INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:735–765
Published online 10 September 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.762

Numerical simulation of dense gas flows on unstructured grids with an implicit high resolution upwind Euler solver

P. Colonna[1,*,†] and S. Rebay[2]

[1]*Faculty of Design, Engineering and Production, Energy Technology Section, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*
[2]*Dip. di Ingegneria Meccanica, Università di Brescia, via Branze 38, 25123 Brescia, Italy*

SUMMARY

The study of the dense gas flows which occur in many technological applications demands for fluid dynamic simulation tools incorporating complex thermodynamic models that are not usually available in commercial software. Moreover, the software mentioned can be used to study very interesting phenomena that usually go under the name of 'non-classical gasdynamics', which are theoretically predicted for high molecular weight fluids in the superheated region, close to saturation. This paper presents the numerical methods and models implemented in a computer code named *zFlow* which is capable of simulating inviscid dense gas flows in complex geometries. A detailed description of the space discretization method used to approximate the Euler equations on unstructured grids and for general equations of state, and a summary of the thermodynamic functions required by the mentioned formulation are also given. The performance of the code is demonstrated by presenting two applications, the calculation of the transonic flow around an airfoil computed with both the ideal gas and a complex equation of state and the simulation of the non-classical phenomena occurring in a supersonic flow between two staggered sinusoidal blades. Non-classical effects are simulated in a supersonic flow of a siloxane using a Peng–Robinson-type equation of state. Siloxanes are a class of substances used as working fluids in organic Rankine cycles turbines. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: Euler solver; unstructured grid; high resolution upwind space discretization; implicit time integration; dense gas; real gas; equations of state; non-classical gasdynamics; BZT fluids; siloxanes; organic Rankine cycle

1. INTRODUCTION

In the numerical simulation of gaseous flows the polytropic ideal gas thermodynamic model is usually employed because of its simplicity and the corresponding low computational cost. This is an acceptable approximation in many cases, but there are applications in which the

*Correspondence to: P. Colonna, Faculty of Design, Engineering and Production, Energy Technology Section, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.
†E-mail: P.Colonna@WbMT.TUDelft.nl

Received 10 December 2003
Revised 2 June 2004

flow conditions lies in a thermodynamic region where the ideal gas approximation cannot be applied (usually defined as the dense gas region).

This is the case, for example, in the study of dense gas flows which may display so-called non-classical gas dynamic phenomena. In a gas region close to saturation at high pressure and temperature and for flows of complex molecules, in an isentropic expansion, the speed of sound increases with decreasing density, as opposed to what is prescribed by applying the ideal gas approximation. Details of the basic theory for non-classical gas dynamics of dense gases can be found for example in References [1, 2]. The opposite trend of the sound speed with respect to density introduces many interesting effects, like the impossibility of compression shocks in that region, the suppression of shock-induced separation and non-conventional shapes of the channel for the supersonic acceleration from supercritical stagnation conditions (see e.g. References [2–9]). Fluids exhibiting non-classical behaviour are called BZT after the initials of the scientists who first formulated the theory.

CFD simulations of dense gases in the classical regime are reported in the literature. Several works describe the study of flows through nozzles [10] or around simple airfoil geometries [11–13]. Two-dimensional simulations of the flow through turbine cascades are treated in References [14, 15]. The complexity of the numerical methods varies from an explicit finite volume method (FVM) applied to the two-dimensional Euler and Navier–Stokes equations [11], to an upwind scheme for the solution of the Euler equations in the axisymmetric form [10], to an implicit, finite-volume upwind scheme for the solution of the two-dimensional Euler and Navier–Stokes equations [12, 15]. Fluid thermodynamic models also varies in complexity from the Van der Waals [10] and Redlich–Kwong-type equations of state (EOS) [14] to the Beatty–Bridgemann [11], Martin–Hou [12, 13] and BWR [10] EOSs.

