INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **49**:183–209
Published online 3 June 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.999

A segregated method for compressible flow computation. Part II: General divariant compressible flows

Guillermo Hauke[1,*,†], Aitor Landaberea[2], Iñaki Garmendia[2] and Javier Canales[3]

[1] *Departamento de Mecánica de Fluidos, Centro Politécnico Superior, C/Maria de Luna 3, 50.018 Zaragoza, Spain*
[2] *Product Engineering Dpt. Fundación INASMET, Mikeletegi 2, 20.009 San Sebastian, Spain*
[3] *Department of Mechanical Engineering, Universidad del País Vasco, Alda. Urquijo, s/n 48.013 Bilbao, Spain*

SUMMARY

Typically, segregated methods have been used for the computation of incompressible flows whereas coupled solvers, for compressible flows. Compared to coupled solvers, segregated methods present the advantage of computational savings in RAM memory and CPU time, although at the cost of an inferior robustness. However, previously published segregated algorithms for general compressible flows are known to present pitfalls, like convergence to wrong solutions, lack of robustness in the presence of strong discontinuities, such as normal and oblique shocks, and complicated boundary condition imposition. Therefore, in this paper a segregated method for non-isothermal compressible flows is proposed that preserves the thermodynamic coupling and overcomes the criticisms of existing methods. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: compressible flow; stabilized finite element methods; segregated methods

1. INTRODUCTION

At present, there is a clear dichotomy between the numerical algorithms for the computation of compressible and incompressible flows. In particular, compressible flows are typically solved via coupled solvers whereas incompressible flows, via segregated strategies.

As pointed out in the first paper of this series [1], the main reasons for that dichotomy are that compressible flows present a strong thermodynamic coupling among the dependent variables, such as the pressure, density and temperature, which does not exist for incom-

[*] Correspondence to: Guillermo Hauke, Departamento de Mecánica de Fluidos, Centro Politécnico Superior, C/Maria de Luna 3, 50.018 Zaragoza, Spain.
[†] E-mail: ghauke@unizar.es

Contract/grant sponsor: Ministerio de Educación y Ciencia
Contract/grant sponsor: University of the Basque Country; contract/grant number: 1/UPV/EHU 00145.345-EA-8246/2000

Received 17 April 2004
Revised 24 January 2005
Accepted 31 January 2005

pressible flows. On the other hand, in incompressible flows, the semi-definite structure of the discrete algorithms finds in regularization or segregated strategies a solution to circumvent this numerical difficulty. The literature and the variety of solutions on this topic is very rich, including artificial compressibility methods [2], penalty methods [3, 4], augmented Lagrangian methods [5–7], projection methods and fractional momentum methods [2, 8–11] and mixed methods [12–14]. For further information, the interested reader can consult References [15–17] and references therein.

In this paper, we focus on the improvement of existing segregated techniques for compressible and incompressible flows. While coupled solvers are typically more robust, segregated solvers have computational advantages such as an inferior memory and CPU time demands, and a higher modularity and flexibility in order to incorporate new physical phenomena modelled by additional partial differential equations. These advantages are even more relevant in the solution of large scale problems and in the application to parallel codes.

Segregated methods can also be classified as implicit or explicit. Implicit algorithms require the solution of linear systems of equations at each time step, whereas explicit methods do not. By avoiding matrix inversions, explicit methods decrease the solution time per step, but the maximum allowed time step is limited by stability conditions, a limit that most implicit methods escape. Up to date, segregated methods for transient incompressible flows are implicit. Therefore, explicit segregated methods find application only for compressible flows or steady incompressible flows (see Reference [18]).

Several strategies have been suggested to develop unified approaches for the computation of compressible and incompressible flows, both, with coupled solvers [19, 20] and segregated approaches [18, 21–24]. However, the success of existing segregated techniques for compressible flows has only been partial. For instance, several passes through the same equation or fractional steps may be required, even for first-order accuracy. Segregated solvers based on pressure Poisson equations may require non-trivial pressure boundary conditions. This can get even more cumbersome and complicated in the intermediate fractional steps. Finally, the existing techniques are not robust when confronted to strong discontinuities, such as normal and oblique shocks, or even converge to wrong solutions.

Therefore, in this paper the work [1] is extended to general divariant compressible flows. The variational formulation is based on stabilized methods, which have been successfully applied to the computation of compressible and incompressible flows (see References [19, 20, 25–41] and references therein). Then a new splitting is proposed, which preserves the thermodynamic coupling, leading to a method that overcomes the problems of previously published methods.

2. THE SYSTEM OF EQUATIONS FOR COMPRESSIBLE FLOW

The equations for compressible flow form an advective–diffusive system, constituted by the continuity, momentum and total energy equations. In conservative form, the system can be written as

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i}^{\text{adv}} = \mathbf{F}_{i,i}^{\text{diff}} + \mathbf{S} \tag{1}$$

where $\mathbf{U}$ represents the vector of conservation variables, $\mathbf{F}_i^{\text{adv}}$ is the advective flux in the ith-direction, $\mathbf{F}_i^{\text{diff}}$ is the diffusive flux in the ith-direction, and $\mathbf{S}$ is the source vector. The inferior comma denotes partial differentiation and the summation convention on repeated indices is applied throughout.

In three-dimensional Cartesian coordinates, the above vectors are

$$\mathbf{U} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix} = \rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{\text{tot}} \end{Bmatrix} \tag{2}$$

$$\mathbf{F}_i^{\text{adv}} = \rho u_i \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_{\text{tot}} \end{Bmatrix} + p \begin{Bmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{Bmatrix} \tag{3}$$

$$\mathbf{F}_i^{\text{diff}} = \begin{Bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} u_j \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q_i \end{Bmatrix} \tag{4}$$

$$\mathbf{S} = \rho \begin{Bmatrix} 0 \\ b_1 \\ b_2 \\ b_3 \\ b_i u_i + r \end{Bmatrix} \tag{5}$$

where u_i are the Cartesian velocity components, ρ the fluid density, p the thermodynamic pressure, $e_{\text{tot}} = e + \frac{1}{2}|\mathbf{u}|^2$ the total energy per unit mass, sum of the specific internal energy e and specific kinetic energy $\frac{1}{2}|\mathbf{u}|^2$. Furthermore, δ_{ij} represents the Kronecker delta, τ_{ij} the viscous stress tensor, q_i the molecular heat flux, b_i the body force per unit mass and r a volumetric heat source.

Using any well-defined set of variables $\mathbf{Y}$, it is possible to rewrite (1) in quasi-linear form as

$$\mathbf{A}_0 \mathbf{Y}_{,t} + \mathbf{A}_i \mathbf{Y}_{,i} = (\mathbf{K}_{ij} \mathbf{Y}_{,j})_{,i} + \mathbf{S} \tag{6}$$

where $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{Y}}$, $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{Y}}^{\text{adv}}$ is the ith Euler–Jacobian matrix, and $\mathbf{K} = [\mathbf{K}_{ij}]$ is the diffusivity matrix where $\mathbf{K}_{ij} \mathbf{Y}_{,j} = \mathbf{F}_i^{\text{diff}}$.

In this paper, special attention is paid to the choice of pressure primitive variables

$$\mathbf{Y} = \begin{Bmatrix} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{Bmatrix} \tag{7}$$

with p the pressure, u_i the Cartesian velocity components and T the absolute temperature. The matrices and vectors for these variables can be found in Reference [20]. This set of variables is endowed with the property that the incompressible limit is well behaved.

2.1. Generalized entropy function and the symmetric form

Symmetric forms of (1) and (6) are those in which the coefficient matrices enjoy the properties

1. $\widetilde{\mathbf{A}}_0 = \mathbf{U}_{,\mathbf{V}}$ is symmetric, positive-definite;
2. $\widetilde{\mathbf{A}}_i = \mathbf{F}^{\mathrm{adv}}_{i,\mathbf{V}}$ is symmetric;
3. $\widetilde{\mathbf{K}} = [\widetilde{\mathbf{K}}_{ij}]$ is symmetric, positive-semidefinite.

It is known that symmetric forms are linked to nonlinear stability principles through the so-called generalized entropy functions, which at the same time engender the entropy variables [42, 43]. This form of the system can be viewed as a canonical form, that can be used up by numerical analysts to design better numerical algorithms.

For instance, appropriately defined finite element methods based on entropy variables can inherit at the discrete level the nonlinear stability principle. Also, as shown in References [19, 20], new methods can be derived for any choice of well defined variables starting from the canonical form based on entropy variables.

In the case of compressible flows, the generalized entropy is just the physical entropy [44, 29], which for a general divariant fluid [45], it is defined as

$$\mathscr{H}(\mathbf{U}) = -\rho(s - s_0) \tag{8}$$

where s is the physical entropy and s_0 is the reference entropy.

