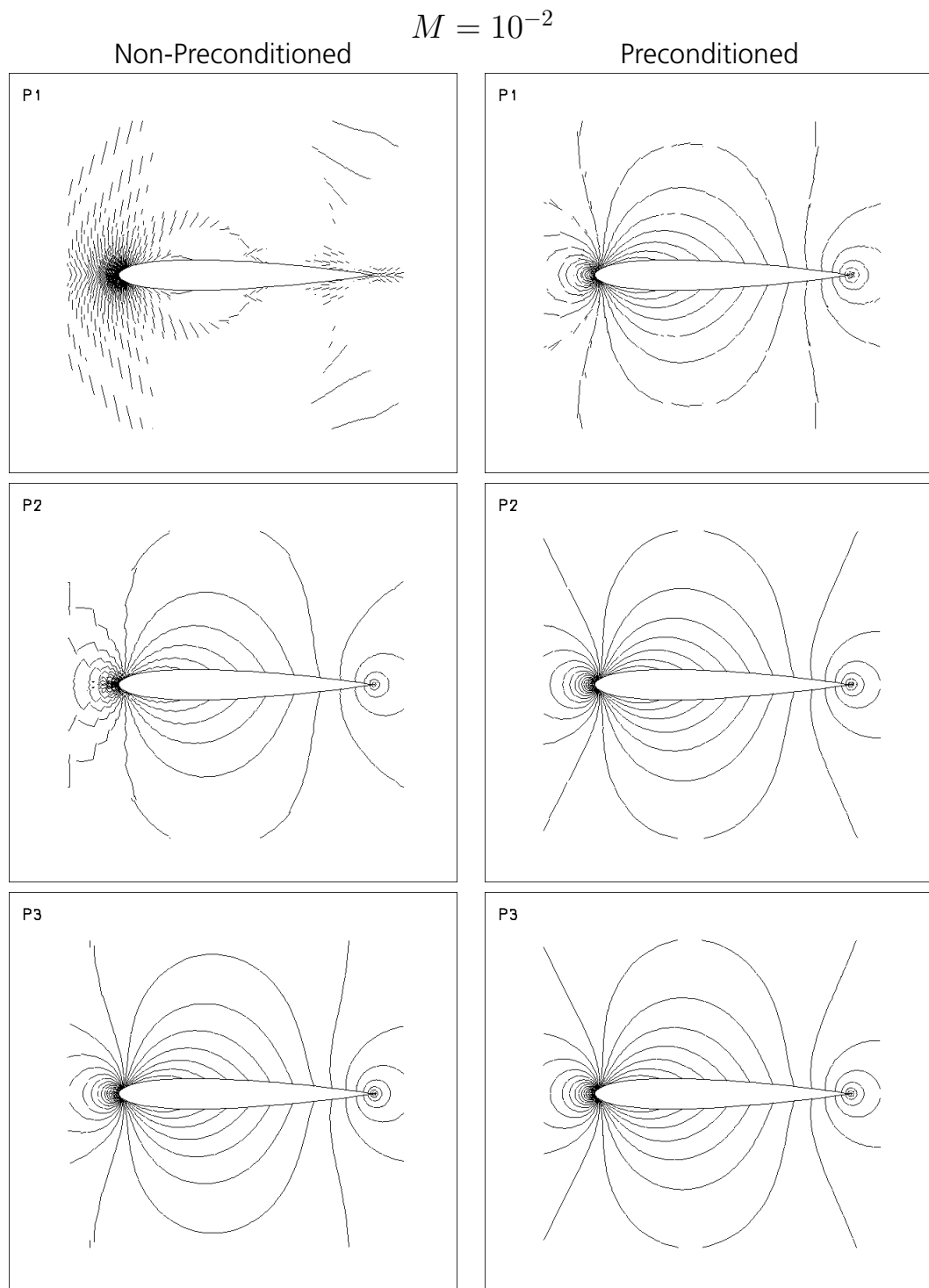


Figure 4.8: Quadrangular grid: contours of normalized pressure for $M = 10^{-2}$. Non-preconditioned (left) and preconditioned (right); linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

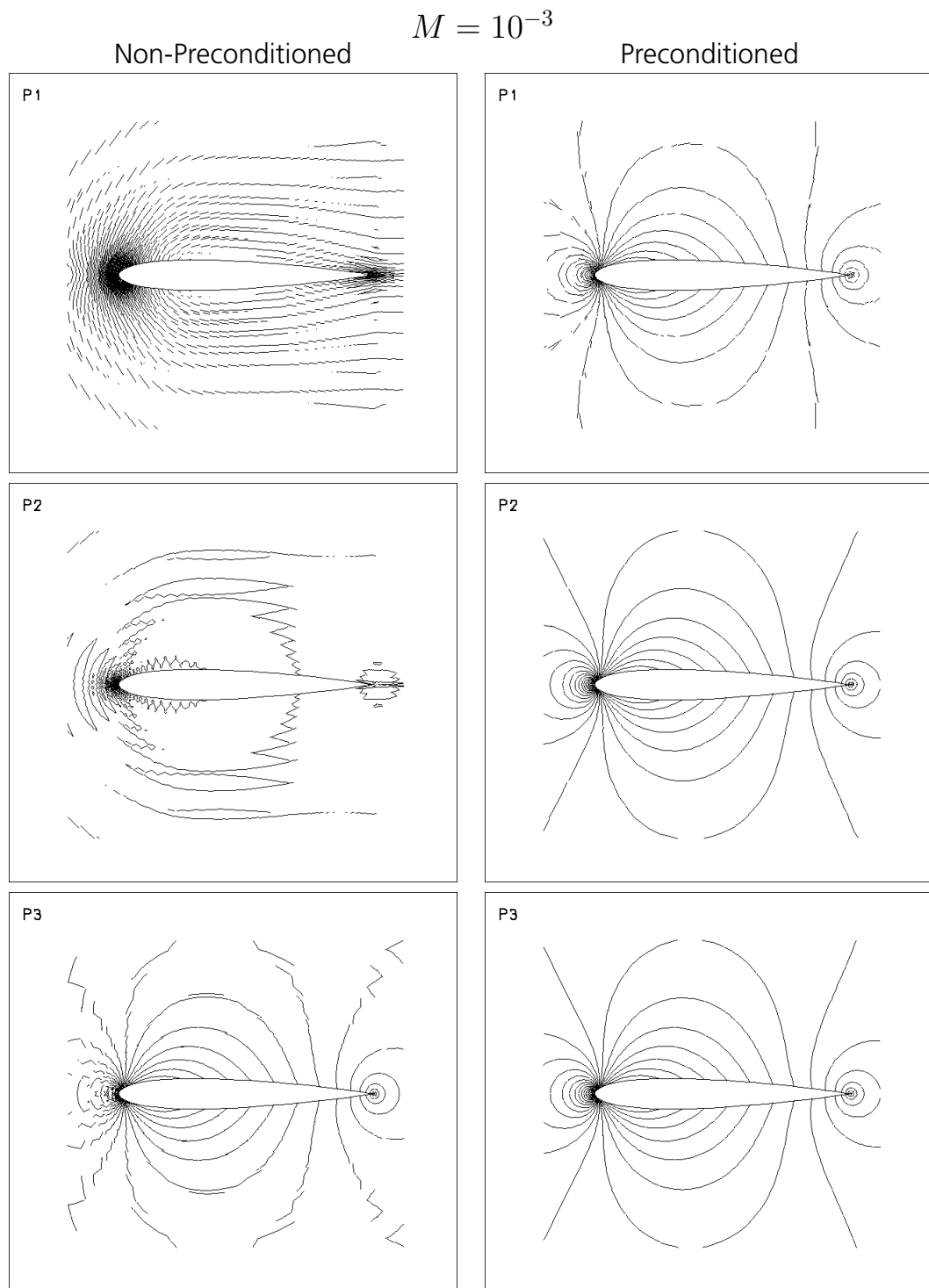


Figure 4.9: Quadrangular grid: contours of normalized pressure for $M = 10^{-3}$. Non-preconditioned (left) and preconditioned (right); linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

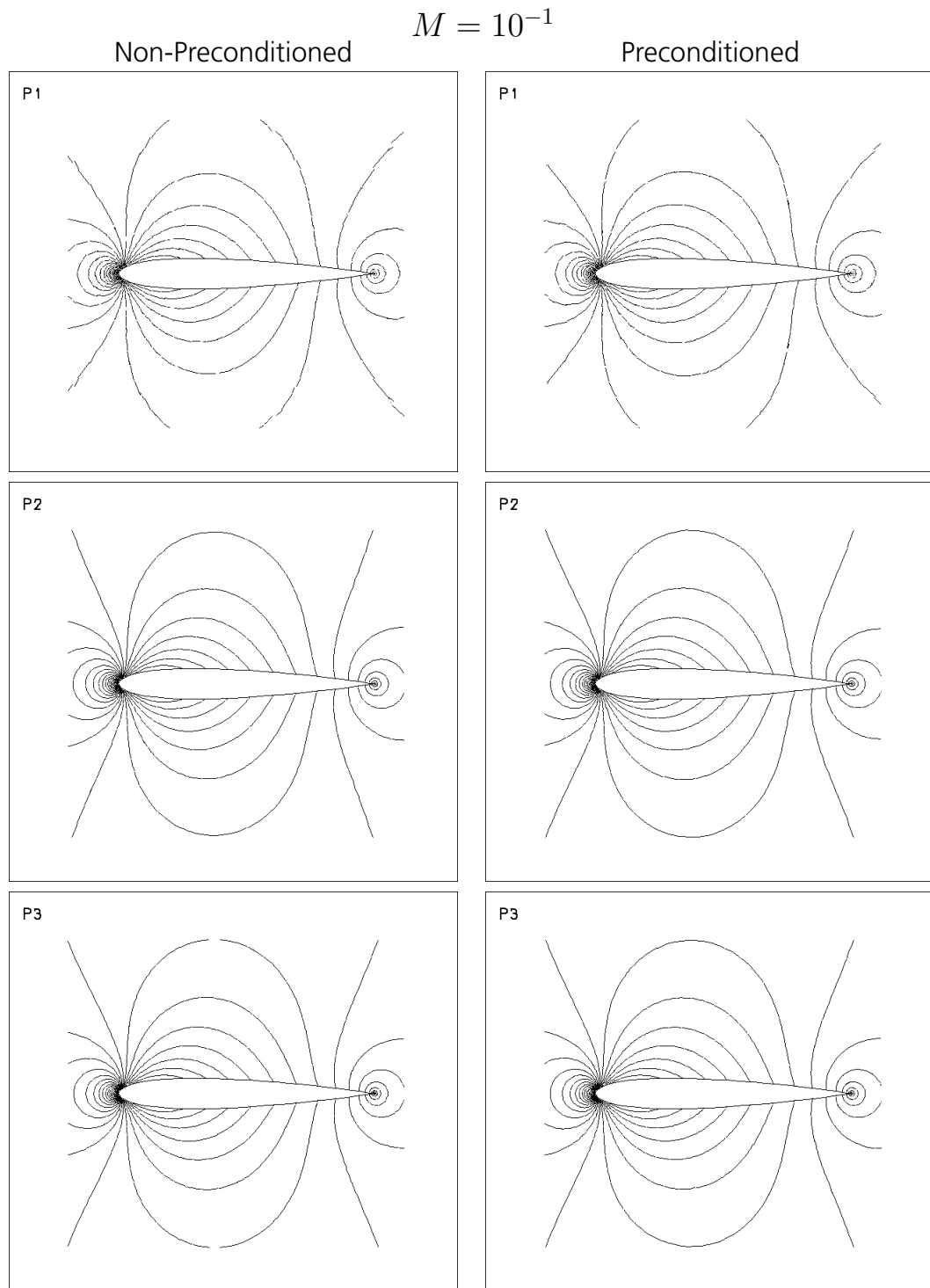


Figure 4.10: Triangular grid: contours of normalized pressure for $M = 10^{-1}$. Non-preconditioned (left) and preconditioned (right); linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

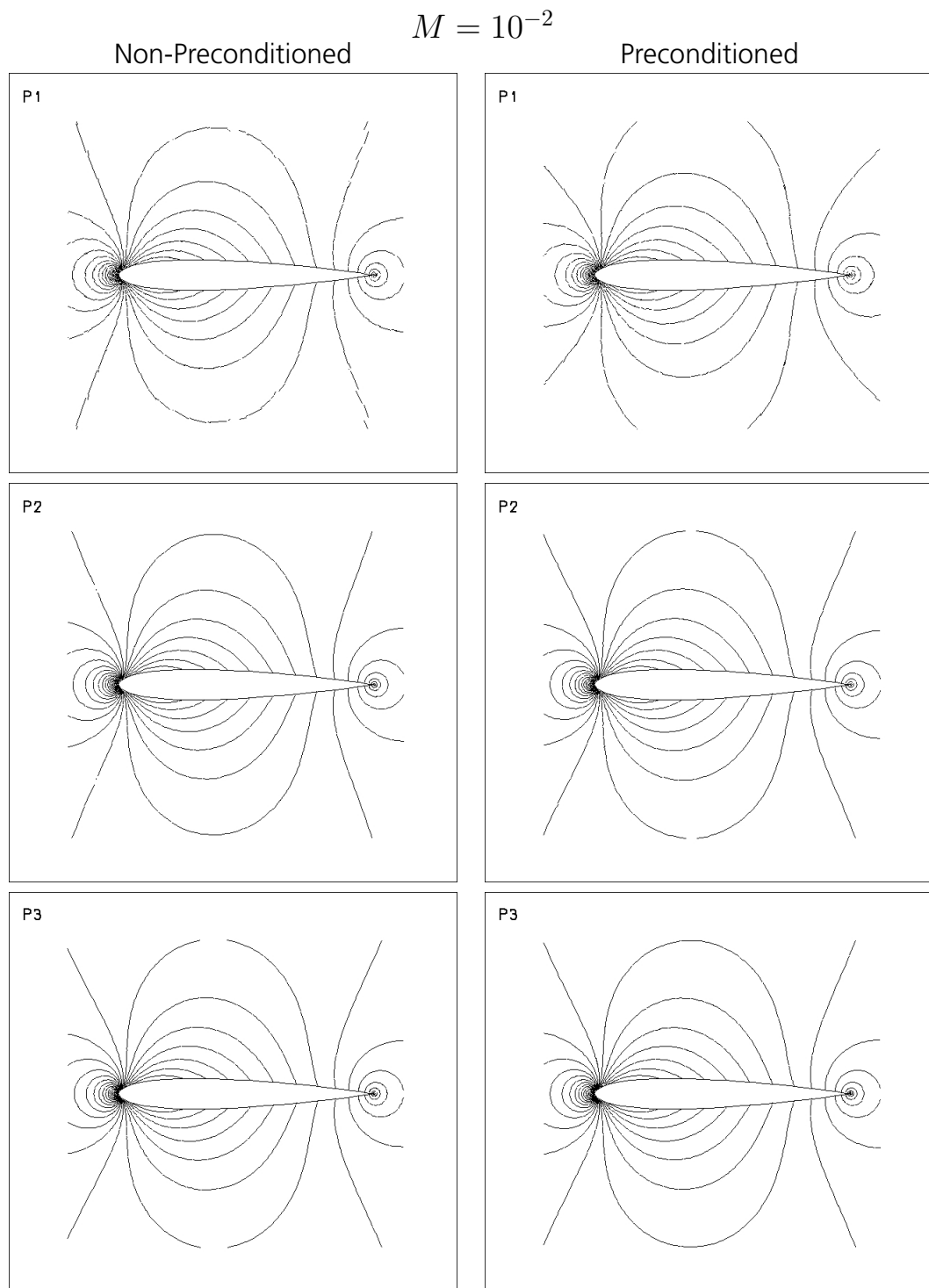


Figure 4.11: Triangular grid: contours of normalized pressure for $M = 10^{-2}$. Non-preconditioned (left) and preconditioned (right); linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

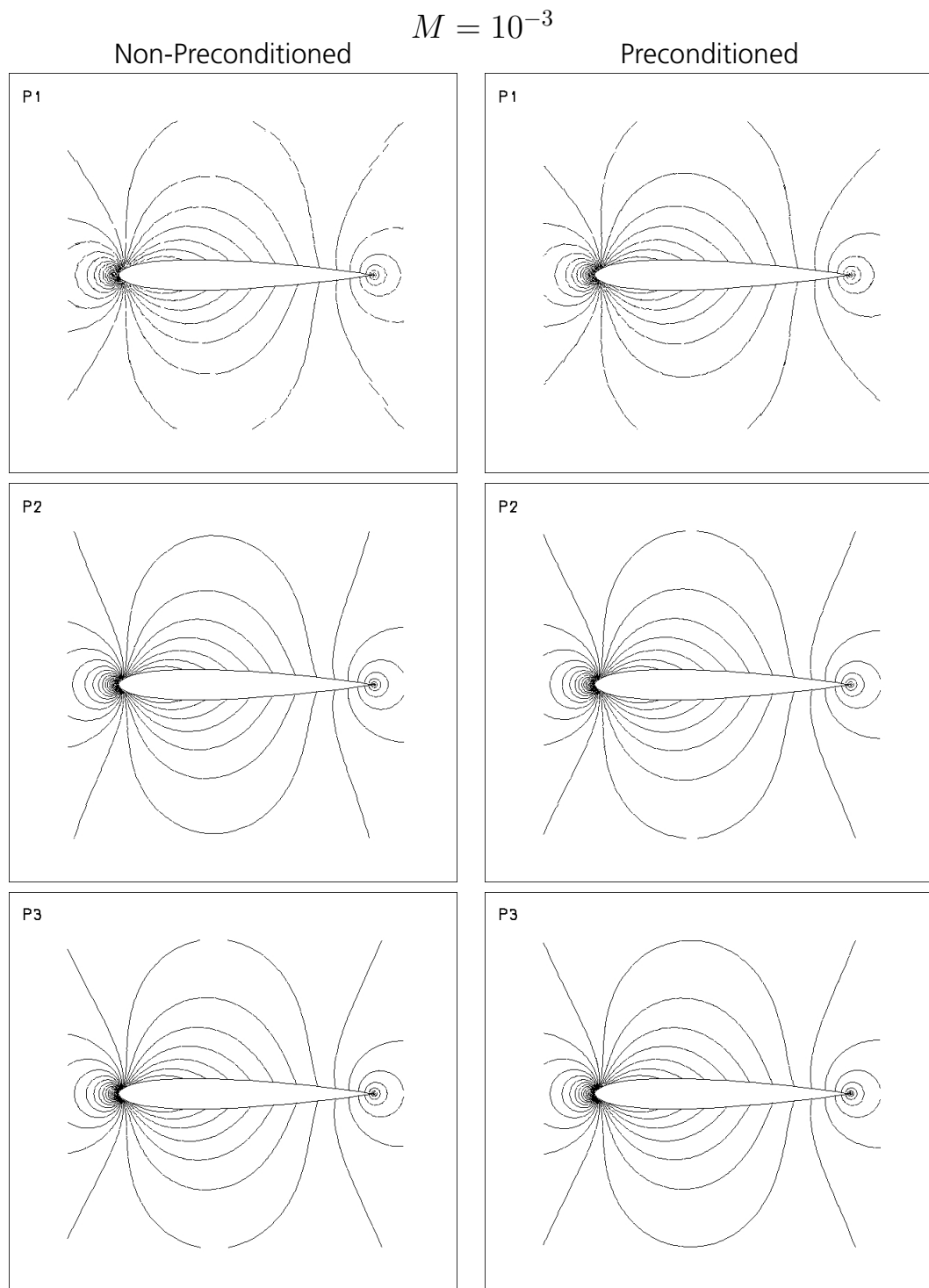


Figure 4.12: Triangular grid: contours of normalized pressure for $M = 10^{-3}$. Non-preconditioned (left) and preconditioned (right); linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

Up to now the analysis of results has been performed in terms of "the quality (in picture norm) of the plots of normalized pressure". Obviously this analysis is a qualitative consideration only. Although the influence of the spatial discretization on the accuracy is quite clear already it deserves a more accurate study.

