ELSEVIER

Comput. Methods Appl. Mech. Engrg. 190 (2001) 5719–5737

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

The solution of the compressible Euler equations at low Mach numbers using a stabilized finite element algorithm

J.S. Wong, D.L. Darmofal, J. Peraire *

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Room 37-451, Cambridge, MA 02139, USA

Received 28 December 2000

Abstract

We present a streamline-upwind/Petrov–Galerkin (SUPG) algorithm for the solution of the compressible Euler equations at low Mach numbers. The Euler equations are written in terms of entropy variables which result in Jacobian matrices which are symmetric. We note that, in the low Mach number limit, the SUPG method with the standard choices for the stabilization matrix fail to provide adequate stabilization. This results in a degradation of the solution accuracy. We propose a stabilization matrix which incorporates dimensional-scaling arguments and which exhibits the appropriate behavior for low Mach numbers. To guide in the derivation of the new stabilization matrix, the non-dimensionalized equations are transformed to a new set of variables that converge, when the characteristic Mach number tends to zero, to the incompressible velocity and pressure variables. The resulting algorithm is capable of accurately computing external flows with free stream Mach numbers as low as 10^{-3}.

Keywords: Euler equations; Low Mach number; Preconditioning; SUPG; Entropy variables

1. Introduction

For low Mach numbers, the compressible Euler and Navier–Stokes equations describe almost incompressible flow. This singular limit of the compressible flow equations is reasonably well understood. Indeed, under the assumption of isentropic flow, and some regularity conditions on the initial and boundary data, the solution of the compressible equations, in the zero Mach number limit, can be shown to satisfy the incompressible flow equations [10,15]. From the computational point of view, accurate solutions of nearly incompressible flows are difficult to obtain. This is due to the very different magnitude of the wave speeds which are present in the system. Our interest is in the development of a compressible flow algorithm which is capable of solving flows ranging from almost incompressible to supersonic regimes. There are several reasons that justify the development of such algorithm. The first and most important is that in many situations, such as high angle of attack aerodynamics, large regions of very low Mach number coexist in the flow domain with regions where the flow is supersonic. Another more practical motivation is that an algorithm that can successfully handle free stream Mach numbers as low as 10^{-3} is well suited for a broad range of applications typically handled with incompressible formulations.

Over the last few years, stabilized finite element algorithms for the solution of the Euler and Navier–Stokes equations have gained increased popularity. Streamline-upwind/Petrov–Galerkin algorithms (SUPG), and some of its relatives, have been presented and analyzed in numerous papers [13,14,19]. These algorithms, although not so widely used as their finite volume counterparts, possess many attractive features. In particular, they have a compact support and can be used with elements of arbitrary order, yielding

* Corresponding author.

0045-7825/01/$ - see front matter
PII: S0045-7825(01)00193-1

solutions of increased accuracy whenever the solution is sufficiently smooth. In addition, there exists a rather well-developed theory for linear problems which can be used to provide design criteria for the development of successful algorithms for non-linear equations.

One of the critical ingredients of SUPG algorithms is the construction of the stabilization matrix τ. Whilst the convergence analysis for the linear problem only dictates that this matrix has to be symmetric positive definite, scale appropriately with the local grid size, and have dimensions of time, much freedom is still available to fully determine it. Only under some very simplified cases (e.g. one-dimensional flow) is the optimal choice of τ unambiguous.

Classical compressible flow formulations, including the existing SUPG algorithms, fail to give adequate numerical solutions when the flow approaches incompressibility. Whenever the Mach number is progressively reduced, keeping the grid size fixed, a degradation in the solution accuracy is observed. This phenomenon has been studied extensively in the context of finite volume algorithms [5,17,22], and several explanations have been proposed. The most common argument is based on the mismatch which occurs, for low Mach numbers, between magnitude of the fluxes in the original equations and the corresponding terms in the numerically added artificial viscosity. Some local preconditioning strategies, aimed at modifying the numerical artificial viscosity to avoid this mismatch, have proven to be extremely successful [2,3,23,24]. Accurate solutions for Mach numbers as low as 10^{-3} have been reported using such preconditioned algorithms. One of the main drawbacks of these preconditioners is their lack of robustness near stagnation points. The reason for this can be traced [4] to a lack of stabilization caused by the eigenvectors of the artificial dissipation matrix becoming nearly parallel.

The formulation of numerical algorithms for the compressible flow equations employing symmetrizing, or entropy, variables has been advocated by several authors, e.g. [1,12]. One of the claims often made is that discrete schemes formulated in entropy variables inherit global entropy stability properties of the original equations. Gustafsson [9] points out that, for a hyperbolic system of equations, the higher the degree of unsymmetry, as measured by the condition number of the transformation required to symmetrize the system, the lesser the well-posedness of the problem, in the sense that perturbations of the initial data influence the solution more. As it turns out, the compressible Euler equations formulated in terms of either primitive or conservative variables become increasingly unsymmetric when the reference Mach number goes to zero. From our perspective, one of the main attractive features derived from the use of entropy variables is that the dissipation operator remains symmetric. This means that a full set of orthogonal eigenvectors can always be found and therefore the stabilization terms remain effective throughout the computational domain. In this respect, the combination of the preconditioning ideas presented in the finite volume context to deal with low Mach number flows combined with a formulation in entropy variables seems very appealing.

In this paper, we propose a specific construction of the stabilization matrix τ for a finite element SUPG formulation of the steady-state Euler equations. The development of such stabilization matrix incorporates low Mach number preconditioning concepts, previously developed in the finite volume context. The ideas presented here extend to the time-dependent and the Navier–Stokes equations in a straightforward manner. The equations are first transformed into a new set of variables which, in the low Mach number limit, can be easily related to the incompressible velocity and pressure. A necessary condition for stability, when the Mach number tends to zero, is obtained by requiring that the asymptotic behavior of the stabilization terms matches that of the terms in the Euler equations. This requirement results in some additional constraints on τ which are not typically satisfied by the classical choices of τ. The proposed algorithm combines the very attractive features of the stabilized finite element formulations with the ability to produce accurate answers over a very large range of Mach numbers.

The paper begins with a brief description of the SUPG formulation for the Euler equations using entropy variables. For simplicity of presentation we will consider the two-dimensional case, but results herein presented extend directly to three dimensions and are also applicable to the Navier–Stokes equations. Next, we discuss the low Mach number scaling arguments which lead to the proposed form of stabilization matrix τ and outline our numerical implementation. Finally, numerical results using linear elements are presented for the flow over isolated cylinders and airfoils with free stream Mach numbers ranging from 10^{-3} to moderate subsonic values. In a forthcoming paper [25], we present results for transonic flows using higher-order elements and a discontinuity-capturing operator.

2. Compressible flow governing equations

2.1. Conservation variables

We start from the time-dependent two-dimensional compressible Euler equations in conservation form

$$\mathbf{U}_{,t} + \mathbf{F}_{1,1} + \mathbf{F}_{2,2} = 0, \tag{1}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{bmatrix}, \quad \mathbf{F}_1 = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ u_1(\rho E + p) \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ u_2(\rho E + p) \end{bmatrix}.$$

In the above expressions, ρ is the density; $\mathbf{u} = [u_1, u_2]^{\mathrm{T}}$ is the velocity vector; E is the specific total energy; p is the pressure; and the comma denotes partial differentiation (e.g. $\mathbf{U}_{,t} = \partial\mathbf{U}/\partial t$, the partial derivative with respect to time, $\mathbf{F}_{i,j} = \partial\mathbf{F}_i/\partial x_j$, the partial derivative with respect to the jth spatial coordinate). The system of equations is closed once the pressure is related to the problem variables through the equation of state, $p = (\gamma - 1)\rho e$, where $e = E - |\mathbf{u}|^2/2$, is the internal energy. Here, γ is the ratio of specific heats which is assumed to be constant.

We will assume that all the above quantities have been non-dimensionalized using reference, or free stream, values for density ρ_*, velocity u_*, and length L. Thus, the dimensional variables, denoted with an overbar, are related to the non-dimensional variables introduced above as:

$$\rho = \frac{\bar{\rho}}{\rho_*}, \quad u_i = \frac{\bar{u}_i}{u_*}, \quad i = 1, 2,$$
$$p = \frac{\bar{p}}{\rho_* u_*^2}, \quad E = \frac{\bar{E}}{u_*^2}, \quad x_i = \frac{\bar{x}_i}{L}, \quad i = 1, 2,$$
$$t = \frac{u_* \bar{t}}{L}.$$

Finally, we introduce the reference speed of sound c_*, and define, for later use, a reference Mach number as $\epsilon = u_*/c_*$.

We note that the equation system (1) can be written as

$$\mathbf{U}_{,t} + \mathbf{A}_1\mathbf{U}_{,1} + \mathbf{A}_2\mathbf{U}_{,2} = 0, \tag{2}$$

where the Jacobian matrices $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{U}}$, $i = 1, 2$, are unsymmetric but have real eigenvalues and a complete set of eigenvectors.

2.2. Entropy variables

We seek a new set of variables $\mathbf{V}$, called entropy variables, such that the change $\mathbf{U} = \mathbf{U}(\mathbf{V})$ applied to (1) give the transformed system

$$\mathbf{U}(\mathbf{V})_{,t} + \mathbf{F}_1(\mathbf{V})_{,1} + \mathbf{F}_2(\mathbf{V})_{,2} = 0, \tag{3}$$

where $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{V}}$ is symmetric positive definite, and $\tilde{\mathbf{A}}_i = \mathbf{A}_i\mathbf{A}_0 = \mathbf{F}_{i,\mathbf{V}}$, $i = 1, 2$, are symmetric.