Then, a change of variables from conservation to entropy variables $\mathbf{U} \mapsto \mathbf{V}$ is introduced by means of

$$\mathbf{V}^{\mathrm{T}} = \frac{\partial \mathscr{H}(\mathbf{U})}{\partial \mathbf{U}} \tag{9}$$

resulting in the set of generalized entropy variables,

$$\mathbf{V} = \frac{1}{T}\left\{\begin{matrix} \mu - \frac{1}{2}|\mathbf{u}|^2 \\ u_1 \\ u_2 \\ u_3 \\ -1 \end{matrix}\right\} \tag{10}$$

The above equation has been written as a function of μ, the electrochemical potential per unit mass. Note that

$$\mu = h - Ts = e + \frac{p}{\rho} - Ts \tag{11}$$

It can be verified that the change of variables $\mathbf{U} \mapsto \mathbf{V}$ induces a Hessian matrix $\widetilde{\mathbf{A}}_0^{-1} = \mathscr{H}_{,\mathbf{UU}}$ which is symmetric and positive-definite. Therefore, the change of variables is well defined and the function $\mathscr{H}(\mathbf{U})$ is a convex function of $\mathbf{U}$. Likewise, the matrices $\widetilde{\mathbf{A}}_i$ is symmetric and $\widetilde{\mathbf{K}} = [\widetilde{\mathbf{K}}_{ij}]$ is symmetric positive-semidefinite (see Reference [45]).

2.2. *Generalized entropy (in)equality*

The stability principle of the Navier–Stokes equations is a combination of the conservation laws and can be obtained by the dot product

$$\mathbf{V} \cdot (\mathbf{U}_{,t} + \mathbf{F}^{\text{adv}}_{i,i} - \mathbf{F}^{\text{diff}}_{i,i} - \mathbf{S}) = 0 \tag{12}$$

That is, the entropy variables are the integration factors that make the above differential exact. The result, derived in Reference [29] for a perfect gas and extended in Reference [45] for a general divariant fluid is the following:

$$\begin{aligned}(\rho s)_{,t} + (\rho s u_i)_{,i} + \left(\frac{-\kappa T_{,i}}{T}\right)_{,i} - \frac{\rho r}{T} &= \left(\frac{\Upsilon(\mathbf{u},\mathbf{u})}{T}\right) + \kappa\,\frac{T_{,i}T_{,i}}{T^2} \\ &\geqslant 0\end{aligned} \tag{13}$$

which is the second law of thermodynamics. Here, $\Upsilon(\mathbf{u},\mathbf{u}) \geqslant 0$ is the viscous dissipation function, $\kappa \geqslant 0$ is the heat conductivity and $T > 0$ is the absolute temperature. The entropy flux is seen to be

$$\sigma u_i = -\rho s u_i = \mathscr{H} u_i \tag{14}$$

Therefore, integration over the domain Ω gives rise to the Clausius–Duhem inequality

$$\int_\Omega (\mathscr{H}_{,t} + (u_i \mathscr{H})_{,i} + \rho r)\,\mathrm{d}\Omega - \int_\Gamma \frac{q_i n_i}{T}\,\mathrm{d}\Gamma \leqslant 0 \tag{15}$$

This implies that under appropriate boundary conditions and body source term, the generalized entropy is a bounded function. Since $\mathscr{H}(\mathbf{U})$ is a convex function of the conservation variables $\mathbf{U}$, the conservation variables are themselves bounded.

2.3. Incompressible limit

The incompressible limit is well behaved for entropy and pressure primitive variables. For more details, the interested reader is referred to References [19, 20].

3. STABILIZED FINITE ELEMENT METHOD

In this paper, the time-discontinuous space–time SUPG and GLS stabilized methods [25, 33, 35] are considered. In the absence of source terms and for linear shape functions both methods coincide.

Consider a space–time domain, where the time interval $I =]0, T[$ is subdivided into N intervals $I_n =]t_n, t_{n+1}[$, $n = 0, 1, \ldots, N-1$. We define for each time interval $Q_n = \Omega \times I_n$ and $P_n = \Gamma \times I_n$, where Ω is the spatial domain and Γ its boundary. Finally, the 'slab' Q_n is decomposed into elements Q_n^e, $e = 1, 2, \ldots, (n_{\text{el}})_n$.

The variational formulation is defined for the set of variables $\mathbf{Y}$ according to References [19, 20]. Within each Q_n, $n = 0, 1, \ldots, N-1$, find $\mathbf{Y} \in \mathscr{S}_{\mathbf{Y}}$ such that $\forall\ \mathbf{W} \in \mathscr{V}_{\mathbf{Y}}$

$$\begin{aligned}
&\int_{Q_n} (-\mathbf{W}_{,t} \cdot \mathbf{U}(\mathbf{Y}) - \mathbf{W}_{,i} \cdot \mathbf{F}_i^{\text{adv}}(\mathbf{Y}) + \mathbf{W}_{,i} \cdot \mathbf{K}_{ij} \mathbf{Y}_{,j} - \mathbf{W} \cdot \mathbf{S})\, \mathrm{d}Q \\
&\quad + \int_{\Omega} (\mathbf{W}(t_{n+1}^-) \cdot \mathbf{U}(\mathbf{Y}(t_{n+1}^-)) - \mathbf{W}(t_n^+) \cdot \mathbf{U}(\mathbf{Y}(t_n^-)))\, \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{(n_{\text{el}})_n} \int_{Q_n^e} (\mathscr{L}^{\mathrm{T}} \mathbf{W}) \cdot \boldsymbol{\tau} (\mathscr{L} \mathbf{Y} - \mathbf{S})\, \mathrm{d}Q \\
&\quad + \sum_{e=1}^{(n_{\text{el}})_n} \int_{Q_n^e} v^h g^{ij} \mathbf{W}_{,i} \cdot \mathbf{A}_0 \mathbf{Y}_{,j}\, \mathrm{d}Q \\
&= \int_{P_n} \mathbf{W} \cdot (-\mathbf{F}_i^{\text{adv}}(\mathbf{Y}) + \mathbf{F}_i^{\text{diff}}(\mathbf{Y})) n_i\, \mathrm{d}P
\end{aligned} \tag{16}$$

The first and last integrals constitute the Galerkin terms expressed as a function of the variables $\mathbf{Y}$, again written in conservative form to ensure that the weak solution is bestowed with the correct Rankine–Hugoniot relations. The jump term is written with the help of the right and left limits,

$$W(t_n^{\pm}) = \lim_{\varepsilon \to 0^{\pm}} W(t_n + \varepsilon) \tag{17}$$

The least-squares contribution is written in terms of the differential operator $\mathscr{L}$ and $\mathscr{L}^{\mathrm{T}}$, which, respectively, are given by

$$\mathscr{L} = \mathbf{A}_0 \frac{\partial}{\partial t} + \mathbf{A}_i \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}\frac{\partial}{\partial x_j}\right) \tag{18}$$

and

$$\mathscr{L}^{\mathrm{T}} = \mathbf{A}_0^{\mathrm{T}} \frac{\partial}{\partial t} + \mathbf{A}_i^{\mathrm{T}} \frac{\partial}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\mathbf{K}_{ij}^{\mathrm{T}}\frac{\partial}{\partial x_j}\right) \tag{19}$$

Note that when entropy variables are used, $\widetilde{\mathscr{L}} = \widetilde{\mathscr{L}}^{\mathrm{T}}$ because of the symmetry of the coefficient matrices and the symmetric form is recovered. We assume

$$\boldsymbol{\tau} = \mathbf{Y}_{,\mathbf{V}}\ \widetilde{\boldsymbol{\tau}} \tag{20}$$

For definitions of the τ matrix see References [31, 35]. In this work, two simplified operators, the diagonal and the non-diagonal, are used (see References [46, 1]).

The fourth integral is the discontinuity capturing operator [47, 19], a function of the artificial diffusivity coefficient,

$$v_{HM}^{h} = \max\left(0, \left[\frac{(\mathscr{L}\mathbf{Y}-\mathbf{S})\cdot\widetilde{\mathbf{A}}_0^{-1}(\mathscr{L}\mathbf{Y}-\mathbf{S})}{g^{ij}\mathbf{Y}_{,i}\cdot\mathbf{A}_0^{DC}\mathbf{Y}_{,j}}\right]^{1/2} - \frac{(\mathscr{L}\mathbf{Y}-\mathbf{S})\cdot\widetilde{\boldsymbol{\tau}}(\mathscr{L}\mathbf{Y}-\mathbf{S})}{g^{ij}\mathbf{Y}_{,i}\cdot\mathbf{A}_0^{DC}\mathbf{Y}_{,j}}\right) \tag{21}$$

$$v_{\mathrm{quad}}^{h} = 2\,\frac{(\mathscr{L}\mathbf{Y}-\mathbf{S})\cdot\widetilde{\boldsymbol{\tau}}\ (\mathscr{L}\mathbf{Y}-\mathbf{S})}{g^{ij}\mathbf{Y}_{,i}\cdot\mathbf{A}_0^{DC}\mathbf{Y}_{,j}} \tag{22}$$

$$v_{\min}^{h} = \min(v_{HM}^{h}, v_{\mathrm{quad}}^{h}) \tag{23}$$

where

$$\mathbf{A}_0^{DC} = \mathbf{V}_{,\mathbf{Y}}^{\mathrm{T}}\widetilde{\mathbf{A}}_0\mathbf{V}_{,\mathbf{Y}} = \mathbf{V}_{,\mathbf{Y}}^{\mathrm{T}}\mathbf{A}_0 \tag{24}$$

Also, g^{ij} denotes the contravariant metric tensor,

$$g^{ij} = [\xi_{k,i}\xi_{k,j}]^{-1} \tag{25}$$

where ξ_k, $k = 1, 2, 3$, are the local element coordinates.