For this purpose Tables 4.1 and 4.2 collect the drag coefficients computed at different Mach numbers ($M = 10^{-1}$, 10^{-2} and 10^{-3}) for P_1 , P_2 and P_3 elements, using the preconditioned and non-preconditioned DG schemes, respectively. In particular, Table 4.1 refers to quadrangular grid while Table 4.2 refers to triangular grid.

Quadrangular grid						
	$M = 10^{-1}$		$M = 10^{-2}$		$M = 10^{-3}$	
	Non-Prec.	Prec.	Non-Prec.	Prec.	Non-Prec.	Prec.
P_1	$5.698 \cdot 10^{-3}$	$1.388 \cdot 10^{-3}$	$3.422 \cdot 10^{-2}$	$1.387 \cdot 10^{-3}$	$1.630 \cdot 10^{-1}$	$1.387 \cdot 10^{-3}$
P_2	$2.616 \cdot 10^{-4}$	$8.414 \cdot 10^{-5}$	$1.307 \cdot 10^{-3}$	$8.440 \cdot 10^{-5}$	$4.453 \cdot 10^{-3}$	$8.441 \cdot 10^{-5}$
P_3	$3.411 \cdot 10^{-5}$	$2.089 \cdot 10^{-5}$	$5.913 \cdot 10^{-5}$	$2.094 \cdot 10^{-5}$	$1.321 \cdot 10^{-4}$	$2.095 \cdot 10^{-5}$

Table 4.1: Drag-coefficients for quadrangular grid using symmetry BCs.

Triangular grid						
	$M = 10^{-1}$		$M = 10^{-2}$		$M = 10^{-3}$	
	Non-Prec.	Prec.	Non-Prec.	Prec.	Non-Prec.	Prec.
P_1	$6.576 \cdot 10^{-4}$	$4.904 \cdot 10^{-4}$	$1.076 \cdot 10^{-3}$	$4.877 \cdot 10^{-4}$	$1.884 \cdot 10^{-3}$	$4.874 \cdot 10^{-4}$
P_2	$2.721 \cdot 10^{-5}$	$2.268 \cdot 10^{-5}$	$4.270 \cdot 10^{-5}$	$2.078 \cdot 10^{-5}$	$5.664 \cdot 10^{-5}$	$2.057 \cdot 10^{-5}$
P_3	$6.374 \cdot 10^{-6}$	$5.440 \cdot 10^{-6}$	$4.728 \cdot 10^{-6}$	$3.038 \cdot 10^{-6}$	$4.917 \cdot 10^{-6}$	$2.785 \cdot 10^{-6}$

Table 4.2: Drag-coefficients for triangular grid using symmetry BCs.

Tables 4.1 and 4.2 show that non-preconditioned drag coefficients of P_1 solutions at $M = 10^{-3}$ are 1 and 2 order of magnitude higher than those at $M = 10^{-1}$ for triangular and quadrangular grids, respectively. This lack of accuracy at lowest Mach number reduces when the polynomial order approximation increases. In particular, the drag coefficients of P_2 solutions become 2 and 10 times higher than those at $M = 10^{-1}$, while P_3 solutions give drag coefficients even 1.3 times lower and 4 times higher than those at $M = 10^{-1}$ for triangular and quadrangular grids, respectively. This confirms that using a high order representation of the unknowns it is possible to preserve the accuracy in the low Mach number limit without preconditioning. Furthermore the results obtained on the triangular grid are more accurate than those obtained on the quadrangular grid. In fact, it is interesting to see that the P_2 and P_3 computations performed on the triangular grid preserve the numerical accuracy for Mach numbers ranging from 10^{-1} to 10^{-3} . Even the P_3 solution at $M = 10^{-3}$ is more accurate than that at $M = 10^{-1}$. Conversely, the corresponding results computed on quadrangular grid show that the numerical accuracy decreases as the Mach number

reduces. The reason for this different behaviour will be explained later, where this trend becomes clearer.

Tables 4.1 and 4.2 show that drag coefficients calculated with preconditioning are independent of the Mach number, even if some differences are present at $M = 10^{-1}$ due to compressibility effects [47]. Furthermore the preconditioning always improves the accuracy of solution. The preconditioned computations performed on triangular grid are more accurate than those computed on quadrangular grid. Nevertheless the improvement of accuracy comparing the preconditioned and the non-preconditioned values is much more marked on the quadrangular grid.

In Figures 4.13 and 4.14 the contours of normalized pressure near the leading edge of the NACA profile are presented in order to investigate the reasons that lead to different drag coefficients for the two spatial discretizations considered in this work. The figures refer to computations performed at $M = 10^{-1}$ on both the triangular and quadrangular grids, using P_1 , P_2 and P_3 elements, with and without preconditioning technique.

Figure 4.13 shows that the loss of accuracy, when computations are performed on quadrangular grid without preconditioning (left column), is found to be generated at the leading edge and is due to entropy generation within the stagnation region. Furthermore, as expected, the dissipation reduces when the polynomial degree increases. In particular, accurate pressure isolines are obtained using P_3 elements. Conversely, the corresponding solutions with preconditioning are less dissipative, even if the differences between preconditioned and non-preconditioned results reduces when using higher polynomial degrees.

In contrast to that, Figure 4.14 shows that preconditioned and non-preconditioned pressure isolines are almost indistinguishables on triangular grid. This means that the entropy generation at the leading edge is strongly reduced by the different spatial discretization.

These two opposite behaviours can be explained with a different effect that each grid has on the Roe's approximate Riemann solver. In fact, as shown in Equations (3.11) and (3.12), the Roe's flux exhibit unbalancing between the central and the dissipative terms at low Mach number, giving accuracy problems [28]. Thus in regions of high gradients this behaviour will be enhanced with further lack of accuracy. In low speed calculations, the region of high gradients occurs near the stagnation points. Then the previous results show that the triangular grid has a balancing effect on these terms, whereas the quadrangular grid does not modify significantly the behaviour of the Roe's flux at low Mach numbers. Notwithstanding the DG method allows to reduce the dissipation produced at the leading edge when high order polynomial approximations are used, independently of the spatial discretization.

Quadrangular grid: $M = 10^{-1}$

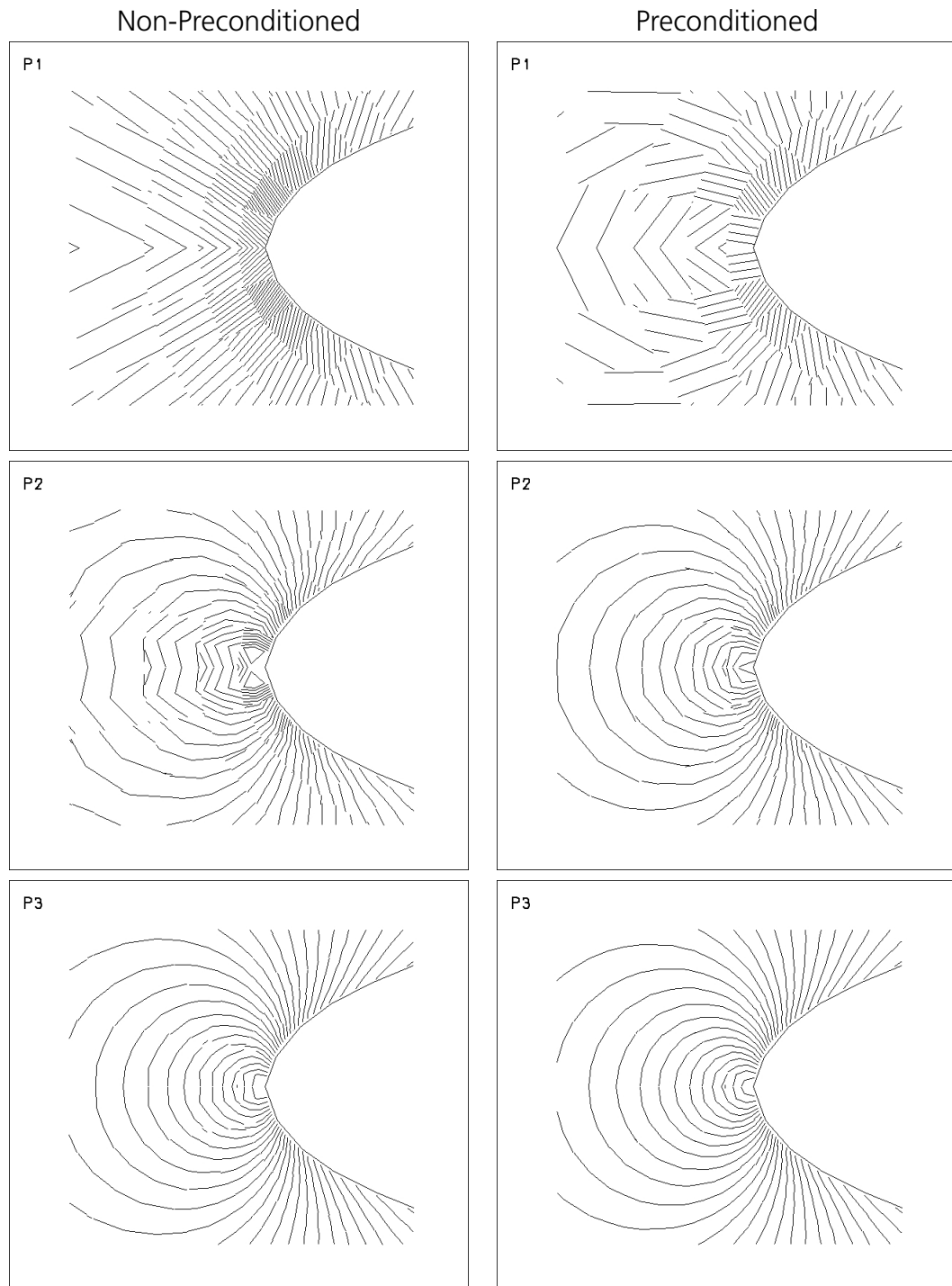


Figure 4.13: Quadrangular grid: contours of normalized pressure near the leading edge for $M = 10^{-1}$. Non-preconditioned (left) and preconditioned (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

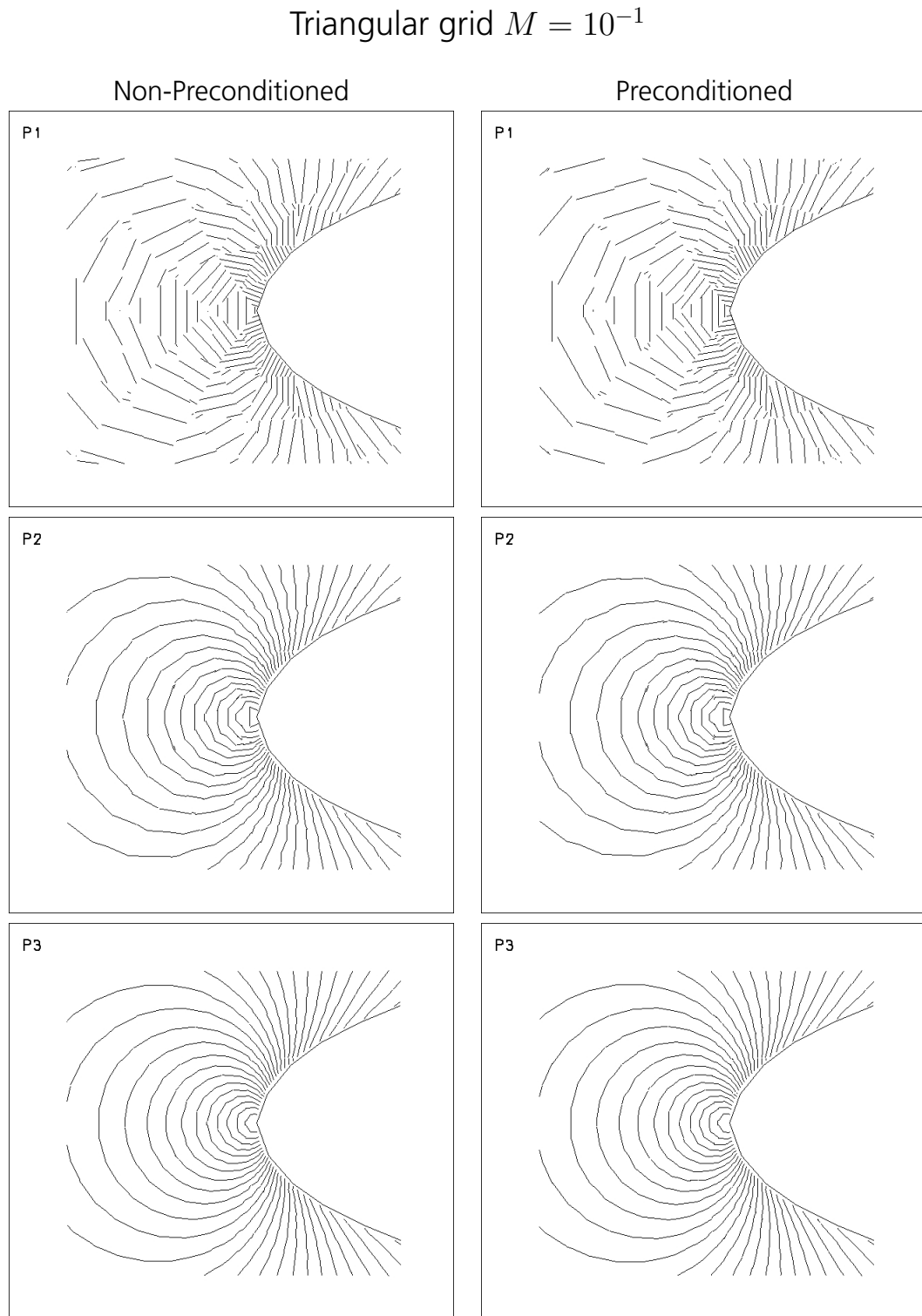


Figure 4.14: Triangular grid: contours of normalized pressure near the leading edge for $M = 10^{-1}$. Non-preconditioned (left) and preconditioned (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

The results presented up to now were obtained using a symmetry technique, see (4.8), to enforce a slip boundary condition at solid walls. With the aim to investigate the effect of the wall treatment on the numerical accuracy a different wall boundary conditions has been implemented: the local pressure condition, see (4.9). The results shown in Table 4.3 and 4.4 refer to quadrangular and triangular grid, respectively.

	Quadrangular grid					
	$M = 10^{-1}$		$M = 10^{-2}$		$M = 10^{-3}$	
	Non-Prec.	Prec.	Non-Prec.	Prec.	Non-Prec.	Prec.
P_1	$4.667 \cdot 10^{-3}$	$1.302 \cdot 10^{-3}$	$2.494 \cdot 10^{-2}$	$1.302 \cdot 10^{-3}$	$1.270 \cdot 10^{-1}$	$1.301 \cdot 10^{-3}$
P_2	$1.280 \cdot 10^{-4}$	$6.621 \cdot 10^{-5}$	$4.540 \cdot 10^{-4}$	$6.641 \cdot 10^{-5}$	$2.225 \cdot 10^{-3}$	$6.642 \cdot 10^{-5}$
P_3	$2.763 \cdot 10^{-5}$	$1.658 \cdot 10^{-5}$	$3.759 \cdot 10^{-5}$	$1.662 \cdot 10^{-5}$	$6.809 \cdot 10^{-5}$	$1.662 \cdot 10^{-5}$

Table 4.3: Drag-coefficients for quadrangular grid using local pressure BCs.

	Triangular grid					
	$M = 10^{-1}$		$M = 10^{-2}$		$M = 10^{-3}$	
	Non-Prec.	Prec.	Non-Prec.	Prec.	Non-Prec.	Prec.
P_1	$5.295 \cdot 10^{-4}$	$4.716 \cdot 10^{-4}$	$5.149 \cdot 10^{-4}$	$4.689 \cdot 10^{-4}$	$5.434 \cdot 10^{-4}$	$4.686 \cdot 10^{-4}$
P_2	$2.177 \cdot 10^{-5}$	$2.157 \cdot 10^{-5}$	$2.017 \cdot 10^{-5}$	$1.966 \cdot 10^{-5}$	$2.001 \cdot 10^{-5}$	$1.946 \cdot 10^{-5}$
P_3	$6.146 \cdot 10^{-6}$	$5.266 \cdot 10^{-6}$	$3.970 \cdot 10^{-6}$	$2.863 \cdot 10^{-6}$	$3.743 \cdot 10^{-6}$	$2.609 \cdot 10^{-6}$

Table 4.4: Drag-coefficients for triangular grid using local pressure BCs.

Comparing Tables 4.1 and 4.3 as well as 4.2 and 4.4 shows that the symmetry condition is more dissipative than the local pressure condition. Nevertheless, the lowest production of entropy at the leading edge was not sufficient enough to overcome the accuracy problem shown above for the quadrangular grid. Finally, Table 4.3 confirms that without preconditioning the numerical accuracy decreases when Mach number reduces, on the quadrangular grid, whereas in Table 4.4 we see that the P_2 and P_3 solutions show an opposite behaviour, which is due to the balancing effect of triangular discretization on the central and dissipative term of Roe's flux at low Mach numbers.

5 Implicit Scheme: Flux Preconditioning technique for the Euler equations

In this chapter we present the preconditioning of the Euler equations in combination with a fully implicit time integration method. In particular, we find that the flux preconditioning approach, which modifies only the dissipative terms of the numerical flux, improves both the accuracy and the rate of convergence of the numerical solution. This formulation is quite simple to implement in any existing implicit DG code, overcomes the time-stepping restrictions of explicit multistage algorithms, is consistent in time and thus applicable to unsteady flows.

5.1 Overview of the Implicit scheme

The time dependent system of the compressible Euler equations becomes very stiff at low Mach number. In very slow flow the condition number increases without bound since the smallest wave speed approaches zero. This slows down the convergence speed of explicit schemes which are typically subject to restrictive limitations on the CFL number. The convergence speed is further reduced by the CFL stability condition for high order discretizations, resulting in inefficient solution techniques for steady state solutions. Time-derivative preconditioning allows to overcome the ill-conditioning of the governing equations, such that the convergence rate of the explicit solver is strongly improved. Nevertheless, when quadratic and cubic elements have been used on triangular grid, the iteration number to reach a steady state solution, was, however, high, as shown in the previous section.

Implicit schemes do not suffer from these time stepping restrictions and significantly larger time steps can be used without hampering the stability of the time integration process. On the other hand, with the implicit schemes the computational effort per time step or iteration is significantly higher than that required by the explicit schemes. Furthermore, implicit schemes require much more memory and are significantly more difficult to implement.

Despite the capability of being stable with CFL numbers much higher than those of the explicit schemes, also implicit methods are adversely affected by the stiffness of Euler equations at low Mach number. In fact, raising the CFL number, the matrix of the linear system to be solved at each time step becomes increasingly ill-conditioned and linear iterative solvers become more and more inefficient.

