A discontinuous Galerkin method for inviscid low Mach number flows

F. Bassi *, C. De Bartolo **, R. Hartmann ***, A. Nigro **

** Dip. di Ingegneria Industriale, Università di Bergamo, viale Marconi 5 24044 - Dalmine (BG), Italy.*

*** Dipartimento di Meccanica, Università della Calabria, Ponte P. Bucci cubo 44/C 87036 - Rende (CS), Italy.*

**** Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany.*

Abstract

In this work we extend the high-order Discontinuous Galerkin (DG) Finite element method to inviscid low Mach number flows. The method here presented is designed to improve the accuracy and efficiency of the solution at low Mach numbers using both explicit and implicit schemes for the temporal discretization of the compressible Euler equations. The algorithm is based on a classical preconditioning technique that in general entails modifying both the instationary term of the governing equations and the dissipative term of the numerical flux function (full preconditioning approach). In the paper we show that full preconditioning is beneficial for explicit time integration while the implicit scheme turns out to be efficient and accurate using just the modified numerical flux function. Thus the implicit scheme could also be used for time accurate computations. The performance of the method is demonstrated by solving an inviscid flow past a NACA0012 airfoil at different low Mach numbers using various degrees of polynomial approximations. Computations with and without preconditioning are performed on different grid topologies to analyze the influence of the spatial discretization on the accuracy of the DG solutions at low Mach numbers.

Key words: Low Mach number Flows; Discontinuous Galerkin finite element method; Preconditioning; Euler Equations, Compressible Flows, Roe Scheme.

1 Introduction

The system of the compressible Euler equations gets increasingly stiff at low Mach numbers and this behaviour, physically due to the large disparity of

wave speeds, strongly influences the numerical solution of such equations. Well known, undesirable effects of low speed flow on most numerical schemes include low convergence speed and loss of accuracy, [1–3]. Two further issues related to the numerical solution of low speed flows concern the choice of proper sets of unknown variables (conservative variables are ill-conditioned at low Mach number, see [4]) and a careful implementation of non reflecting boundary conditions.

Several preconditioning techniques, applied to the governing equations and to their discretization, have been developed in the past to cope with the stiffness and accuracy problems. These techniques basically modify the acoustic wave speeds premultiplying the time derivative terms of the governing equations by a preconditioning matrix. The resulting effect is that the condition number of the inviscid flux Jacobian matrices is drastically reduced, and hence the convergence speed of time-stepping or iterative procedures is significantly improved. For the large family of upwind schemes, preconditioning enters also in the formulation of numerical flux functions in order to properly balance the artificial dissipation implied by the numerical flux formulation, [2,3,5]. Some of the most recognized local preconditioners for inviscid and viscous flows were proposed by Choi and Merkle [6], Turkel [7,8], Lee and van Leer [9] and Weiss and Smith [10]. As the preconditioning destroys the time accuracy, it can be applied to steady-state simulations only. To overcome this limitation, dual time-stepping technique may be employed [10]. In the past, numerous studies have been devoted to these topics; a complete review of preconditioning techniques is given in [7–9,11].

As regards the set of dependent variables, it has been shown in [4] that the conservative incompressible formulation is well defined only for the entropy variables and the primitive variables including pressure. It has also been shown that these two sets of variables are best suited for solving practical problems, with the primitive variables being more accurate than the entropy variables for low speed and incompressible flow computations. For these reasons the primitive variables are often preferred for low Mach number computations [6,10,12,13] and they have also been used to develop numerical schemes well suited for both compressible and incompressible flows.

In this context, we note that Schneider et al. [14] and Klein et al. [15], devised a numerical scheme for zero Mach number computations based on conservative variables. In view of this the issue of what is the best choice of dependent variables for solving flow problems ranging from very subsonic to supersonic speeds might not be considered as settled.

Finally, as reported in [11,16,17], efficient and accurate implementations of preconditioning techniques also require to minimize spurious reflections at far field boundaries and this can be achieved by setting suitable combinations of

variables at far field boundaries.

In this paper we present a preconditioned DG discretization of the 2D compressible Euler equations suitable to compute low Mach number inviscid flows. The conservative Euler equations are written in terms of primitive variables and iterated to steady state using both explicit and implicit schemes. In the explicit case preconditioning affects both the time derivative terms of the governing equations, through the action of the Weiss and Smith preconditioning matrix [10], and the numerical dissipation of the Roe's Riemann solver used to compute the numerical flux (full preconditioning technique). In the implicit case we have found that preconditioning only needs to be applied to the numerical flux function (flux preconditioning technique). Thus the implicit scheme could directly be used to compute unsteady low Mach number flows without resorting to dual time stepping techniques.

To the author's knowledge a few papers have appeared in the literature describing DG solutions of low Mach number flows and such papers do not report on using any form of preconditioning. Luo et al. [18] have performed numerical experiments up to a Mach number of 10^{-2} while Feistauer and Kucera [19] have extended the simulation of compressible inviscid flows to a Mach number of 10^{-4}.

This paper aims at giving more insight on employing DG discretizations for low Mach number flows. In particular, we consider the DG discretization of the Euler equations written in the most appropriate set of variables, we show that preconditioning clearly improves both the accuracy and efficiency of the DG solvers, and, finally, we examine in detail the accuracy of solutions for different topologies of computational grids.

The outline of the paper is as follows. In Section 2 we present the preconditioned form of the compressible Euler equations using primitive variables. In Section 3 we describe the DG discretization of the governing equations, the boundary conditions and the preconditioned numerical flux function. In Section 4 we give some detail on the explicit and implicit time stepping schemes. The performance of the numerical scheme is then demonstrated in Section 5 by computing an inviscid flow around a NACA0012 airfoil for different low Mach numbers, grid topologies and degrees of polynomial approximation. Finally, a few conclusions are drawn in Section 6.

2 Governing 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

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{1}$$

where $\mathbf{w}$ is the state vector of conservative variables, and $\mathbf{F} = \mathbf{F}(\mathbf{f}, \mathbf{g})$ is the inviscid flux vector given by

$$\mathbf{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{bmatrix}.$$

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 $\gamma = c_p/c_v$.

Transforming the compressible Euler equations given in Eq.(1) from conservative variable to primitive variables we obtain

$$\mathbf{P}\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{2}$$

where the state vector $\mathbf{q}$ in primitive variables, and the transformation matrix $\mathbf{P} = \frac{\partial \mathbf{w}}{\partial \mathbf{q}}$ is given by

$$\mathbf{q} = \begin{bmatrix} p \\ u \\ v \\ T \end{bmatrix}, \qquad \mathbf{P} = \begin{pmatrix} \rho_p & 0 & 0 & \rho_T \\ \rho_p u & \rho & 0 & \rho_T u \\ \rho_p v & 0 & \rho & \rho_T v \\ \rho_p H - 1 & \rho u & \rho v & \rho_T H + \rho c_p \end{pmatrix}.$$

By assuming that the fluid obeys the perfect gas state equation, ρ can be calculated as $\rho{=}p/T$ and the derivatives of ρ are given by

$$\rho_p{=}\left.\frac{\partial \rho}{\partial p}\right|_{T=const.} = 1/T, \qquad \rho_T{=}\left.\frac{\partial \rho}{\partial T}\right|_{p=const.} = -\rho/T.$$

In a second step the transformation matrix $\mathbf{P}$ in Eq. (2) is replaced by a preconditioning matrix $\mathbf{\Gamma}$ resulting in the following preconditioned compressible Euler equations,

$$\mathbf{\Gamma}\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F} = 0. \tag{3}$$

The matrix $\mathbf{\Gamma}$ used in the present work is the local preconditioning matrix of Weiss and Smith [10] written in the following form:

$$\mathbf{\Gamma} = \begin{pmatrix} \theta & 0 & 0 & \rho_T \\ \theta u & \rho & 0 & \rho_T u \\ \theta v & 0 & \rho & \rho_T v \\ \theta H - 1 & \rho u & \rho v & \rho_T H + \rho c_p \end{pmatrix}, \tag{4}$$

where Θ is given by

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

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

$$U_r = \begin{cases} \varepsilon c, & \text{if } |\mathbf{v}| < \varepsilon c, \\ |\mathbf{v}|, & \text{if } \varepsilon c < |\mathbf{v}| < c, \\ c, & \text{if } |\mathbf{v}| > c, \end{cases} \tag{5}$$

where c is the acoustic speed and ε is a small number included to prevent singularities at stagnation points. Choosing $\varepsilon = O(M)$, the low Mach preconditioning ensures that the convective and acoustic wave speeds are of similar magnitude, proportional to the flow speed [20].

In the next section we will show how preconditioning enters in the formulation of the numerical flux function in the normal direction at Gauss integration points on inter-element faces. Hence it is worthwhile introducing here the wave speeds of the preconditioned Euler equations in the direction of the unit vector $\mathbf{n}$, which are given by the eigenvalues of $\mathbf{\Gamma}^{-1}(\frac{\partial \mathbf{f}}{\partial \mathbf{q}} n_1 + \frac{\partial \mathbf{g}}{\partial \mathbf{q}} n_2)$, where $\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{q}}$ are the inviscid flux jacobians with respect to the primitive variables, and n_1 and n_2 are the components of the unit vector $\mathbf{n} = (n_1, n_2)^T$. The propagation speeds in this direction are

$$\lambda_1 = \lambda_2 = u_n, \quad \lambda_3 = u_n' + c', \quad \lambda_4 = u_n' - c',$$

where

$$u_n = \mathbf{v} \cdot \mathbf{n} \tag{6}$$

$$\begin{aligned} u_n' &= u_n(1-\alpha), \\ c' &= \sqrt{\alpha^2 u_n^2 + U_r^2}, \\ \alpha &= \frac{1-\beta U_r^2}{2}, \\ \beta &= \left(\rho_p + \frac{\rho_T}{\rho C_p}\right), \\ \rho_p &= \left.\frac{\partial \rho}{\partial p}\right|_{T=const.}. \end{aligned} \tag{7}$$

For an ideal gas $\beta = 1/c^2$. At low speed as $U_r \to 0$, $\alpha \to 1/2$, and all the eigenvalues become of the same order as u_n. For the non-preconditioned system ($\alpha = 0, u_n' = u_n, c' = U_r = c$), $\mathbf{\Gamma}$ reduces to the transformation matrix $\mathbf{P}$ between conservative and primitive variables, and Eq.(3) becomes the conservative formulation of the Euler equations in terms of primitive variables.