Following [11], we introduce a scalar entropy function $H(\mathbf{U}) = -\rho g(s)$, where s is the non-dimensional entropy $s = \ln(p/\rho^\gamma)$. The required change of variables is obtained by taking

$$\mathbf{V} = H_{,\mathbf{U}}^{\mathrm{T}} = \frac{g'}{e}\begin{bmatrix} e(\gamma - g/g') - |\mathbf{u}|^2/2 \\ u_1 \\ u_2 \\ -1 \end{bmatrix}. \tag{4}$$

The conditions $g' > 0$ and $g''/g' < \gamma^{-1}$, ensure that $H(\mathbf{U})$ is a convex function and therefore $\mathbf{A}_0^{-1} = \mathbf{V}_{,\mathbf{U}} = H_{,\mathbf{UU}}$, and $\mathbf{A}_0$, are symmetric positive definite. We consider in Appendix A the variables resulting from two particular choices of the function $g(s)$. The system (3), can thus be written in symmetric quasi-linear form as

$$\mathbf{A}_0\mathbf{V}_{,t} + \tilde{\mathbf{A}}_1\mathbf{V}_{,1} + \tilde{\mathbf{A}}_2\mathbf{V}_{,2} = 0. \tag{5}$$

Barth [1] noted that it is always possible to construct matrices $\tilde{\mathbf{R}}_i$, $i = 1, 2$ whose columns are the scaled right eigenvectors of $\mathbf{A}_i$, $i = 1, 2$, respectively, such that

$$\mathbf{A}_0 = \tilde{\mathbf{R}}_1\tilde{\mathbf{R}}_1^{\mathrm{T}} = \tilde{\mathbf{R}}_2\tilde{\mathbf{R}}_2^{\mathrm{T}}, \tag{6}$$

and

$$\mathbf{A}_i = \tilde{\mathbf{R}}_i\mathbf{\Lambda}_i\tilde{\mathbf{R}}_i^{-1}, \quad \tilde{\mathbf{A}}_i = \tilde{\mathbf{R}}_i\mathbf{\Lambda}_i\tilde{\mathbf{R}}_i^{\mathrm{T}} \quad \text{for } i = 1, 2. \tag{7}$$

An explicit expression for $\tilde{\mathbf{R}}_i$, $i = 1, 2$, is given in Appendix B.

3. Low Mach number limit

The low Mach number limit of the Euler equations is best studied by rewriting the equations in terms of the variables (p, u_1, u_2, s) [21]. This yields the following system:

$$\begin{bmatrix} \frac{1}{\rho c}\frac{\partial p}{\partial t} \\ \frac{\partial u_1}{\partial t} \\ \frac{\partial u_2}{\partial t} \\ \frac{\partial s}{\partial t} \end{bmatrix} + \begin{bmatrix} u_1 & c & 0 & 0 \\ c & u_1 & 0 & 0 \\ 0 & 0 & u_1 & 0 \\ 0 & 0 & 0 & u_1 \end{bmatrix} \begin{bmatrix} \frac{1}{\rho c}\frac{\partial p}{\partial x_1} \\ \frac{\partial u_1}{\partial x_1} \\ \frac{\partial u_2}{\partial x_1} \\ \frac{\partial s}{\partial x_1} \end{bmatrix} + \begin{bmatrix} u_2 & 0 & c & 0 \\ 0 & u_2 & 0 & 0 \\ c & 0 & u_2 & 0 \\ 0 & 0 & 0 & u_2 \end{bmatrix} \begin{bmatrix} \frac{1}{\rho c}\frac{\partial p}{\partial x_2} \\ \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_2} \\ \frac{\partial s}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8}$$

In matrix form, this system can be written as

$$\mathbf{Z}_{,t} + \mathbf{C}_1\mathbf{Z}_{,1} + \mathbf{C}_2\mathbf{Z}_{,2} = 0, \tag{9}$$

where the matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are symmetric and $\mathrm{d}\mathbf{Z} = [\mathrm{d}\phi, \mathrm{d}u_1, \mathrm{d}u_2, \mathrm{d}s]^{\mathrm{T}}$ with $\mathrm{d}\phi = \mathrm{d}p/\rho c$. If we assume that the flow is isentropic, which is certainly satisfied for low Mach numbers when the free stream is isentropic, we can express c and p as a function of ρ, i.e.

$$p = \frac{\rho^\gamma}{\gamma\epsilon^2} \quad \text{and} \quad c = \frac{\rho^{(\gamma-1)/2}}{\epsilon},$$

to obtain [8],

$$\phi = \frac{2}{(\gamma-1)\epsilon}(\rho^{(\gamma-1)/2} - 1) = \frac{2}{\gamma-1}\left(c - \frac{1}{\epsilon}\right).$$

In terms of ϕ, u_1 and u_2, the continuity and momentum isentropic compressible flow equations become

$$\begin{aligned} &\phi_{,t} + u_1\phi_{,1} + \left(\frac{\gamma-1}{2}\phi + \frac{1}{\epsilon}\right)u_{1,1} + u_2\phi_{,2} + \left(\frac{\gamma-1}{2}\phi + \frac{1}{\epsilon}\right)u_{2,2} = 0, \\ &u_{1,t} + \left(\frac{\gamma-1}{2}\phi + \frac{1}{\epsilon}\right)\phi_{,1} + u_1u_{1,1} + u_2u_{1,2} = 0, \\ &u_{2,t} + u_1u_{2,1} + \left(\frac{\gamma-1}{2}\phi + \frac{1}{\epsilon}\right)\phi_{,2} + u_2u_{2,2} = 0. \end{aligned} \tag{10}$$

Note that, in assuming isentropic flow, we have reduced the number of dependent variables by eliminating entropy from the system of governing equations. Eq. (10), look now similar to the incompressible flow equations

$$\begin{aligned}
&\hat{u}_{1,1} + \hat{u}_{2,2} = 0,\\
&\hat{u}_{1,t} + \hat{p}_{,1} + \hat{u}_1\hat{u}_{1,1} + \hat{u}_2\hat{u}_{1,2} = 0,\\
&\hat{u}_{2,t} + \hat{u}_1\hat{u}_{2,1} + \hat{p}_{,2} + \hat{u}_2\hat{u}_{2,2} = 0,
\end{aligned} \tag{11}$$

where $\hat{\mathbf{u}} = [\widehat{u_1}, \widehat{u_2}]^{\mathrm{T}}$ denotes the velocity and $\hat{p}$ the pressure. When $\epsilon \to 0$, the equivalence between (10) and (11), under some regularity assumptions on the initial and boundary data, can be rigorously shown [10]. In particular, we have that for $\epsilon \to 0$, $u_i \to \widehat{u_i}$ and $((\gamma-1)\phi/2 + 1/\epsilon)\phi_{,i} \to \hat{p}_{,i}$ for $i = 1, 2$. To obtain bounded derivatives when $\epsilon \to 0$, it follows that $\phi_{,i}$ and $u_{1,1} + u_{2,2}$ must be $\mathrm{O}(\epsilon)$. We also note that the velocity components u_i and its spatial derivatives $u_{i,j}$, $i, j = 1, 2$ are $\mathrm{O}(1)$. Finally, it can be shown that ϕ is $\mathrm{O}(\epsilon)$ and that ϕ/ϵ well defined in the limit $\epsilon \to 0$.

We now turn our attention to the steady-state Euler equations for isentropic flow and derive some asymptotic estimates which are valid in the low Mach number limit. The equations for this reduced problem are

$$\mathbf{C}_1^I \mathbf{Z}_{,1}^I + \mathbf{C}_2^I \mathbf{Z}_{,2}^I = 0, \tag{12}$$

where

$$\mathbf{Z}^I = \begin{bmatrix} \phi \\ u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{C}_1^I \begin{bmatrix} u_1 & c & 0 \\ c & u_1 & 0 \\ 0 & 0 & u_1 \end{bmatrix}, \quad \mathbf{C}_2^I = \begin{bmatrix} u_2 & 0 & c \\ 0 & u_2 & 0 \\ c & 0 & u_2 \end{bmatrix}.$$

Using the above estimates we have that, for $\epsilon \to 0$,

$$\mathbf{Z}_{,1}^I \sim \mathbf{Z}_{,2}^I \sim \begin{bmatrix} \mathrm{O}(\epsilon) \\ \mathrm{O}(1) \\ \mathrm{O}(1) \end{bmatrix}, \quad \mathbf{C}_1^I \sim \begin{bmatrix} \mathrm{O}(1) & \mathrm{O}(\epsilon^{-1}) & 0 \\ \mathrm{O}(\epsilon^{-1}) & \mathrm{O}(1) & 0 \\ 0 & 0 & \mathrm{O}(1) \end{bmatrix}, \quad \mathbf{C}_2^I \sim \begin{bmatrix} \mathrm{O}(1) & 0 & \mathrm{O}(\epsilon^{-1}) \\ 0 & \mathrm{O}(1) & 0 \\ \mathrm{O}(\epsilon^{-1}) & 0 & \mathrm{O}(1) \end{bmatrix}. \tag{13}$$

Thus, for the individual terms of the isentropic Euler equations,

$$\mathbf{C}_1^I \mathbf{Z}_{,1}^I \sim \mathbf{C}_2^I \mathbf{Z}_{,2}^I \sim \begin{bmatrix} \mathrm{O}(\epsilon^{-1}) \\ \mathrm{O}(1) \\ \mathrm{O}(1) \end{bmatrix} \quad \text{as } \epsilon \to 0. \tag{14}$$

4. Variational formulation for the steady-state problem

We now consider the compressible steady problem in conservation form expressed in terms of symmetrizing variables. The conservative form of the equations is taken to be the starting point because we are ultimately interested in an algorithm that can be used over the whole range of speed regimes, including situations where the solution may contain discontinuities. The problem is defined in a domain Ω with boundary Γ by

$$\mathbf{F}_1(\mathbf{V})_{,1} + \mathbf{F}_2(\mathbf{V})_{,2} = 0 \quad \text{in } \Omega, \tag{15}$$

$$\tilde{\mathbf{A}}_n^- \mathbf{V} = \tilde{\mathbf{A}}_n^- \mathbf{g} \quad \text{on } \Gamma \setminus \Gamma_a, \tag{16}$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_a. \tag{17}$$

For simplicity, the domain boundary is assumed to be made up of an impermeable solid wall Γ_a, and a computational far field boundary $\Gamma \setminus \Gamma_a$. In (16) and (17), $\mathbf{n} = [n_1, n_2]^{\mathrm{T}}$ is the outward unit normal vector to Γ, and $\tilde{\mathbf{A}}_n = \mathbf{A}_n\mathbf{A}_0$, $\mathbf{A}_n = \mathbf{A}_1 n_1 + \mathbf{A}_2 n_2$. Finally, $\tilde{\mathbf{A}}_n^- = \mathbf{A}_n^-\mathbf{A}_0$, and $\mathbf{A}_n^-$ denotes the negative definite part of $\mathbf{A}_n$.