5.2 Overview of Flux preconditioning technique

The fully preconditioning technique was found to be very effective to improve the convergence rate of the multistage explicit solver. The time derivative preconditioning allowed the Euler equations to converge at a rate independent of the Mach number. The characteristic based boundary conditions were used at the inlet and outlet to overcome convergence problems due to the reflections of the acoustic waves impinging on the far field boundary.

For the implicit method it can be expected that time derivative preconditioning is of minor importance because its contribution to the linear system matrix of the implicit time discretization reduces as the CFL numbers increases. Instead, flux preconditioning is important in the implicit scheme for exactly the same reasons why it was important for the explicit scheme, i.e. to improve the accuracy of the numerical flux at low Mach number. Moreover, we notice that in the implicit scheme flux preconditioning affects also the linear system matrix through the Jacobian of the residuals. As a matter of fact, the results presented below show that the linear solver becomes much more efficient and this indicates a significant improvement of the condition number of the system matrix.

Finally we remark that, lacking any time derivative preconditioning, higher order implicit schemes could directly be used to compute unsteady low Mach number flows.

5.3 Time discretization scheme

The DG space discretization of Equation (3.2) results in the following global system of equations:

$$\mathbf{M}_\Gamma \frac{d\mathbf{Q}}{dt} + \mathbf{R} = 0, \quad (5.1)$$

where \mathbf{Q} and \mathbf{R} are the global vectors of degrees of freedom (dof) and of residuals and \mathbf{M}_Γ stands for the discretization of the first integral of Equation (3.2) (with Γ in place of $\bar{\Gamma}$). Hence \mathbf{M}_Γ is a block diagonal matrix where the block corresponding to one element couples all the dof of all variables within the element (the coupling among dof of different variables is due to the action of Γ).

Using the implicit backward Euler scheme for the time discretization Equation (5.1), upon linearizing a time level n we are led to the following linear system of equations:

$$\underbrace{\left[\frac{\mathbf{M}_\Gamma}{\Delta t} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}} \right]}_{\mathbf{B}} \Delta \mathbf{Q}^n = -\mathbf{R}^n, \quad (5.2)$$

where $\Delta \mathbf{Q}^n = \mathbf{Q}^{n+1} - \mathbf{Q}^n$, $\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}}$ is the Jacobian matrix of the DG space discretization and \mathbf{B} denotes the global system matrix.

The matrix \mathbf{B} can be regarded as an $N_\kappa \times N_\kappa$ block sparse matrix where N_κ is the number of elements in T_h and the rank of each block is $M \times N_{dof}^\kappa$, where N_{dof}^κ is the number of dof for each of the M primitive variables in the generic element κ . Thanks to the DG discretization here adopted the dof of a generic element κ are only coupled with those of the neighbouring elements and the number of nonzero blocks for each (block) row κ of the matrix \mathbf{B} is therefore equal to the number of elements surrounding the element κ plus one.

The Jacobian matrix of the DG discretization has been computed analytically (except for the computation of the dissipative part of the numerical flux that have been computed numerically) without any approximation and, using very large time steps, the method can therefore achieve quadratic convergence in the computation of steady state solutions. For the backward Euler scheme and in the limit $\Delta t \rightarrow \infty$ Equation (5.2) is in fact identical to one iteration of the Newton method applied to the steady discrete problem.

To solve Equation (5.2) we can use one of the numerous methods (direct or iterative, sequential or parallel) available in the PETSc [48] library (Portable Extensible Toolkit for Scientific Computations), the software upon which the DG codes rely for the purpose of parallelization.

For the computations presented below we have used the restarted GMRES algorithm with ILU(0) preconditioning available in PETSc. The parameters required by the GMRES solver have been set as follows:

- number of search directions equal to 60 for the computations on the quadrangular grid and 140 for the solutions on the triangular mesh,
- number of restarts equal to 1 and relative tolerance to stop the iterative solution equal to 10^{-6} .

5.4 Jacobian of the preconditioned numerical flux function

In this work, we consider the preconditioned Roe's flux:

$$\mathbf{H}(\mathbf{q}^+, \mathbf{q}^-) = \frac{1}{2} \left(\mathbf{F}(\mathbf{q}^+) + \mathbf{F}(\mathbf{q}^-) - \tilde{\mathbf{F}}_{\bar{\Gamma}}(\mathbf{q}^+, \mathbf{q}^-) \right) \quad (5.3)$$

where $\tilde{\mathbf{F}}_{\bar{\Gamma}} = \bar{\Gamma} |\tilde{\mathbf{A}}_{\bar{\Gamma}}| \Delta \mathbf{q}$ is the dissipation term. For the non-preconditioned system and an ideal gas Equation (5.3) reduces to the Roe's flux-difference splitting when Roe-averaged values are used to evaluate $\bar{\Gamma} |\tilde{\mathbf{A}}_{\bar{\Gamma}}|$.

According to Equation (5.3) the Jacobian of the preconditioned Roe's flux is given by:

$$\begin{aligned} \frac{\partial \mathbf{H}}{\partial \mathbf{q}^+} &= \frac{1}{2} \left(\mathbf{A}^+ - \frac{\partial \tilde{\mathbf{F}}_{\bar{\Gamma}}}{\partial \mathbf{q}^+} \right), \\ \frac{\partial \mathbf{H}}{\partial \mathbf{q}^-} &= \frac{1}{2} \left(\mathbf{A}^- - \frac{\partial \tilde{\mathbf{F}}_{\bar{\Gamma}}}{\partial \mathbf{q}^-} \right), \end{aligned}$$

where \mathbf{A}^+ and \mathbf{A}^- are the Jacobian matrices of the interior, \mathbf{q}^+ , and exterior, \mathbf{q}^- , states on $\partial \kappa$ defined as $\frac{\partial \mathbf{F}(\mathbf{q}^+)}{\partial \mathbf{q}^+}$ and $\frac{\partial \mathbf{F}(\mathbf{q}^-)}{\partial \mathbf{q}^-}$, respectively, and $\frac{\partial \tilde{\mathbf{F}}_{\bar{\Gamma}}}{\partial \mathbf{q}^+}$ and $\frac{\partial \tilde{\mathbf{F}}_{\bar{\Gamma}}}{\partial \mathbf{q}^-}$ are the derivatives of the dissipation term $\tilde{\mathbf{F}}_{\bar{\Gamma}}$ with respect to the same states.

From Equation (4.6), we note that the dissipative part of the preconditioned flux difference-splitting scheme contains absolute value functions and thus it is non differentiable. In this work the Jacobian dissipation terms for the preconditioned and non-preconditioned versions of Roe's flux are computed by difference quotients.

5.5 Boundary Conditions

At far-field we employ the non-preconditioned boundary condition [41] as the flux preconditioning technique does not modify the time derivative of the Euler equations.

At the airfoil surface we impose the local pressure condition instead of the symmetry condition as the results obtained using the explicit scheme show that the latter is more dissipative.

At the boundary of the domain, the numerical flux function must be consistent with the boundary conditions of the problem. In practice, this is accomplished by properly defining a boundary state which accounts for the boundary data and, together with internal state, allows to compute the numerical fluxes.

The computation of the Jacobian on the boundary must account for the dependence of the external state, \mathbf{q}^b , on the boundary data and on the internal state itself, \mathbf{q}^+ .

5.6 Results

In order to demonstrate the performance of the flux preconditioning technique for the implicit DG method we look at i) the convergence of the residuals and ii) the accuracy of the solutions. Simulations have been performed at Mach numbers $M = 10^{-1}$, $M = 10^{-2}$ and $M = 10^{-3}$ using linear, quadratic and cubic elements on both the triangular and the quadrangular grid.

- As regards the convergence of the residuals we present plots of the residuals versus number of iterations and residuals versus CPU time.
The convergence history of each variable is plotted in terms of the L_2 norm of the residuals. The residuals are normalized with respect to the first residual.
- As regards the accuracy we present the plots of drag versus CPU time.
The plots for both triangular and quadrangular grids presented give some insight into the effects of flux preconditioning on both the accuracy and the computational effort.

5.6.1 Convergence

The Figure 5.1 compares the history of residuals versus the number of “Newton” iterations of Equation (5.2), computed on the quadrangular grid with and without flux preconditioning.

We remark that the graphs of Figure 5.1 merely show the effect of the fixed GMRES parameters (number of Krylov-subspace vectors, number of restarts and relative tolerance to stop iterative solution) on the convergence of the global “Newton” iterations and if these parameters are enough to ensure quadratic convergence of residuals.

At $M = 10^{-1}$ (left column) both the preconditioned and non-preconditioned DG schemes converge at about the same convergence rate almost independently of the polynomial degree. Furthermore the corresponding preconditioned and non-preconditioned residual histories decrease about the same order of magnitude. In particular, the residuals of the velocity components are indistinguishable, whereas

the preconditioned residuals of pressure and temperature decrease about one orders of magnitude less than the corresponding non-preconditioned ones.

At $M = 10^{-2}$ (middle column) and at $M = 10^{-3}$ (right column), we notice that the preconditioned scheme always displays quadratic convergence, whilst this is not the case for the non-preconditioned scheme with the same GMRES parameters. The effect is appreciable at $M = 10^{-2}$ and more evident at $M = 10^{-3}$.

The comparison between the residual decay of each variable at $M = 10^{-1}$ and at $M = 10^{-2}$ as well as at $M = 10^{-2}$ and at $M = 10^{-3}$ shows that, at the lowest Mach numbers, the residual decays of the velocity components reduce one order of magnitude with and without preconditioning, whereas the residuals of thermodynamic variables reduce one and two orders of magnitude for the non-preconditioned and the preconditioned solutions, respectively.

We conclude that, with the chosen GMRES parameters, the flux preconditioning technique allows to reduce the number of iterations needed to reach the full convergence of each variable as compared to the non-preconditioned solutions. This is due to the effect of preconditioning on the linear system matrix through the Jacobian of residuals. In particular, the full convergence of the residuals was reached in about 10 iterations independently of both Mach number and polynomial degree.

Finally, whereas all the residual decays of the non-preconditioned DG method are of $O(M)$, the preconditioned residual decays of velocity components and thermodynamic variables are of $O(M)$ and $O(M^2)$, respectively, when Mach number tends to zero, because of round-off errors.

The Figure 5.2 compares the history of residuals versus CPU time (seconds), computed on the quadrangular grid with and without flux preconditioning. Comparing Figures 5.2 and 5.1 at $M = 10^{-1}$ (left column), we observe that preconditioning improves the efficiency of GMRES solver and this can be explained again with the improved conditioning of the linear system matrix.

Similar results are found at $M = 10^{-2}$ and at $M = 10^{-3}$. The comparison with the residual histories at $M = 10^{-1}$ shows that the convergence rate reduces without preconditioning whereas preconditioned convergences are much less dependent on the Mach number.

We can conclude that, using the non-preconditioned Roe's flux, the overhead in terms of CPU time increases as the Mach number gets smaller and the polynomial degree raises.

Quadrangular grid: Residuals vs. Number of Iterations

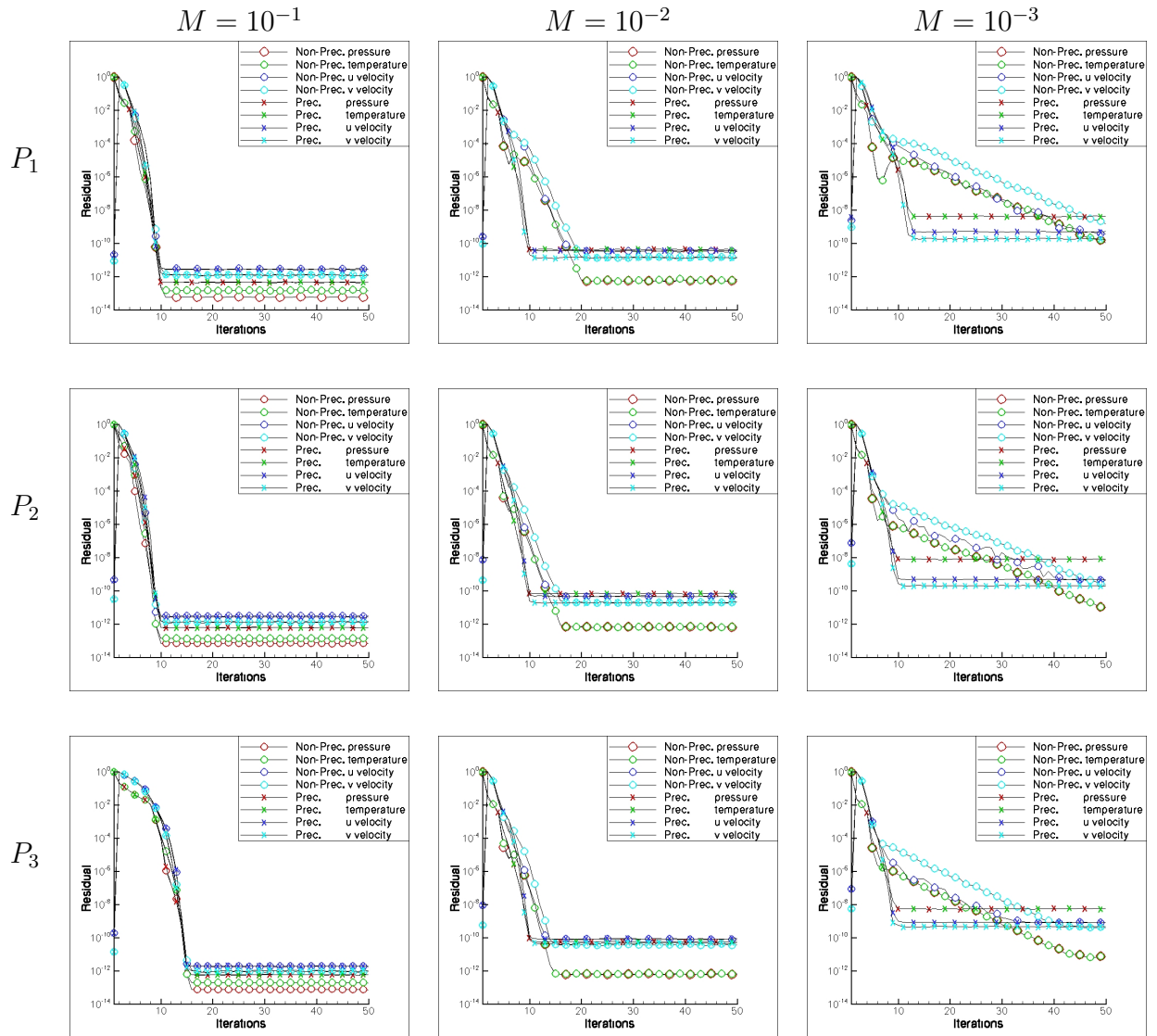


Figure 5.1: Residuals vs. number of iterations history for the quadrangular grid. $M = 0.1$ (left column), $M = 0.01$ (middle column) and $M = 0.001$ (right column). Linear elements (top row), quadratic elements (middle row) and cubic elements (bottom row).

Quadrangular grid: Residuals vs. CPU time

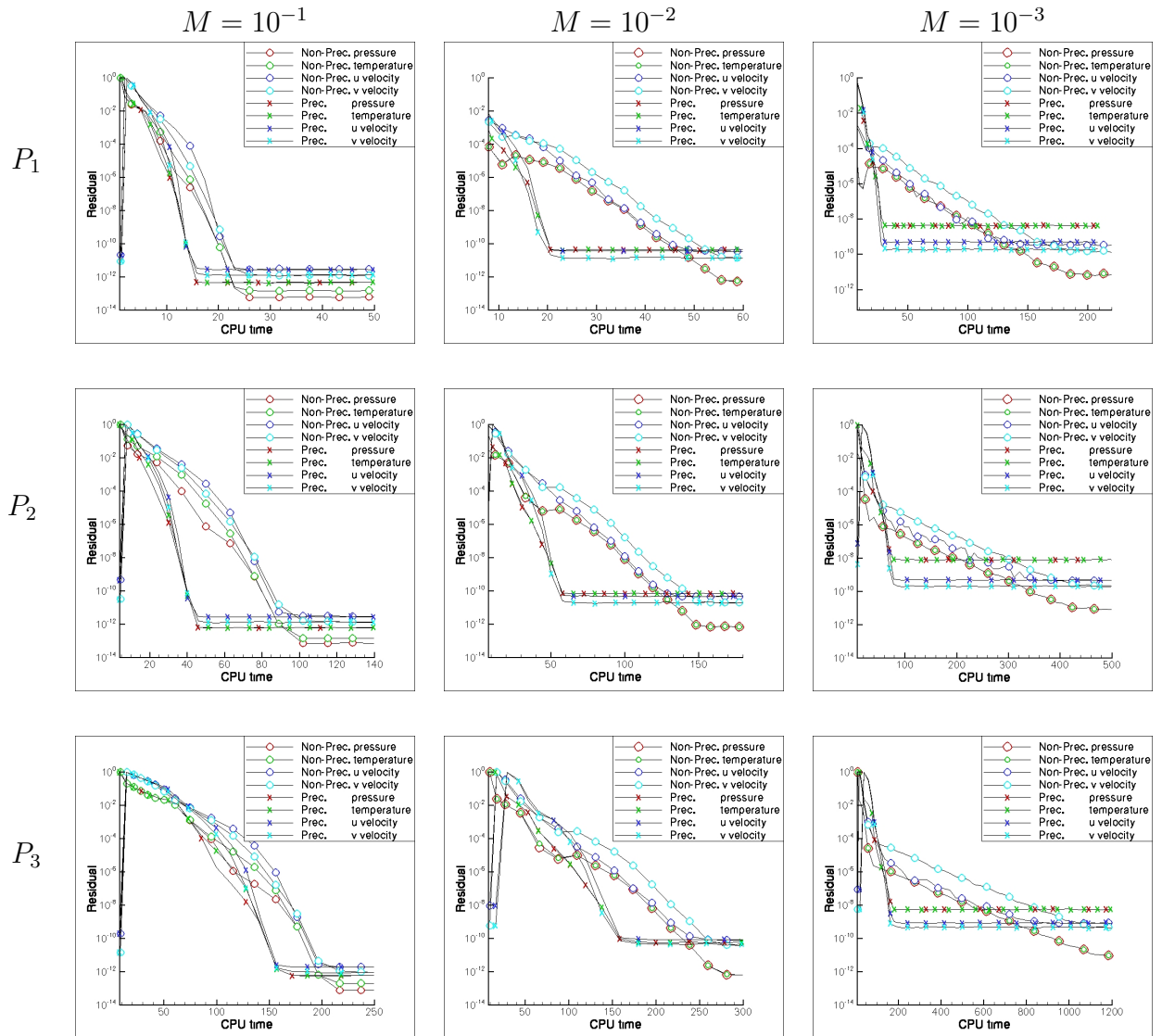


Figure 5.2: Residuals vs. CPU time history for the quadrangular grid. $M = 0.1$ (left column), $M = 0.01$ (middle column) and $M = 0.001$ (right column). Linear elements (top row), quadratic elements (middle row) and cubic elements (bottom row).

Coming to the results on the triangular grid, the Figure 5.3 compares the history of residuals versus the number of “Newton” iterations, with and without flux preconditioning.

In this case the numbers of Krylov-subspace vectors and iterations are higher than those of the computations on the quadrangular grid and ensure quadratic convergence up to $M = 10^{-2}$ for both the preconditioned and the non-preconditioned schemes. Instead, at $M = 10^{-3}$ we still have Newton convergence for the preconditioned computations but not for the non-preconditioned ones. In particular, the figure shows that with preconditioning the full convergence was reached in about 10 iterations independently of both the Mach number and the polynomial degree, in perfect agreement with the results on the quadrangular grid, see Figure 5.1. Hence, we conclude that the preconditioned DG scheme allows to obtain Newton convergence both for quadrangular and triangular element shapes.