Other documented simulations are focused specifically on BZT effects in fluid flows. Nozzle and simple geometries are treated e.g. in References [8, 16–18]. Shock tube flows are treated e.g. in References [19–22]. In References [16, 17] an FVM by Jameson is used to solve the two-dimensional inviscid equations. In References [8, 19–21] a predictor–corrector total variational diminishing (TVD) scheme is used to solve the time-dependent, two-dimensional Euler equations. In Reference [22] the upwind scheme of Reference [23] was adopted to solve the three-dimensional Euler equations. A transonic small-disturbance approximation of the Navier–Stokes equations is solved with a Murman and Cole method in Reference [18]. In all the cited works dealing with BZT effects, either the Van der Waals [8, 16, 17, 19] or the Martin–Hou [18, 20–22] EOSs were adopted. Two-dimensional flows exhibiting non-classical effects through turbine cascades were simulated in References [9, 24, 25]. In Reference [24], the Van der Waals model was employed while the Martin–Hou EOS was adopted in References [9, 25]. In Reference [24] the FVM by Jameson was used, while the McCormack scheme was adopted in Reference [25] and a second-order time-space accurate flux-limited method by Davis was employed in Reference [9] to solve the Euler equations. In all the publications mentioned, structured grids were employed for the space discretization except for the works in References [17, 22] where unstructured grids were used.

The study of non-classical gas dynamic phenomena has been one of the driving forces for the development of a CFD code named *zFlow*, which integrates a modern numerical discretization scheme with accurate thermodynamic models to describe dense gas effects. The objective of this paper is to present the theory, implementation, validation and the preliminary results obtained with *zFlow*. The code is presently limited to the solution of the Euler

equations, i.e. viscous effects are not considered, but can nevertheless give substantial information on the non-classical gasdynamic effects mentioned above. Other interesting phenomena which could be strongly influenced by non-classical effects, such as for example the interaction between a shock impinging on a solid wall and the boundary layer, cannot be studied unless viscous effects (and turbulence) are considered. The extension of the code to the solution of the full Reynolds-averaged Navier–Stokes equations is however planned for the near future.

The main features characterizing the CFD code *zFlow* are hereby summarized. The spatial approximation of the Euler equations is constructed with an high resolution finite volume method suitable for general unstructured and hybrid grids. The method can be regarded as an hybrid between the finite element (FE) and finite volume (FV) methods in that the finite volume metric quantities are constructed on the basis of the Lagrangian polynomial shape functions typically used in finite element methods [26]. As a consequence it can be proved that this type of FV schemes with a centred numerical flux function is identical to a classical FE discretization [27]. The high-resolution upwind discretization is constructed on the basis of the Roe approximate Riemann solver (ARS) [28–30] generalized to the case of fluids characterized by arbitrary equations of state according to the method of Vinokur and Montagnè [31]. The interested reader may find a comparison of some possible approaches for the construction of generalized ARS in Reference [32], see also e.g. References [23, 33–35]. It is important to remark that the use of unstructured grids allows the straightforward treatment of domains of arbitrarily complex geometry. Another important feature of *zFlow* is represented by the adopted implicit time integration scheme, which allows for the computation of steady-state solutions in a much more efficient way with respect to conventional explicit schemes. The gain in computational efficiency is crucial when complex equation of state are needed for an accurate flow simulation. The CFD code *zFlow* integrates a rich thermodynamic library for the calculation of properties of pure fluids and mixtures which has been extended to include the particular thermodynamic functions requested by the implicit upwind flow solver (see Reference [36]). To the knowledge of the authors, the use in a CFD code of thermodynamic models like the modified Keenan–Keyes EOS for water [37, 38], the Haar–Gallagher EOS for ammonia [37, 39], the Starling EOS for light hydrocarbons [37, 40] and the Peng–Robinson–Strijeck–Vera (PRSV) cubic equation of state for pure fluids and its extension to mixtures by means of the Wong–Sandler mixing rules is documented here for the first time.

In Section 2 of this paper, particular emphasis is given to the discussion of high-resolution upwind formulation for unstructured grids and general equations of state as implemented in *zFlow*. The boundary treatment in the case of general gases is also thoroughly presented. Section 3 gives an overview of the additional thermodynamic functions that the thermodynamic library must provide to the CFD solver. Section 4 presents the results of a validation test—a well known test case for inviscid flows involving a transonic flow of air around an airfoil is compared to a *zFlow* calculation in which nitrogen is modelled with the PRSV EOS. Section 5 shows an evaluation of the computational efficiency of the implemented implicit backward Euler time integration scheme. A first assessment of the *zFlow* capability of simulating non-classical gas dynamic effects is illustrated in Section 6 and compared with similar calculations by other researchers. In the example, a two-dimensional supersonic flow between two staggered sinusoidal blades is calculated. Fluids belonging to the siloxanes class are currently employed as working fluids in organic Rankine cycles (ORC).