4. SEGREGATED FORMULATION

Substitution of the finite element spaces in the above weak form leads to a set of nonlinear equations. This system of equations is coupled and can be written as a residual which depends on the nodal unknowns $\mathbf{y}$ at time t_n and t_{n+1},

$$\mathbf{R}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}) = \mathbf{0} \tag{26}$$

For compressible flows, coupled solvers are typically employed to drive the nonlinear residual to zero via a series of linear problems. An interesting choice to achieve convergence in a systematic and general way is to introduce predictor multi-corrector algorithms [25, 34, 48], where an update of the solution $\Delta\mathbf{y}$ is computed at each iteration according to

$$\begin{aligned} \mathbf{M}\Delta\mathbf{y} &= -\mathbf{R} \\ \mathbf{y}^{\text{new}} &= \mathbf{y}^{\text{old}} + \Delta\mathbf{y} \end{aligned} \tag{27}$$

Ideally, the matrix $\mathbf{M}$ is the consistent tangent, that is, the Jacobian of the residual with respect to the unknowns,

$$\mathbf{M} = \frac{\partial\mathbf{R}}{\partial\mathbf{y}_{(n+1)}} \tag{28}$$

The consistent tangent gives the highest convergence rates within the radius of convergence, that is, when the initial guess is sufficiently close to the exact solution. However, when this is not the case, approximations of $\mathbf{M}$ may result in more robust algorithms and higher convergence rates or, simply, give rise to more convenient algorithms.

In this paper, segregated solvers for the problem (27) are analysed. In particular, the vector of unknowns is split into three parts

$$\mathbf{Y} = \begin{Bmatrix} \mathbf{Y}_\rho \\ \mathbf{Y}_u \\ \mathbf{Y}_e \end{Bmatrix} \tag{29}$$

For instance, for pressure primitive variables

$$\mathbf{Y}_\rho = \{ p \} \tag{30}$$

$$\mathbf{Y}_u = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \tag{31}$$

$$\mathbf{Y}_e = \{ T \} \tag{32}$$

The nodal unknowns and the equations can be segregated in a similar manner

$$\mathbf{y} = \begin{Bmatrix} \mathbf{y}_\rho \\ \mathbf{y}_u \\ \mathbf{y}_e \end{Bmatrix} \tag{33}$$

$$\mathbf{R} = \begin{Bmatrix} \mathbf{R}_\rho \\ \mathbf{R}_u \\ \mathbf{R}_e \end{Bmatrix} \tag{34}$$

Then, the Newton–Raphson method (27) can be expressed as

$$\begin{bmatrix} \mathbf{M}_{\rho\rho} & \mathbf{M}_{\rho u} & \mathbf{M}_{\rho e} \\ \mathbf{M}_{u\rho} & \mathbf{M}_{uu} & \mathbf{M}_{ue} \\ \mathbf{M}_{e\rho} & \mathbf{M}_{eu} & \mathbf{M}_{\mathrm{ee}} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{y}_{\rho} \\ \Delta\mathbf{y}_{u} \\ \Delta\mathbf{y}_{e} \end{Bmatrix} = - \begin{Bmatrix} \mathbf{R}_{\rho} \\ \mathbf{R}_{u} \\ \mathbf{R}_{e} \end{Bmatrix} \tag{35}$$

In the next subsections, several staggered iterative strategies to tackle the above problem are examined.

4.1. Naive iterative methods based on primitive variables

The obvious way to solve the system (35) is to apply the methods presented in the first paper [1].

Diagonal iterative method (*Algorithm* 1): This algorithm consists of neglecting the off-diagonal terms of the tangent matrix **M** and, thus, only considering the diagonal blocks. That is

$$\begin{aligned} \mathbf{M}_{\rho\rho}\Delta\mathbf{y}_{\rho} &= -\mathbf{R}_{\rho} \\ \mathbf{M}_{uu}\Delta\mathbf{y}_{u} &= -\mathbf{R}_{u} \\ \mathbf{M}_{\mathrm{ee}}\Delta\mathbf{y}_{e} &= -\mathbf{R}_{e} \end{aligned} \tag{36}$$

The algorithm is displayed in Box 1, where i denotes the iteration counter, $i_{\max}$ the number of corrector passes, and the residual at iteration i is denoted by

$$\mathbf{R}_{\rho}^{(i)} = \mathbf{R}_{\rho}(\mathbf{y}_{(n+1)}^{(i)}; \mathbf{y}_{(n)}) \tag{37}$$

$$\mathbf{R}_{u}^{(i)} = \mathbf{R}_{u}(\mathbf{y}_{(n+1)}^{(i)}; \mathbf{y}_{(n)}) \tag{38}$$

$$\mathbf{R}_{e}^{(i)} = \mathbf{R}_{e}(\mathbf{y}_{(n+1)}^{(i)}; \mathbf{y}_{(n)}) \tag{39}$$

Jacobi or Gauss–Seidel iterative method (*Algorithm* 2): This is a little more sophisticated version of the diagonal algorithm, where the coupling blocks are driven into the right-hand side and carried on in an explicit manner (see Box 2). In the Jacobi method, the off-diagonal blocks are evaluated at the previous corrector pass, whereas in the Gauss–Seidel method, the most up-to-date blocks are employed for that

$$\begin{aligned} \mathbf{M}_{\rho\rho}\Delta\mathbf{y}_{\rho} &= -\mathbf{R}_{\rho} - \mathbf{M}_{\rho u}\Delta\mathbf{y}_{u} - \mathbf{M}_{\rho e}\Delta\mathbf{y}_{e} \\ \mathbf{M}_{uu}\Delta\mathbf{y}_{u} &= -\mathbf{R}_{u} - \mathbf{M}_{u\rho}\Delta\mathbf{y}_{\rho} - \mathbf{M}_{ue}\Delta\mathbf{y}_{e} \\ \mathbf{M}_{\mathrm{ee}}\Delta\mathbf{y}_{e} &= -\mathbf{R}_{e} - \mathbf{M}_{e\rho}\Delta\mathbf{y}_{\rho} - \mathbf{M}_{eu}\Delta\mathbf{y}_{u} \end{aligned} \tag{40}$$

However, these methodologies, that work perfectly for the isothermal case and have been applied successfully to low Mach number flows (see Reference [49]), present convergence

Box 1. Predictor multi-corrector algorithm. Diagonal iterative method.

```
(Predictor phase)
$\mathbf{y}_\rho^{(0)} = \mathbf{y}_{\rho(n)}$
$\mathbf{y}_u^{(0)} = \mathbf{y}_{u(n)}$
$\mathbf{y}_e^{(0)} = \mathbf{y}_{e(n)}$
(Multi-corrector phase)
for $i = 0, 1, \ldots, i_{\max} - 1$
    $\mathbf{M}_{\rho\rho}\Delta\mathbf{y}_\rho^{(i)} = -\mathbf{R}_\rho^{(i)}$
    $\mathbf{M}_{uu}\Delta\mathbf{y}_u^{(i)} = -\mathbf{R}_u^{(i)}$
    $\mathbf{M}_{\mathrm{ee}}\Delta\mathbf{y}_e^{(i)} = -\mathbf{R}_e^{(i)}$
    $\mathbf{y}_\rho^{(i+1)} = \mathbf{y}_\rho^{(i)} + \Delta\mathbf{y}_\rho^{(i)}$
    $\mathbf{y}_u^{(i+1)} = \mathbf{y}_u^{(i)} + \Delta\mathbf{y}_u^{(i)}$
    $\mathbf{y}_e^{(i+1)} = \mathbf{y}_e^{(i)} + \Delta\mathbf{y}_e^{(i)}$
$\mathbf{y}_{\rho(n+1)} = \mathbf{y}_\rho^{(i_{\max})}$
$\mathbf{y}_{u(n+1)} = \mathbf{y}_u^{(i_{\max})}$
$\mathbf{y}_{e(n+1)} = \mathbf{y}_e^{(i_{\max})}$
```

Box 2. Predictor multi-corrector algorithm. Jacobi iterative method.

```
(Predictor phase)
$\mathbf{y}_\rho^{(0)} = \mathbf{y}_{\rho(n)}$
$\mathbf{y}_u^{(0)} = \mathbf{y}_{u(n)}$
$\mathbf{y}_e^{(0)} = \mathbf{y}_{e(n)}$
(Multi-corrector phase)
for $i = 0, 1, \ldots, i_{\max} - 1$
    $\mathbf{M}_{\rho\rho}\Delta\mathbf{y}_\rho^{(i)} = -\mathbf{R}_\rho^{(i)} - \mathbf{M}_{\rho u}\Delta\mathbf{y}_u^{(i-1)} - \mathbf{M}_{\rho e}\Delta\mathbf{y}_e^{(i-1)}$
    $\mathbf{M}_{uu}\Delta\mathbf{y}_u^{(i)} = -\mathbf{R}_u^{(i)} - \mathbf{M}_{u\rho}\Delta\mathbf{y}_\rho^{(i-1)} - \mathbf{M}_{ue}\Delta\mathbf{y}_e^{(i-1)}$
    $\mathbf{M}_{\mathrm{ee}}\Delta\mathbf{y}_e^{(i)} = -\mathbf{R}_e^{(i)} - \mathbf{M}_{e\rho}\Delta\mathbf{y}_\rho^{(i-1)} - \mathbf{M}_{eu}\Delta\mathbf{y}_u^{(i-1)}$
    $\mathbf{y}_\rho^{(i+1)} = \mathbf{y}_\rho^{(i)} + \Delta\mathbf{y}_\rho^{(i)}$
    $\mathbf{y}_u^{(i+1)} = \mathbf{y}_u^{(i)} + \Delta\mathbf{y}_u^{(i)}$
    $\mathbf{y}_e^{(i+1)} = \mathbf{y}_e^{(i)} + \Delta\mathbf{y}_e^{(i)}$
$\mathbf{y}_{\rho(n+1)} = \mathbf{y}_\rho^{(i_{\max})}$
$\mathbf{y}_{u(n+1)} = \mathbf{y}_u^{(i_{\max})}$
$\mathbf{y}_{e(n+1)} = \mathbf{y}_e^{(i_{\max})}$
```

problems for general compressible flows. One of the causes can be traced back to the fact that for compressible flows, the complete mass matrix is positive-definite, whereas the lone diagonal block of the energy equation $\mathbf{M}_{\mathrm{ee}}$ is not. Also, another cause for this behaviour is the strong coupling between the thermodynamic variables.