As for the solutions on the quadrangular grid, the residuals of velocity components reduce as $O(M)$, whilst the preconditioned and non-preconditioned residuals of the thermodynamic variables reduce as $O(M^2)$ and $O(M)$, respectively, respectively, because of round-off errors.

The Figure 5.4 compares the history of residuals versus CPU time (seconds), computed on the triangular grid with and without flux preconditioning.

The general behaviour is similar to that of Figure 5.2 with the noticeable exceptions of P_1 and P_2 solutions at $M = 10^{-1}$ and perhaps a slightly lower gain of CPU time using the preconditioning technique. This effect was already observed in the solutions obtained with the explicit scheme. However, we can conclude that the flux preconditioning technique improves the efficiency of the implicit scheme at the lowest Mach numbers also for the triangular grid.

5.6.2 Accuracy

The drag coefficient is a global quantity of aerodynamic interest that can be used to monitor both the convergence and the accuracy of the solutions with and without preconditioning.

The Figures 5.5 and 5.6 show the values of drag coefficient as a function of CPU time (seconds) for the quadrangular and triangular grids, respectively.

The Figure 5.5 summarizes the influence of the Mach number and the polynomial degree on the drag coefficient. Over all, from this figure we can appreciate that the preconditioned DG solution is more accurate than the non-preconditioned one.

As expected, without preconditioning the drag coefficient at convergence increases as the Mach number reduces, whereas the drag coefficients with preconditioning are independent of the Mach number. Hence, the difference in accuracy between the preconditioned and non-preconditioned solutions increases as the Mach number tends to zero.

Increasing the polynomial degree the accuracy improves. For a given Mach number, the drag coefficient at convergence reduces as the polynomial degree increases

Triangular grid: Residuals vs. Number of Iterations

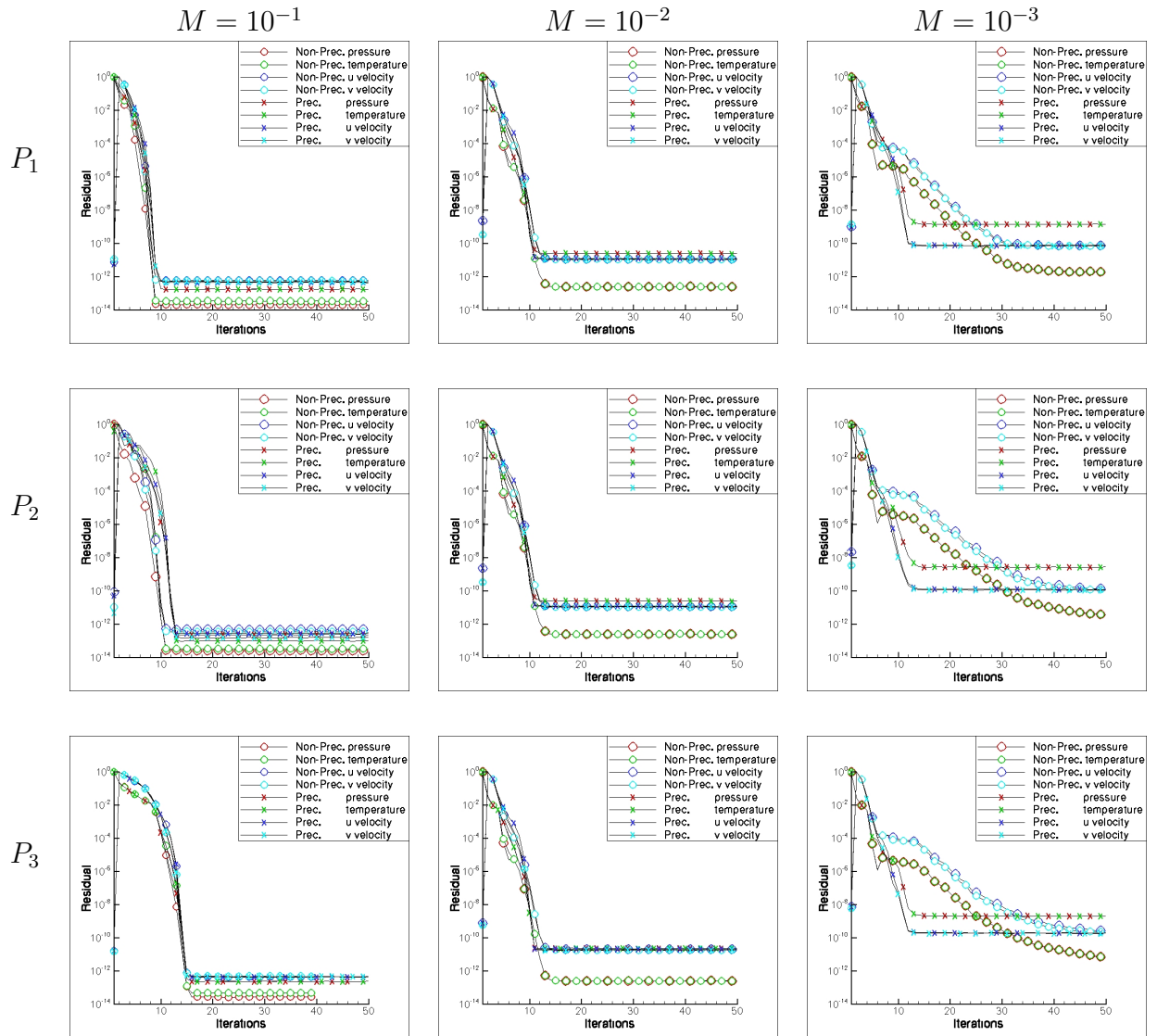


Figure 5.3: Residuals vs. number of iterations history for the triangular grid. $M = 0.1$ (left column), $M = 0.01$ (middle column) and $M = 0.001$ (right column). Linear elements (top row), quadratic elements (middle row) and cubic elements (bottom row).

Triangular grid: Residuals vs. CPU time

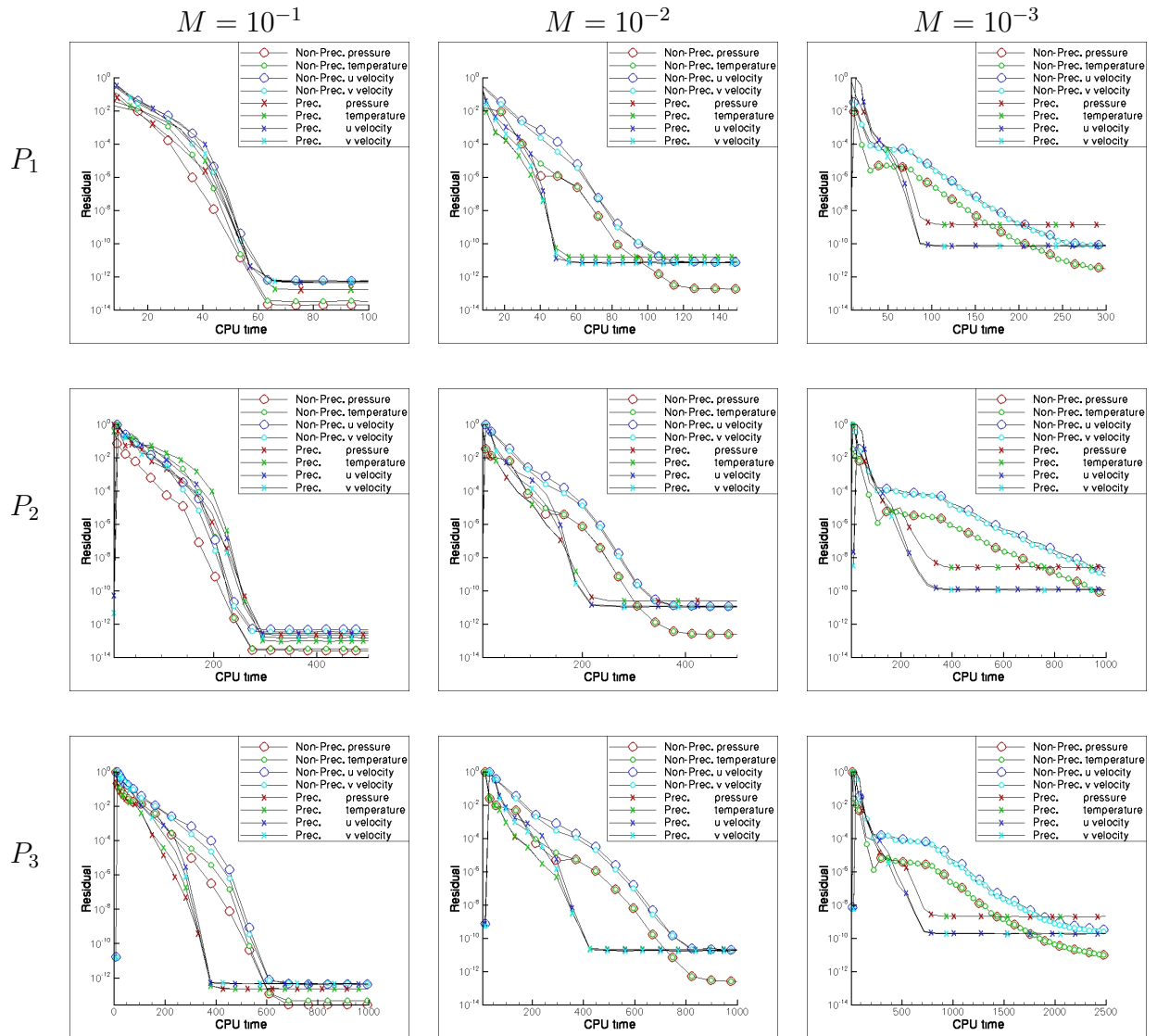


Figure 5.4: Residuals vs. CPU time history for the triangular grid. $M = 0.1$ (left column), $M = 0.01$ (middle column) and $M = 0.001$ (right column). Linear elements (top row), quadratic elements (middle row) and cubic elements (bottom row).

and the difference in accuracy between preconditioned and non-preconditioned solutions also reduces.

Both preconditioned and non-preconditioned DG solutions yield comparable drag convergence histories as long as the drag coefficients at convergence are not very different. In such cases the preconditioning allows to somewhat reduce the computational effort.

Figure 5.6 shows the value of drag coefficient as a function of CPU time for the triangular grid. The above comments about the influence of Mach number and of polynomial degree approximation apply also to the results of the DG computations on the triangular grid.

Nevertheless, it is worth noting that the DG discretization on triangular grid yields remarkably accurate solutions at low Mach number even without preconditioning. In particular, the preconditioned and non-preconditioned drag coefficients are very close to each other and in some cases are almost indistinguishable.

Notwithstanding the CPU time needed for the convergence of drag coefficients using the preconditioned algorithm is lower than that without preconditioning. In some cases, however, the difference is negligible.

The Tables 5.1, 5.2 and 5.3 summarize the results obtained on the quadrangular and the triangular grids with and without preconditioning. For the three Mach numbers considered and P_1 , P_2 and P_3 elements, the CPU time and the drag coefficients at convergence are presented.

The tables demonstrate that the flux preconditioning technique improves the accuracy of solutions, especially for the quadrangular grid.

The comparison between preconditioned and non-preconditioned values of CPU time on the triangular grid confirms that the flux preconditioning technique reduces the computational effort needed to reach the convergence of the drag coefficient. This effect is not so evident on the quadrangular grid due to the inaccuracy of the non-preconditioned solutions at the lowest Mach numbers when simulations are performed using P_1 and P_2 elements.

However, it is worth noting that the values of CPU time obtained on quadrangular and triangular grids with preconditioning are independent of the Mach number (except of the P_3 solution at $M = 10^{-3}$ on the triangular grid and the P_2 solution at $M = 10^{-3}$ on the quadrangular grid).

As a final comment, we remark that on both grids the computational effort for the convergence of the drag coefficient is significantly lower than the CPU time required for the full convergence of residuals, see Figures 5.2 and 5.4. In this respect, we observe that, according to Lee [21], the minimum CPU time required for the convergence of the drag coefficients nearly corresponds to a decay of 5 orders of magnitude of the residuals.

Quadrangular grid: drag vs. CPU time

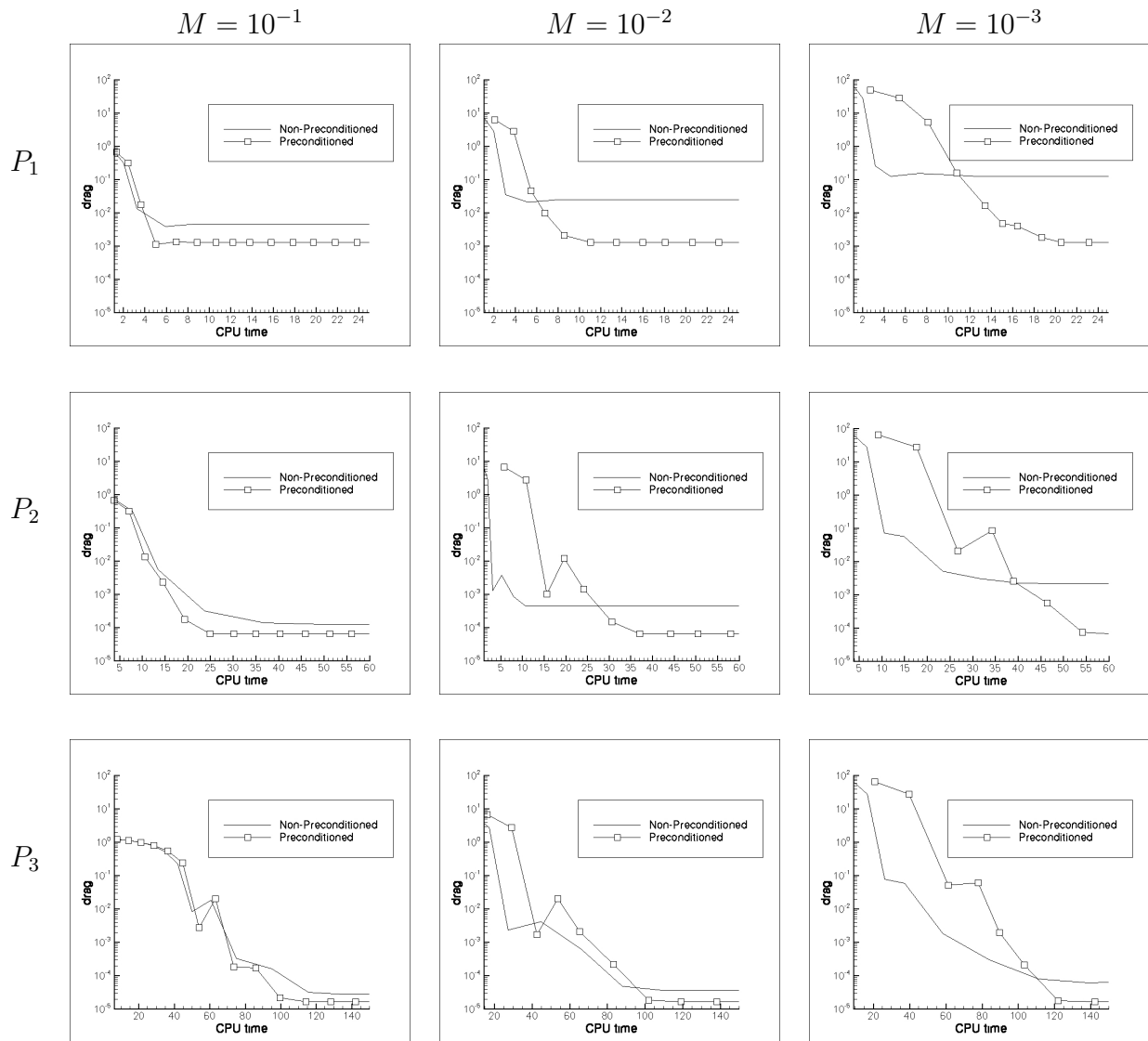


Figure 5.5: drag vs. CPU time history for the quadrangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). Linear elements (top row), quadratic elements (middle row) and cubic elements (bottom row).

Triangular grid: drag vs. CPU time

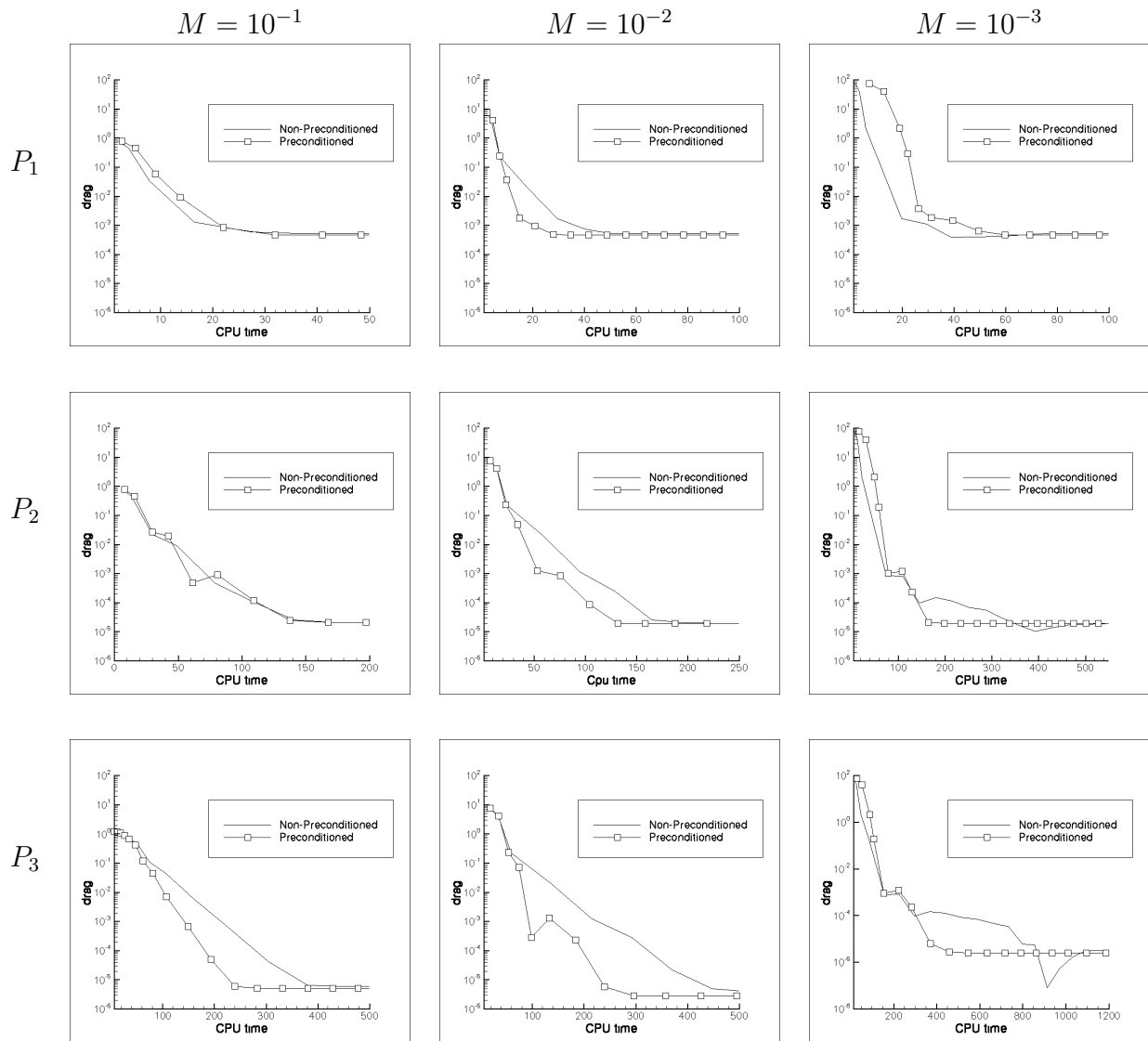


Figure 5.6: drag vs. CPU time history for the triangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). linear Elements (top row), quadratic elements (middle row) and cubic elements (bottom row).