Let the spatial domain Ω, be discretized into non-overlapping elements T_e, such that $\Omega = \cup T_e$, and $T_e \cap T_{e'} = \emptyset$, $e \neq e'$. We consider the space of functions $\mathscr{V}_h$, defined over the discretization and consisting of the continuous functions which are piecewise linear over each element

$$\mathscr{V}_h = \{\mathbf{W} \,|\, \mathbf{W} \in (C^0(\Omega))^4, \mathbf{W}|_{T_e} \in (\mathscr{P}_1(T_e))^4 \ \forall T_e \in \Omega\}.$$

The SUPG algorithm can then be written as: Find $\mathbf{V}_h \in \mathscr{V}^h$ such that for all $\mathbf{W} \in \mathscr{V}^h$,

$$B(\mathbf{V}_h, \mathbf{W})_{\text{gal}} + B(\mathbf{V}_h, \mathbf{W})_{\text{supg}} + B(\mathbf{V}_h, \mathbf{W})_{\text{bc}} = 0, \tag{18}$$

where the forms $B(\cdot,\cdot)_{\text{gal}}, B(\cdot,\cdot)_{\text{supg}}$ and $B(\cdot,\cdot)_{\text{bc}}$ account for the Galerkin, SUPG stabilization, and boundary condition terms, respectively, and are defined as:

$$B(\mathbf{V}, \mathbf{W})_{\text{gal}} = \int_\Omega (-\mathbf{W}_{,1} \cdot \mathbf{F}_1(\mathbf{V}) - \mathbf{W}_{,2} \cdot \mathbf{F}_2(\mathbf{V}))\, \mathrm{d}\Omega, \tag{19}$$

$$B(\mathbf{V}, \mathbf{W})_{\text{supg}} = \int_\Omega (\tilde{\mathbf{A}}_1 \mathbf{W}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{W}_{,2}) \cdot \boldsymbol{\tau} (\tilde{\mathbf{A}}_1 \mathbf{V}_{,1} + \tilde{\mathbf{A}}_2 \mathbf{V}_{,2})\, \mathrm{d}\Omega, \tag{20}$$

and

$$B(\mathbf{V}, \mathbf{W})_{\text{bc}} = \int_{\Gamma_a} \mathbf{W} \cdot \mathbf{F}_{\text{w}}(\mathbf{V}; \mathbf{n})\, \mathrm{d}s + \int_{\Gamma \setminus \Gamma_a} \mathbf{W} \cdot \mathbf{F}_{\text{ff}}(\mathbf{V}, \mathbf{g}; \mathbf{n})\, \mathrm{d}s, \tag{21}$$

where $\boldsymbol{\tau}$ is the stabilization matrix. The numerical flux function on the impermeable wall boundary $\mathbf{F}_{\text{w}}$, is simply $[0, pn_1, pn_2, 0]^{\text{T}}$ while the numerical flux function on the far field boundary $\mathbf{F}_{\text{ff}}$, is defined by

$$\mathbf{F}_{\text{ff}}(\mathbf{V}_-, \mathbf{V}_+; \mathbf{n}) = \frac{1}{2}(\mathbf{F}_n(\mathbf{V}_-) + \mathbf{F}_n(\mathbf{V}_+)) - \frac{1}{2}|\mathbf{A}_n(\mathbf{V}_{\text{Roe}}(\mathbf{V}_-, \mathbf{V}_+))|(\mathbf{U}(\mathbf{V}_+) - \mathbf{U}(\mathbf{V}_-)).$$

Here, $|\mathbf{A}_n(\mathbf{V})| = \mathbf{A}_n^+(\mathbf{V}) - \mathbf{A}^-(\mathbf{V})$ is the absolute value of $\mathbf{A}_n$ evaluated at $\mathbf{V}$, and $\mathbf{V}_{\text{Roe}}(\mathbf{V}_+, \mathbf{V}_-)$, is the Roe average [18], between the states $\mathbf{V}^+$ and $\mathbf{V}^-$.

4.1. Standard definitions for $\boldsymbol{\tau}$

In order to uniquely specify the SUPG algorithm (18), an appropriate expression for the stabilization matrix $\boldsymbol{\tau}$ needs to be given. We recall that an appropriate $\boldsymbol{\tau}$ should be symmetric, positive definite, have dimensions of time, and scale linearly with the element size [13].

For a triangle T_e, we introduce the mapping $\mathbf{x} = \mathbf{x}(\xi)$, between the master triangle $\hat{T}_e$, in the parametric space $\boldsymbol{\xi} = [\xi_1, \xi_2]^{\text{T}}$, and $T_e \in \Omega$ in the mapped space $\mathbf{x} = [x_1, x_2]^{\text{T}}$, as illustrated in Fig. 1. We define:

$$\mathbf{B}_1 = (x_{2,\xi_2}\tilde{\mathbf{A}}_1 - x_{1,\xi_2}\tilde{\mathbf{A}}_2)/\mathbf{J},$$

$$\mathbf{B}_2 = (x_{1,\xi_1}\tilde{\mathbf{A}}_2 - x_{2,\xi_1}\tilde{\mathbf{A}}_1)/\mathbf{J},$$

$$\mathbf{B}_3 = ((x_{2,\xi_1} - x_{2,\xi_2})\tilde{\mathbf{A}}_1 - (x_{1,\xi_2} - x_{1,\xi_1})\tilde{\mathbf{A}}_2)/\mathbf{J},$$

and the Jacobian of the mapping, $\mathbf{J} = x_{1,\xi_1}x_{2,\xi_2} - x_{2,\xi_1}x_{1,\xi_2}$. The following definition for $\boldsymbol{\tau}$ can be shown to satisfy all of the above requirements:

$$\boldsymbol{\tau} = \mathbf{A}_0^{-1}(|\mathbf{B}_1|^p + |\mathbf{B}_2|^p + |\mathbf{B}_3|^p)^{-1/p}. \tag{22}$$

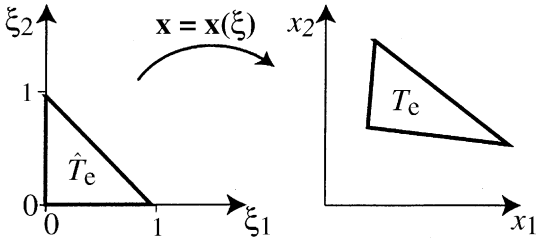

Fig. 1. Mapping between the master triangle $\hat{T}_e$ and a typical element T_e.

The most common choice for p has been 2, which necessitates the evaluation of a matrix square root. This choice has been advocated by Shakib et al. [19], whereas Barth [1], has shown that the choice $p = 1$ is computationally advantageous and, in practice, gives very similar results.

These choices of stabilization matrix, work well provided that the flow Mach number is not too low. For problems involving significant regions of low Mach number flow the solution degrades and becomes less and less accurate as the Mach number is decreased. In the following sections we will address this issue by first, identifying the source of the problem, and second, devising a new formulation for $\boldsymbol{\tau}$ which does not suffer from this drawback and which leads to a formulation which can handle very low Mach numbers accurately.

5. A stabilized formulation for isentropic flows

Turkel et al. [22] have employed a scaling analysis to determine the appropriate low Mach number behavior of local preconditioners. In their application, the local preconditioner modifies the upwind dissipation of the underlying finite volume or finite difference discretization. In this section, we will consider a stabilized SUPG formulation applied directly to the isentropic Eq. (12). We will extend the analysis presented in [22] to our finite element formulation and, in particular, derive the appropriate scaling for the stabilization matrix in the limit of vanishing Mach number.

Introducing the finite element space,

$$\mathscr{V}_h^I = \{\mathbf{W}^I \,|\, \mathbf{W}^I \in (C^0(\Omega))^3, \mathbf{W}^I|_{T_e} \in (\mathscr{P}_1(T_e))^3 \ \forall T_e \in \Omega\}.$$

We consider the following SUPG formulation: Find $\mathbf{Z}_h^I \in \mathscr{V}_h^I$ such that for all $\mathbf{W}^I \in \mathscr{V}_h^I$,

$$B^I(\mathbf{Z}_h^I, \mathbf{W}^I)_{\text{gal}} + B^I(\mathbf{Z}_h^I, \mathbf{W}^I)_{\text{supg}} + B^I(\mathbf{Z}_h^I, \mathbf{W}^I)_{\text{bc}} = 0, \tag{23}$$

$$B^I(\mathbf{Z}^I, \mathbf{W}^I)_{\text{gal}} = \int_\Omega (\mathbf{W}^I \cdot (\mathbf{C}_1^I \mathbf{Z}_{,1}^I + \mathbf{C}_2^I \mathbf{Z}_{,2}^I))\, \mathrm{d}\Omega, \tag{24}$$

where

$$B^I(\mathbf{Z}^I, \mathbf{W}^I)_{\text{supg}} = \int_\Omega (\mathbf{C}_1^I \mathbf{W}_{,1}^I + \mathbf{C}_2^I \mathbf{W}_{,2}^I) \cdot \bar{\boldsymbol{\tau}}^I (\mathbf{C}_1^I \mathbf{Z}_{,1}^I + \mathbf{C}_2^I \mathbf{Z}_{,2}^I)\, \mathrm{d}\Omega, \tag{25}$$

and $B^I(\cdot,\cdot)_{\text{bc}}$ is a term incorporating the desired boundary conditions. Note that here, the Galerkin term (24) has been integrated by parts, thus, $B^I(\cdot,\cdot)_{\text{bc}}$ also includes the additional boundary terms.