4.2. *Iterative techniques based on mixed variables*

In order to overcome the difficulties explained above, several strategies are proposed firstly. To attain a positive-definite block for the energy equation, the total energy per unit volume ρe_{tot} is employed as the dependent variable for the energy equation. Then, the temperature T is calculated according to different update algorithms.

Three variants A, B and C were tested.

(A) The temperature is computed from the total energy density, so the whole change of total energy is assigned to the internal energy. For example, for a calorifically perfect fluid, the temperature will be updated according to

$$\rho e_{\mathrm{tot}}^{(i+1)} = \rho^{(i)}(c_v T^{(i+1)} + \tfrac{1}{2}|\mathbf{u}^{(i)}|^2) \tag{41}$$

or

$$T^{(i+1)} = f(\mathbf{y}_e^{(i+1)}, \mathbf{y}_\rho^{(i)}, \mathbf{y}_u^{(i)}) \tag{42}$$

The algorithm is displayed in Box 3.

Box 3. Predictor multi-corrector algorithm based on mixed variables for versions A and B.

```
(Predictor phase)
$\mathbf{y}_\rho^{(0)} = \mathbf{y}_{\rho(n)}$
$\mathbf{y}_u^{(0)} = \mathbf{y}_{u(n)}$
$\mathbf{y}_e^{(0)} = \mathbf{y}_{e(n)}$
(Multi-corrector phase)
for $i = 0, 1, \ldots, i_{\max} - 1$
    $\mathbf{M}_{\rho\rho} \Delta \mathbf{y}_\rho^{(i)} = -\mathbf{R}_\rho^{(i)}$
    $\mathbf{M}_{uu} \Delta \mathbf{y}_u^{(i)} = -\mathbf{R}_u^{(i)}$
    $\mathbf{M}_{\mathrm{ee}} \Delta \mathbf{y}_e^{(i)} = -\mathbf{R}_e^{(i)}$
    $\mathbf{y}_\rho^{(i+1)} = \mathbf{y}_\rho^{(i)} + \Delta \mathbf{y}_\rho^{(i)}$
    $\mathbf{y}_u^{(i+1)} = \mathbf{y}_u^{(i)} + \Delta \mathbf{y}_u^{(i)}$
    $\mathbf{y}_e^{(i+1)} = \mathbf{y}_e^{(i)} + \Delta \mathbf{y}_e^{(i)}$
    Compute $T^{(i+1)}$
$\mathbf{y}_{\rho(n+1)} = \mathbf{y}_\rho^{(i_{\max})}$
$\mathbf{y}_{u(n+1)} = \mathbf{y}_u^{(i_{\max})}$
$\mathbf{y}_{e(n+1)} = \mathbf{y}_e^{(i_{\max})}$
```

(B) The update of the temperature is computed from the update of the total energy density taking into account the already computed variation of the kinetic energy given by the momentum equations, i.e.

$$\Delta \mathbf{y}_e^{(i)} \approx \rho \Delta e_{\text{tot}}^{(i)} = \rho \Delta (e+k)^{(i)} = \rho (c_v \Delta T^{(i)} + \Delta k^{(i)}) \tag{43}$$

or

$$T^{(i+1)} = f(\rho^{(i)}, \Delta e^{(i)}, \Delta k^{(i)}) \tag{44}$$

In this way, the variation of total energy is distributed into variation of thermal energy and kinetic energy. See Box 3.

(C) This third algorithm is similar to the semi-implicit CBS algorithm proposed by Codina *et al.* for pressure primitive variables [24]. In particular, the temperature and the density are eventually updated according to

$$T^{(i+1)} = f(\mathbf{y}_e^{(i+1)}, \mathbf{y}_\rho^{(i+1)}, \mathbf{y}_u^{(i+1)}) \tag{45}$$

and

$$\rho^{(i+1)} = \frac{p^{(i+1)}}{RT^{(i+1)}} \tag{46}$$

The algorithm is expressed with more detail in Box 4, where some intermediate thermodynamic variables are computed along the steps. First, from the energy equation, an update for the total energy is obtained, and the intermediate temperature T_g, calculated. Then from the continuity equation, which depends only on the density and velocity, the pressure is updated and an intermediate density ρ_g is computed. Finally, the velocity is updated and the final temperature and density, calculated.

Several numerical tests have shown that strategy (B) can lead to incorrect temperature fields. On the other hand, the solution given by algorithms (A) and (C) is very sensitive to the pressure accuracy. And although these algorithms can also be used to solve low Mach number flows [49], as already pointed out in Reference [24], for flows with strong nonlinearities, such as flows with shock waves, convergence problems may appear when pressure primitive variables are used.

4.3. General segregated method with thermodynamic coupling

The ill-convergence of the above algorithms has been finally traced back to the strong coupling between the thermodynamic variables. Thermodynamically segregated formulations are not able to mimic this behaviour. Therefore, in order to cure the problem, a coupled treatment of the continuity and energy equations is proposed.

So let us now introduce the splitting

$$\mathbf{Y} = \begin{Bmatrix} \mathbf{Y}_{\rho e} \\ \mathbf{Y}_u \end{Bmatrix} \tag{47}$$

Box 4. Predictor multi-corrector algorithm based on mixed variables for version C.

```
(Predictor phase)
$\mathbf{y}_\rho^{(0)} = \mathbf{y}_{\rho(n)}$
$\mathbf{y}_u^{(0)} = \mathbf{y}_{u(n)}$
$\mathbf{y}_e^{(0)} = \mathbf{y}_{e(n)}$
(Multi-corrector phase)
for $i = 0, 1, \ldots, i_{\max} - 1$
    $\mathbf{M}_{\mathrm{ee}} \Delta \mathbf{y}_e^{(i)} = -\mathbf{R}_e^{(i)}$
        $\mathbf{y}_e^{(i+1)} = \mathbf{y}_e^{(i)} + \Delta \mathbf{y}_e^{(i)}$
        $\mathbf{T}_g = f(\mathbf{y}_e^{(i+1)}, \mathbf{y}_\rho^{(i)}, \mathbf{y}_u^{(i)})$
    $\mathbf{M}_{\rho\rho} \Delta \mathbf{y}_\rho^{(i)} = -\mathbf{R}_\rho^{(i)}$
        $\mathbf{y}_\rho^{(i+1)} = \mathbf{y}_\rho^{(i)} + \Delta \mathbf{y}_\rho^{(i)}$
        $\rho_g = \mathbf{p}^{(i+1)} / R\mathbf{T}_g$
    $\mathbf{M}_{uu} \Delta \mathbf{y}_u^{(i)} = -\mathbf{R}_u^{(i)}$
        $\mathbf{y}_u^{(i+1)} = \mathbf{y}_u^{(i)} + \Delta \mathbf{y}_u^{(i)}$
        $\mathbf{T}^{(i+1)} = f(\mathbf{y}_e^{(i+1)}, \mathbf{y}_\rho^{(i+1)}, \mathbf{y}_u^{(i+1)})$
        $\rho^{(i+1)} = \mathbf{p}^{(i+1)} / R\mathbf{T}^{(i+1)}$
$\mathbf{y}_{\rho(n+1)} = \mathbf{y}_\rho^{(i_{\max})}$
$\mathbf{y}_{u(n+1)} = \mathbf{y}_u^{(i_{\max})}$
$\mathbf{y}_{e(n+1)} = \mathbf{y}_e^{(i_{\max})}$
```

For instance, this splitting yields for pressure primitive variables to

$$\mathbf{Y}_{\rho e} = \begin{Bmatrix} p \\ T \end{Bmatrix} \tag{48}$$

$$\mathbf{Y}_u = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \tag{49}$$

The nodal unknowns and the equations are ordered in a similar manner

$$\mathbf{y} = \begin{Bmatrix} \mathbf{y}_{\rho e} \\ \mathbf{y}_u \end{Bmatrix} \tag{50}$$

$$\mathbf{R} = \begin{Bmatrix} \mathbf{R}_{\rho e} \\ \mathbf{R}_u \end{Bmatrix} \tag{51}$$

Box 5. Predictor multi-corrector algorithm. Diagonal iterative method with thermodynamic coupling.

```
(Predictor phase)
$\mathbf{y}_{\rho e}^{(0)} = \mathbf{y}_{\rho e(n)}$
$\mathbf{y}_{u}^{(0)} = \mathbf{y}_{u(n)}$
(Multi-corrector phase)
for $i = 0,\ 1, \ldots, i_{\max} - 1$
    $\mathbf{M}_{\rho e \rho e} \Delta \mathbf{y}_{\rho e}^{(i)} = -\mathbf{R}_{\rho e}^{(i)}$
    $\mathbf{M}_{uu} \Delta \mathbf{y}_{u}^{(i)} = -\mathbf{R}_{u}^{(i)}$
    $\mathbf{y}_{\rho e}^{(i+1)} = \mathbf{y}_{\rho e}^{(i)} + \Delta \mathbf{y}_{\rho e}^{(i)}$
    $\mathbf{y}_{u}^{(i+1)} = \mathbf{y}_{u}^{(i)} + \Delta \mathbf{y}_{u}^{(i)}$
$\mathbf{y}_{\rho e(n+1)} = \mathbf{y}_{\rho e}^{(i_{\max})}$
$\mathbf{y}_{u(n+1)} = \mathbf{y}_{u}^{(i_{\max})}$
```

where the residual $\mathbf{R}_{\rho e}$ gathers the contributions of the continuity and energy equations, whereas $\mathbf{R}_u$ gathers the momentum equations.