$M = 10^{-1}$								
Quadrangular grid					Triangular grid			
Preconditioned			Non-Preconditioned		Preconditioned		Non-Preconditioned	
CPU	Drag		CPU	Drag	CPU	Drag	CPU	Drag
P_1	10	$1.301 \cdot 10^{-3}$	10	$4.667 \cdot 10^{-3}$	35	$4.716 \cdot 10^{-4}$	40	$5.295 \cdot 10^{-4}$
P_2	30	$6.620 \cdot 10^{-5}$	50	$1.280 \cdot 10^{-4}$	170	$2.157 \cdot 10^{-5}$	170	$2.177 \cdot 10^{-5}$
P_3	120	$1.659 \cdot 10^{-5}$	140	$2.766 \cdot 10^{-5}$	300	$5.266 \cdot 10^{-6}$	450	$6.149 \cdot 10^{-6}$

Table 5.1: CPU time and drag coefficients with and without flux preconditioning technique at $M = 10^{-1}$.

$M = 10^{-2}$								
Quadrangular grid					Triangular grid			
Preconditioned			Non-Preconditioned		Preconditioned		Non-Preconditioned	
CPU	Drag		CPU	Drag	CPU	Drag	CPU	Drag
P_1	12	$1.300 \cdot 10^{-3}$	10	$2.495 \cdot 10^{-2}$	35	$4.689 \cdot 10^{-4}$	60	$5.419 \cdot 10^{-4}$
P_2	37	$6.640 \cdot 10^{-5}$	12	$4.540 \cdot 10^{-4}$	170	$1.966 \cdot 10^{-5}$	225	$2.017 \cdot 10^{-5}$
P_3	120	$1.663 \cdot 10^{-5}$	120	$3.760 \cdot 10^{-5}$	300	$2.863 \cdot 10^{-6}$	500	$3.971 \cdot 10^{-6}$

Table 5.2: CPU time and drag coefficients with and without flux preconditioning technique at $M = 10^{-2}$.

$M = 10^{-3}$								
Quadrangular grid					Triangular grid			
Preconditioned			Non-Preconditioned		Preconditioned		Non-Preconditioned	
CPU	Drag		CPU	Drag	CPU	Drag	CPU	Drag
P_1	11	$1.300 \cdot 10^{-3}$	14	$1.271 \cdot 10^{-1}$	60	$4.686 \cdot 10^{-4}$	80	$5.434 \cdot 10^{-4}$
P_2	55	$6.640 \cdot 10^{-5}$	50	$2.225 \cdot 10^{-3}$	170	$1.946 \cdot 10^{-5}$	500	$2.001 \cdot 10^{-5}$
P_3	120	$1.663 \cdot 10^{-5}$	150	$6.809 \cdot 10^{-5}$	500	$2.609 \cdot 10^{-6}$	1200	$3.743 \cdot 10^{-6}$

Table 5.3: CPU time and drag coefficients with and without flux preconditioning technique at $M = 10^{-3}$.

6 Cancellation problem

The preconditioned Euler equations have a serious convergence problem at very low Mach numbers and there is a Mach number limit below which converged solutions could not be obtained [21]. This is attributed to cancellation errors that occur due to accumulation effects of round-off errors. Round-off errors are mainly determined by the precision of the floating-point variables and are thus inevitable [49]. However, cancellation errors can be avoided to a certain extent by a proper manipulation of the independent variables of the fluid: the so-called splitting technique [50].

6.1 Behaviour of governing equations at low Mach numbers

It is difficult or impossible to obtain a fully converged temperature field at a Mach number below 10^{-6} , while the pressure field and velocity field can be obtained at a much lower Mach number. This is due to the fact that the cancellation errors in the energy equation grow faster than those in the continuity equation and the momentum equation, as the Mach number decreases. The calculation with an approximate jacobian shows that the off-diagonal element related to the pressure change magnifies the round-off errors and prevents the energy equation from converging [49]. The problem of the cancellation error can be minimized formulating the governing equations in terms of perturbation variables [50, 51].

Reference quantities are introduced in the equations for the thermodynamic variables and the computations are performed for the fluctuations. The governing equations are unaltered and the method can be used in conjunction with standard numerical strategies, like preconditioning. Some previous works used the perturbation analysis to accurately compute low Mach numbers flows [28, 30, 52, 53, 54, 55]. Recently, S. H. Lee [21] analyzed the relationship between the relative treatments of variables and convergence rate for the preconditioned Euler equations and reported that, in conjunction with perturbed analysis, a higher precision of floating-point variables plays a significant role in reducing the cancellation problem.

6.2 Round-off error and relative treatment of the variables

When solving the compressible Euler and Navier-Stokes equations for low Mach numbers flows, the variations of thermodynamic quantities (like pressure p , temperature T and density ρ) become small with respect to their stagnation values. The consequence is that round-off errors occur during numerical computations. Sesterhenn *et al.* [50] showed that this arises from the calculation of the pressure gradient, Δp , and that the round-off error increases proportional to the square of Mach number. Furthermore, they illustrated that the round-off error not only occurs in the momentum equation but also affects the energy equation, due to the contribution of the kinetic energy to the total energy. In particular this is true also if one adopts primitive formulations of the governing equations with, for example, the temperature as the principle variable. Thus the cancellation error which occurs as an accumulation effect of round-off errors plays a significant role in computing low Mach number flows. Several previous studies [29, 30, 52, 53, 54, 55] show that this problem can be alleviated by employing the concept of gauge pressure, in which the pressure is decomposed into a constant reference pressure and a relative pressure. Sesterhenn *et al.* [50] extended the relative treatment to all variables and flux vectors. Nevertheless Lee findings [21] showed that this approach produced a slight improvement in the convergence process of the energy equation only while

the precision of floating-point variables was a much more important factor in the calculations of the temperature field at very low Mach numbers.

In this work the relative thermodynamic variables p' and T' are defined as,

$$\begin{aligned} p' &= p - p_\infty, \\ T' &= T - T_\infty, \end{aligned} \quad (6.1)$$

where p_∞ and T_∞ are the freestream pressure and temperature, respectively. Furthermore, the momentum fluxes are defined considering the relative pressure p' . Then the primitive variables \mathbf{q} and the cartesian components \mathbf{f} and \mathbf{g} of the convective flux function \mathbf{F} are redefined as follow:

$$\mathbf{q} = \begin{pmatrix} p' \\ u \\ v \\ T' \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u u + p' \\ \rho u v \\ \rho H u \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v v + p' \\ \rho H v \end{pmatrix}. \quad (6.2)$$

The relative formulation of the preconditioned governing equations obtained using Equations (6.2) are mathematically equivalent to the original ones, see (4.1). In particular, the preconditioning matrix (4.2) is not modified and the ideal gas law is maintained.

6.3 Results

To investigate the influence of the splitting technique on the accuracy of the DG method applied to inviscid low Mach number flows, we present the numerical results obtained by applying the Fully Preconditioning technique to the Euler equations with and without splitting using the explicit scheme. To this end, we perform computations at different very low Mach numbers, using P_1 , P_2 and P_3 elements, on the triangular grid.

The discussion of the results obtained is split in two different sections, in order to highlight separately two different aspects: the convergence and the accuracy of the solutions.

- Convergence.

The residual histories versus iteration number were computed to evaluate the effect of the splitting technique on the convergence of the solution process. The iteration history is measured in terms of the L_2 norm of the residuals. The residual history is shown separately for each governing equations to evaluate the effect of the splitting technique on each convergence characteristic of the preconditioned Euler equations (pressure p , temperature T , horizontal and vertical velocity components u and v , respectively).

In all figures the residual values are normalized such that the first residual equals 1.

- Accuracy

Concerning the accuracy contours of the normalized pressure, temperature and absolute values of velocity, with and without splitting technique at two very low Mach numbers $M = 10^{-5}$ and $M = 10^{-6}$ are shown. In fact the cancellation problem becomes evident at different Mach numbers depending on the variable considered and on the numerical accuracy desired.

6.3.1 Convergence

In this section the convergence histories of the preconditioned scheme at different very low Mach numbers and for different polynomial degrees with and without splitting technique are presented. Figure 6.1 shows the convergence history of pressure at $M = 10^{-2}$, $M = 10^{-4}$ and $M = 10^{-6}$ for linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements, comparing the results obtained with (left) and without (right) splitting technique.

In Figure 6.1 the influence of the splitting technique on the reduction of the pressure residual can be clearly seen. Examining the left column of Figure 6.1, that refers to the solutions without splitting technique, we can make two observations. The first is that, for a given polynomial degree, the lower the Mach number, the smaller the reduction of the pressure residual. The second is that, for a given Mach number, the higher the polynomial degree, the smaller the reduction of the pressure residual, even if this influence is less evident than the first.

However, both the influences of Mach number and polynomial degree on the decrease of the residual have not allowed to obtain a pressure field at the lowest Mach number $M = 10^{-6}$ using the highest polynomial degree P_3 .

Examining now the right column of the same figure, that refers to the solutions with splitting technique, we see that the reduction of the pressure residual is now independent of the Mach number. The dependency of the residual reduction on the polynomial degree is the same previously observed without the relative treatment of variables.

Different is the case of the temperature, see Figure (6.2). Here we see that even if we use the splitting technique, the residuals of temperature reduce less as compared to the residuals of pressure because they stagnate at a level closer to the starting value. In particular, the decay of the temperature residual strongly reduces when Mach number goes to zero. The reason of this behaviour can be found in the order of magnitude of the convective vector as the Mach number goes to zero. Due to the reference values adopted in this work to non-dimensionalize the governing equations, the orders of magnitude of the non-dimensionalized quantities are as follows:

$$u, v \sim O(M), \quad \rho, p, T, H \sim O(1), \quad p' \sim O(M^2).$$

Then the convection vectors in the x and y direction (6.2) can be expressed as follows:

$$\mathbf{f}, \mathbf{g} \sim \begin{pmatrix} O(M) \\ O(M^2) \\ O(M^2) \\ O(M) + O(M^3) \end{pmatrix}.$$

We see that the range of the order of magnitude in the energy equation is wider than that in the other equations. Thereby, the temperature suffers more from the cancellation problem than the other variables, due to the contribution of the kinetic energy to the total enthalpy [50]. We encountered serious problems in calculating the temperature field at $M = 10^{-6}$ using P_3 elements. In particular, it was not possible to perform computations without using the relative treatment of the variables.

Finally, Figures (6.3) and (6.4) compare the decays of residuals of the u and v velocity components at the Mach numbers $M = 10^{-2}$, 10^{-4} and 10^{-6} . The normalized residuals of both velocity components exactly coincide using the splitting technique, whereas there is a clear influence of the Mach number on the residual decay for the computations performed without splitting. Figures (6.3) and (6.4) also show the influence of the polynomial degree on the reduction of the residuals computed with and without splitting.

We note that, using the splitting technique, while the residual decays of pressure, u and v velocity components are sufficient enough to obtain accurate flow variable distributions, the residual reduction of the temperature does not always allow to obtain a fully converged temperature field.

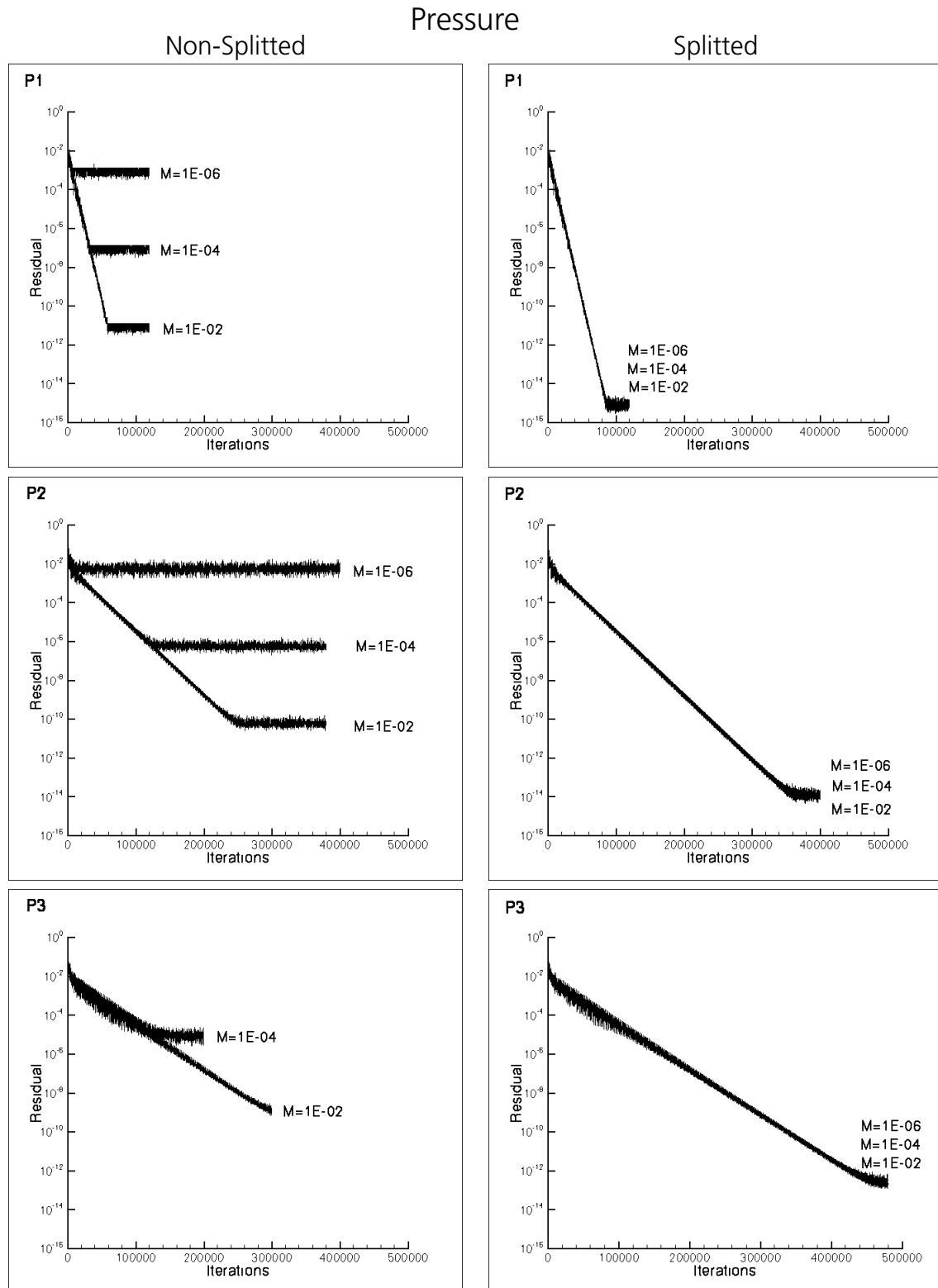


Figure 6.1: Convergence history of the pressure for $M = 10^{-2}$, $M = 10^{-4}$ and $M = 10^{-6}$ for linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements, with (left) and without (right) splitting technique.

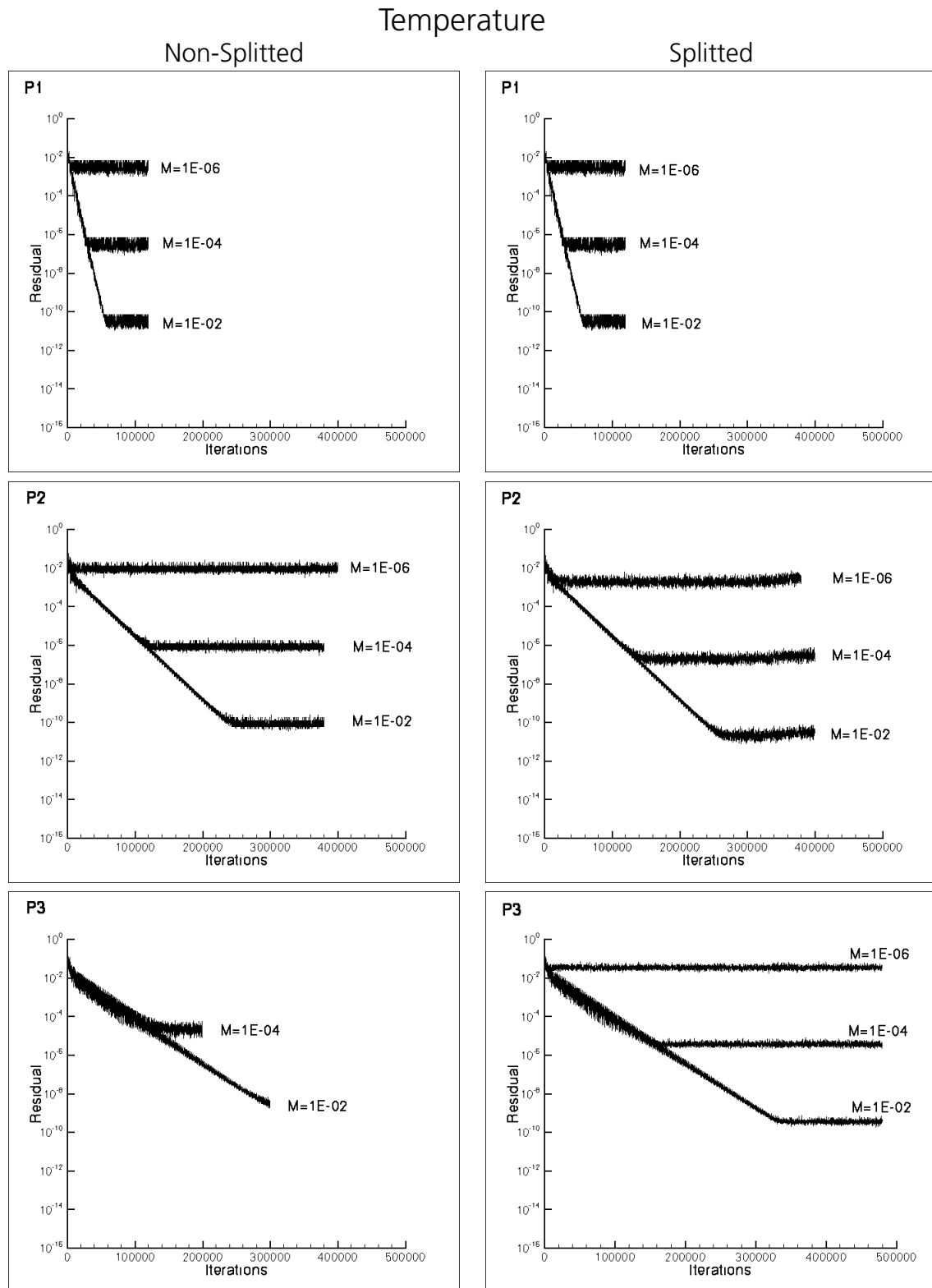


Figure 6.2: Convergence history of the temperature for $M = 10^{-2}$, $M = 10^{-4}$ and $M = 10^{-6}$ for linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements, with (left) and without (right) splitting technique.