Our desire is to construct a stabilized finite element method which admits solutions, $\mathbf{Z}_h^I$, with the same low Mach number asymptotic behavior as the solutions of the isentropic Euler equations, $\mathbf{Z}^I$. In addition, we require that the stabilization operator (25) must scale like the Galerkin and boundary operators (24) as the Mach number is reduced. Specifically, these requirements imply that

$$\mathbf{C}_1^I \mathbf{Z}_{h,1}^I + \mathbf{C}_2^I \mathbf{Z}_{h,2}^I \sim \begin{bmatrix} \mathrm{O}(\epsilon^{-1}) \\ \mathrm{O}(1) \\ \mathrm{O}(1) \end{bmatrix} \quad \text{for } \epsilon \to 0, \tag{26}$$

and that,

$$\mathbf{C}_i^I \cdot \bar{\boldsymbol{\tau}}^I (\mathbf{C}_1^I \mathbf{Z}_{h,1} + \mathbf{C}_2^I \mathbf{Z}_{h,2}) \sim \mathbf{C}_1^I \mathbf{Z}_{h,1}^I + \mathbf{C}_1^I \mathbf{Z}_{h,1}^I \quad \text{for } i = 1,2, \ \epsilon \to 0. \tag{27}$$

Expressions (26) and (27), provide an additional condition, on the asymptotic behavior of the components of the stabilization matrix $\bar{\boldsymbol{\tau}}^I$ that should be satisfied for $\epsilon \to 0$. This condition can be written as,

$$\begin{bmatrix} \mathrm{O}(1) & \mathrm{O}(\epsilon^{-1}) & \mathrm{O}(\epsilon^{-1}) \\ \mathrm{O}(\epsilon^{-1}) & \mathrm{O}(1) & 0 \\ \mathrm{O}(\epsilon^{-1}) & 0 & \mathrm{O}(1) \end{bmatrix} \begin{bmatrix} \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{13} & \tau_{23} & \tau_{33} \end{bmatrix} \begin{bmatrix} \mathrm{O}(\epsilon^{-1}) \\ \mathrm{O}(1) \\ \mathrm{O}(1) \end{bmatrix} \sim \begin{bmatrix} \mathrm{O}(\epsilon^{-1}) \\ \mathrm{O}(1) \\ \mathrm{O}(1) \end{bmatrix}. \tag{28}$$

5.1. Asymptotic behavior of $\boldsymbol{\tau}$ using standard definitions

We consider now the standard definition of the stabilization matrix, given in (22), applied to our simplified problem (23), and show that it fails to satisfy condition (28). Specifically, for one-dimensional problems, the standard stabilization (22) is, for $p = 1$,

$$\bar{\boldsymbol{\tau}}^I_{1d} = \kappa |C^I_{1d}|^{-1},$$

where κ is a constant proportional to the element size. Based on the above-assumed behavior of the solution, it can be shown that for $\epsilon \to 0$,

$$\bar{\boldsymbol{\tau}}^I_{1d} \sim \begin{bmatrix} O(\epsilon) & O(\epsilon^2) \\ O(\epsilon^2) & O(\epsilon) \end{bmatrix}. \tag{29}$$

It can easily be verified that this stabilization matrix, does not satisfy the condition (28), which in the one-dimensional case is

$$\begin{bmatrix} O(1) & O(\epsilon^{-1}) \\ O(\epsilon^{-1}) & O(1) \end{bmatrix} \begin{bmatrix} \bar{\tau}_{11} & \bar{\tau}_{12} \\ \bar{\tau}_{12} & \bar{\tau}_{22} \end{bmatrix} \begin{bmatrix} O(\epsilon^{-1}) \\ O(I) \end{bmatrix} \sim \begin{bmatrix} O(\epsilon^{-1}) \\ O(I) \end{bmatrix}. \tag{30}$$

Therefore, this form of $\bar{\boldsymbol{\tau}}^I$, will fail to provide adequate stabilization when $\epsilon \to 0$. It can also be shown that, in the multi-dimensional case, the standard choice of stabilization matrix (22), does not have the appropriate asymptotic behavior for $\epsilon \to 0$, either.

When working directly with entropy, or conservative variables, the analysis is more complicated because the link between these variables and the incompressible velocity and pressure is less transparent. It is observed in practice that the standard stabilization scheme (22), when used with the equations written in terms of entropy variables, produce solutions which deteriorate severely when the Mach number is reduced. In [7], the use of the standard finite volume upwind scheme, using Roe's dissipation, is shown to give in the incompressible limit, solutions in which the pressure variations do not scale like the Mach number squared. This first-order finite volume upwind scheme, can be thought of, at least in one dimension, as the result of using a stabilization matrix of the form (22), directly with conservative variables.

6. Alternative definition of for low Mach number flow

Our objective is to derive a stabilization matrix $\boldsymbol{\tau}$, for the variational formulation in entropy variables (18). An approach which has been advocated in the design of low Mach preconditioners for finite volume schemes [5,22], has been to derive an appropriate stabilization matrix $\bar{\boldsymbol{\tau}}$, for the system (8), and then transform it to conservative variables.

It is apparent that, even with the additional constraint given by condition (28), the stabilization matrix is not uniquely defined and some freedom is still available in constructing it. In addition to the requirements placed on the construction of the standard forms of $\bar{\boldsymbol{\tau}}$, i.e. (22), we place the following three constraints:

(i) The entropy equation, which decouples from the other equations, is stabilized as an independent quantity convected with the velocity.

(ii) The vorticity equation, which, in the incompressible limit, decouples from the other equations after taking the curl of the velocity evolution equations, is stabilized as an independent quantity convected with the velocity.

(iii) The resulting algorithm has the correct low Mach number scaling as discussed in the previous section.

In order to incorporate the above conditions, we consider the Euler equation corresponding to the variational formulation (23), augmented with the entropy equation,

$$\mathbf{C}_1\mathbf{Z}_{,1} + \mathbf{C}_2\mathbf{Z}_{,2} = (\mathbf{C}_1\bar{\boldsymbol{\tau}}(\mathbf{C}_1\mathbf{Z}_{,1} + \mathbf{C}_2\mathbf{Z}_{,2}))_{,1} + (\mathbf{C}_2\bar{\boldsymbol{\tau}}(\mathbf{C}_1\mathbf{Z}_{,1} + \mathbf{C}_2\mathbf{Z}_{,2}))_{,2}. \tag{31}$$

We shall further assume that, only for the purpose of deriving $\boldsymbol{\tau}$, the matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are locally constant and therefore the above system of equations can be treated as if it was linear.

Requirement (i) forces the last row and column of $\bar{\boldsymbol{\tau}}$ to be zero, except for the diagonal entry which corresponds to a velocity time scale. Requirement (ii) implies that the rest of the matrix must also be of diagonal form. The entries in the second and third rows must be equal and are also associated with a velocity time scale. Therefore, we find that $\bar{\boldsymbol{\tau}}$, for the complete system must be of the form

$$\bar{\boldsymbol{\tau}} = h_e \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & b \end{bmatrix}, \tag{32}$$

where $b = 1/|\mathbf{u}|$ and h_e is the size of the element. In order to satisfy the low Mach number scaling, it is clear that $b \sim \mathrm{O}(1)$, and therefore a must be $\mathrm{O}(\epsilon^2)$. The simplest choice, having the right dimensions, is $a = |\mathbf{u}|/c^2$. For very high speed flows, the stabilization should transition to a pure streamwise upwinding such that $a \to 1/|\mathbf{u}|$. In practice, the specific definition of $\bar{\boldsymbol{\tau}}$ which we use is,

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \tau_{\mathrm{a}} & 0 & 0 & 0 \\ 0 & \tau_{\mathrm{c}} & 0 & 0 \\ 0 & 0 & \tau_{\mathrm{c}} & 0 \\ 0 & 0 & 0 & \tau_{\mathrm{c}} \end{bmatrix}, \tag{33}$$

where τ_{c} is the convective timescale and τ_{a} is the acoustic timescale. Specifically, for the convective timescale, we employed a form proposed in [16] for scalar convection,

$$\tau_{\mathrm{c}}^{-1} = \sum_{i=1}^{3} \frac{|\mathbf{l}_i \cdot \bar{\mathbf{u}}|}{\mathbf{l}_i \cdot \mathbf{l}_i},$$

where $\bar{\mathbf{u}}$ is the average velocity in the element, and $\mathbf{l}_i$ is the vector between the nodes of side i of the element. We then define the acoustic timescale as,

$$\tau_{\mathrm{a}}^{-1} = \tau_{\mathrm{c}}^{-1} + \frac{c^2}{h_e|\bar{\mathbf{u}}|}.$$

Using these definitions, the acoustic timescale, τ_{a}, has the correct low Mach number behavior required by the asymptotic analysis. Also, τ_{a} behaves appropriately for large local Mach numbers where it returns to the convective timescale τ_{c}.

Finally, the required form of $\boldsymbol{\tau}$ can be obtained by transforming the system (31) to entropy variables. In principle, this could be accomplished by using the transformation matrix $\mathbf{S} = \mathbf{V}_{,\mathbf{Z}}$. Inserting $\mathrm{d}\mathbf{Z} = \mathbf{S}^{-1}\mathrm{d}\mathbf{V}$ into (31) and multiplying through by $\mathbf{S}^{-\mathrm{T}}$, gives,

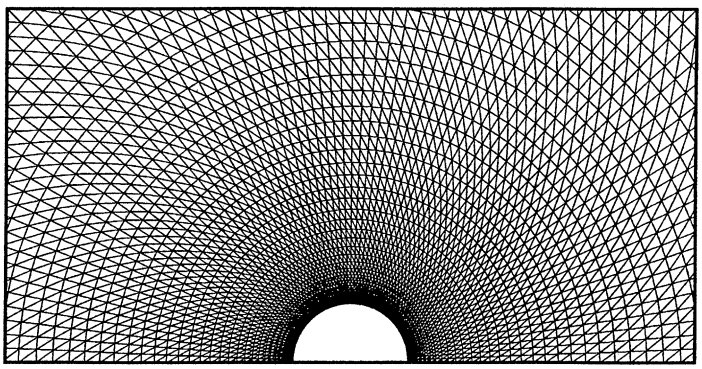

Fig. 2. Detail of the medium size mesh used for computations.