We note, that all formulae above are given in non-dimensionalized variables based on the following reference values: the reference length l_r, density ρ_r, pressure p_r and constant gas R_r. Reference values for the other quantities are derived from these by dimensional relationships.

3 The preconditioned DG discretization

Multiplying Eq. (3) by a vector-valued test function $\mathbf{v}$ and integrating by parts, we obtain the weak formulation:

$$\int_\Omega \mathbf{v}^T \mathbf{\Gamma} \frac{\partial \mathbf{q}}{\partial t} \,\mathrm{d}\mathbf{x} - \int_\Omega \nabla \mathbf{v}^T \cdot \mathbf{F} \,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} \mathbf{v}^T \mathbf{F} \cdot \mathbf{n} \,\mathrm{d}s = 0 \qquad \forall \mathbf{v} \in H^1(\Omega) \tag{8}$$

where Ω is the domain with boundary $\partial\Omega$, and $\mathbf{n}$ is the unit outward normal vector. To discretize in space, we define $\mathbf{V}_h^p$ 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

$$\mathbf{V}_h^p = \left\{ \mathbf{v} \in L^2(\Omega) : \mathbf{v} \mid_\kappa \in P_p, \kappa \in T_h \right\},$$

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

$$\sum_{\kappa \epsilon T_h} \left\{ \int_\kappa \mathbf{v}_h^T \mathbf{\Gamma} \frac{\partial \mathbf{q}_h}{\partial t} \, \mathrm{d}\mathbf{x} - \int_\kappa \nabla \mathbf{v}_h^T \cdot \mathbf{F} \, \mathrm{d}\mathbf{x} \right.$$
$$\left. + \int_{\partial\kappa \backslash \partial\Omega} \mathbf{v}_h^{+^T} \mathbf{H}_i \left(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n} \right) \, \mathrm{d}s + \int_{\partial\kappa \cap \partial\Omega} \mathbf{v}_h^{+^T} \mathbf{H}_b \left(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n} \right) \, \mathrm{d}s \right\} = 0 \qquad (9)$$

for all $\mathbf{v}_h \in \mathbf{V}_h^p$, where $\mathbf{H}_i \left(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n} \right)$ and $\mathbf{H}_b \left(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n} \right)$ are numerical flux functions defined on interior and boundary faces, respectively. $\mathbf{H}_i$ takes into account the possible discontinuities of $\mathbf{q}_h$ at element interfaces. On interior edges $\partial\kappa \backslash \partial\Omega$, $\mathbf{H}_i$ depends on the elements interior state $\mathbf{q}_h^+$ and on the neighbouring elements state $\mathbf{q}_h^-$. On boundary edges $\partial\kappa \cap \partial\Omega$, $\mathbf{H}_b$ depends on the interior state $\mathbf{q}_h^+$ and a consistent boundary state $\mathbf{q}_h^b$. We note that $\mathbf{H}_b$ may be different from $\mathbf{H}_i$.
We note that due to the quasi-linear form of the time derivative term of Eq. (2) and due to the preconditioning (4) applied an explicit time stepping scheme based on (9) is not time-accurate nor conservative in space-time. However, having reached a steady state solution the time derivatives vanish. In fact, for steady state solutions the numerical scheme (9) is conservative which can be seen by setting $\mathbf{v}_h^+ \equiv 1$ in (9).

The spatial DG discretization of Eq. (9) results in the following global system of equations:

$$\mathbf{M}_\Gamma \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} + \mathbf{R} = 0, \qquad (10)$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the global vectors of degrees of freedom (dof) and of residuals respectively, and $\mathbf{M}_\Gamma$ stands for the discretization of the first integral of Eq. (9). Hence, $\mathbf{M}_\Gamma$ 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 $\mathbf{\Gamma}$).

3.1 Boundary treatment

In the following we give some details on the boundary treatment. In particular, an appropriate representation of the possibly curved boundary geometry and an appropriate discretization of boundary conditions employed are essential for preserving the numerical accuracy and improving the convergence speed of the solution process in the low Mach number limit.

3.1.1 Geometry representation

A high order accurate solution on relatively coarse grids can be obtained only if a corresponding high order approximation of the geometry is employed. In this work, the geometric continuity of the element edges belonging to the boundary $\partial\Omega$ is guaranteed by a mapping based on Lagrangian polynomial functions $\phi_j(\boldsymbol{\xi})$ and Lagrangian node coordinates $\mathbf{x}^{(j)}$ and is given by

$$\mathbf{x} = \sum_j \mathbf{x}^{(j)} \phi_j(\boldsymbol{\xi}) \qquad \forall \boldsymbol{\xi} \in \hat{\kappa}, \tag{11}$$

where $\boldsymbol{\xi}$ is the independent variable on the reference element $\hat{\kappa}$. Notice that the Lagrangian nodes are placed on the real geometry of the boundary.

3.1.2 Boundary conditions

When $\partial\kappa$ belongs to $\partial\Omega$ the boundary fluxes, denoted by $\mathbf{H}_b\left(\mathbf{q}^+, \mathbf{q}^b, \mathbf{n}\right)$, are chosen to weakly prescribe the boundary conditions of the problem. Here, $\mathbf{n}$ is the unit outward normal vector, $\mathbf{q}^+$ is the interior state at the boundary and $\mathbf{q}^b$ is computed according to the conditions that must be satisfied on the boundary.

- Far-field

  At far-field a complete set of characteristic boundary conditions [21], and a set of simplified non-reflecting boundary conditions [11] are employed for the non-preconditioned and the preconditioned DG scheme, respectively. In particular, for the preconditioned scheme, at the inflow boundary the state $\mathbf{q}^b$ has the same pressure as $\mathbf{q}^+$, whereas the velocity vector and the temperature is prescribed based on the freestream values. Conversely, at the outflow boundary, the state $\mathbf{q}^b$ has the same temperature and velocity vector as $\mathbf{q}^+$, whereas the pressure is prescribed based on the freestream value. We remark that the simplified non-reflecting boundary conditions require a far-field boundary well far away from the

aerodynamic surface in order to get efficient and accurate solutions.

- Slip wall

  The wall boundary condition employed is based on following boundary state:

$$\begin{aligned} p^b &= p^+, \\ u^b &= u^+ - (\mathbf{v}\cdot\mathbf{n})^+ n_1, \\ v^b &= v^+ - (\mathbf{v}\cdot\mathbf{n})^+ n_2, \\ T^b &= T^+, \end{aligned} \tag{12}$$

  where n_1 and n_2 are the components of the unit outward normal $\mathbf{n} = (n_1, n_2)^T$. The conditions imposed on the velocity components 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:

$$\mathbf{H}_b\left(\mathbf{q}_h^+, \mathbf{q}_h^b, \mathbf{n}\right) = \mathbf{F}\left(\mathbf{q}_h^b\right)\cdot\mathbf{n}.$$

  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.

3.2 Flux difference splitting

The numerical flux $\mathbf{H}_i(\mathbf{q}^+, \mathbf{q}^-, \mathbf{n})$ appearing in Eq. (9) is computed based on a preconditioning of the artificial dissipation term of the Roe's approximate Riemann solver [22]. In terms of primitive quantities $\mathbf{q}$, the value of $\mathbf{H}_i$ at each face is given by

$$\mathbf{H}_i\left(\mathbf{q}^+, \mathbf{q}^-, \mathbf{n}\right) = \frac{1}{2}\left(\mathbf{F}(\mathbf{q}^+)\cdot\mathbf{n} + \mathbf{F}(\mathbf{q}^-)\cdot\mathbf{n} - \tilde{\mathbf{F}}_\Gamma\left(\mathbf{q}^+, \mathbf{q}^-, \mathbf{n}\right)\right), \tag{13}$$

where $\tilde{\mathbf{F}}_\Gamma$ is given by

$$\tilde{\mathbf{\Gamma}}|\tilde{\mathbf{A}}_\Gamma|\Delta\mathbf{q}. \tag{14}$$

Here, $\Delta\mathbf{q} = \mathbf{q}^- - \mathbf{q}^+$ and the matrix $|\tilde{\mathbf{A}}_\Gamma|$ is defined in terms of the preconditioned eigenvalues and eigenvectors by

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

The symbol $\tilde{}$ denotes that the matrices are computed using the Roe-averaged variables [23] and the subscript Γ that the diagonal matrix of eigenvalues and the modal matrix are derived from the preconditioned system, where $\tilde{\mathbf{\Lambda}}_\Gamma$ is the diagonal matrix of the preconditioned eigenvalues, and $\tilde{\mathbf{T}}_\Gamma$ diagonalizes the matrix $\mathbf{\Gamma}^{-1}(\frac{\tilde{\partial}\mathbf{F}}{\partial \mathbf{q}} \cdot \mathbf{n})$. We note, that for the non-preconditioned system, Eq.(13) reduces to the standard Roe's flux difference splitting.