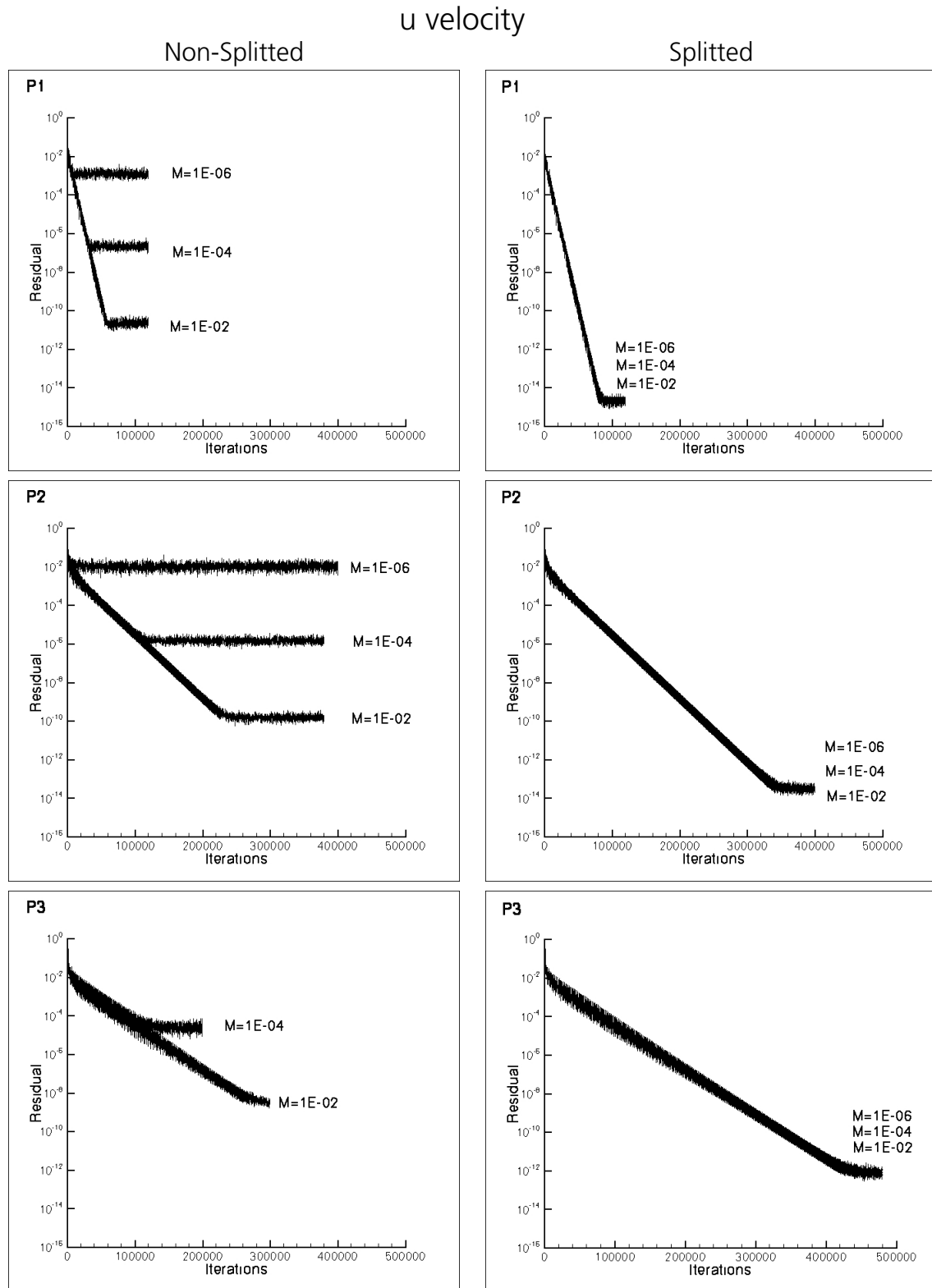


Figure 6.3: Convergence history of the horizontal velocity component for $M = 10^{-2}$, $M = 10^{-4}$ and $M = 10^{-6}$ for linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements, with (left) and without (right) splitting technique.

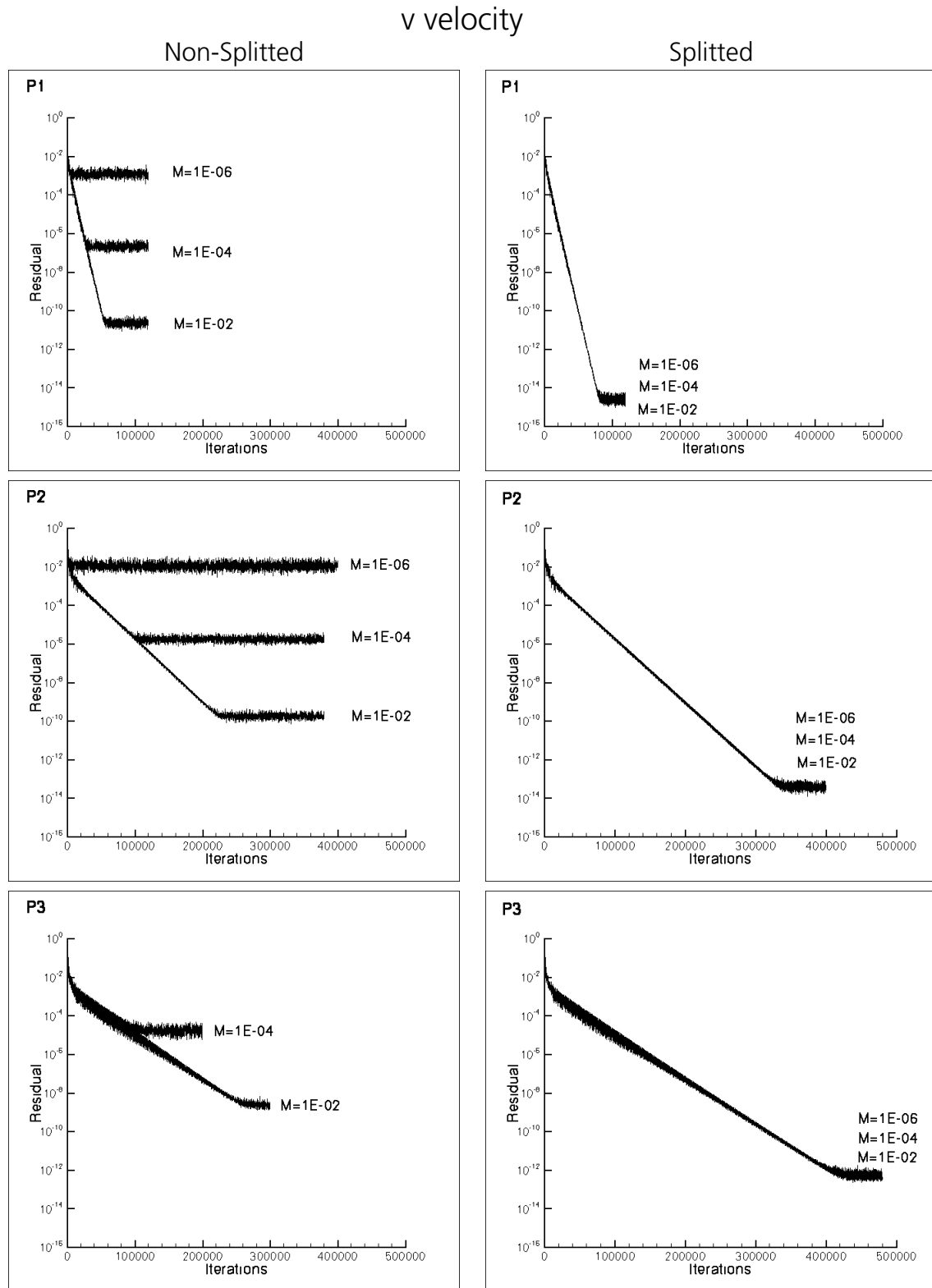


Figure 6.4: Convergence history of the vertical velocity component for $M = 10^{-2}$, $M = 10^{-4}$ and $M = 10^{-6}$ for linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements, with (left) and without (right) splitting technique.

6.3.2 Accuracy

In this section the accuracy of the preconditioned solutions for different Mach numbers and different polynomial degrees with and without splitting technique is analyzed. Figures 6.5, 6.6 and 6.7 show the normalized contours of the pressure, temperature and velocity vector with and without splitting technique at $M = 10^{-5}$ using P_1 , P_2 and P_3 elements. We see that on the basis of normalized pressure and absolute value of velocity isolines there are no differences between the splitted and the non-splitted solutions, whereas isolines of temperature begin to deteriorate using P_3 elements without splitting.

The solutions at $M = 10^{-6}$, see Figures 6.8, 6.9 and 6.10, show more clearly how the splitting technique improves the numerical accuracy in the low Mach number limit. Here, the P_1 solutions obtained without splitting exhibit numerical oscillations, and the results worse as the polynomial degree increases. This is due to the higher number of computations performed when the higher order approximations are used. In other words, the larger the number of computations with rounding errors occurring at each computation, the worse the solution. In fact, like for the P_3 solution at $M = 10^{-6}$ it was not possible to obtain a converged solution for lower Mach number, regardless of the polynomial degree.

From these results we see that the relative treatment of variables is fundamental to obtain convergence of continuity and momentum equations at very low Mach numbers, although the energy equation still does not converge.

Nevertheless the splitting technique allows to obtain accurate pressure and velocity isolines even for extremely low Mach numbers, $M = 10^{-15}$, see Figure 6.11, independently of the accuracy of numerical solution, thus extending the DG scheme to the incompressible limit.

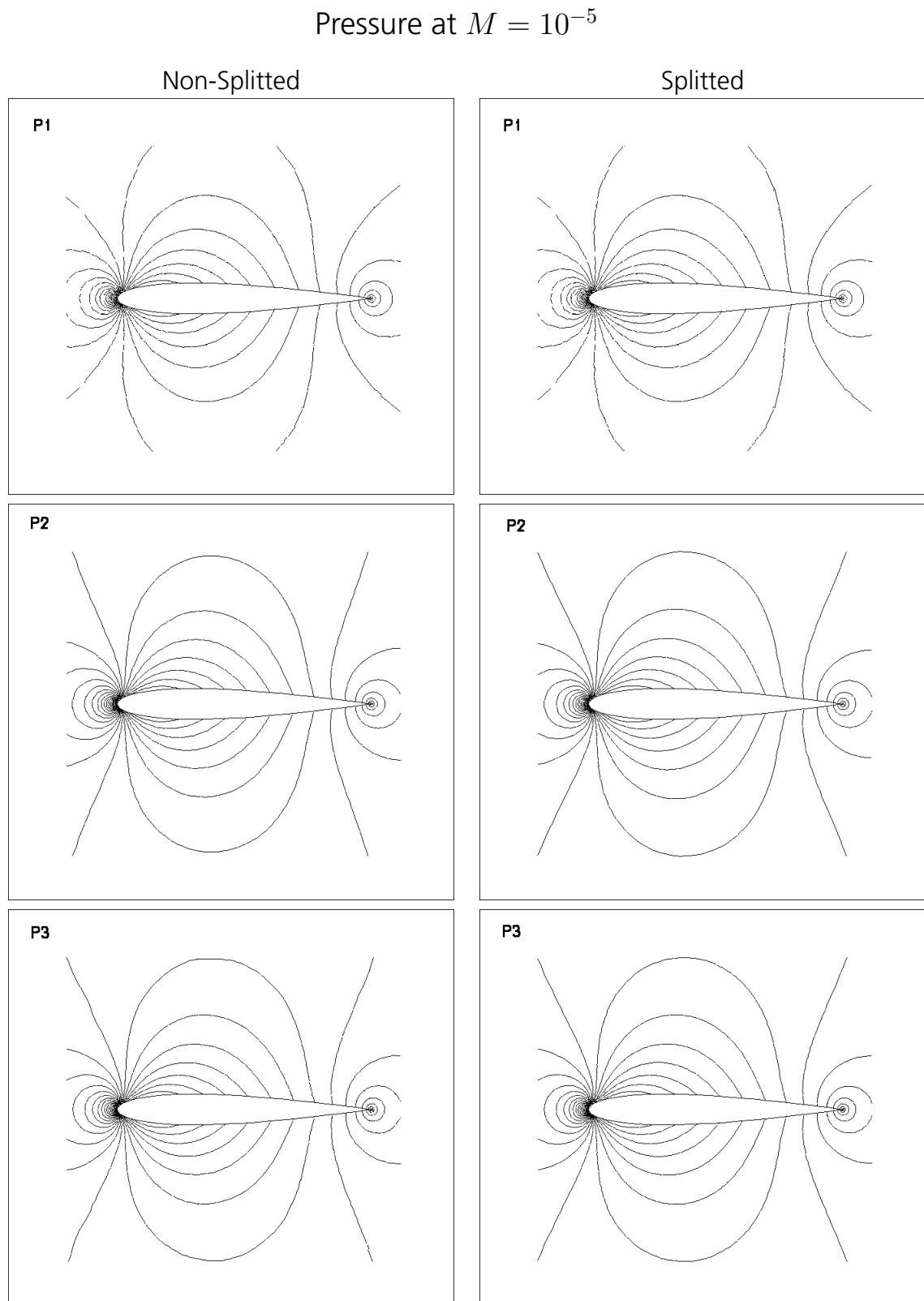


Figure 6.5: Contours of normalized pressure for $M = 10^{-5}$. Non-splitted (left) and splitted (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

Temperature at $M = 10^{-5}$

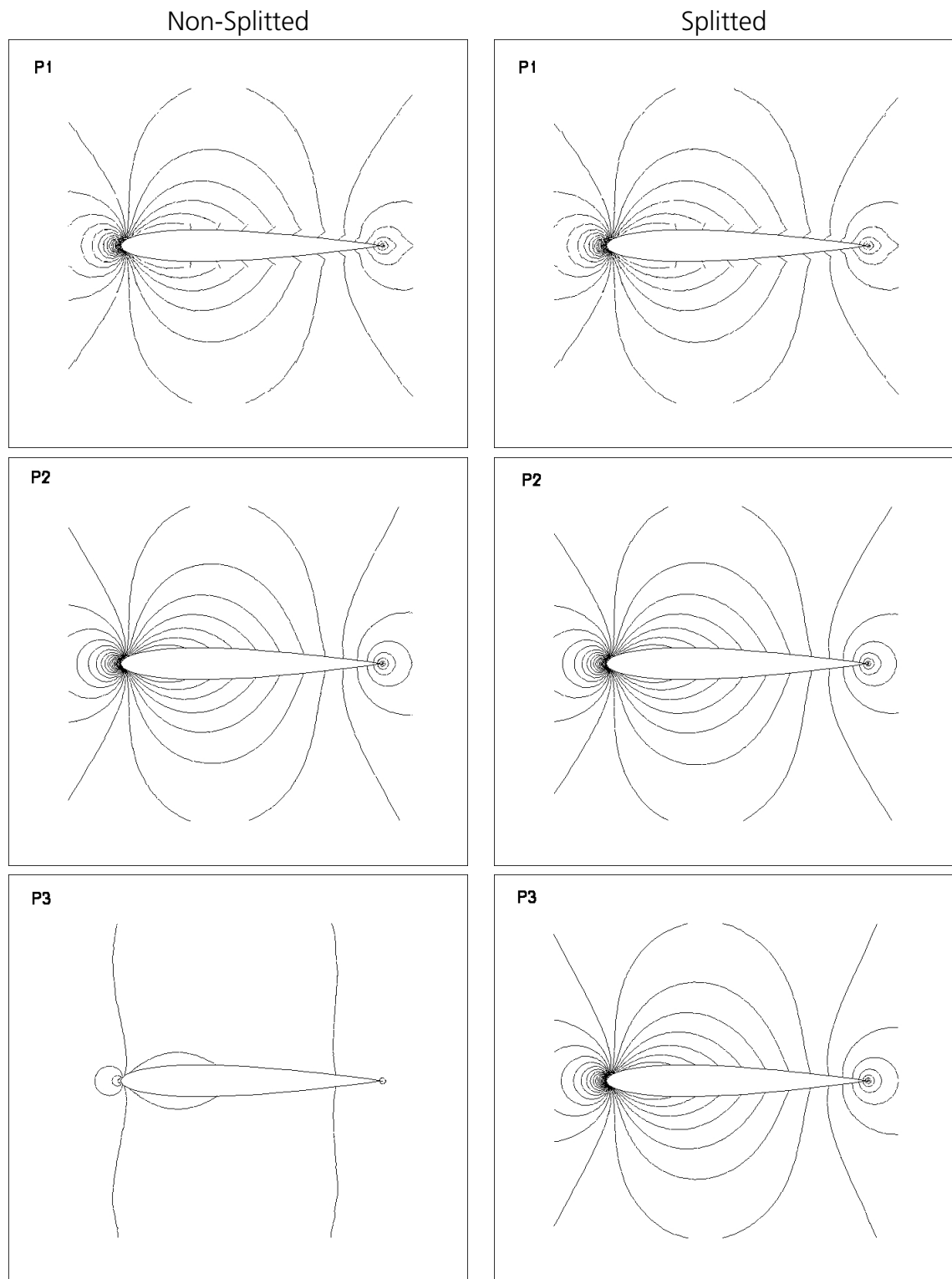


Figure 6.6: Contours of normalized temperature for $M = 10^{-5}$. Non-splitted (left) and splitted (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

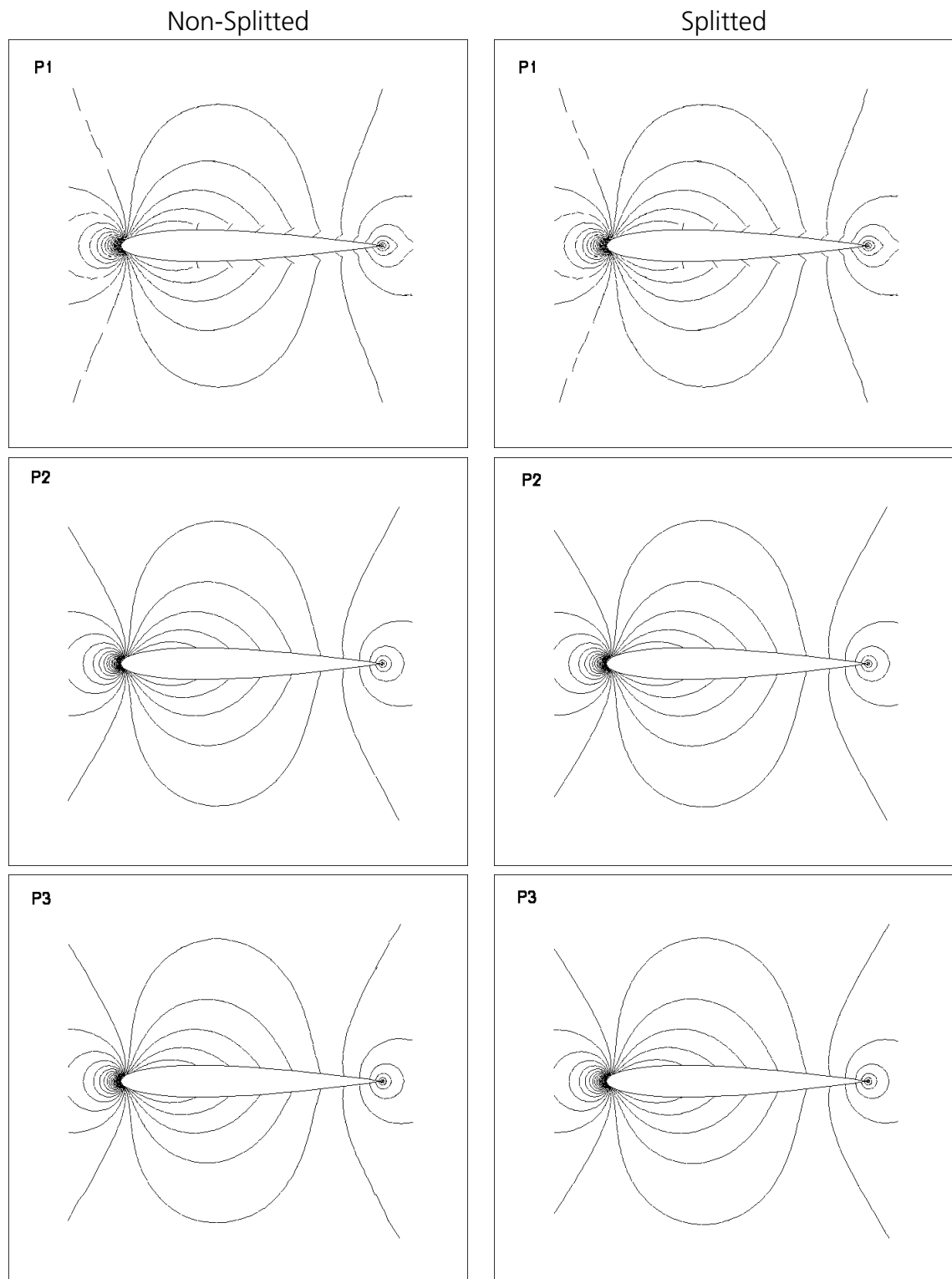
Velocity at $M = 10^{-5}$ 

Figure 6.7: Contours of normalized velocity for $M = 10^{-5}$. Non-splitted (left) and splitted (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