$$\mathbf{S}^{-\mathrm{T}}\mathbf{C}_1\mathbf{S}^{-1}\mathbf{V}_{,1} + \mathbf{S}^{-\mathrm{T}}\mathbf{C}_2\mathbf{S}^{-1}\mathbf{V}_{,2} = (\mathbf{S}^{-\mathrm{T}}\mathbf{C}_1\mathbf{S}^{-1}\mathbf{S}\bar{\boldsymbol{\tau}}\mathbf{S}^{\mathrm{T}}(\mathbf{S}^{-\mathrm{T}}\mathbf{C}_1\mathbf{S}^{-1}\mathbf{V}_{,1} + \mathbf{S}^{-\mathrm{T}}\mathbf{C}_2\mathbf{S}^{-1}\mathbf{V}_{,2}))_{,1} + (\mathbf{S}^{-\mathrm{T}}\mathbf{C}_2\mathbf{S}^{-1}\mathbf{S}\bar{\boldsymbol{\tau}}\mathbf{S}^{\mathrm{T}}(\mathbf{S}^{-\mathrm{T}}\mathbf{C}_1\mathbf{S}^{-1}\mathbf{V}_{,1} + \mathbf{S}^{-\mathrm{T}}\mathbf{C}_2\mathbf{S}^{-1}\mathbf{V}_{,2}))_{,2}, \tag{34}$$

where the transformed matrix $\boldsymbol{\tau}$ can be readily identified as

$$\boldsymbol{\tau} = \mathbf{S}\bar{\boldsymbol{\tau}}\mathbf{S}^{\mathrm{T}}. \tag{35}$$

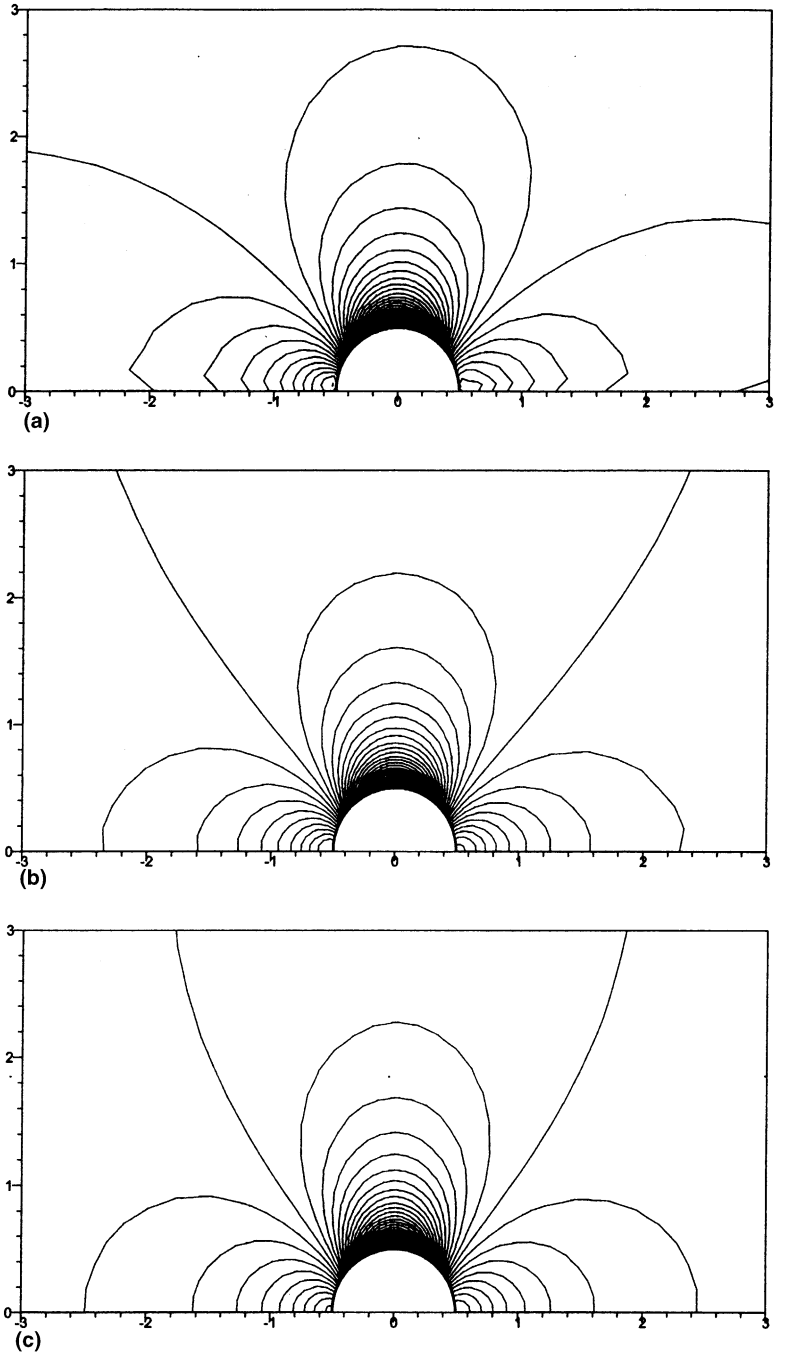

Fig. 3. Static pressure contours computed on the coarse mesh using the SUPG algorithm with the standard choice of stabilization matrix given by (22), for a free stream Mach number of (a) 0.01, (b) 0.1, and (c) 0.38.

We note that the transformed matrices $\mathbf{S}^{-\mathrm{T}}\mathbf{C}_i\mathbf{S}^{-1}$, $i = 1, 2$ are not equal to $\tilde{\mathbf{A}}_i$, $i = 1, 2$ and indeed $\mathbf{S}^{-\mathrm{T}}\mathbf{S}^{-1}$ is not equal to $\mathbf{A}_0$. This is due to the fact that the entropy equation in (8), completely decouples from the rest of the system. Therefore, one can multiply the first three components of $\mathrm{d}\mathbf{Z}$, i.e. $\mathrm{d}p/\rho c$, $\mathrm{d}u_1$, $\mathrm{d}u_2$, by a scalar factor, and the fourth component of $\mathrm{d}\mathbf{Z}$, i.e. $\mathrm{d}s$, by different scalar factor without changing the Jacobian matrices $\mathbf{C}_i$, $i = 1, 2$ or $\bar{\boldsymbol{\tau}}$, in (31). It is not hard to find the scalar factors that one should use to define modified $\mathrm{d}\mathbf{Z}$ variables so that the resulting $\mathbf{S}$ matrix would give $\mathbf{S}^{-\mathrm{T}}\mathbf{C}_i\mathbf{S}^{-1} = \tilde{\mathbf{A}}_i, i = 1, 2$. An alternative

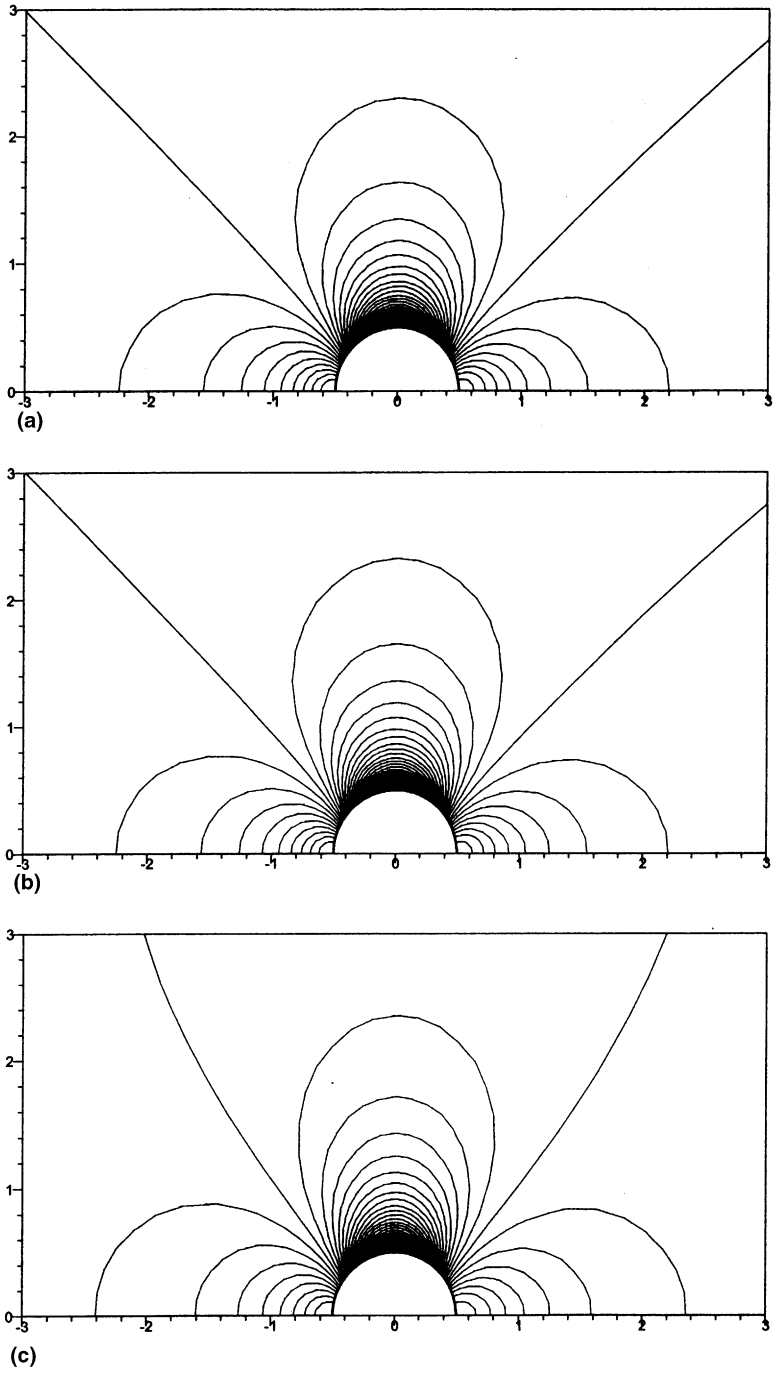

Fig. 4. Static pressure contours computed on the coarse mesh using the SUPG algorithm with the proposed choice of stabilization matrix given by (35), for a free stream Mach number of (a) 0.01, (b) 0.1, and (c) 0.38.

procedure for evaluating $\mathbf{S}$, which avoids the use of modified variables is given in Appendix C. Once the appropriate transformation matrix has been evaluated, τ is found using expression (35).

7. Numerical results

In this section we present some numerical results that illustrate the performance of the proposed algorithm. For all the examples, we have solved problem (18), which is expressed in terms of entropy variables. The non-linear set of equations resulting from the discretization of (18) is solved by a Newton–Raphson iteration using exact linearization. For some of the simulations however, it was necessary to damp the Newton–Raphson iteration during the first few iterations. The solution of the non-symmetric non-linear system of equations required for each iteration was solved using an enhanced BiCGstab(2) algorithm [6,20] together with a block ILU(k) preconditioning, with k either 0 or 1. For all the examples we have considered the two choices of entropy variables given in Appendix A, and we have not found any appreciable differences in the computed results. We have followed [1], and employed a linear representation of the solution over each triangular element, but have used a quadratic mapping to more accurately represent the geometry. We have found that using quadratic interpolation of the geometry greatly improves the quality of the solution and only incurs a minimal incremental cost.