4 Time discretization of the Euler equations

4.1 Explicit time stepping scheme

The semidiscrete system Eq.(10) is discretized in time based on an explicit multistage time-stepping method. In order to overcome the restrictive explicit CFL stability limit, both the local time–stepping and the preconditioning techniques have been used to improve the convergence speed to steady state solutions.

The solution is advanced from time t to time $t + \Delta t$ with an s-stage SSP Runge-Kutta scheme [24], given by

$$\begin{aligned} &\mathbf{Q}^0 = \mathbf{Q}^t, \\ &\mathbf{Q}^i = \sum_{k=0}^{i-1} \alpha_{ik} \mathbf{Q}^k + \beta_{ik} \Delta t \mathbf{M}_\Gamma^{-1} \mathbf{R}(\mathbf{Q}^k) \qquad i = 1, 2, ..., s, \\ &\mathbf{Q}^{t+\Delta t} = \mathbf{Q}^s, \end{aligned} \tag{15}$$

where i is the stage counter for the s-stage scheme and α_{ik} and β_{ik} are the multistage coefficients for the ith stage.

The local time step Δt on each element κ is computed by considering the CFL stability condition:

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

where the preconditioned convective spectral radii Λ_c^x and Λ_c^y are defined as

$$\Lambda_c^x = \left(|\bar{u}'| + \bar{c}_x'\right) \Delta S^x \ ,$$
$$\Lambda_c^y = \left(|\bar{v}'| + \bar{c}_y'\right) \Delta S^y .$$

The variables ΔS^x and ΔS^y represent the projections of the element κ onto the x and y axis, respectively, whereas $\bar{u}'$, $\bar{c}_x'$ and $\bar{v}'$, $\bar{c}_y'$ are obtained applying Equations (7) along the x and y directions and using the mean values of the flow quantities on each element κ.

4.2 Implicit time stepping scheme

We have found that the implicit time stepping scheme can be used to compute efficiently and accurately low Mach number flows even in absence of time-derivative preconditioning. Hence, in Eq.(9) the matrix $\mathbf{\Gamma}$ reduces to the transformation matrix between conservative and primitive variables, $\mathbf{P}$, and the DG space discretization results in the following global system of equations:

$$\mathbf{M}_P \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}t} + \mathbf{R} = 0. \tag{16}$$

The implicit backward Euler time discretization of Eq. (16) can be written as

$$\underbrace{\left[\frac{\mathbf{M}_P}{\Delta t} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}}\right]}_{\mathbf{B}} \Delta \mathbf{Q}^n = -\mathbf{R}^n, \tag{17}$$

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 has 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 $\Delta t \to \infty$ Equation (17) is in fact identical to one iteration of the Newton method applied to the steady discrete problem.

Finally, we mention that to solve Equation (17) we can use either direct or iterative linear solvers. For all the computations presented below we have used the GMRES iterative solver available in the PETSc [25] library. By default this solver employs the ILU(0) preconditioning and in the following the GMRES solver is meant to be used with ILU(0) preconditioning for the computations performed with and without low Mach number preconditioning.

5 Numerical results

In this section, we present some numerical results demonstrating the performance of the proposed preconditioned DG discretization. To this end, we consider an inviscid flow past a NACA0012 airfoil at zero angle of attack comparing the DG discretizations with and without preconditioning. DG solutions on different grids, for different low Mach numbers ($M = 10^{-1}$, $M = 10^{-2}$ and $M = 10^{-3}$) and using linear (P_1), quadratic (P_2) and cubic (P_3) elements are performed. Two grid topologies (quadrangular and triangular) are used in order to investigate the behavior of both the standard and the preconditioned DG method for different element shapes. Fig. 1 shows the computational grids. The quadrangular grid is a C-type grid with 1792 elements, and the triangular grid consists of the triangles obtained by splitting each quadrangle in two parts. The distance of the far-field boundary from the profile is about 55 chords. All computations are performed in double precision, storing 16 significant digits.

The computational results are organized in two subsections, one focusing on the convergence of the residuals and the other on the accuracy of the converged solutions. The convergence speed of the solution process is presented in terms of the normalized L_2 norm of the residuals versus the number of iterations and versus the CPU time. The accuracy of the converged solutions is analyzed both qualitatively and quantitatively. First, the normalized pressure fields are presented for a qualitative comparison. Then, for the quantitative analysis, the scaling of computed pressure fluctuations as the Mach number reduces is compared with the M^2 theoretical scaling, and the computed drag coefficients are compared with the theoretical one which is zero for the subsonic inviscid flow considered.

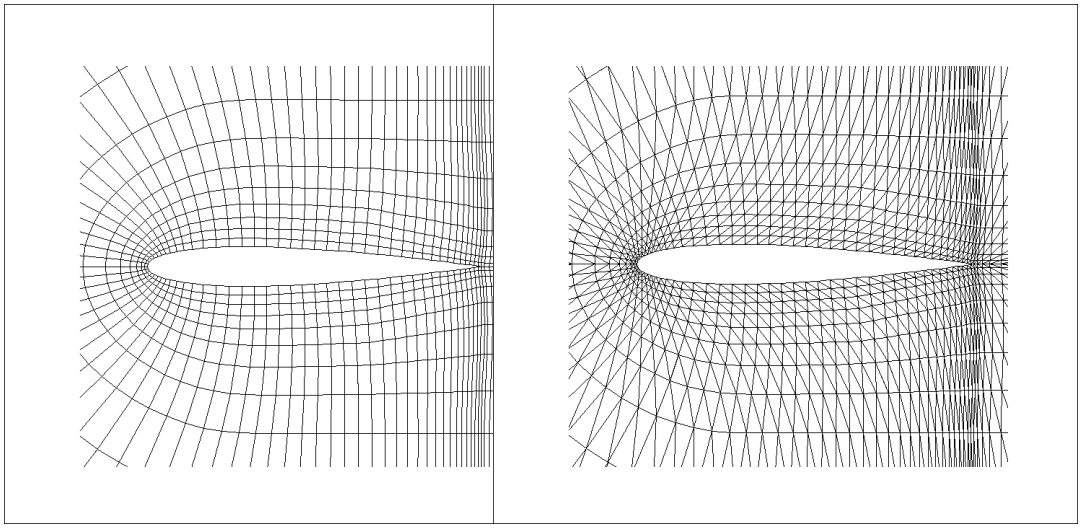

Fig. 1. Computational Grids

5.1 *Effects of preconditioning on convergence speed*

The convergence histories are shown only for the quadrangular grid as similar histories are obtained on the triangular grid. The results are presented first for the full preconditioning approach and then for the flux preconditioning technique.

5.1.1 *Explicit time stepping results*

In Fig. 2 we compare the residual histories with and without preconditioning. The plots show that the preconditioning technique leads to an acceleration of convergence in comparison to the non-preconditioned solution. For a given polynomial approximation, the convergence speed without preconditioning reduces as the Mach number approaches zero, while it is independent of the Mach number with preconditioning. For a given Mach number, the efficiency of both the preconditioned and the non-preconditioned explicit methods reduces due to the CFL stability condition. Considering the fully converged solutions, we see that, for a given polynomial degree, the lower the Mach number, the smaller the preconditioned residual decay. This is due to round-off errors and resulting cancellation errors that have a larger effect on the preconditioned scheme than on the non-preconditioned one as it is evident for $M = 10^{-1}$ and quadratic elements, see also [26]. This behaviour can be explained considering that with preconditioning the settings used to compute the artificial dissipation of Roe's flux cause truncation error to grow more than the corresponding non-preconditioned one. Notwithstanding, in all cases the preconditioned residual decays were sufficient enough to obtain accurate solutions. In addition, the oscillations that appear in the convergence histories are due to the vorticity produced at the leading and at the trailing edge during

the wave reflections [27].