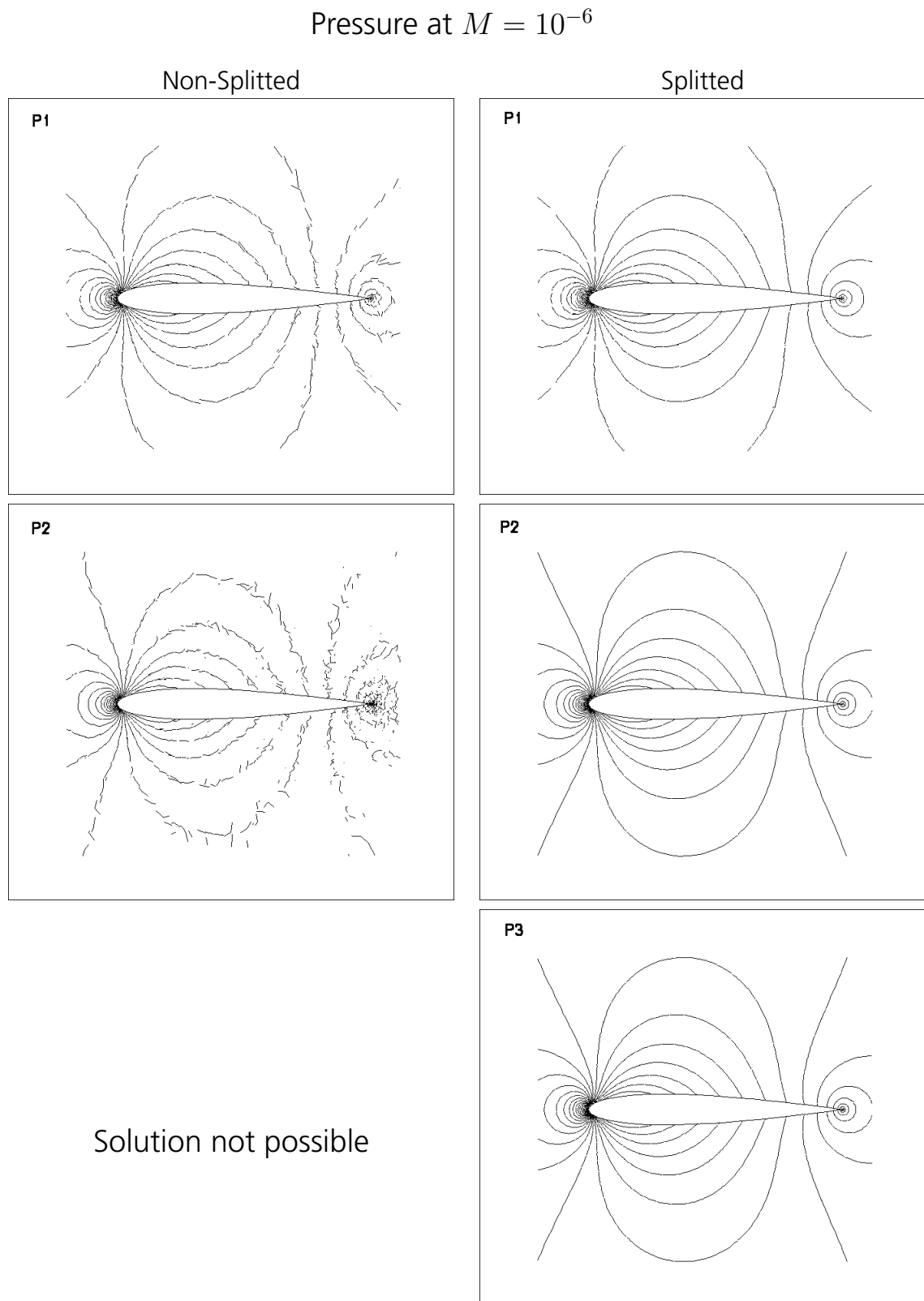


Figure 6.8: Contours of normalized pressure for $M = 10^{-6}$. Non-splitted (left) and splitted (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

Temperature at $M = 10^{-6}$

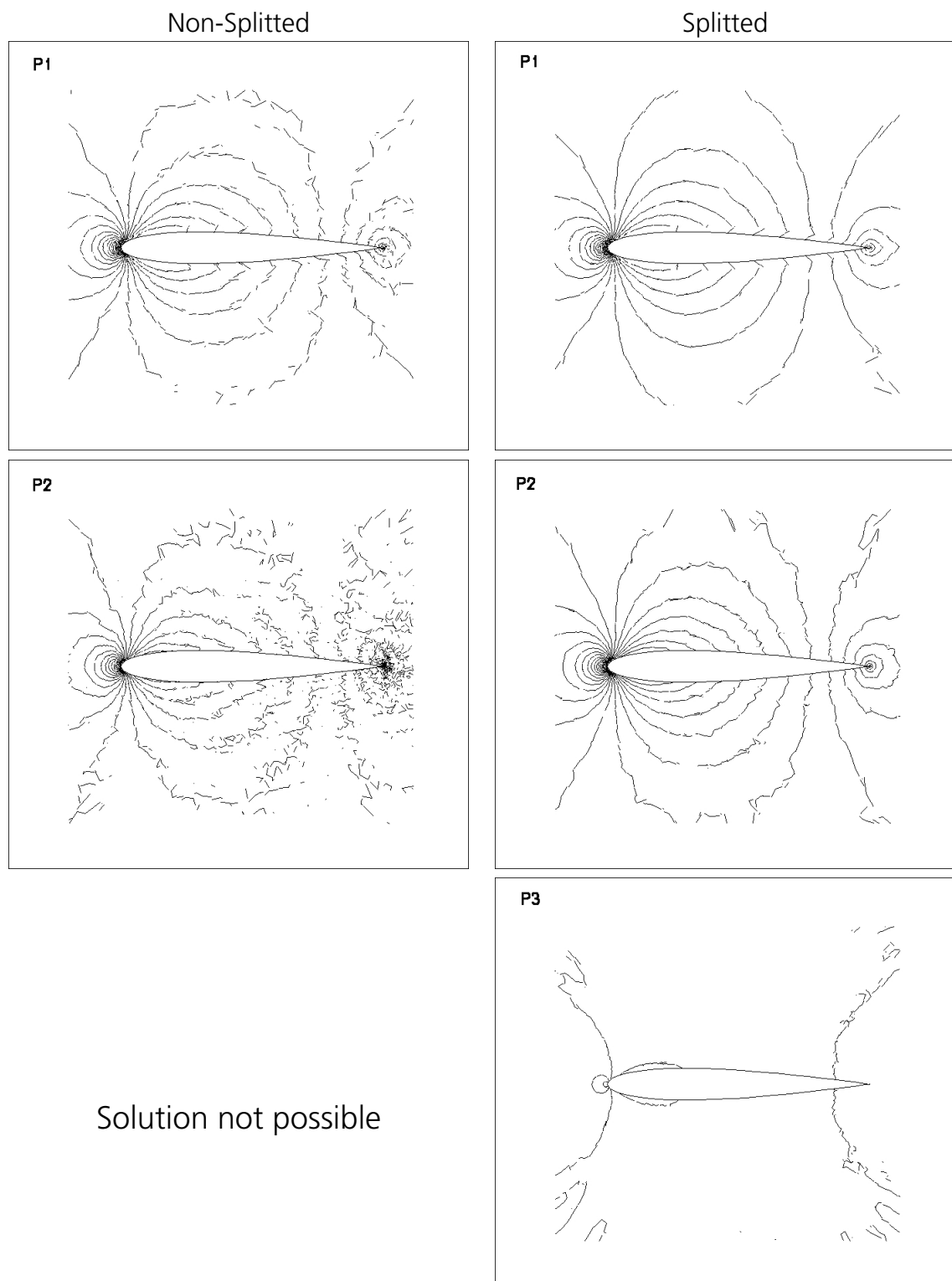


Figure 6.9: Contours of normalized temperature for $M = 10^{-6}$. Non-splitted (left) and splitted (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

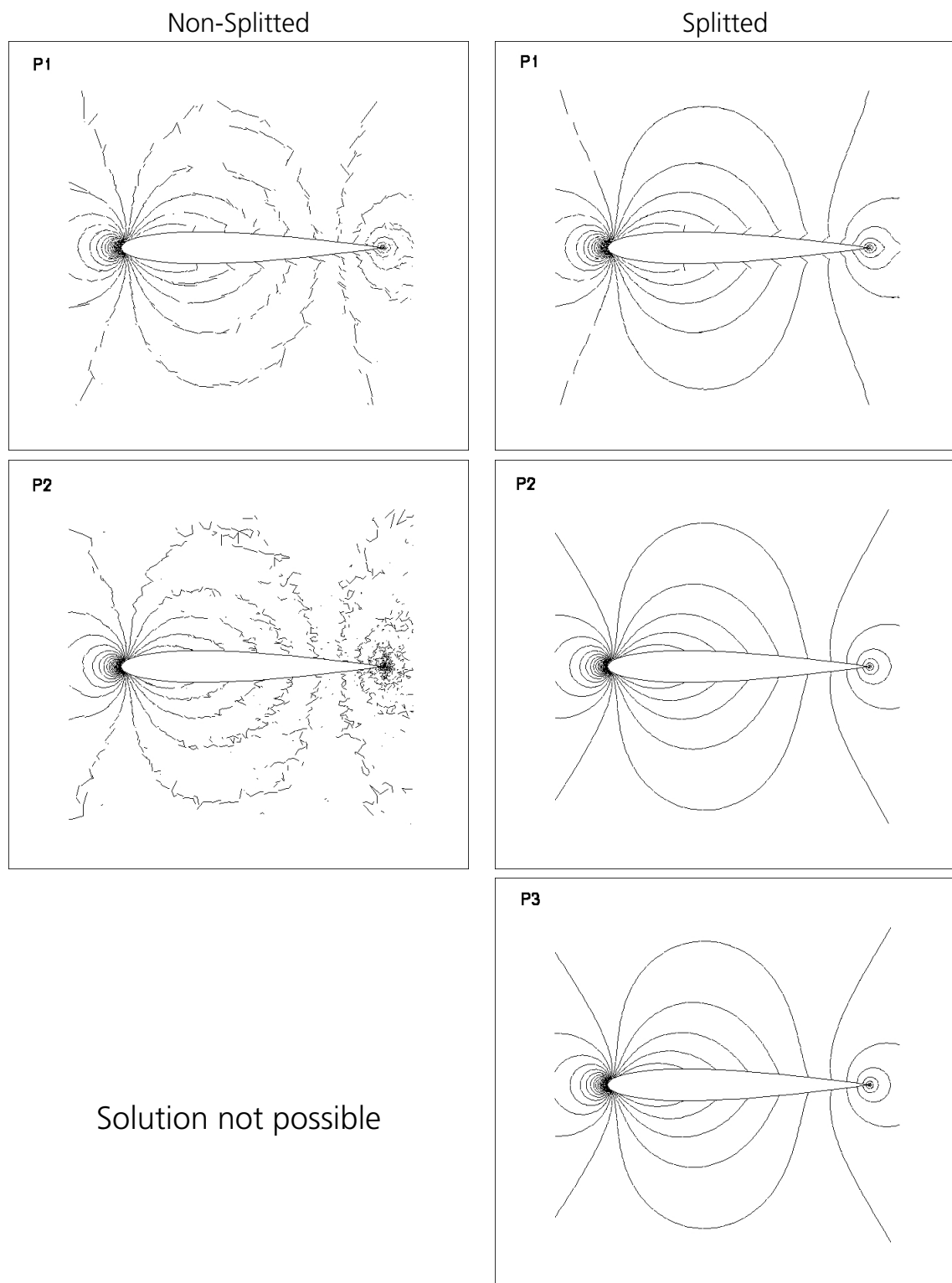
Velocity at $M = 10^{-6}$ 

Figure 6.10: Contours of normalized velocity for $M = 10^{-6}$. Non-splitted (left) and splitted (right). Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

Splitting technique: $M = 10^{-15}$

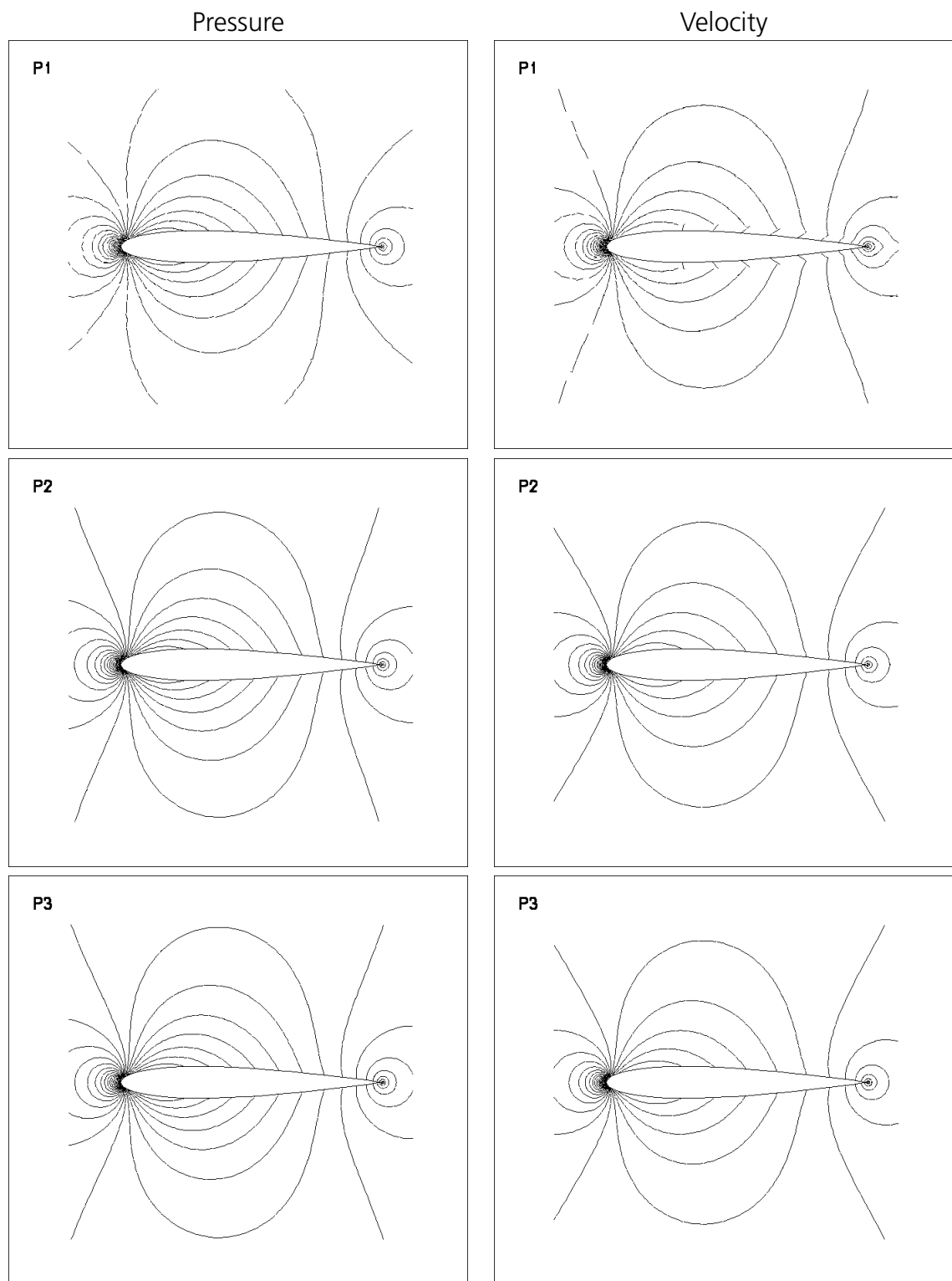


Figure 6.11: Contours of normalized pressure (righth column) and velocity (left column) for $M = 10^{-15}$ with splitting technique. Linear (P_1 top), quadratic (P_2 middle) and cubic (P_3 bottom) elements.

Conclusions

The goal of this research was to give a contribution to the development of high order accurate Discontinuous Galerkin finite element method (DG) for compressible flows at all speeds. In particular, in this work we extended the DG method to low Mach number flows, in order to obtain a significant augmentation of the overall numerical performance of the scheme when computations are performed to predict in this very challenging flow regime.

Numerical studies were performed to asses the potentiality of the DG method in solving inviscid low Mach number flows. Furthermore, two different low Mach number preconditioning techniques were applied to the compressible Euler equations, expressed in terms of primitive variables, for both the explicit and implicit time discretization schemes, in order to improve the efficiency and the accuracy of the numerical scheme. Finally the DG method was extended to the incompressible limit formulating the governing equations in terms of perturbation variables.

Computations were performed at different low Mach numbers using linear, quadratic and cubic elements on quadrangular and triangular grids in order to analyze the influence of the polynomial degree and spatial discretization (triangular or quadrangular) on the convergence rate and numerical accuracy, when the Mach number tends to zero. The method yielding a minimal amount of dissipation has proven to be very effective in the solution of a classical two-dimensional test case for the Euler equations in the low Mach number limit.

The results show that accurate solutions on relatively coarse meshes can be computed by using a high-order representation of the unknowns and of the geometry of the boundary. In particular, very accurate solutions were obtained on the triangular grid up to $M = 10^{-3}$ using quadratic and cubic elements, while cubic elements allowed to preserve the accuracy on the quadrangular grid up to $M = 10^{-2}$ only, highlighting the influence of the spatial discretization on the lack of accuracy exhibit by the upwind schemes at low Mach numbers.

Concerning the convergence rate, the explicit solver was very inefficient for low Mach number computations as the stability conditions strongly restrict the time step size.