The Newton–Raphson method can be written as

$$\begin{bmatrix} \mathbf{M}_{\rho e \rho e} & \mathbf{M}_{\rho e u} \\ \mathbf{M}_{u \rho e} & \mathbf{M}_{uu} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{y}_{\rho e} \\ \Delta \mathbf{y}_{u} \end{Bmatrix} = - \begin{Bmatrix} \mathbf{R}_{\rho e} \\ \mathbf{R}_{u} \end{Bmatrix} \tag{52}$$

This system is approximated with a diagonal iterative method, where the coupling blocks are ignored

$$\begin{aligned} \mathbf{M}_{\rho e \rho e} \Delta \mathbf{y}_{\rho e} &= -\mathbf{R}_{\rho e} \\ \mathbf{M}_{uu} \Delta \mathbf{y}_{u} &= -\mathbf{R}_{u} \end{aligned} \tag{53}$$

The matrices for this strategy can be found in Appendix A and the algorithm is outlined in Box 5.

5. NUMERICAL EXAMPLES

In this section several examples of compressible flows are numerically solved with the segregated algorithm presented previously. In all cases the perfect gas state equation is assumed with $\gamma = 1.4$ and $c_v = 716.5$. For inviscid flows, the Hughes–Mallet discontinuity capturing operator (21) has been used while for viscous compressible flows, the minimum operator (23). All computations have been performed in a personal computer with bilinear quadrilateral elements, standard 2×2 Gaussian quadrature and a frontal solver as a linear solver.

5.1. Normal shock

An inviscid normal $M=2$ shock is analysed to demonstrate that the formulation is conservative. The mesh consists of 21×1 square elements covering the domain $-2.1\leqslant x\leqslant 2.1$. The shock is placed at $x=0$ and the initial conditions are as follows:

$$x<0\begin{cases} M=2\\ u_1=1\\ \rho=1\\ p=0.178571\end{cases}\qquad x>0\begin{cases} M=0.577350\\ u_1=0.375000\\ \rho=2.666667\\ p=0.803572\end{cases}\tag{54}$$

The velocity, temperature and density have been specified at the inlet section and natural boundary conditions have been assumed at the outlet. The vertical velocity component has been set to zero in the entire domain.

Figure 1 shows the density variation across the shock where it can be observed that the most accurate results are attained with the non-diagonal stabilizing matrix. The same comment can be extended to the rest of variables like for example the horizontal velocity represented in Figure 2.

The convergence of the algorithm can be evaluated by the evolution of the normalized residuals associated to the continuity and energy equations given in Figures 3 and 4. Generally speaking, the formulation with the non-diagonal tau matrix results in a more robust algorithm since it allows the use of higher CFL numbers which leads to an acceleration of the convergence rate (see Figure 3).

In addition, for the diagonal tau matrix, increasing of the number of corrector passes to two (see Figure 4) does not give a better convergence rate. In the case of the non-diagonal matrix, the convergence rate is improved initially with two corrector passes but at the end, the residuals maintain constant at higher levels than those achieved with only one pass through the corrector loop. In any case, the use of two corrector passes is not recommended since it requires a higher computational effort.

Finally, compared to the corresponding coupled algorithm (see Reference [46]) the segregated version presents a lower convergence rate.

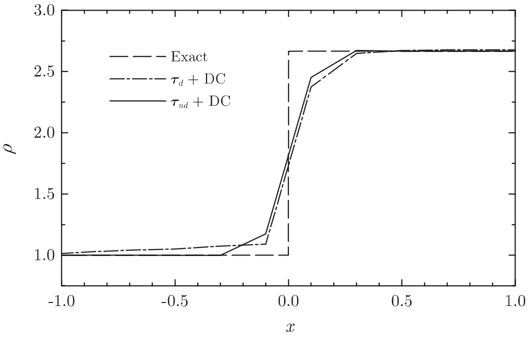

Figure 1. Normal shock wave $M=2$. Density.

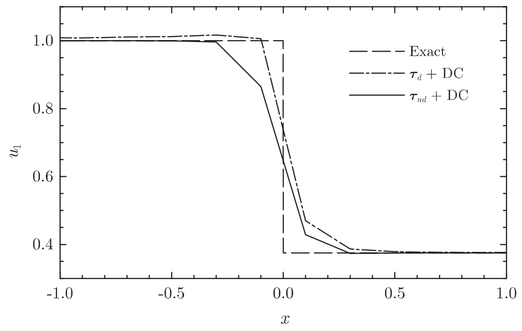

Figure 2. Normal shock wave $M = 2$. Velocity.

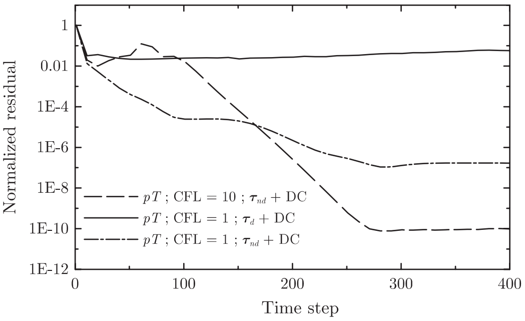

Figure 3. Normal shock wave $M = 2$. Convergence for Δt global, $i_{\max} = 1$.

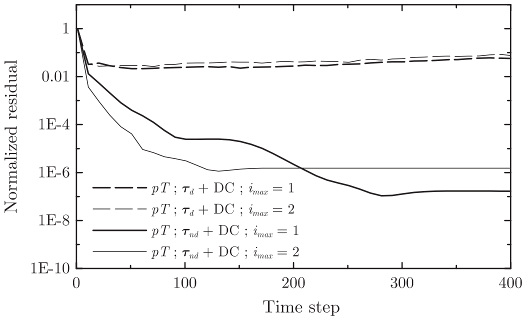

Figure 4. Normal shock wave $M = 2$. Convergence for CFL $= 1$, Δt global.

5.2. Oblique shock wave

This problem consists of an inviscid uniform flow with $M = 2$ which is sharply turned at an angle of 10° by a wall. An oblique shock develops from the corner forming an angle of 29.3° with the wall, as sketched in Figure 5.

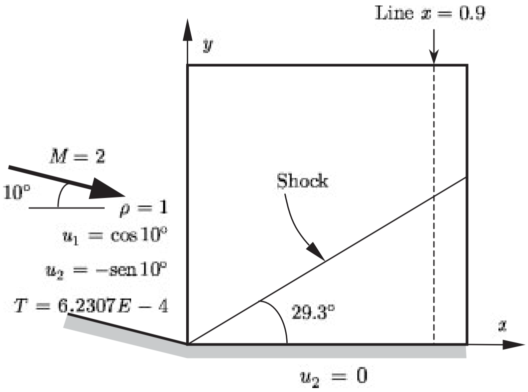

Figure 5. Oblique shock wave $M = 2$. Problem set up.

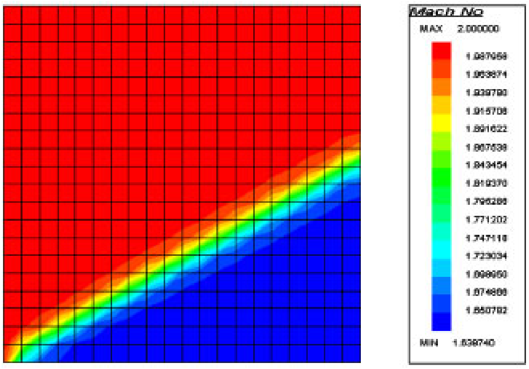

Figure 6. Oblique shock wave $M = 2$. Mach number.

A mesh of 20×20 square elements has been used to discretize the computational domain defined over a unit square. The velocity vector, temperature and density have been especified at the inlet and top boundaries; zero normal velocity has been set at the wall; and none at the outflow.

A contour of the Mach number distribution is shown in Figure 6 where the shock can be clearly distinguished. The density and horizontal velocity variations along the vertical line $x = 0.9$ are plotted in Figures 7 and 8, respectively. As can be observed, the formulation with the non-diagonal matrix gives better accuracy than the diagonal matrix. This problem was not detected when using the coupled solver [46].