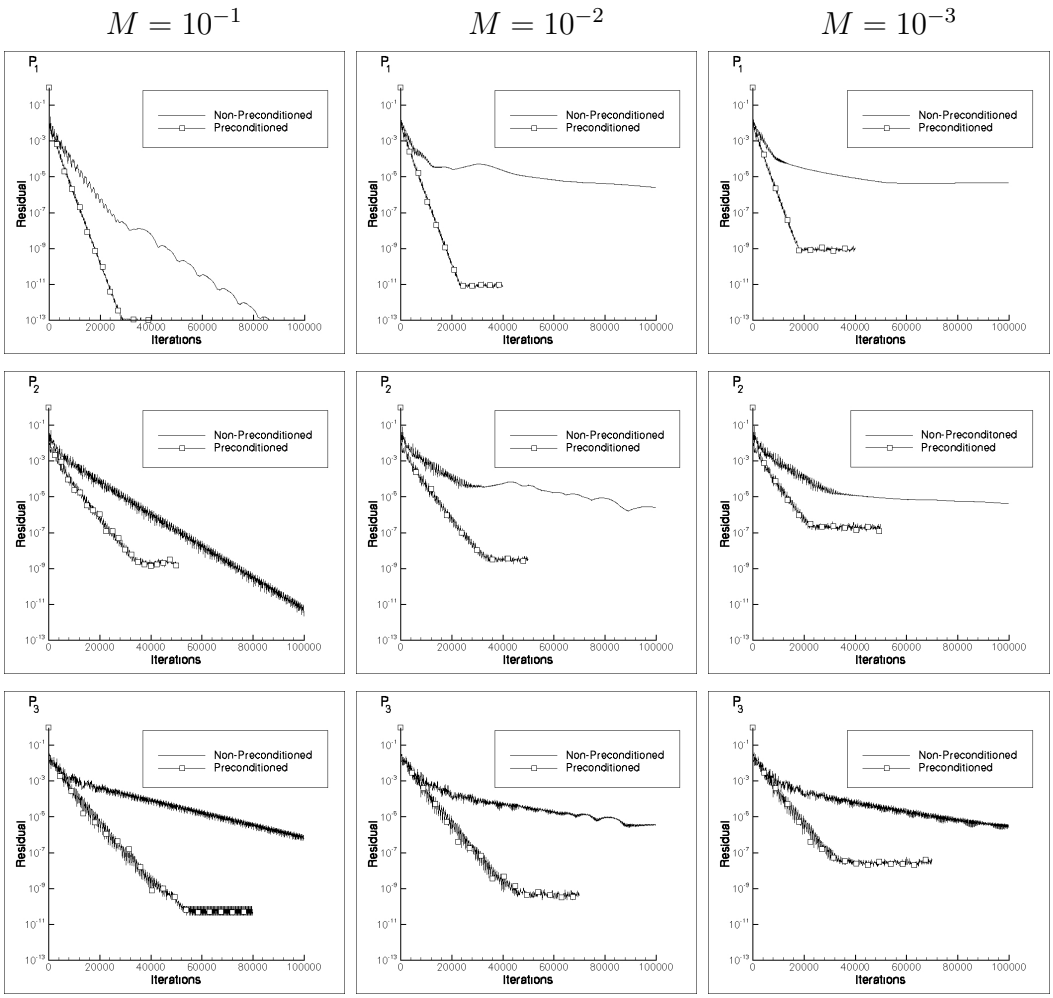

Fig. 2. History of the nonlinear residuals vs. the number of iteration steps for the quadrangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). Linear (P_1 top row), quadratic (P_2 middle row) and cubic (P_3 bottom row) elements.

The effectiveness of preconditioning in accelerating the convergence is illustrated in Fig. 3 where we compare the convergence histories of a preconditioned computation at $M = 10^{-3}$ and of a subsonic computation at $M = 0.4$ which does not require preconditioning. The results clearly show that preconditioning effectively recovers the same (or even better) efficiency of a classical TVD Runge-Kutta scheme, subject to the typical CFL condition for high-order DG discretizations, applied to a well-conditioned problem.

5.1.2 *Implicit time stepping results*

The Fig. 4 compares the history of residuals versus the number of "Newton" iteration steps of Eq. (17) with and without flux preconditioning. We remark that the graphs of Fig. 4 merely show the effect of the fixed GMRES parameters (number of Krylov-subspace vectors = 60, number of restarts = 1 and relative tolerance to stop iterative solution= 10^{-6}) on the convergence of the global "Newton" iterations and if these parameters are enough to ensure quadratic convergence of residuals.

The plots show that both the non-preconditioned and the preconditioned implicit schemes converge. Nevertheless, the use of non-preconditioned Jacobians shows a deterioration in the convergence rate at large Courant numbers as the Mach numbers gets smaller. 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 $M = 10^{-3}$. Hence, 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, with preconditioning the full convergence of the residuals was reached quadratically in about 10 iterations independently of both Mach number and polynomial degree.

Finally, 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, whereas all the residual decays of the non-preconditioned DG method reduce of $O(M)$, the preconditioned residual decays of velocity components and thermodynamic variables reduce of $O(M)$ and $O(M^2)$, respectively, when Mach number tends to zero, because of round-off errors. The round-off errors can be alleviated by introducing the gauge-pressure [6].

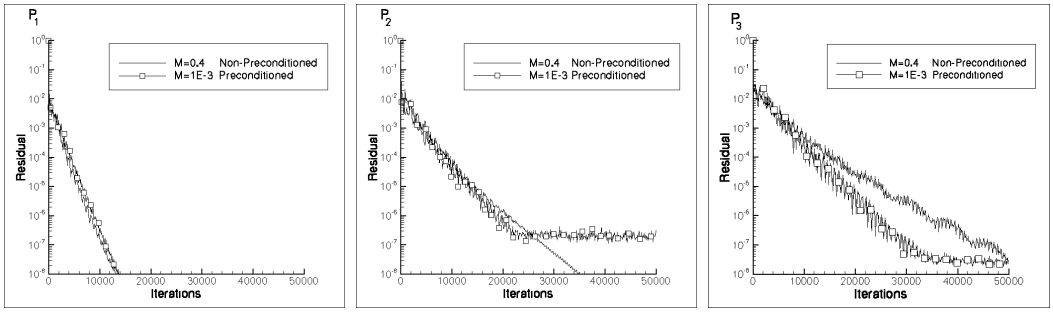

Fig. 3. History of nonlinear residuals for $M = 10^{-3}$ and $M = 0.4$ with and without preconditioning, respectively. Linear (P_1 left), quadratic (P_2 middle) and cubic (P_3 right) elements.

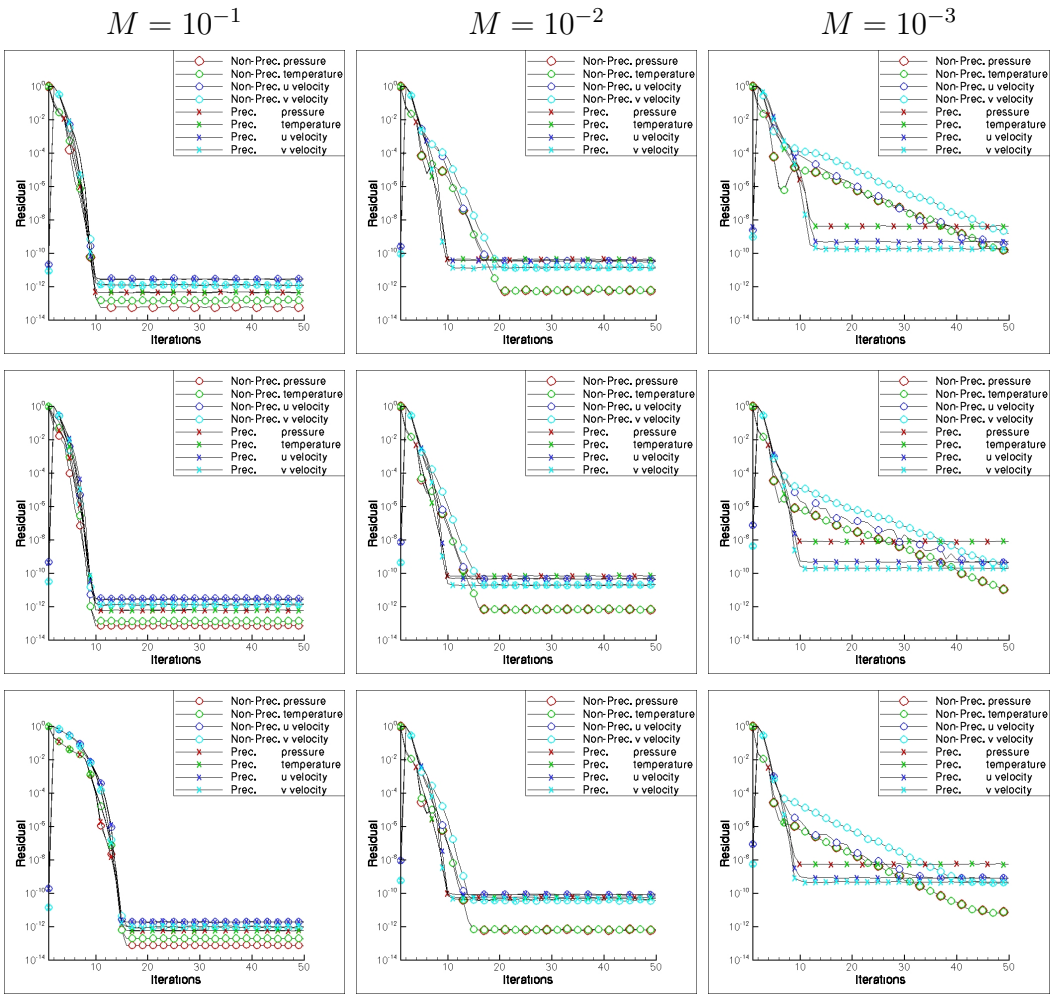

Fig. 4. History of residuals vs. number of iterations for the quadrangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). Linear (P_1 top row), quadratic (P_2 middle row) and cubic (P_3 bottom row) elements.

The Fig. 5 compares the history of residuals versus CPU time (seconds), computed on the quadrangular grid with and without flux preconditioning. Overall, the plots confirm that preconditioning improves the efficiency of the implicit solver. For a given polynomial approximation, the convergence rate without preconditioning reduces as the Mach number goes to zero, while it is almost independent of the Mach number with preconditioning. Furthermore, for a given Mach number, using the preconditioned Roe's flux, the overhead, in terms of CPU time, significantly reduces as the polynomial degree increases in comparison to the non-preconditioned solution. This effect is greatest at lower Mach number and reduces as the Mach number gets larger.