7.1. Example 1: Flow over cylinder

In this example, we consider the flow about a cylinder and perform several numerical tests. Because of the symmetry of the problem at the low Mach numbers considered here, only half of the domain is considered for the solution. We have used a sequence of three meshes of triangular elements containing 1271, 4941, and 19,481 nodes, respectively, to discretize to computational domain. The medium and fine meshes are obtained by uniformly subdividing the coarse mesh. Fig. 2 shows a detail of the medium size mesh near the cylinder.

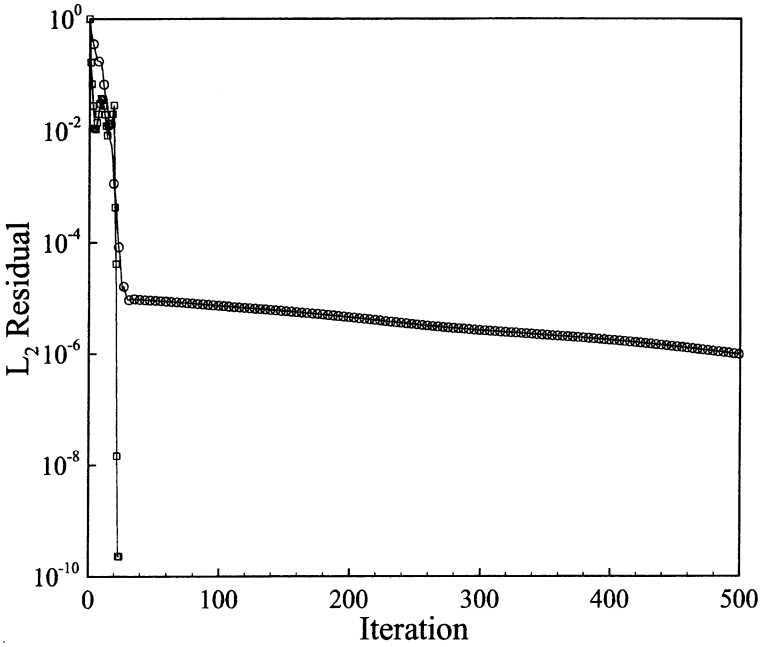

Fig. 5. Convergence history of the Newton-Rapson iteration residual for the coarse mesh solution at free stream Mach number of 0.01 using: (a) τ given by (22) (circles), and (b) τ given by (35) (squares).

In the first test, we consider the SUPG algorithm (18), with the standard choice of τ given by (22) for $p = 1$, and solve for the flow about the cylinder on the coarse mesh for free stream Mach numbers of 0.38, 0.1 and 0.01. Fig. 3 shows the pressure contours for the three solutions. The degradation in the solution accuracy when the Mach number is reduced is clearly apparent. We could not obtain any numerical solutions below a free stream Mach number of 0.01, and for this Mach number, the iteration would only converge if a damped Newton–Raphson iteration was used. For comparison purposes, we show in Fig. 4 the pressure contours obtained, for the same flow conditions and mesh, when the proposed form of τ, given

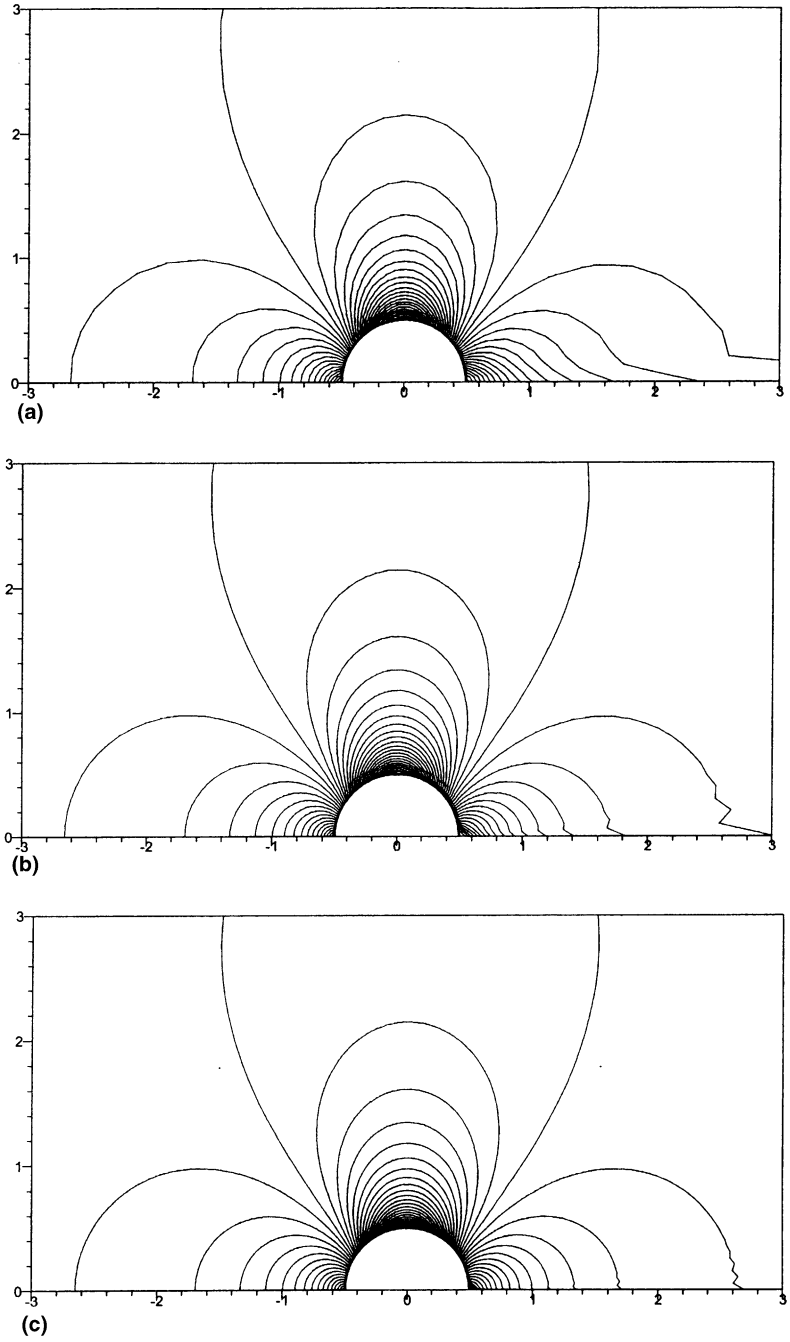

Fig. 6. Mach contours for a free stream Mach number of 0.38 computed on the: (a) coarse, (b) medium, and (c) fine meshes.

in (35), is used. No qualitative degradation of the solution is observed and in fact, we observe that, the solutions for Mach numbers of 0.1 and 0.01 look almost identical, as expected. For illustrative purposes, we show in Fig. 5, the L_2 norm of the Newton–Raphson iteration residual versus the number of iterations, for the 0.01 free stream Mach number case, using both the standard and proposed choices of τ. Quadratic convergence is obtained for the proposed τ while a severe deterioration in the convergence rate is observed for the standard choice of τ due to the necessity to use a damped iteration.

In the second test, we perform a mesh convergence analysis at various Mach numbers. Fig. 6 shows the Mach number contours computed on the coarse, medium and fine mesh for a free stream Mach number of

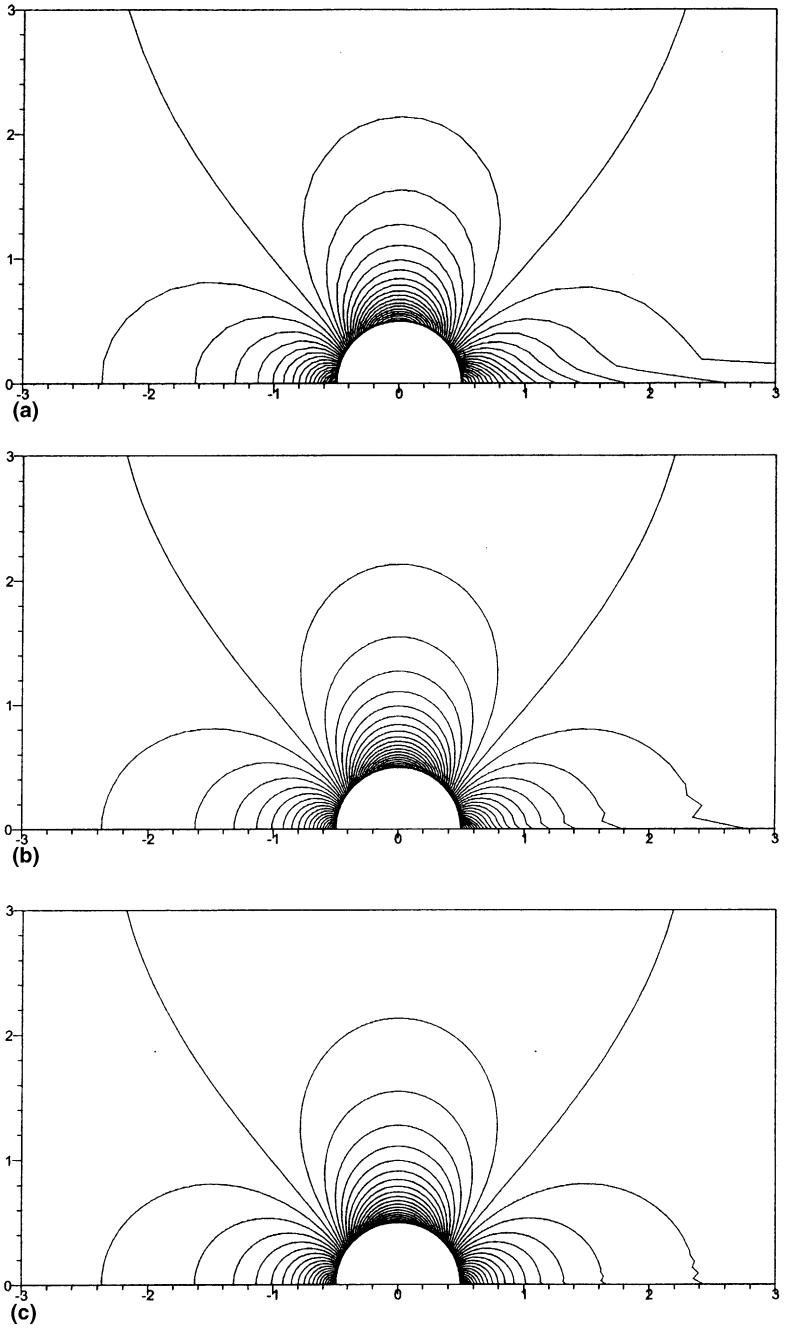

Fig. 7. Mach contours for a free stream Mach number of 0.001 computed on the: (a) coarse, (b) medium, and (c) fine meshes.