2. NUMERICAL SOLUTION OF THE EULER EQUATIONS ON UNSTRUCTURED GRIDS FOR COMPLEX EQUATIONS OF STATE

This section illustrates the discretization techniques of the Euler equations implemented in the computer program *zFlow*. A brief review of the theory of the equations governing the motion of inviscid compressible flows, including the treatment of the initial and boundary conditions, is first given. The finite element–finite volume (FEFV) space discretization technique used in *zFlow* is then presented. Particular emphasis is given to the high-resolution upwind formulation suitable for general unstructured and hybrid grids and general EOS and to the procedure used to enforce the boundary conditions in the case of general EOS. The explicit and implicit time integration methods available in the code are also shown.

It should be noted that the methodologies hereby described have been developed by various authors of the CFD community in the past decade and can be found, albeit in a somewhat scattered fashion, in the available CFD literature. No claim of originality is therefore made by the authors from a methodological perspective. The material which follows has therefore to be regarded as a brief review of the subject which allows for a self-contained reading of the paper.

2.1. Governing equations

The Euler equations can be written in the so-called 'divergence' or 'conservative' form as,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{F}(\mathbf{u}) = 0 \tag{1}$$

where the conservative variable vector $\mathbf{u} \in \mathbb{R}^{d+2}$, the flux function $\mathbf{F}(\mathbf{u}) \in \mathbb{R}^d \otimes \mathbb{R}^{d+2}$, $\mathbf{x} \in \Omega \subset \mathbb{R}^d$, $t \in [0, T] \subset \mathbb{R}$, and d denotes the number of space dimensions. Here and in the following, bold symbols are used to indicate vectors both in the physical space $\mathbb{R}^d$ and in the so-called 'state space' $\mathbb{R}^{d+2}$. The distinction between the two cases is explicitly indicated when not clear from the context. Bold capitals instead indicate quantities belonging to the $\mathbb{R}^d \otimes \mathbb{R}^{d+2}$ such as the flux $\mathbf{F}(\mathbf{u})$. The conservative variable vector $\mathbf{u}$ and the cartesian components $\mathbf{f}_j(\mathbf{u})$ of the flux $\mathbf{F}(\mathbf{u})$ are given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho e^t \\ \rho q_i \end{bmatrix}, \qquad \mathbf{f}_j(\mathbf{u}) = \begin{bmatrix} \rho q_j \\ (\rho e^t + P) q_j \\ \rho q_i q_j + P \delta_{ij} \end{bmatrix} \tag{2}$$

where ρ denotes the density, e^t the stagnation (or total) energy (i.e. the sum of the internal and kinetic energy), P the pressure, q_j the cartesian components of the velocity vector $\mathbf{q}$, and δ_{ij} the Kroenecker symbol. System (1) represents $d+2$ partial differential equations in the $d+2$ unknowns $\mathbf{u}$ if the pressure is regarded as a known function $\Pi(\mathbf{u})$ given by

$$\Pi(\mathbf{u}) = \Pi(\rho, \rho e^t, \rho \mathbf{q}) = P\left(\rho, \frac{\rho e^t}{\rho} - \frac{|\rho \mathbf{q}|^2}{2\rho^2}\right) = P(\rho, e) \tag{3}$$

where $P(\rho, e)$ is a thermodynamic equation of state (see Appendix A for more details).

A particularly convenient starting point for the construction of a space-discretized version of the Euler equations is the so-called 'weak' or 'variational' form,

$$\int_\Omega \phi \frac{\partial \mathbf{u}}{\partial t}\, \mathrm{d}\Omega - \int_\Omega \nabla_{\mathbf{x}}\phi \cdot \mathbf{F}(\mathbf{u})\, \mathrm{d}\Omega + \oint_{\partial\Omega} \phi \mathbf{n} \cdot \mathbf{F}(\mathbf{u})\, \mathrm{d}\sigma = 0 \quad \forall \phi \in V \tag{4}$$

which must be satisfied for any function ϕ belonging to a suitable function space V. The weak form can be obtained by multiplying the differential form (1) by a test function $\phi \in V$, integrating over the domain Ω and performing an integration by parts of the term containing the divergence of $\mathbf{F}(\mathbf{u})$.