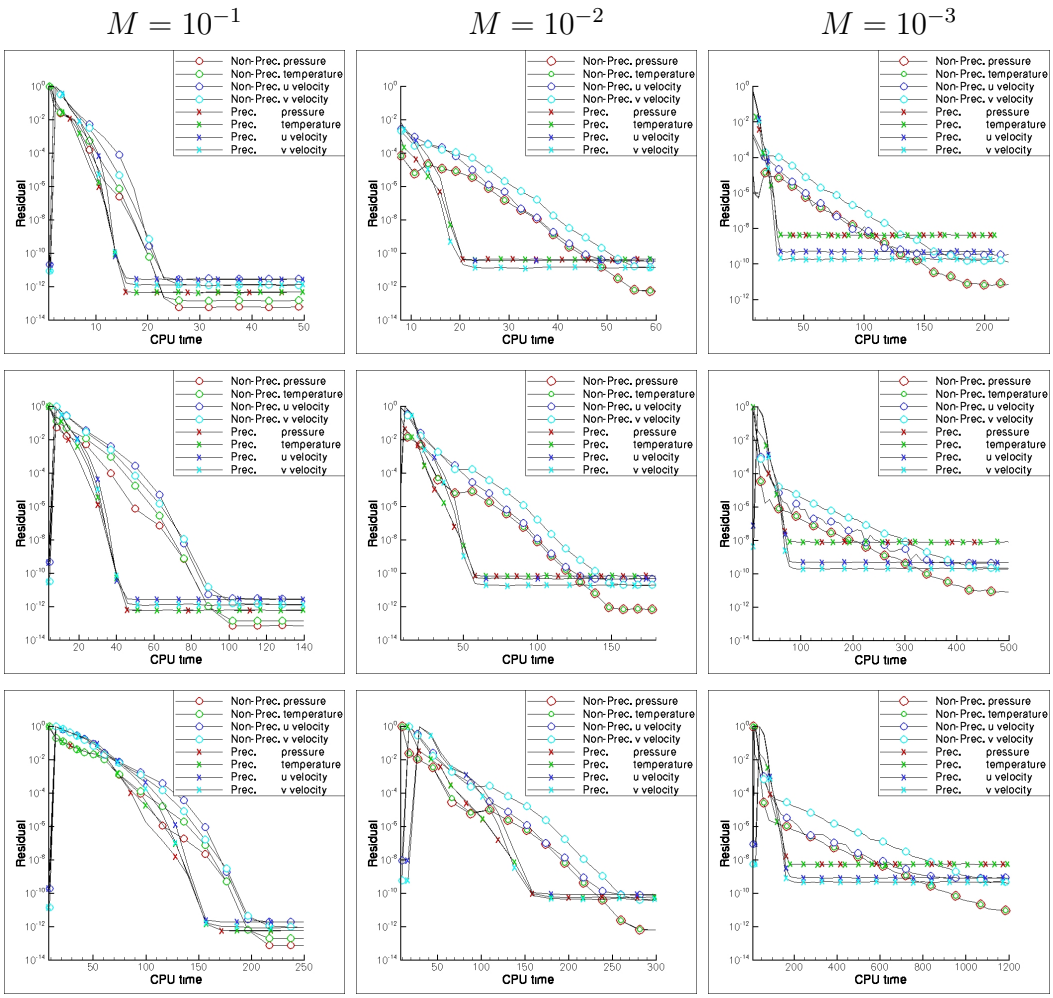

Fig. 5. History of residuals vs. CPU time for the quadrangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). Linear (P_1 top row), quadratic (P_2 middle row) and cubic (P_3 bottom row) elements.

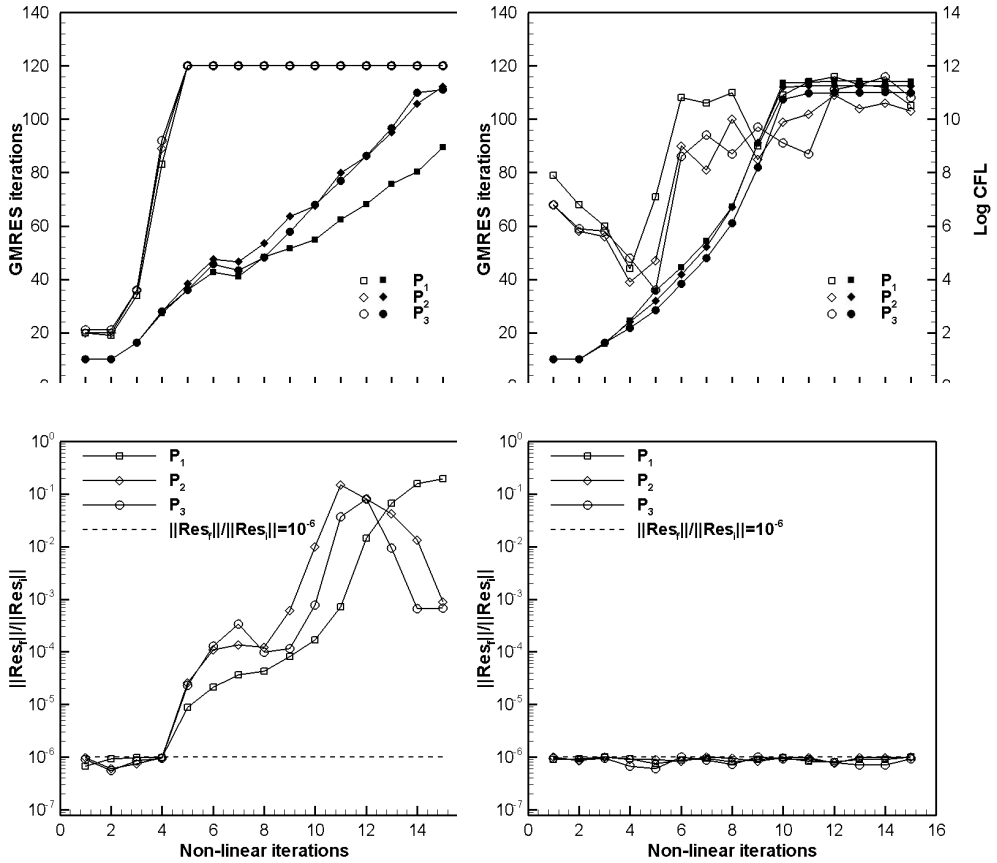

Fig. 6. Behaviour of GMRES solver with (right column) and without (left column) low Mach number preconditioning.

The Fig. 6 summarizes the performance of the GMRES solver with (right column) and without (left column) low Mach number preconditioning. The graphs show the results for the P_1, P_2 and P_3 solutions at $M = 10^{-2}$. Similar results hold also for $M = 10^{-1}$ and $M = 10^{-3}$. The plots on the top row show the number of GMRES iterations (open symbols) and the logarithm of CFL number (solid symbols), while those on the bottom row show the ratio between the L_2 norms of the last and the first residual of the GMRES iterative solution. The quantity on the X-axis is the number of non-linear iterations. The graphs of Fig. 6 suggest that (i) increasing the CFL number the computations performed without low Mach number preconditioning rapidly use up the maximum number of GMRES iterations without satisfying the required six-order drop of residuals, and that (ii) the low Mach number preconditioned solutions require somewhat less than 120 GMRES iterations to solve the linear system within each time step, even for the highest CFL numbers. Moreover we notice that the different behaviour of the low Mach number preconditioned and non-preconditioned solutions is even more evident for the lowest Mach number. Finally, we mention that the cost to compute the analytical Jacobian with respect to the computational cost of a full time step using 120 GM-

RES iterations is around 20%, 28% and 35% for the P1, P2 and P3 solutions, respectively.

5.2 Effects of preconditioning on the solution accuracy

In this section we examine the accuracy of the (fully) converged solutions. We observe that, whilst the time-derivative preconditioning matrix $\mathbf{\Gamma}$ basically improves the convergence speed of low Mach number computations, the solution accuracy is essentially determined by the preconditioning of the Riemann solver. The two preconditioning strategies here presented use the same flux difference splitting scheme, and thereby give the same results in terms of accuracy of solution.

5.2.1 Normalized pressure

In the following we present the contour plots of the normalized pressure, defined as $p_{norm} = (p - p_{min}) / (p_{max} - p_{min})$, computed on the quadrangular and triangular grids.

We begin by showing the results on the quadrangular grid. Fig. 7 shows the normalized pressure isolines of the non-preconditioned solutions at $M = 10^{-1}$, $M = 10^{-2}$ and $M = 10^{-3}$, for P_1, P_2 and P_3 elements. Fig. 8 shows the corresponding results of the preconditioned solutions but only at $M = 10^{-3}$, as the preconditioned results are independent of the Mach number, as will be shown in the next section.

Overall, from Figs. 7 and 8 we see that the preconditioned solutions are more accurate than the corresponding non-preconditioned ones. In particular, at $M = 10^{-1}$ (left column), the P_1 solution is inaccurate without preconditioning. This loss of accuracy is less evident using P_2 elements, whereas for P_3 elements there are no visible differences in terms of normalized pressure isolines. At $M = 10^{-2}$ (middle column) at least P_3 elements are required to obtain an acceptable level of accuracy without preconditioning, whereas at $M = 10^{-3}$ (left column) there is a clear difference between the preconditioned and the non-preconditioned solutions even if P_3 elements are used. Hence, 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. Furthermore, for a given Mach number, the higher the polynomial degree, the lower is the difference between the preconditioned and the non-preconditioned solutions. In such cases the preconditioning allows to significantly reduce the computational effort.