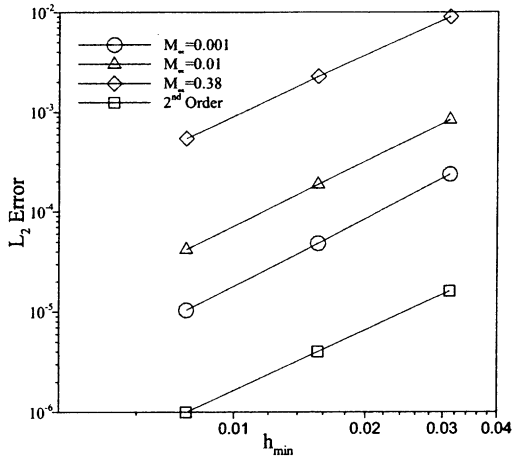

Fig. 8. Grid convergence plot showing the solution error norm versus the grid size parameter using the proposed algorithm and for free stream Mach numbers of 0.001, 0.01 and 0.38.

0.38. Analogous results, but now for a Mach number of 0.001 are shown in Fig. 7. The qualitative behavior of the solution, at both Mach numbers, does not present any anomalies. In Fig. 8, a plot of the solution error norm versus grid size is shown. Here, solutions at Mach numbers of 0.38, 0.01, and 0.001 are compared and second-order convergence is observed in all cases, showing no deterioration in the convergence rate as the Mach number is reduced. The norm of the solution error considered was $[\int_\Omega (\mathbf{V}-\mathbf{V}^h)\mathbf{A}_0(\mathbf{V}-\mathbf{V}^h)\,d\Omega]^{1/2}$, where $\mathbf{V}$ is a reference solution computed using a quadratic approximation on a highly refined grid of 77,361 nodes [25].

7.2. Example 2: Flow over an airfoil

In this example, the proposed scheme was used to simulate the flow over NACA 0012 airfoil at Mach numbers of 0.01 and 0.6, and at an angle of attack of 2°. An unstructured triangular mesh of 19,948 nodes

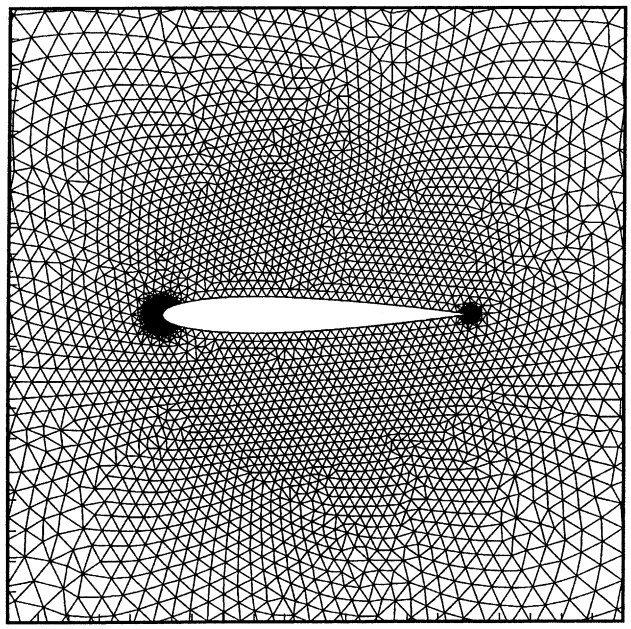

Fig. 9. Detail of the unstructured triangular the mesh used for the airfoil computations.

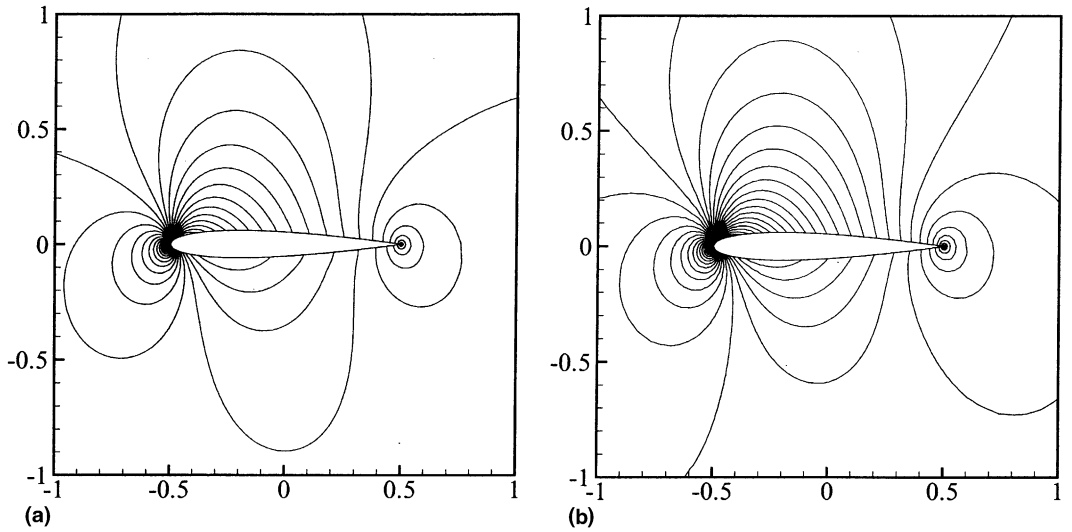

Fig. 10. Mach number contours for the solution of the flow over NACA 0012 at an angle of attack of 2° for a free stream Mach number of: (a) 0.01, and (b) 0.6.

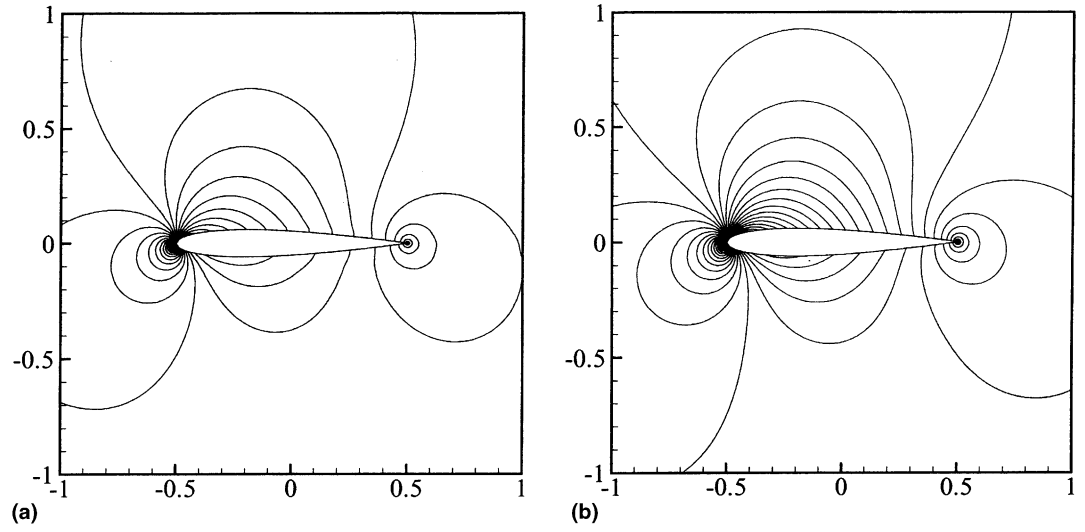

Fig. 11. Static pressure contours for the solution of the flow over NACA 0012 at an angle of attack of 2° for a free stream Mach number of: (a) 0.01, and (b) 0.6.

was used for the simulation. Fig. 9 shows a coarser mesh from which the 19,948 node mesh can be obtained by uniformly subdividing each element into four elements. The computed Mach number and pressure contours for the two flow conditions are shown in Figs. 10 and 11, respectively. The qualitative behavior of the solution looks excellent, and again, no solution accuracy or numerical anomalies are observed when very low Mach number flows are considered.

8. Conclusions

We have presented a method for constructing a SUPG stabilization matrix τ, for the computation of low Mach number flows. The proposed stabilization matrix represents a dramatic improvement over alterna-

tive, more standard, choices. We contend that this matrix also has advantages for more general high speed flows, since regions of low Mach number, are unavoidable in practically all flows of engineering interest.

The proposed method incorporates some of the developments in the area of low Mach number preconditioning for finite volume algorithms which have occurred over the last few years. However, one essential difference is that the dissipation, or stabilizing, operator is always symmetric and has a full set of orthogonal eigenvectors due to the use of symmetrizing entropy variables in our formulation. As a consequence, the lack of robustness of the finite volume techniques, often associated with the eigenvectors of the dissipation operator becoming nearly parallel in the vicinity of stagnation points, has not been encountered with the present formulation.

It is clear that the choice of τ presented in Section 6 incorporates some degree of arbitrariness. However, condition (28) represents a necessary condition for stability when the Mach number tends to zero. The proposed form of τ has the advantage of being simple, thus allowing an explicit calculation of the tangent matrix needed for the Newton–Raphson iteration.

We have seen from the examples that for the Mach numbers as low as 10^{-3}, no deterioration in the convergence rate is observed when the mesh size is reduced. We have also found no evidence that the linear systems that need to be solved, or indeed the non-linear iteration, are worse conditioned, or harder to solve, than those obtained for higher Mach numbers. We point out however, that if the variables are non-dimensionalized so that typical pressure variations over the domain are $O(1)$, then the value of the pressure itself will be $O(\epsilon^{-2})$. As a consequence one will unavoidably run out into precision problems for sufficiently low Mach numbers. In practice, we have found that eight-byte floating point numbers are adequate for Mach numbers as low as 10^{-3}. We also point out that at very low Mach numbers the flow solution has a very small sensitivity on the Mach number and as a consequence, flows with Mach numbers as low as 10^{-3}, can, for most practical purposes, be considered as incompressible.