The implementation of a full preconditioning of the Euler equations for the explicit scheme allowed to overcome the stiffness problem. This technique improves both the convergence rate and the accuracy, preconditioning the time-derivative of the governing equations and modifying accordingly the numerical flux function, respectively. Because time-derivative preconditioning destroys the time accuracy of the system, this algorithm can be used to solve steady flows only. Inviscid flow computations around a NACA 0012 profile at zero angle of attack suggest that the preconditioning always improves the accuracy of the numerical solution as compared to the non-preconditioned scheme. This effect is particularly visible for the results on the quadrangular grid. Furthermore, the efficiency of the explicit solver is improved with preconditioning that gives convergence rates independent of the Mach number. Finally, two different wall conditions have been implemented to evaluate the influence of the wall treatment on the prediction of drag coefficients.

The flux preconditioning of the Euler equations for the implicit scheme was implemented to overcome the CFL condition which has a strong impact on the efficiency of the explicit scheme previously used for high order polynomial approximations. This approach modifies the numerical flux function only and leaves unaltered the instationary term of the governing equations. This formulation is quite simple to implement, overcomes the time-stepping restrictions of the explicit scheme, is consistent in time and thus applicable to unsteady flows too. This technique allows to strongly improve the convergence rate as compared to the explicit full preconditioning technique because of the high robustness of the implicit scheme in the case of stiff equation systems. In particular, for a given polynomial degree, the implicit solver gives convergence rates almost independent of the Mach number. Furthermore, we have shown that for a given level of accuracy the preconditioning of the numerical fluxes allows to reduce the computational efforts expressed in terms of CPU time.

Finally, the problem of cancellation error has been minimized by formulating the governing equations in terms of perturbed variables. Reference quantities have been introduced in the equations for the thermodynamic variables and the computations were performed for the fluctuations. The governing equations are unaltered and the method was used in conjunction with the preconditioning. Computations were performed using the full preconditioning technique on triangular grid at very low Mach numbers and using linear, quadratic and cubic elements. The results suggest

that the relative treatment is fundamental to obtain convergence of the continuity and momentum equations at very low Mach numbers and that the cancellation error becomes more dominant when increasing the degree of the polynomial approximation. As the energy equation suffers more from the cancellation error than the other equations, the accuracy of temperature was improved only to some extent by the relative treatment of the variables.

A Primitive variables

In the following we give the formulae and matrices for the transformation between conservative variables $\mathbf{w} = (\rho, \rho u, \rho v, \rho E)^T$ and primitive variables $\mathbf{q} = (p, u, v, T)^T$.

Given \mathbf{w} in conservative variables we compute

$$\mathbf{q} = \begin{pmatrix} p \\ u \\ v \\ T \end{pmatrix} = \begin{pmatrix} (\gamma - 1) \rho e \\ \rho u / \rho \\ \rho v / \rho \\ (\gamma - 1) \frac{\rho e}{\rho} \end{pmatrix},$$

where the static specific energy e for a perfect gas is calculated from the conservative variables as:

$$\rho e = \rho E - \frac{1}{2\rho} [(\rho u)^2 + (\rho v)^2].$$

Given \mathbf{q} in primitive variables we compute

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} = \begin{pmatrix} p/T \\ (p/T) u \\ (p/T) v \\ p/(\gamma - 1) + p/(2T) (u^2 + v^2) \end{pmatrix}.$$

Thereby, the transformation matrix from conservative to primitive variables $\mathbf{\Gamma}$ is given by

$$\mathbf{\Gamma} = \frac{\partial \mathbf{w}}{\partial \mathbf{q}} = \begin{pmatrix} \rho_p & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & \rho_T u \\ \rho_p v & 0 & \rho & \rho_T v \\ \rho_p H - 1 & \rho u & \rho v & \rho_T H + \rho c_p \end{pmatrix},$$

with

$$\begin{aligned} \rho_p &= \frac{1}{T}, \\ \rho_T &= -\frac{p}{T^2} = -\frac{\rho}{T}, \end{aligned}$$

and thus

$$\begin{aligned} H &= e + \frac{1}{2}q^2 + \frac{p}{\rho} = (c_v + 1)T + \frac{1}{2}q^2 = c_p T + \frac{1}{2}q^2, \\ \rho_p H - 1 &= \frac{1}{T} \left[(c_v + 1)T + \frac{1}{2}q^2 \right] - 1 = c_v + \frac{1}{2}\rho_p q^2, \\ \rho_T H + \rho c_p &= -\frac{\rho}{T} \left[\left(c_p T + \frac{1}{2}q^2 \right) + \rho c_p \right] = \frac{1}{2}\rho_T q^2, \end{aligned}$$

where

$$\begin{aligned} e &= c_v T, \\ q^2 &= u^2 + v^2. \end{aligned}$$

Finally, the transformation matrix from primitive to conservative variables $\mathbf{\Gamma}^{-1}$ is given by

$$\mathbf{\Gamma}^{-1} = \frac{\partial \mathbf{q}}{\partial \mathbf{w}} = \begin{pmatrix} \frac{1}{2}(\gamma - 1)q^2 & -(\gamma - 1)u & -(\gamma - 1)v & \gamma - 1 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 \\ \frac{1}{\rho}(\gamma - 1)(q^2 - E) & -(\gamma - 1)\frac{u}{\rho} & -(\gamma - 1)\frac{v}{\rho} & \frac{\gamma - 1}{\rho} \end{pmatrix}.$$

B Roe Numerical Flux

The discrete, inviscid flux vectors are evaluated by flux difference splitting schemes. This scheme evaluates the convective flux on an interface between two neighbouring elements based on (in general discontinuous) interior and exterior states by solving the Riemann (shock tube) problem. In order to reduce the computational effort for the exact solution of the Riemann problem (Godunov's scheme), approximate Riemann solver were developed. In particular, Roe's method is often applied because of its high accuracy. The Roe's approximate Riemann solver is based on the decomposition of the flux difference over a face of the control volume into a sum of wave contributions, while ensuring the conservation properties of the Euler equations, for flux difference splitting see[2].

In terms of conservative variables \mathbf{w} the value of \mathbf{H} at each Gauss quadrature point on each face is given by

$$\mathbf{H} = \frac{1}{2} (\mathbf{F}^+ + \mathbf{F}^-) - \frac{1}{2} |\tilde{\mathbf{A}}| \Delta \mathbf{w}, \quad (B.1)$$

where \mathbf{F}^+ and \mathbf{F}^- are fluxes computed using the solution vectors \mathbf{w}^+ and \mathbf{w}^- on each (the interior and the exterior) side of the face and $\Delta \mathbf{w} = \mathbf{w}^- - \mathbf{w}^+$. In the above equation $|\tilde{\mathbf{A}}| \Delta \mathbf{w}$ can be viewed as a dissipation term. The matrix $|\tilde{\mathbf{A}}|$ denotes the so-called Roe matrix or dissipation matrix, and is equal to the Jacobian $\partial \mathbf{F} / \partial \mathbf{w}$ where the flow variables are replaced by the so-called Roe-averaged variables, denoted by the $\tilde{}$ symbol.

These are computed from the interior and exterior and right state by the formulae, see [2, 56],

$$\begin{aligned}
\tilde{\rho} &= \sqrt{\rho^+ \rho^-}, \\
\tilde{u} &= \frac{u^+ \sqrt{\rho^+} + u^- \sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\
\tilde{v} &= \frac{v^+ \sqrt{\rho^+} + v^- \sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\
\tilde{H} &= \frac{H^+ \sqrt{\rho^+} + H^- \sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\
\tilde{c} &= \sqrt{(\gamma - 1) \left(\tilde{H} - \tilde{q}^2/2 \right)}, \\
\tilde{u}_n &= \tilde{u} n_1 + \tilde{v} n_2, \\
\tilde{q}^2 &= \tilde{u}^2 + \tilde{v}^2.
\end{aligned}$$

The dissipation matrix $|\tilde{\mathbf{A}}|$ is also defined by

$$|\tilde{\mathbf{A}}| = \tilde{\mathbf{T}} |\tilde{\mathbf{\Lambda}}| \tilde{\mathbf{T}}^{-1}$$

where $|\tilde{\mathbf{\Lambda}}| = \text{diag}(\tilde{u}_n, \tilde{u}_n, \tilde{u}_n + \tilde{c}, \tilde{u}_n - \tilde{c})$ is the matrix of eigenvalues evaluated using Roe's averaging, as well as the matrix of left, $\tilde{\mathbf{T}}^{-1}$, and right, $\tilde{\mathbf{T}}$, eigenvectors, remembering that $\tilde{\mathbf{T}}$ is the modal matrix that diagonalizes the matrix $\tilde{\mathbf{A}}$.

The Roe's numerical flux already expressed in equations (B.1) can be rewritten in terms of diagonalization of the Roe matrix as:

$$\mathbf{H} = \frac{1}{2} (\mathbf{F}^+ + \mathbf{F}^-) - \frac{1}{2} \tilde{\mathbf{T}} |\tilde{\mathbf{\Lambda}}| \tilde{\mathbf{T}}^{-1} \Delta \mathbf{w}$$

This formulation makes clearer the decomposition into waves in Roe's scheme.

Bibliography

- [1] B. Van Leer. *Flux-vector splitting for the Euler equations*. Technical Report 81-11, ICASE, 1981.
- [2] P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.
- [3] B. Van Leer. Upwind-difference methods for Aerodynamic problems governed by the Euler equations. *Lectures in Applied Mathematics*, 22, 1985.
- [4] P. L. Roe. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics*, 18:337–365, 1986.
- [5] B. Van Leer, J. Thomas, P. Roe, and R. Newsome. A comparison of numerical flux formulas for the Euler and Navier-Stokes equations. *AIAA Paper*, 87-1104, 1987.
- [6] W. K. Anderson, R. D. Rausch, and D. L. Bonhaus. Implicit/multigrid algorithms for incompressible turbulent flows on unstructured grids. *Journal of Computational Physics*, 128:391–408, 1996.
- [7] N. A. Pierce and M. B. Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. *Journal of Computational Physics*, 136:425–445, 1997.
- [8] D. J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *Journal of Computational Physics*, 145:141–165, 1998.
- [9] D. J. Mavriplis and S. Pirzadeh. Large-scale parallel unstructured mesh computations for 3-d high lift analysis. *AIAA Journal of Aircraft*, 36:987–998, 1999.
- [10] D. W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, and M. J. Hemsch. Data summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop. *Journal of Aircraft*, 40(5):875–882, 2003.
- [11] K. R. Laflin, J. C. Vassberg, R. A. Wahls, J. H. Morrison, O. Brodersen, M. Rakowitz, E. N. Tinoco, and J. Godard. Summary of data from the Second AIAA CFD Drag Prediction Workshop. *AIAA Paper*, 2004-0555, 2004.

- [12] R. Hartmann. Discontinuous Galerkin methods for compressible flows: higher order accuracy, error estimation and adaptivity. In H. Deconinck and M. Ricchiuto, editors, *VKI LS 2006-01: CFD-Higher Order Discretization Methods, November 14-18, 2005, Rhode Saint Genese, Belgium*. Von Karman Institute for Fluid Dynamics, 2005.
- [13] D. Vigneron, G. Deliége, and J. A. Essers. Low Mach number local preconditioning for unsteady viscous finite volumes simulations on 3D unstructured grids. *ECCOMAS Computational Fluid Dynamics Conference 2006*, pages 1–14, TU Delft Netherland, 2006.
- [14] G. Volpe. Performance of compressible flow codes at low Mach numbers. *AIAA Journal*, 31:49–56, 1993.
- [15] M. Feistauer and V. Kucera. Solution of compressible flow with all Mach numbers. *Proceedings of the Conference ECCOMAS CFD*, 2006.
- [16] H. Luo, J. Baum, and R. Lohner. A fast p-multigrid Discontinuous Galerkin Method for compressible flow at all speeds. *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reeno, 2006.
- [17] F. Bassi, A. Crivellini, D. A. Di Pietro, and S. Rebay. A high-order Discontinuous Galerkin solver for 3D aerodynamic turbulent flows. *Proceedings of the Conference ECCOMAS CFD*, 2006.
- [18] R. Hartmann et al. *PADGE Parallel Adaptive Discontinuous Galerkin Environment*. Technical Reference. DLR Braunschweig, 2007. In preparation.
- [19] W. Bangerth, R. Hartmann, and G. Kanschat. *deal.II Differential Equations Analysis Library, Technical Reference*. <http://www.dealii.org/>, 6.0 edition, Sept. 2007. First edition 1999.
- [20] W. Bangerth, R. Hartmann, and G. Kanschat. *deal.II A general purpose object oriented finite element library*. *ACM Transactions on Mathematical Software*, 33(4), 2007. To appear. Available as technical report ISC-06-02-MATH, Texas A&M University, 2006.
- [21] S. H. Lee. Convergence characteristics of preconditioned Euler equations. *Journal of Computational Physics*, 208:266–288, 2005.
- [22] D. Gottlieb and S. A. Orszag. *Numerical Analysis of Spectral Methods*. SIAM, Philadelphia, 1977.
- [23] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, New York, 1987.
- [24] I. Babuska, B.A. Szabo, and I. N. Katz. The p-version of the finite element method. *SIAM Journal on Numerical Analysis*, 18(3):515–545, 1981.

- [25] W. H. Reed and T. R. Hill. *Triangular mesh methods for the neutron transport equation*. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [26] B. Cockburn, G. Karniadakis, and C. Shu. *Discontinuous Galerkin methods: theory, computation and applications*, volume 11 of Lecture Notes in Computational Science and Engineering. Springer, 2000.
- [27] E. Turkel, A. Fiterman, and B. Van Leer. Preconditioning and the limit of the compressible to the incompressible flow equations for finite difference schemes. *Frontiers of Computational Fluid Dynamics*, 1994:215–234, Caughey DA, Hafez MM editors, 1994.
- [28] H. Guillard and C. Viozat. On the behaviour of upwind schemes in the low Mach number limit. *Computers Fluids*, 28:63–86, 1999.
- [29] D. Lee. Local Preconditioning of the Euler and Navier-Stokes equations. *PhD Thesis*, University of Michigan, 1996.
- [30] Y. H. Choi and C. L. Merkle. The application of preconditioning in viscous flows. *Journal of computational Physics*, 105:207–233, 1993.
- [31] E. Turkel. Review of preconditioning methods for fluid dynamics. *Applied Numerical Mathematics*, 12:257–284, 1993.
- [32] E. Turkel. Preconditioning techniques in computational fluid dynamics. *Annual review of Fluid Mechanics*, 31:385–416, 1999.
- [33] D. Lee and B. Van Leer. Progress in local preconditioning of the Euler and Navier-Stokes equations. *AIAA Paper*, 93-3328, 1993.
- [34] J. Weiss and W. A. Smith. Preconditioning applied to variable and constant density flows. *AIAA Journal*, 33:2050–2057, 1995.
- [35] R. Radespiel, E. Turkel, and N. Kroll. Assessment of preconditioning methods. *DLR-FB*, 95-29, 1995.
- [36] G. Hauke and T. J. R. Hughes. A comparative study of different sets of variables for solving compressible and incompressible flows. *Comput. Methods Appl. Mech. Eng.*, 153:1–44, 1998.
- [37] B. Engquist and A. Majda. Absorbing boundary conditions for numerical simulation of waves. *Mathematics of Computations*, 31:629–651, 1997.
- [38] D. Givoli. Non-reflecting boundary conditions. *Journal of Computational Physics*, 94:1–29, 1991.
- [39] A. Bayliss and E. Turkel. Far field boundary conditions for compressible flow. *Journal of Computational Physics*, 48:182–199, 1982.

- [40] B. Gustafsson. Far field boundary conditions for time-dependent hyperbolic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(5):812–828, 1988.
- [41] D. L. Whitfield and J. M. Janus. Three-dimensional unsteady Euler equations solution using flux vector Splitting. *AIAA Paper*, 84-1552, 1984.
- [42] F. Bassi and S. Rebay. High order accurate discontinuous finite element solution of the 2d Euler equations. *Journal of Computational Physics*, 138:251–285, 1997.
- [43] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA Paper*, 81-1259, 1981.
- [44] B. Van Leer, C. H. Tai, and K. G. Powell. Design of optimally smoothing multistage schemes for the Euler equations. *AIAA Paper*, 89-1933, 1989.
- [45] H. T. Chang, H. S. Jiann, and B. Van Leer. Optimal multistage schemes for Euler equations with residual smoothing. *AIAA Journal*, 33:1008–1016, 1995.
- [46] R. Courant, K. O. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der Physik. *Math Ann.*, 100:32–74, 1928.
- [47] R. Heinrich. Implementation of a preconditioning technique in the hybrid flow solver TAU⁺. *PAMM*, 1:337–338, 2002.
- [48] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, B. F. Smith, and H. Zhang. *PETSc Web page*. <http://www.mcs.anl.gov/petsc>, 2001.
- [49] S. H. Lee. Cancellation problem of preconditioning method at low Mach numbers. *Journal of Computational Physics*, 225:1199–1210, 2007.
- [50] J. Sesterhenn, B. Müller, and H. Thomann. On the cancellation problem in calculating compressible low Mach number flows. *Journal of Computational Physics*, 151:597–615, 1999.
- [51] B. Müller and P. Jenny. Improving the reliability of low Mach number flow computations. *CFD Journal*, 9(1):518–528, 2001.
- [52] R. Klein. Semi-implicit extension of a Godunov-type scheme based on low Mach number Asymptotics I: one dimensional flow. *Journal of Computational Physics*, 121:213 – 237, 1995.
- [53] H. Bijl and P. Wesseling. Unified formulation for compressible and incompressible flows by using multi-integrated moments II multi-dimensional version for compressible and incompressible flows. *Journal of Computational Physics*, 141:153–173, 1998.

- [54] H. Guillard and A. Murrone. On the behaviour of upwind schemes in the Low Mach number limit: II Godunov type schemes. *Computers & Fluids*, 33:655–675, 2004.
- [55] Th. Schneider, N. Botta, J. J. Geratz, and R. Klein. Extension of finite volume compressible flow solver to multi-dimensional, variable density zero Mach number flow. *Journal of Computational Physics*, 155:248–286, 1999.
- [56] P. L. Roe and J. Pike. *Efficient construction and utilization of approximate Riemann solutions*. Computing Methods in Applied Sciences and Engineering. R. Glowinski and J. L. Lions (eds.), North Holland Publishing, The Netherlands, 1984.

IB 124-2008/1

**Discontinuous Galerkin Methods for inviscid low Mach
number flows**

A. Nigro

Verteiler:

Institut für Aerodynamik und Strömungstechnik, BS	1	Exemplar
Institut für Aerodynamik und Strömungstechnik, GÖ	...	1	Exemplar
Verfasser je	1	Exemplar
Prof. Dr.-Ing. habil. C. Rossow	1	Exemplar
Prof. Dr.-Ing. N. Kroll	1	Exemplar
Dr.-Ing. D. Schwamborn	1	Exemplar
Dr. R. Hartmann	1	Exemplar
Prof. G. Olivetti (University of Calabria)	1	Exemplar
Prof. G. Florio (University of Calabria)	1	Exemplar
Ing. C. De Bartolo (University of Calabria)	1	Exemplar
Prof. F. Bassi (University of Bergamo)	1	Exemplar
Technische Informationsbibliothek Hannover	1	Exemplar
Deutsche Bibliothek Frankfurt am Main	2	Exemplare
Zentralbibliothek	2	Exemplare
Reserve	5	Exemplare

21 Exemplare