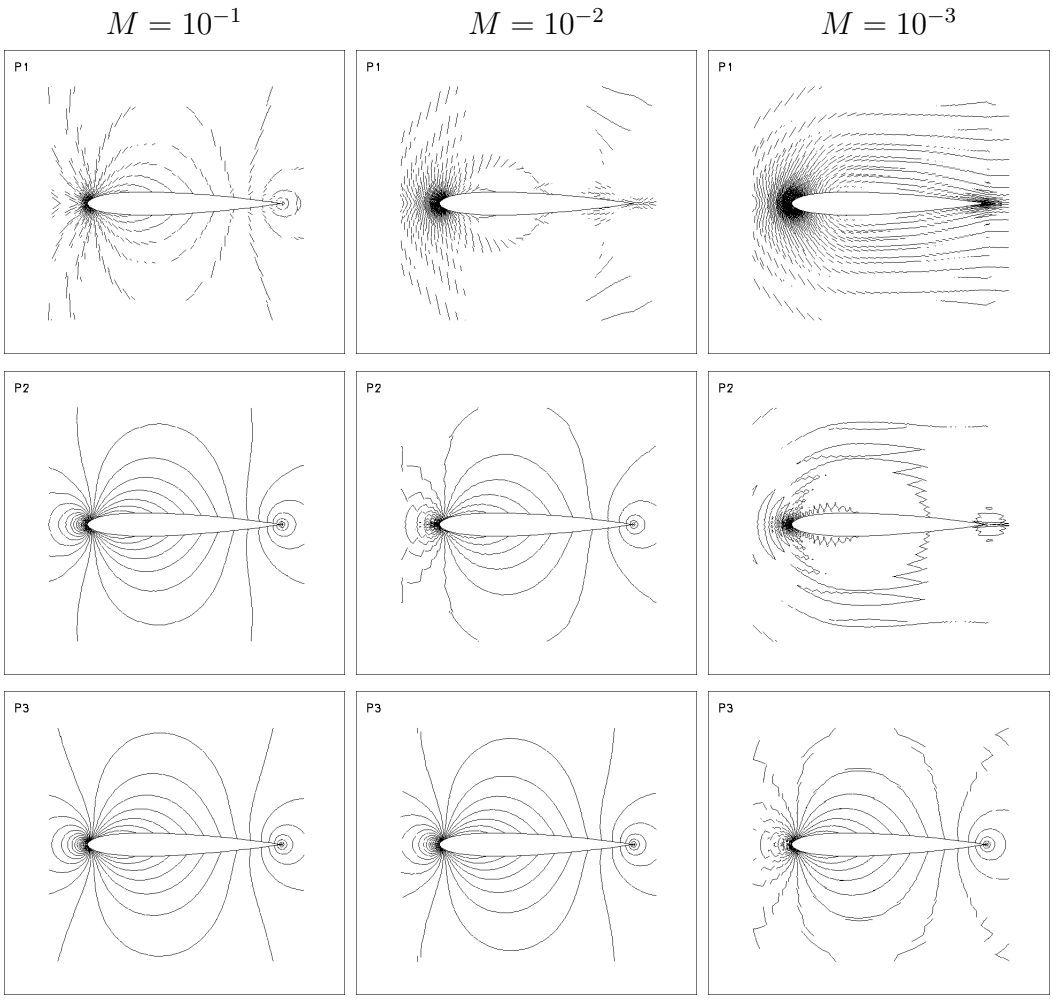

Fig. 7. Contours of normalized pressure without preconditioning for the quadrangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). Linear (P_1 top row), quadratic (P_2 middle row) and cubic (P_3 bottom row) elements.

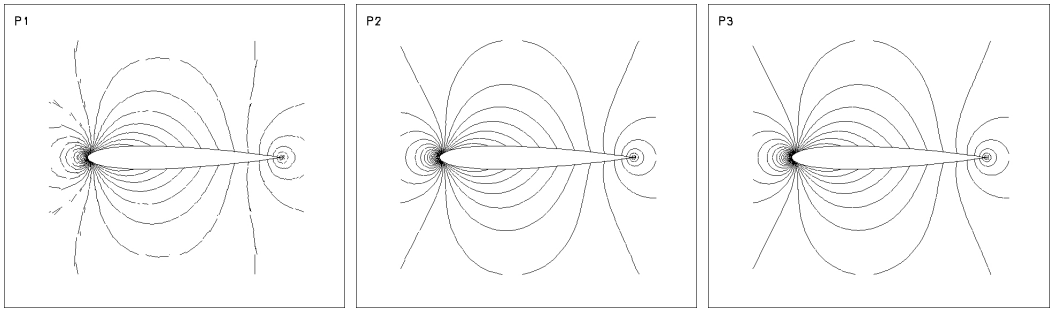

Fig. 8. Contours of normalized pressure with preconditioning for the quadrangular grid at $M = 10^{-3}$. Linear (P_1 left), quadratic (P_2 middle) and cubic (P_3 right) elements.

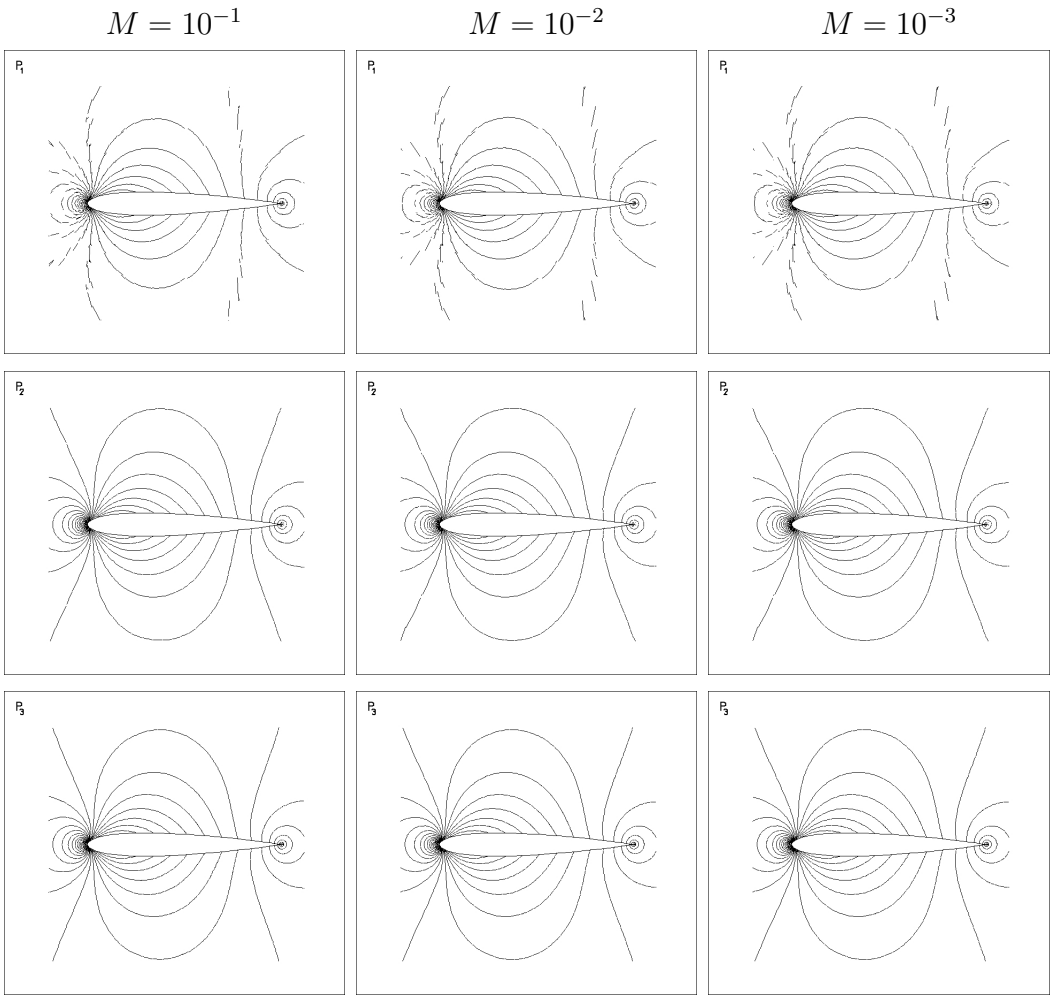

Fig. 9. Contours of normalized pressure without preconditioning for triangular grid. $M = 10^{-1}$ (left column), $M = 10^{-2}$ (middle column) and $M = 10^{-3}$ (right column). Linear (P_1 top row), quadratic (P_2 middle row) and cubic (P_3 bottom row) elements.

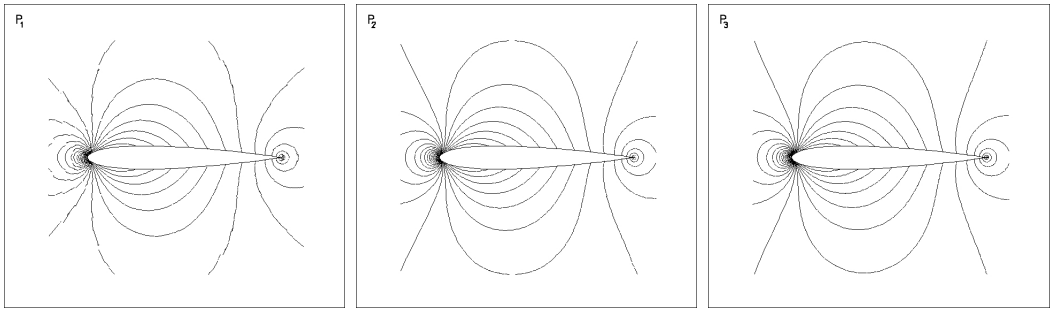

Fig. 10. Contours of normalized pressure with preconditioning for triangular grid at $M = 10^{-3}$. Linear (P_1 left), quadratic (P_2 middle) and cubic (P_3 right) elements.