The Euler equations must be supplemented by consistent initial and boundary conditions. The initial data is simply an arbitrary field $\mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0(\mathbf{x})$. To impose the boundary condition in the context of the weak or variational formulation, the flux function $\mathbf{n} \cdot \mathbf{F}(\mathbf{u})$ appearing in the boundary integral of (4) must be evaluated with the 'boundary state' $\mathbf{u}^{\mathrm{b}}$. The evaluation of $\mathbf{u}^{\mathrm{b}}$ for a fluid characterized by an arbitrary equation of state is described in Section 2.3. The weak formulation with weakly prescribed boundary conditions therefore reads,

$$\int_\Omega \phi \frac{\partial \mathbf{u}}{\partial t}\, \mathrm{d}\Omega - \int_\Omega \nabla_{\mathbf{x}}\phi \cdot \mathbf{F}(\mathbf{u})\, \mathrm{d}\Omega + \oint_{\partial\Omega} \phi \mathbf{n} \cdot \mathbf{F}(\mathbf{u}^{\mathrm{b}})\, \mathrm{d}\sigma = 0 \quad \forall \phi \in V \tag{5}$$

2.2. Space discretization

The first step needed for the construction of a discrete version of a partial differential problem is the geometrical discretization of the domain Ω which must be approximated as a collection of non-overlapping elements $\{E\}$, namely triangles or quadrilateral in 2D, and tetrahedra, prisms, pyramids and hexahedra in 3D. The set $\{E\}$ is referred to as the 'grid', 'mesh' or the 'triangulation' of the domain (even when the elements are not actual triangles) and is denoted by the symbol $\mathscr{T}_h$. Ω_e indicates the domain associated to element E (i.e. the set of points internal or on the boundary of E). For every vertex (hereafter also called node) of $\mathscr{T}_h$, Ω_i refers to the union of the domains Ω_e of the elements E which contain the node i, and $\partial\Omega_i$ refers to the boundary of Ω_i. Γ_i denotes the set $\Omega_i \cap \partial\Omega$. The set of all the nodes of $\mathscr{T}_h$ is denoted by $\mathscr{K}$ and that of the boundary nodes by $\mathscr{K}^{\partial}$. The nodes inside or on the boundary of Ω_i are instead denoted by $\mathscr{K}_i$, $\mathscr{K}_{i\neq}$ denotes the set $\mathscr{K}_i$ except node i, and $\mathscr{K}_i^{\partial} = \mathscr{K}_i \cap \mathscr{K}^{\partial}$ (see Figure 1).

The adopted space-discretization method closely follows the ideas introduced in Reference [26] and subsequently extended in References [27, 41], and can be regarded as an hybrid finite element/finite volume (FE/FV) scheme. As a preliminary step towards the description of the discretization method used in this work, the FV and the FE space discretization of the Euler equations are therefore briefly reviewed. The notation is the one employed in Reference [41].

The FV schemes can be constructed by searching an approximate solution of the weak formulation (5) among functions $\mathbf{u}_h$ which belong to the finite-dimensional space V_h^0 of piecewise constant functions over a set of non-overlapping regions $\mathscr{C}_k$ which cover the entire domain Ω. A convenient basis of the space V_h^0 is given by the characteristic functions $I_k(\mathbf{x})$ which are null for $\mathbf{x} \notin \mathscr{C}_k$ and are equal to 1 for $\mathbf{x} \in \mathscr{C}_k$. A generic function $\mathbf{u}_h(\mathbf{x}, t) \in V_h^0$ or $\phi_h(\mathbf{x}) \in V_h^0$ can in fact be written as

$$\phi_h(\mathbf{x}) = \sum_{k \in \mathscr{K}} I_k(\mathbf{x}) \phi_k, \quad \mathbf{u}_h(\mathbf{x}, t) = \sum_{k \in \mathscr{K}} I_k(\mathbf{x}) \mathbf{u}_k(t)$$