Following the guidelines suggested by the previous example, the algorithm has been operated with the global time step strategy and one corrector pass. A condition CFL = 1 has been needed in order to attain convergence.

Regarding the convergence of the algorithm, Figure 9 shows the residual evolution versus time step. Again the non-diagonal matrix leads to better convergence rates.

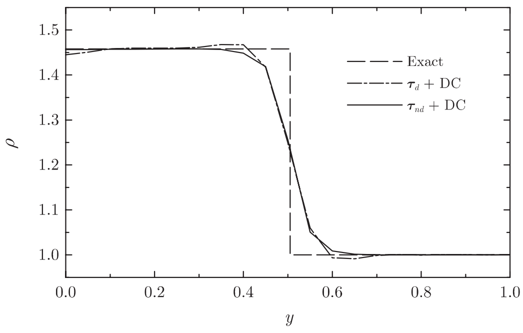

Figure 7. Oblique shock wave $M=2$. Density along $x=0.9$.

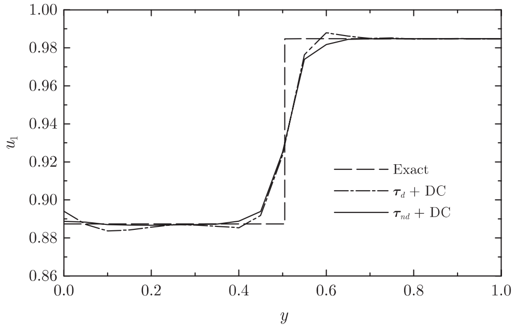

Figure 8. Oblique shock wave $M=2$. x velocity component along $x=0.9$.

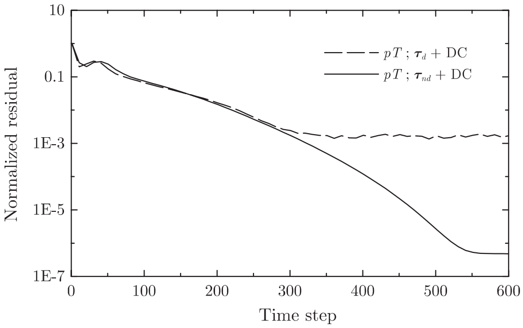

Figure 9. Oblique shock wave $M=2$. Residual convergence. Global time stepping, $\text{CFL}=1$, $i_{\max}=1$.

5.3. *Supersonic flow $M=3$ over a flat plate*

This problem, also known as Carter's flat plate problem, solves the $M=3$ viscous flow which encounters a flat plate at a zero angle of incidence resulting in the generation of a shock and a boundary layer starting from the leading edge of the plate. Carter's numerical solution is published in Reference [50]. The set up of the problem is depicted in Figure 10 where the

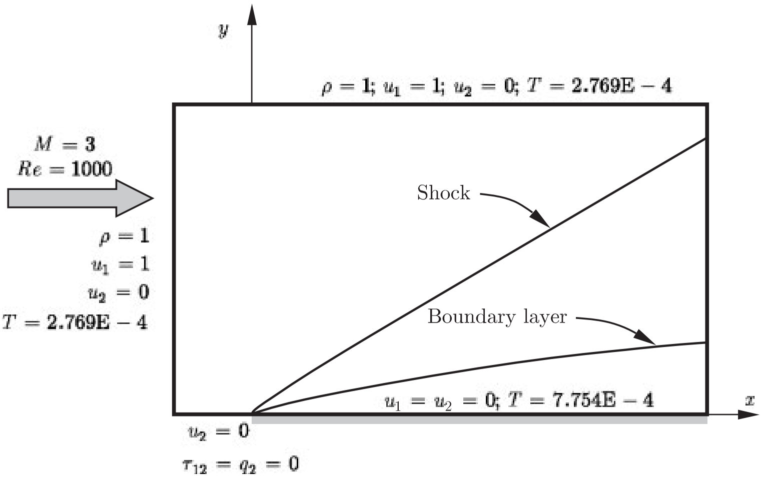

Figure 10. Flat plate $M = 3$. Problem set up.

Reynolds number based on the free stream values and the unit length is 1000. The following Sutherland's law for the viscosity has been assumed:

$$\mu = \frac{0.0906\ T^{1.5}}{T + 0.0001406} \tag{55}$$

The computational domain extends over $-0.2 \leqslant x \leqslant 1.2$ and $0 \leqslant y \leqslant 0.8$ with the origin placed at the upstream corner of the plate. Several meshes have been employed though only the results with a mesh of 112×64 (7168) square elements will be presented.

As boundary conditions, the velocity components, density and temperature have been prescribed at the inflow and top boundaries; zero vertical velocity, tangential viscous traction and heat flux have been specified at the symmetry line; and along the wall the no-slip condition together with the stagnation temperature

$$T_s = T_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2\right) \tag{56}$$

have been applied. The Prandtl number is 0.72.

The algorithm has been operated with non-diagonal tau matrix, global time strategy, the discontinuity operator and a condition $\mathrm{CFL} = 1$ for convergence.

Figure 11 shows the contours of pressure, Mach number, density and temperature. The distributions agree well with the results published by other authors (see References [20, 24]).

To analyse the accuracy of the solution, the pressure, temperature, density and the vertical velocity component along the line $x = 1$ are plotted in Figure 12. The variables have been normalized with respect to the free-stream values. The results are compared with those presented by Carter [50] and Codina *et al.* [24] and the correspondence is fully satisfactory. Pressure is only compared to data from Reference [24] since the pressure distribution obtained by Carter presents oscillations specially near the plate. The formulation with the

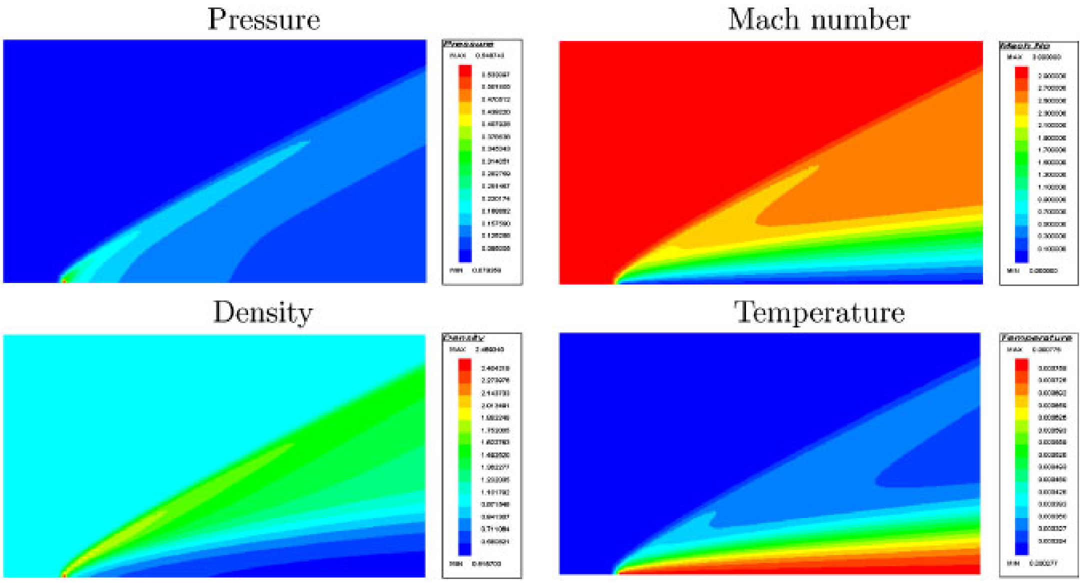

Figure 11. Flat plate $M = 3$. Pressure, Mach number, density and temperature.

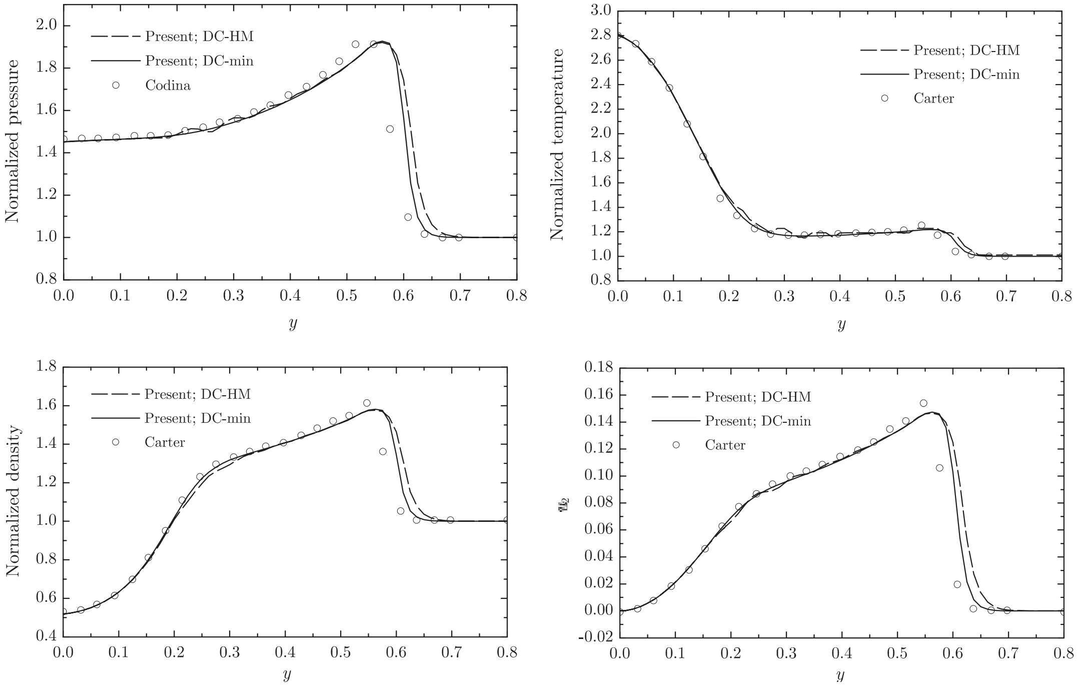

Figure 12. Flat plate $M = 3$. Pressure, temperature, density, y velocity component along $x = 1$.