Figs. 9 and 10 show the isolines of the normalized pressure for the triangular grid with and without preconditioning, respectively. Overall, it is worth noting that the DG discretization on triangular grid yields remarkably accurate solutions at low Mach even without preconditioning. In particular, the preconditioned and the non-preconditioned contours of normalized pressure are almost indistinguishable using P_2 and P_3 elements, whereas some differences can be seen in the P_1 solutions. However, we see that the DG discretization on the triangular grid avoids the accuracy degradation of the solutions as the Mach number reduces even for the lowest order approximation.

These results cannot be explained only by the doubled number of elements in the triangular grid. In fact in Fig. 11 we see that a computation at $M = 10^{-3}$ on a globally refined quadrangular grid with 7168 elements produces results which are still far worse than the corresponding results on the triangular grid with half the number of elements, shown in Fig. 9 (right column). Nevertheless, the difference in accuracy between results computed on the two grid types reduces as the polynomial degree increases.

The origin of the inaccuracy of the non-preconditioned solutions can be understood looking in detail at the normalized pressure contours around the leading edge of the airfoil. In Fig. 12 we compare the solutions for $M = 10^{-3}$ computed on the refined quadrangular grid and the triangular grid, using P_1, P_2 and P_3 elements. This Figure clearly shows that approaching the stagnation point the solution degrades because in this region the poorly scaled dissipation term of the Roe's Riemann solver is badly affected by the magnitude of inter-element jumps. This effect reduces by increasing the degree of polynomial approximation. More importantly, and consistently with the results of Fig. 9, the loss of accuracy around the leading edge is much higher in the solutions on the quadrangular grid. The marked influence of the geometrical shape of the elements on the accuracy of the Roe's flux in the low Mach number limit is an issue that needs deeper investigation. The asympotic analysis recently performed by Rieper [28] for the first-order Roe scheme might indicate that low order DG schemes face the same problems as the standard finite volume upwind schemes: at low Mach number they only work on triangular elements.

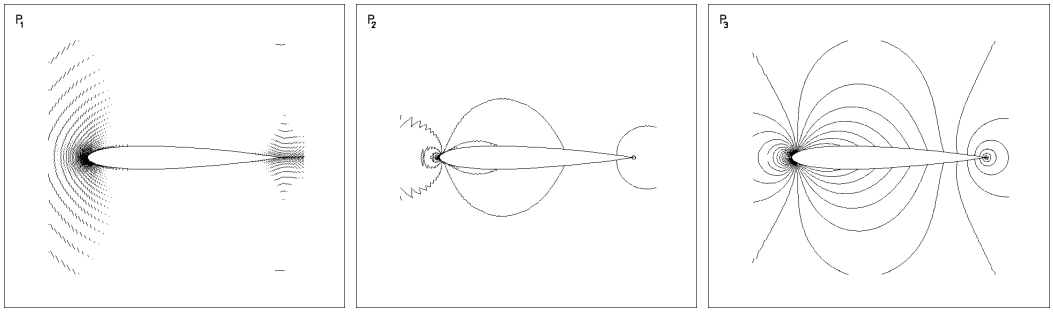

Fig. 11. Contours of normalized pressure without preconditioning on the refined quadrangular grid at $M = 10^{-3}$. Linear (P_1 left), quadratic (P_2 middle) and cubic (P_3 right) elements.

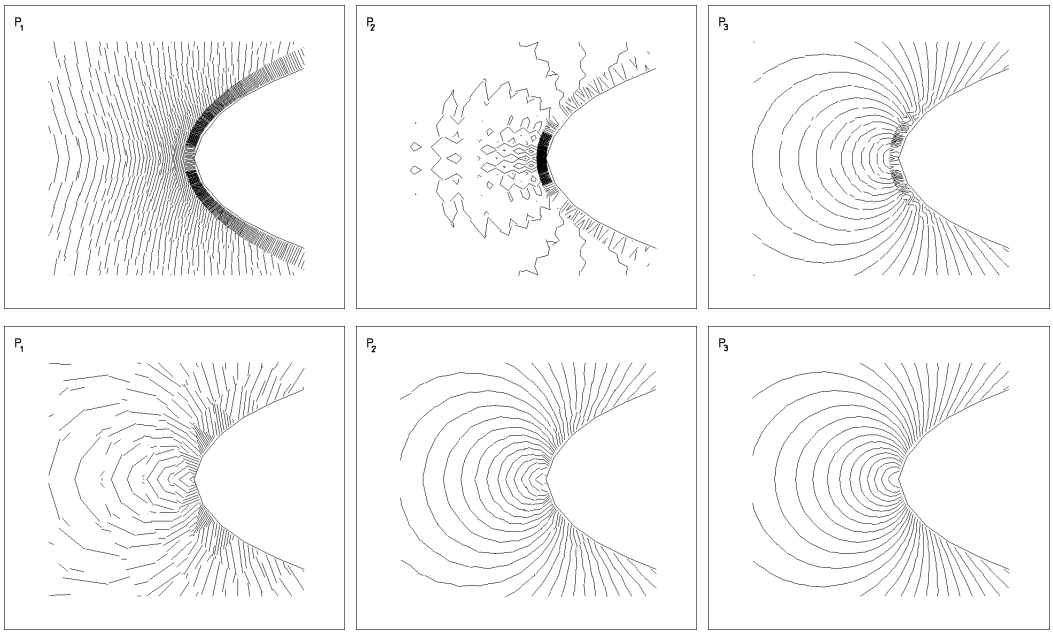

Fig. 12. Contours of normalized pressure without preconditioning at $M = 10^{-3}$. Refined quadrangular grid (top row), triangular grid (bottom row). Linear (P_1 left column), quadratic (P_2 middle column) and cubic (P_3 right column) elements.

5.2.2 Pressure fluctuations

Fig. 13 shows the pressure fluctuations $(p_{max} - p_{min})/p_{max}$ versus the Mach number for linear, quadratic and cubic elements on quadrangular (left) and triangular (right) grids without preconditioning. From the plots, we observe that the pressure fluctuations on the quadrangular grid do not scale with the square of the Mach number as they should do. The accuracy of solution deteriorates as the Mach number goes to zero. Nevertheless, the high order approximation allows to obtain more accurate results. In contrast to the lack of accuracy shown for the quadrangular grid, the pressure fluctuations on the triangular

grid (right) are proportional to the square of the Mach number. In particular, there is a very good agreement between numerical and theoretical results using P_2 and P_3 elements, whereas the P_1 pressure fluctuations are slightly less accurate. Fig. 14 shows the pressure fluctuations $(p_{max} - p_{min})/p_{max}$ versus the Mach number with preconditioning. Comparing corresponding plots in Fig. 13 and Fig. 14 we see that the preconditioning improves the accuracy of the solutions, especially on the quadrangular grid. In perfect agreement with the theory, the pressure fluctuations scale exactly with the square of the Mach number for all spatial discretizations.

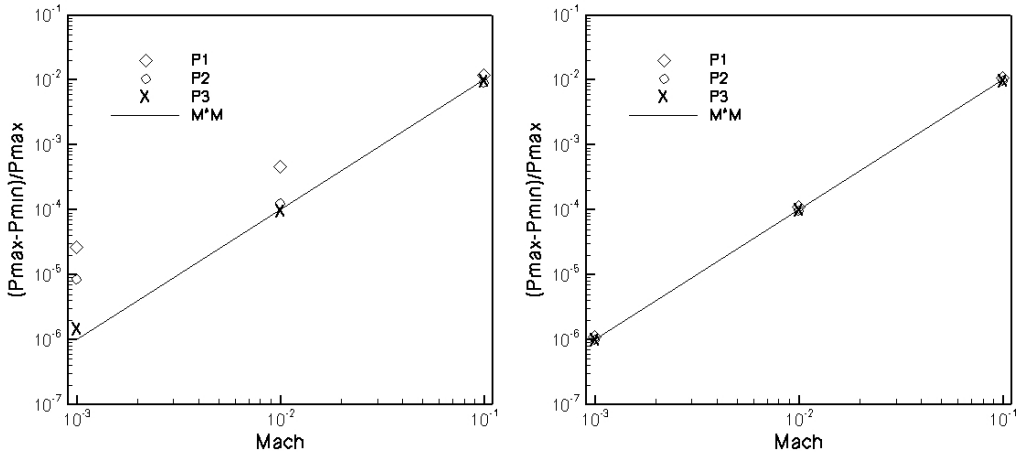

Fig. 13. Pressure fluctuations vs. Mach number for linear (P_1), quadratic (P_2) and cubic (P_3) elements without preconditioning. Quadrangular grid (left), triangular grid (right). For comparison, the theoretical behavior, M^2, is represented by a solid line.

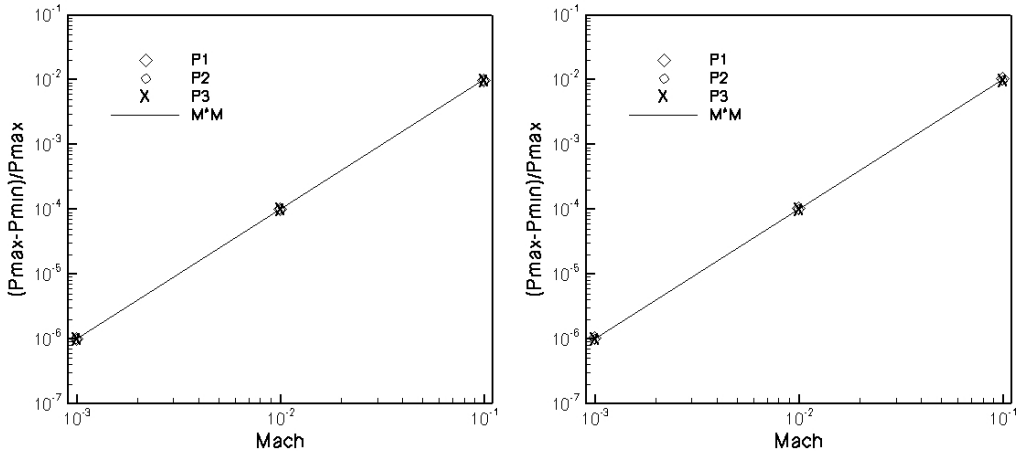

Fig. 14. Pressure fluctuations vs. Mach number for linear (P_1), quadratic (P_2) and cubic (P_3) elements with preconditioning. Quadrangular grid (left), triangular grid (right). For comparison, the theoretical behavior, M^2, is represented by a solid line.