Acknowledgements

The authors would like to acknowledge the support provided by the NASA Langley Research Center through grant NAG1-2122, and the NASA Dryden Flight Research Center through grant NAG4-157, and the MIT-Singapore Alliance, to carry out the research reported in this article.

Appendix A

We consider two particular choices for the function $g(s)$ in (4). First, we take $g(s) = s/(\gamma - 1)$ which results in the following set of entropy variables

$$\mathbf{V} = H_{,\mathbf{U}}^{\mathrm{T}} = \begin{bmatrix} \frac{\gamma+1}{\gamma-1} - \frac{s}{\gamma-1} - \frac{\rho E}{p} \\ \frac{\rho u_1}{p} \\ \frac{\rho u_2}{p} \\ \frac{-\rho}{p} \end{bmatrix}. \tag{A.1}$$

This set of variables has been advocated in [12] and has the feature that it also symmetrizes the Navier–Stokes equation including heat conducting terms. The second set of variables that we consider is [11]

$$\mathbf{V} = H_{,\mathbf{U}}^{\mathrm{T}} = \sqrt{\rho p^{(1-2\gamma)/\gamma}} \begin{bmatrix} E + \frac{p}{\rho} \\ -u_1 \\ -u_2 \\ 1 \end{bmatrix}, \tag{A.2}$$

which corresponds to the choice $g(s) = \mathrm{e}^{s/2\gamma}$. This choice has the particular feature that the fluxes $\mathbf{F}_1$ and $\mathbf{F}_2$, expressed as functions of $\mathbf{V}$, are homogeneous functions of degree $-2\gamma/(\gamma - 1)$.

Appendix B

For an arbitrary unit vector $\mathbf{n} = [n_1, n_2]^{\mathrm{T}}$, the matrix $\mathbf{A}_n = n_1\mathbf{A}_1 + n_2\mathbf{A}_2$, can be expressed as $\mathbf{A}_n = \mathbf{R}_n\mathbf{\Lambda}_n\mathbf{R}_n^{-1}$, where $\mathbf{R}_n$ is the matrix of right eigenvectors

$$\mathbf{R}_n = \begin{bmatrix} 1 & 0 & 1 & 1 \\ u_1 & -n_2 & u_1 + n_1 c & u_1 - n_1 c \\ u_2 & n_1 & u_2 + n_2 c & u_2 - n_2 c \\ \frac{|\mathbf{u}|^2}{2} & -n_2 u_1 + n_1 u_2 & \frac{|\mathbf{u}|^2}{2} + u_n c + \frac{c^2}{(\gamma-1)} & \frac{|\mathbf{u}|^2}{2} - u_n c + \frac{c^2}{(\gamma-1)} \end{bmatrix}$$

and $\mathbf{\Lambda}_n$ is the diagonal matrix of eigenvalues given by $\mathrm{diag}(\mathbf{\Lambda}_n) = \{u_n, u_n, u_n + c, u_n - c\}$, with $u_n = \mathbf{u}\cdot\mathbf{n}$.

The scaled matrix of eigenvectors $\tilde{\mathbf{R}}_n$, in (6) and (7) is given by $\tilde{\mathbf{R}}_n = \mathbf{R}_n\mathbf{Y}$, where $\mathbf{Y}$ is the diagonal matrix of scaling factors. For the entropy variables in (A.1), corresponding to the entropy function $H(\mathbf{U}) = -\rho s/(\gamma - 1)$, $\mathbf{Y}$ has the form

$$\mathbf{Y} = \sqrt{\frac{\rho}{2\gamma}} \begin{bmatrix} \sqrt{2(\gamma-1)} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

whereas for the entropy variables in (A.2), corresponding to $H(\mathbf{U}) = -\rho e^{s/2\gamma}$, the matrix of scaling factors becomes

$$\mathbf{Y} = \sqrt{\frac{\rho}{2(\gamma-1)}} e^{-s/2\gamma} \begin{bmatrix} \sqrt{2(\gamma-1)} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Thus, the matrix $\tilde{\mathbf{A}}_n = n_1\tilde{\mathbf{A}}_1 + n_2\tilde{\mathbf{A}}_2$ is given by $\tilde{\mathbf{A}}_n = \tilde{\mathbf{R}}_n\mathbf{\Lambda}_n\tilde{\mathbf{R}}_n^{\mathrm{T}}$.

Appendix C

Consider the matrix $\mathbf{C}_n = n_1\mathbf{C}_1 + n_2\mathbf{C}_2$, for an arbitrary unit vector $\mathbf{n} = [n_1, n_2]^{\mathrm{T}}$. The orthogonal matrix of right eigenvectors of $\mathbf{C}_n$, is given by

$$\mathbf{P}_n = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & -n_2 & n_1/\sqrt{2} & -n_1/\sqrt{2} \\ 0 & n_1 & n_2/\sqrt{2} & -n_2/\sqrt{2} \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Thus $\mathbf{C}_n = \mathbf{P}_n\mathbf{\Lambda}_n\mathbf{P}_n^{\mathrm{T}}$ and the matrix $\mathbf{S} = \mathbf{V}_{,\mathbf{Z}}$, in (35), can be obtained as $\mathbf{S} = \tilde{\mathbf{R}}_n^{-\mathrm{T}}\mathbf{P}_n^{\mathrm{T}}$. We note that, although $\tilde{\mathbf{R}}_n$ and $\mathbf{P}_n$ depend on the choice of $\mathbf{n}$, $\mathbf{S}$ does not. It can be verified that $\mathbf{S}$ satisfies $\mathbf{S}^{-\mathrm{T}}\mathbf{S}^{-1} = \mathbf{A}_0$, and that $\mathbf{S}^{-\mathrm{T}}\mathbf{C}_n\mathbf{S}^{-1} = \tilde{\mathbf{A}}_n$ for any $\mathbf{n}$.

References

[1] T.J. Barth, Numerical Methods for Gasdynamic Systems on Unstructured Meshes, Lecture Notes in Computational Science and Engineering, 1998, pp. 195–284.
[2] D. Choi, C.L. Merkle, Application of time-iterative schemes to incompressible flow, AIAA J. 23 (1985) 1518–1524.
[3] Y.H. Choi, C.L. Merkle, The application of preconditioning in viscous flows, J. Comput. Phys. 105 (1993) 203–223.
[4] D.L. Darmofal, P.J. Schmid, The importance of eigenvectors for local preconditioners of the Euler equations, J. Comput. Phys. 127 (1996) 346–362.
[5] D.L. Darmofal, B. Van Leer, Local preconditioning: manipulating mother nature to fool father time, in: M. Hafez, D.A. Caughey (Eds.), Computing the Future II: Advances and Prospects for Computational Aerodynamics, Wiley, New York, 1998, pp. 211–239.
[6] D.R. Fokkema, Subspace methods for linear, nonlinear and eigenproblems, Ph.D. Thesis, Utrech University, 1996.

[7] H. Guillard, C. Viozat, On the behavior of upwind schemes in the low Mach number limit, Comput. Fluids 28 (1999) 63–86.
[8] J. Guerra, B. Gustafsson, A numerical method for incompressible and compressible flow problems with smooth solutions, J. Comput. Phys. 63 (1986) 377–397.
[9] B. Gustaffson, Unsymmetric hyperbolic systems and the Euler equations at low mach numbers, J. Sci. Comput. 2 (2) (1987).
[10] B. Gustafsson, H. Stoor, Navier–Stokes equations for almost incompressible flow, SIAM J. Numer. Anal. 28 (1991) 1523–1547.
[11] A. Harten, On the symmetric for of systems of conservation laws with entropy, J. Comput. Phys. 49 (1983) 151–164.
[12] T.J.R. Hughes, L.P. Franca, M. Mallet, A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics, Comp. Methods Appl. Mech. Engrg. 54 (1986) 223–234.
[13] T.J.R. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective diffusive systems, Comput. Methods Appl. Mech. Engrg. 58 (1986) 305–328.
[14] T.J.R. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin Least-Squares method for advective diffusive equations, Comput. Methods Appl. Mech. Engrg. 73 (1989) 173–189.
[15] S. Klainerman, A. Majda, Compressible and incompressible fluids, Commun. Pure Appl. Math. 34 (1981) 629–651.
[16] L. Machiels, A.T. Patera, J. Peraire, Y. Maday, A general framework for finite element of posteriori error control: Application to linear anad nonlinear convection-dominated problems, in: ICFD Conference on Numerical Methods for Fluid Dynamics, Oxford, England, March 31–April 3, 1998.
[17] C.L. Reed, D.A. Anderson, Application of low speed preconditioning to the compressible Navier–Stokes equations, AIAA Paper 97-0873, 1997.
[18] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[19] F. Shakib, T.J.R. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 89 (1991) 141–219.
[20] G.L.G. Sleijpen, H.A. van der Vorst, D.R. Fokkema, BiCGstab(1) and other hybrid Bi-CG methods, Numer. Algorithms 7 (1994) 75–109.
[21] E. Turkel, Symmetrization of fluid dynamic matrices with application, Math. Comput. 27 (1973) 729–736.
[22] E. Turkel, A. Fiterman, B. Van Leer, Preconditioning and the limit to the incompressible flow equations for finite differences schemes, in: M. Hafez, D.A. Caughey (Eds.), Computing the Future: Advances and Prospects for Computational Aerodynamics, Wiley, New York, 1994, pp. 215–234.
[23] B. Van Leer, W.T. Lee, P.L. Roe, Characteristic time-stepping or local preconditioning of the Euler equations, AIAA Paper 91-1552, 1991.
[24] J.M. Weiss, W.A. Smith, Preconditioning applied to variable and constant density flows, AIAA J. 33 (1995) 2050–2057.
[25] J.S. Wong, D.L. Darmofal, J. Peraire, High-order finite element discretization for the compressible Euler and Navier-Stokes equations, FDRL TR-01-1, Dept. of Aeronautics & Astronautics, MIT, April, 2001.