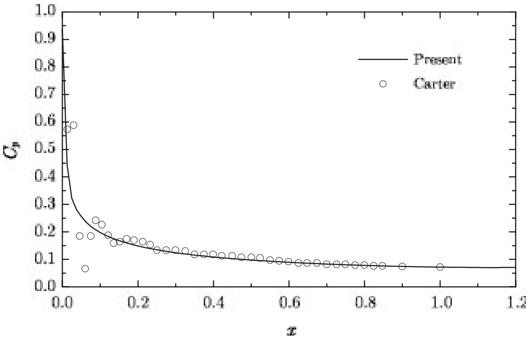

Figure 13. Flat plate $M = 3$. Pressure coefficient.

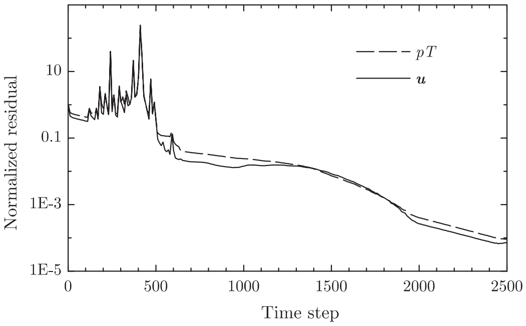

Figure 14. Flat plate $M = 3$. Residual.

Hughes–Mallet discontinuity operator exhibits slight oscillations in the subsonic region which disappear when the minimum operator is activated. In both cases, the solution seems to be somewhat over-diffusive in the shock with respect to the distribution of the referenced authors.

The obtained pressure coefficient along the wall is compared to that published by Carter in Figure 13 showing a smoother variation without oscillations in the case of the algorithm developed in the present work.

The normalized residual evolution is plotted in Figure 14, where some oscillations appear along the first time steps. Again, the segregated approach presents lower convergence rate than the associated coupled algorithm (see Reference [46]).

6. CONCLUSIONS

A segregated algorithm for solving general compressible flows has been presented. The approach generalizes the methodology developed in the first part for isothermal incompressible flows [1] to general divariant fluids. The present algorithm solves the conservation equations via a stabilized finite element method formulated with the set of pressure primitive variables. Then, several predictor multi-corrector algorithms are proposed to solve the system segregatedly.

The results show that simple versions of the segregated algorithm, where the mass and energy conservation equations are treated separately, can only be applied to the resolution of smooth low Mach number flows. Therefore, in order to attain convergence in compressible flows with strong discontinuities, the coupling between the thermodynamic variables must be preserved.

Thus, a thermodynamically coupled approach is proposed which overcomes the drawbacks of previously published methods. For instance, the imposition of boundary conditions including the pressure is straightforward throughout all the corrector passes. In order to achieve convergence, only one pass per equation is necessary. And the method is robust in the presence of strong discontinuities.

Regarding the stabilization matrix it can be concluded that, compared to the diagonal stabilizing matrix, the choice of the non-diagonal stabilization matrix gives better accuracy as well as an increased robustness of the algorithm. This sensitivity to the stabilizing matrix is not present in the coupled solver.

Compared to the coupled approach, segregated methods may present significant advantages, such as considerable RAM memory savings.

APPENDIX A: COEFFICIENT MATRICES FOR THE SEGREGATED FORMULATION WITH THERMODYNAMIC COUPLING

As a function of v the specific volume, the isothermal expansion coefficient can be expressed as

$$\beta_T = -\frac{1}{v}\left(\frac{\partial v}{\partial p}\right)_T = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial p}\right)_T \tag{A1}$$

and the isobaric compressibility coefficient,

$$\alpha_p = \frac{1}{v}\left(\frac{\partial v}{\partial T}\right)_p = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p \tag{A2}$$

Also, let us introduce the auxiliary variables

$$e_1^p = \left(\frac{\partial \rho}{\partial p}\right)_T e_{\text{tot}} + \rho\left(\frac{\partial e}{\partial p}\right)_T \tag{A3}$$

$$e_4^p = \left(\frac{\partial \rho}{\partial T}\right)_p e_{\text{tot}} + \rho\left(\frac{\partial e}{\partial T}\right)_p \tag{A4}$$

Also, the viscous coefficients are μ^{visc}, λ^{visc} and

$$\chi^{\text{visc}} = \lambda^{\text{visc}} + 2\mu^{\text{visc}} \tag{A5}$$

$$\begin{aligned}\mathbf{M}_{\rho e\rho eab} = {} & \frac{1}{\Delta t}\int_{\Omega^e} N_a^e \mathbf{A}_{0\rho e\rho e} N_b^e \, \mathrm{d}\Omega \\ & + \int_{\Omega^e} N_a^e \mathbf{A}_{i\rho e\rho e} N_{b,i}^e \, \mathrm{d}\Omega \\ & + \int_{\Omega^e} N_{a,i}^e \mathbf{K}_{ij\rho e\rho e} N_{b,j}^e \, \mathrm{d}\Omega \\ & + \int_{\Omega^e} N_{a,i}^e \mathbf{A}_i \boldsymbol{\tau} \mathbf{A}_j N_{b,j}^e \, \mathrm{d}\Omega \bigg|_{\rho e\rho e} \\ & + \int_{\Omega^e} v^e g^{ij} N_{a,i}^e \mathbf{A}_{0\rho e\rho e} N_{b,j}^e \, \mathrm{d}\Omega \end{aligned} \tag{A6}$$

$$\begin{aligned}\mathbf{M}_{uuab} = {} & \frac{1}{\Delta t}\int_{\Omega^e} N_a^e \mathbf{A}_{0uu} N_b^e \, \mathrm{d}\Omega \\ & + \int_{\Omega^e} N_a^e \mathbf{A}_{iuu} N_{b,i}^e \, \mathrm{d}\Omega \\ & + \int_{\Omega^e} N_{a,i}^e \mathbf{K}_{ijuu} N_{b,j}^e \, \mathrm{d}\Omega \\ & + \int_{\Omega^e} N_{a,i}^e \mathbf{A}_i \boldsymbol{\tau} \mathbf{A}_j N_{b,j}^e \, \mathrm{d}\Omega \bigg|_{uu} \\ & + \int_{\Omega^e} v^e g^{ij} N_{a,i}^e \mathbf{A}_{0uu} N_{b,j}^e \, \mathrm{d}\Omega \end{aligned} \tag{A7}$$

The block matrices for the mass conservation and total energy equations are:

$$\mathbf{A}_{0\rho e\rho e} = \begin{bmatrix} \rho\beta_T & -\rho\alpha_p \\ e_1^p & e_4^p \end{bmatrix} \tag{A8}$$

$$\mathbf{A}_{1\rho e\rho e} = \begin{bmatrix} \rho\beta_T u_1 & -\rho\alpha_p u_1 \\ u_1 e_2^p & u_1 e_4^p \end{bmatrix} \tag{A9}$$

$$\mathbf{A}_{2\rho e\rho e} = \begin{bmatrix} \rho\beta_T u_2 & -\rho\alpha_p u_2 \\ u_2 e_2^p & u_2 e_4^p \end{bmatrix} \tag{A10}$$

$$\mathbf{A}_{3\rho e\rho e} = \begin{bmatrix} \rho\beta_T u_3 & -\rho\alpha_p u_3 \\ u_3 e_2^p & u_3 e_4^p \end{bmatrix} \tag{A11}$$

$$\mathbf{K}_{11\rho e\rho e} = \mathbf{K}_{22\rho e\rho e} = \mathbf{K}_{33\rho e\rho e} = \begin{bmatrix} 0 & 0 \\ 0 & \kappa \end{bmatrix} \tag{A12}$$

$$\mathbf{K}_{ij\rho e\rho e} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad i \neq j \tag{A13}$$