5.2.3 *Drag coefficients*

In this section we evaluate the accuracy of the preconditioned and non-preconditioned solutions in terms of computed drag coefficients. In Tables 1 and 2 we 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. In particular, Table 1 refers to the quadrangular grid while Table 2 refers to the triangular grid.

For both spatial discretizations the preconditioning always improves the accuracy of solution, making the drag coefficients independent of the Mach number. Some differences are present at $M = 10^{-1}$ due to compressibility effects [29]. Furthermore the non-preconditioned drag coefficients show that in the low Mach number limit accurate solutions on a relatively coarse grid can be obtained only if a higher order polynomial discretization is employed.

Finally, we observe that the improvement of the accuracy due to the preconditioning is more marked for the computations performed on the quadrangular grid. In this respect, it is worth noting that, according to results shown in the previous section, the DG discretization on the triangular grid yields drag coefficients remarkably accurate and almost independent of the Mach number even without preconditioning. Notwithstanding, the computational effort needed for the convergence of the drag coefficient using the preconditioning algorithm is significantly lower than that without preconditioning.

	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 1
Drag-coefficients on the quadrangular grid.

	Triangular grid					
	$M = 10^{-1}$		$M = 10^{-2}$		$M = 10^{-3}$	
	Non-Prec.	Prec.	Non-Prec.	Prec.	Non-Prec.	Prec.
P_1	$7.183 \cdot 10^{-4}$	$4.979 \cdot 10^{-4}$	$8.026 \cdot 10^{-4}$	$4.988 \cdot 10^{-4}$	$8.130 \cdot 10^{-4}$	$4.988 \cdot 10^{-4}$
P_2	$3.290 \cdot 10^{-5}$	$2.701 \cdot 10^{-5}$	$3.472 \cdot 10^{-5}$	$2.710 \cdot 10^{-5}$	$3.490 \cdot 10^{-5}$	$2.710 \cdot 10^{-5}$
P_3	$1.038 \cdot 10^{-5}$	$7.511 \cdot 10^{-6}$	$1.102 \cdot 10^{-5}$	$7.519 \cdot 10^{-6}$	$1.108 \cdot 10^{-5}$	$7.519 \cdot 10^{-6}$

Table 2
Drag-coefficients on the triangular grid.

6 Conclusions

In this work we have presented the main features of a preconditioned DG discretization for inviscid low Mach number computations. The method solves the conservative Euler equations in terms of primitive variables using both an explicit and an implicit scheme for the temporal discretization. The algorithm employs the low Mach number preconditioning of both the time-derivative term of the governing equations and of the numerical flux function using the explicit time integration, and the preconditioning of numerical flux function only for the implicit scheme. Numerical results have been presented solving the 2D compressible Euler equations at low Mach numbers. Computations were performed at different low Mach numbers using linear, quadratic and cubic elements on quadrangular and triangular grids. In all cases, the method significantly improves the speed of convergence. In particular, the implicit scheme turns out to be efficient using just the modified numerical flux function, and then it could also be used for time accurate computations. Furthermore, it has been shown that preconditioning enhances the accuracy of the numerical solution. In particular, the computations indicate that the preconditioning of the upwind numerical flux function is mandatory to obtain accurate solutions on a relatively coarse quadrangular grid. In contrast to that, the DG discretization on the triangular grid yields remarkably accurate solutions even without preconditioning. A theoretical investigation of these results is the subject of ongoing work.

References

[1] G. Volpe. Performance of compressible flow codes at low mach numbers. AIAA Journal, 31:49–56, 1993.

[2] E. Turkel, A. Fiterman, B. van Leer. Preconditioning and the limit of the compressible to the incompressible flow equations for finite difference schemes. In: Caughey DA, Hafez MM, editors. Forntiers of Computational Fluid Dynamics. Wiley, Chichester, 1994:215-234, 1994.

[3] H. Guillard and C. Viozat. On the Behavior of Upwind Schemes in the Low Mach Number Limit. Computers & Fluids, Vol. 28, pp. 63-86, 1999.

[4] 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.

[5] D. Lee. Local Preconditioning of the Euler Equations and Navier-Stokes Equations. PhD Thesis, University of Michigan, 1996

[6] Y. H. Choi and C. L. Merkle. The Application of Preconditioning in Viscous Flows. J. Comput. Physics, Vol. 105, pp. 207-233, 1993.

[7] E. Turkel. Review of Preconditioning Methods for Fluid Dynamics. Applied Numerical Mathematics, Vol. 12, pp. 257-284, 1993.

[8] E. Turkel. Preconditioning Techniques in Computational Fluid Dynamics. Annu. Re. Fluid Mech., Vol. 31, pp. 385-416, 1999.

[9] D. Lee and B. Van Leer. Progress in Local Preconditioning of the Euler and Navier-Stokes Equations. AIAA Paper 93-3328, 1993.

[10] J. Weiss and W. A. Smith. Preconditioning Applied to Variable and Constant Density Flows. AIAA Journal , Vol.33, pp. 2050-2057, 1995.

[11] Radespiel, R.; Turkel, E.; Kroll, N.:Assessment of Preconditioning Methods. DLR-FB 95–29 (1995).

[12] S. H. Lee. Convergence characteristics of preconditioned Euler Equations. J. Comput. Physics, Vol. 208, pp. 266-288, 2005.

[13] D. Vigneron, G. Deliége, 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] T. Schneider, N. Botta, K. J. Geratz, R. Klein. Extension of Finite Volume Compressible Flow Solvers to Multi-dimensional, Variable Density Zero Mach Number. J. Comput. Physics, Vol. 155, pp. 248–286, 1999.

[15] R. Klein, N. Botta, Th. Schneider, C.-D. Munz, S. Roller, A. Meister, L. Hoffmann, Th. Sonar. Asymptotic adaptive methods for multi-scale problems in fluid mechanics. J. Eng. Math., Vol. 39, pp. 261–343, 2001.

[16] P. Moinier, M.B. Giles. Compressible Navier-Stokes Equations for Low Mach Number Applications. ECCOMAS Computational Fluid Dynamics Conference 2001, pages 1-14, Swansea, Wales, UK, September 4-7, 2001.

[17] B. Müller, Influence of Inlet and Outlet Boundary Conditions in Internal Flow Computations at Low Mach Numbers, in Desideri, J.-A., Hirsch,C., Le Tallec,P., Pandolfi,M., Periaux,J. (Eds.), "Computational Fluid Dynamics '96", Proc. 3rd ECCOMAS CFD Conference, Paris, 9-13 Sept. 1996, John Wiley, Chichester, 1996, pp. 637-643.

[18] H. Luo and J. Baum, R. Lohner. A Fast p-Multigrid Discontinuous Galerkin Method for Compressible Flows at All Speeds. 44th AIAA Aerospace Sciences Meeting and Exhibit, Reeno, 2006.

[19] M. Feistauer, V. Kucera. On a robust discontinuous Galerkin technique for the solution of compressible flow. J. Comput. Physics, Vol 224, Issue 1, pp. 208-221, 2007.

[20] E. Turkel,. Preconditioned Methods for Solving Incompressible and Low Speed Compressible Equations. J. Comput. Physics, Vol. 72, pp. 277-298, 1987

[21] D. L. Whitfield and J. M. Janus. Three-dimensional unsteady Euler equations solution using flux vector Splitting. AIAA Paper, 84-1552, 1984.

[22] P.L. Roe. Characteristic Based Schemes for the Euler Equations. Annual Review of Fluid Mechanics,Vol. 18, pp. 337-365, 1986.

[23] P.L. Roe. Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. J. Comput. Physics., Vol 43, pp. 357-352, 1981.

[24] R. J. Spiteri and S.J. Ruuth. A new Class of Optimal High-Horder Strong-Stability-Presevring Time Discretization Methods, Technical Report CS-2001-01, Acadaia University, Wolfville, Nova Scotia, Canada, 2001.

[25] S. Balay and K. Buschelman and W. D. Gropp and D. Kaushik and M. G. Knepley and L. Curfman McInnes and B. F. Smith and H. Zhang. `PETSc` Web page. http://www.mcs.anl.gov/petsc, 2001.

[26] A. Nigro, Discontinuous Galerkin methods for inviscid low Mach number flows. DLR-IB 124–2008/1.

[27] D. Lee. Criteria for local Euler Preconditioning. J. Comput. Physics, Vol. 144, pp. 423-459, 1998.

[28] F. Rieper. The influence of cell geometry on the accuracy of upwind schemes in the low Mach number regime. J. Comput. Physics, accepted for publication, 2008.

[29] R. Heinrich. Implementation of a preconditioning technique in the hybrid flow solver TAU. PAMM, Vol. 1, pp. 337-338, 2002.