And the block matrices to build the momentum equations are

$$\mathbf{A}_{0uu} = \rho \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A14}$$

$$\mathbf{A}_{1uu} = \rho \begin{bmatrix} 2u_1 & 0 & 0 \\ u_2 & u_1 & 0 \\ u_3 & 0 & u_1 \end{bmatrix} \tag{A15}$$

$$\mathbf{A}_{2uu} = \rho \begin{bmatrix} u_2 & u_1 & 0 \\ 0 & 2u_2 & 0 \\ 0 & u_3 & u_2 \end{bmatrix} \tag{A16}$$

$$\mathbf{A}_{3uu} = \rho \begin{bmatrix} u_3 & 0 & u_1 \\ 0 & u_3 & u_2 \\ 0 & 0 & 2u_3 \end{bmatrix} \tag{A17}$$

$$\mathbf{K}_{11uu} = \begin{bmatrix} \chi^{\text{visc}} & 0 & 0 \\ 0 & \mu^{\text{visc}} & 0 \\ 0 & 0 & \mu^{\text{visc}} \end{bmatrix} \tag{A18}$$

$$\mathbf{K}_{22uu} = \begin{bmatrix} \mu^{\text{visc}} & 0 & 0 \\ 0 & \chi^{\text{visc}} & 0 \\ 0 & 0 & \mu^{\text{visc}} \end{bmatrix} \tag{A19}$$

$$\mathbf{K}_{33uu} = \begin{bmatrix} \mu^{\text{visc}} & 0 & 0 \\ 0 & \mu^{\text{visc}} & 0 \\ 0 & 0 & \chi^{\text{visc}} \end{bmatrix} \tag{A20}$$

$$\mathbf{K}_{12uu} = \mathbf{K}_{21uu}^{\mathrm{T}} = \begin{bmatrix} 0 & \lambda^{\mathrm{visc}} & 0 \\ \mu^{\mathrm{visc}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A21}$$

$$\mathbf{K}_{13uu} = \mathbf{K}_{31uu}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & \lambda^{\mathrm{visc}} \\ 0 & 0 & 0 \\ \mu^{\mathrm{visc}} & 0 & 0 \end{bmatrix} \tag{A22}$$

$$\mathbf{K}_{23uu} = \mathbf{K}_{32uu}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \lambda^{\mathrm{visc}} \\ 0 & \mu^{\mathrm{visc}} & 0 \end{bmatrix} \tag{A23}$$

ACKNOWLEDGEMENTS

This work has been partially sponsored by the Ministerio de Educación y Ciencia, Spain and by the University of the Basque Country, Spain project 1/UPV/EHU 00145.345-EA-8246/2000.

REFERENCES

1. Hauke G, Landaberea A, Garmendia I, Canales J. A segregated method for compressible flow computation. Part I: isothermal compressible flows. *International Journal for Numerical Methods in Fluids*, submitted.
2. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
3. Hughes TJR, Liu WK, Brooks A. Finite element analysis of incompressible flows by the penalty function formulation. *Journal of Computational Physics* 1979; **30**:1–60.
4. Reddy JN. On penalty function methods in the finite-element analysis of flow problems. *International Journal for Numerical Methods in Fluids* 1982; **2**:151–171.
5. Segal A. On the numerical solution of the Stokes equations using the finite element methods. *Computer Methods in Applied Mechanics and Engineering* 1979; **19**:165–185.
6. Fortin M, Fortin A. A generalization of Uzawa's algorithm for the solution of the Navier–Stokes equations. *Communications in Applied Numerical Methods* 1985; **1**:205–208.
7. Simo JC, Armero F. Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier–Stokes and Euler equations. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**: 111–154.
8. Benim AC, Zinser W. A segregated formulation of Navier–Stokes equations with finite elements. *Computer Methods in Applied Mechanics and Engineering* 1986; **57**:223–237.
9. Rice JG, Schnipke RJ. An equal-order velocity-pressure formulation that does not exhibit spurious pressure modes. *Computer Methods in Applied Mechanics and Engineering* 1986; **58**:135–149.
10. Shaw CT. Using a segregated finite element scheme to solve the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1991; **12**:81–92.
11. Haroutunian V, Engelman MS, Hasbani I. Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems. *International Journal for Numerical Methods in Fluids* 1993; **17**: 323–348.
12. Taylor C, Hood P. A numerical solution of the Navier–Stokes equations using the finite element methods. *Computer & Fluids* 1973; **1**:73–100.
13. Huyakorn RS, Taylor C, Lee RL, Gresho PM. A comparison of various mixed-interpolation finite elements in the velocity-pressure formulation of the Navier–Stokes equations. *Computer & Fluids* 1978; **6**:25–35.
14. Lee RL, Gresho PM, Sani RL. Smoothing techniques for certain primitive variable solutions of the Navier–Stokes equations. *International Journal for Numerical Methods in Engineering* 1979; **14**:1785–1804.

15. Gunzburger MD. *Finite Element Methods for Viscous Incompressible Flows*. Academic Press Inc.: New York, 1989.
16. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: New York, 1991.
17. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method*: *Advection-Diffusion and Isothermal Laminar Flow*. Wiley: England, 1998.
18. Zienkiewicz OC. Morgan K, Satya Sai BVK, Codina R, Vasquez M. A general algorithm for compressible and incompressible flow—Part II. Tests on the explicit form. *International Journal for Numerical Methods in Fluids* 1995; **20**:887–913.
19. Hauke G, Hughes TJR. A unified approach to compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1994; **113**:389–395.
20. Hauke G, Hughes TJR. A comparative study of different sets of variables for solving compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1998; **153**:1–44.
21. Zienkiewicz OC, Wu J. Incompressibility without tears—How to avoid restrictions of mixed formulation. *International Journal for Numerical Methods in Engineering* 1991; **32**:1189–1203.
22. Zienkiewicz OC, Wu J. A general explicit or semi-explicit algorithm for compressible and incompressible flows. *International Journal for Numerical Methods in Engineering* 1992; **35**:457–479.
23. Zienkiewicz OC, Codina R. A general algorithm for compressible and incompressible flow—Part I. The split, characteristic-based scheme. *International Journal for Numerical Methods in Fluids* 1995; **20**:869–885.
24. Codina R, Vazquez M, Zienkiewicz OC. A general algorithm for compressible and incompressible flow—Part III. The semi-implicit form. *International Journal for Numerical Methods in Fluids* 1998; **27**:13–32.
25. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
26. Franca LP, Frey SL. Stabilized finite element methods: II. The incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:209–233.
27. Franca LP, Hughes TJR. Convergence analyses of Galerkin least-squares methods for symmetric advective–diffusive forms of the Stokes and incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1993; **105**:285–298.
28. Hughes TJR, Tezduyar TE. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering* 1984; **45**: 217–284.
29. Hughes TJR, Franca LP, Mallet M. 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. *Computer Methods in Applied Mechanics and Engineering* 1986; **54**:223–234.
30. Hughes TJR, Mallet M, Mizukami A. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering* 1986; **54**:341–355.
31. Hughes TJR, Mallet M. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advection-diffusion systems. *Computer Methods in Applied Mechanics and Engineering* 1986; **58**:305–328.
32. Hughes TJR, Franca LP, Mallet M. A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advective–diffusive systems. *Computer Methods in Applied Mechanics and Engineering* 1987; **63**:97–112.
33. Hughes TJR, Franca LP, Hulbert G. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:173–189.
34. Shakib F, Hughes TJR. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms. *Computer Methods in Applied Mechanics and Engineering* 1991; **87**:35–58.
35. Shakib F, Hughes TJR, Johan Z. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1991; **89**:141–219.
36. Brueckner FP, Heinrich JC. Petrov–Galerkin finite element model for compressible flows. *International Journal for Numerical Methods in Engineering* 1991; **32**:255–274.
37. Le Beau GJ, Ray SE, Aliabadi SK, Tezduyar TE. SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. *Computer Methods in Applied Mechanics and Engineering* 1993; **104**:397–422.
38. Hansbo P. Explicit streamline diffusion finite element methods for the compressible Euler equations in conservation variables. *Journal of Computational Physics* 1993; **109**:274–288.
39. Hughes TJR, Hauke G, Jansen K, Johan Z. In *Stabilized Finite Element Methods in Fluids: Inspirations, Origins, Status and Recent Developments.* Hughes TJR, Oñate E, Zienkiewicz OC (eds). A book dedicated to Robert L. Taylor (*CIMNE*, Barcelona, 1994).

40. Soulaimani A, Fortin M. Finite element solution of compressible viscous flows using conservative variables. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:319–350.
41. Taylor CA, Hughes TJR, Zarins CK. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering* 1998; **158**:155–196.
42. Godunov SK. The problem of a generalized solution in the theory of quasi-linear equations and in gas dynamics. *Russian Mathematical Surveys* 1962; **17**:145–156.
43. Mock M. Systems of conservation laws of mixed type. *Journal of Differential Equations* 1980; **37**:70–88.
44. Harten A. On the symmetric form of systems of conservation laws with entropy. *Journal of Computational Physics* 1983; **49**:151–164.
45. Chalot F, Hughes TJR, Shakib F. Symmetrization of conservation laws with entropy for high-temperature hypersonic computations. *Computer Systems in Engineering* 1990; **1**:495–521.
46. Hauke G. Simple stabilizing matrices for the computation of compressible flows in primitive variables. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:6881–6893.
47. Hughes TJR, Mallet M. A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective–diffusive systems. *Computer Methods in Applied Mechanics and Engineering* 1986; **58**:329–336.
48. Hughes, Pister, Taylor. Implicit–Explicit finite elements in non-linear transient analysis. *Computer Methods in Applied Mechanics and Engineering* 1979; **17–18**:159–182.
49. Landaberea A. Un método segregado para la resolución de las ecuaciones de dinámica de fluidos mediante el método de los elementos finitos. *Ph.D. Thesis*, Universidad del País Vasco, 2002.
50. Carter JE. Numerical solutions of the Navier–Stokes equations for the supersonic laminar flow over a two-dimensional compression corner. *NASA Technical Report*, *NASA TR-R-385*, 1972.