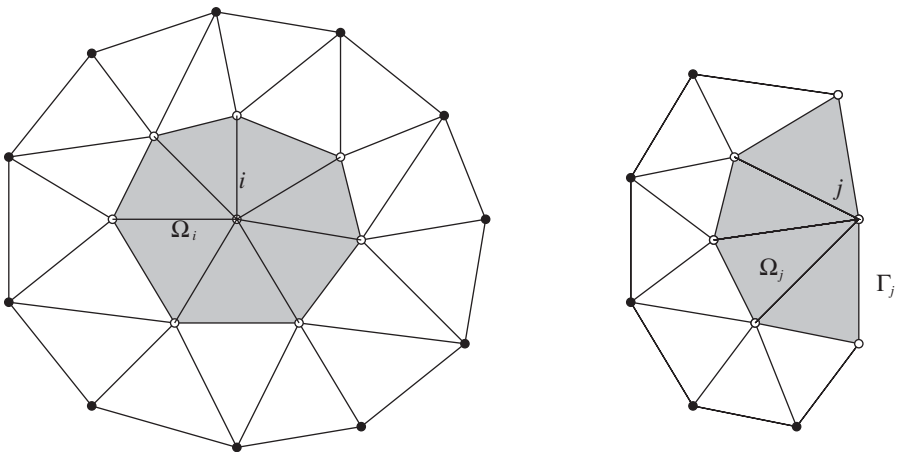

Figure 1. Simple 2D triangulations showing the domains Ω_i and Ω_j associated with an internal (left) and a boundary (right) point, respectively. The white circles indicate points $i \in \mathscr{K}_i$ and $j \in \mathscr{K}_j$.

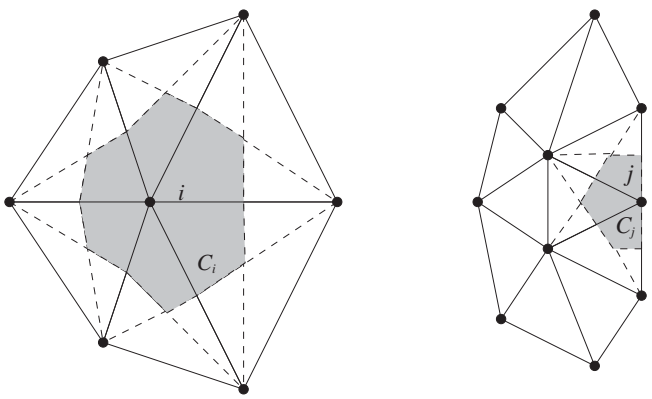

Figure 2. Simple 2D triangulations showing the domains $\mathscr{C}_i$ and $\mathscr{C}_j$ associated with an internal (left) and a boundary (right) point, respectively.

where the expansion coefficients $\mathbf{u}_k(t)$ and ϕ_k are simply the values of $\mathbf{u}_h(\mathbf{x},t)$ or $\phi_h(\mathbf{x})$ for $\mathbf{x} \in \mathscr{C}_k$. The so-called node-centred (or vertex-centred) FV schemes are obtained if the volumes are associated to the nodes $\mathscr{K}$ as depicted in Figure 2. A discrete version of the weak form (5) can be obtained by replacing the functions $\phi \in V$ and $u \in V$ with functions $\phi_h \in V_h^0$ and $u_h \in V_h^0$ thus obtaining, for a generic cell $\mathscr{C}_i$, the equation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathscr{C}_i} \mathbf{u}_h \,\mathrm{d}\Omega + \sum_{k \in \mathscr{K}_{i,\neq}} \int_{\Sigma_{ik}} \mathbf{n}\cdot\mathbf{F}_i \,\mathrm{d}\sigma + \int_{\Sigma_i} \mathbf{n}\cdot\mathbf{F}(\mathbf{u}^{\mathrm{b}}) \,\mathrm{d}\sigma = 0 \tag{6}$$

where $\mathbf{F}_i = \mathbf{F}(\mathbf{u}_i)$ and the boundary $\partial\mathscr{C}_i$ of $\mathscr{C}_i$ has been regarded as the union of interfaces $\Sigma_{ik} = \partial\mathscr{C}_i \cap \partial\mathscr{C}_k$ and, if the cell $\mathscr{C}_i$ is adjacent to the boundary of the domain, of the boundary face $\Sigma_i = \partial\mathscr{C}_i \cap \partial\Omega$.

Note that the flux function appearing in the interface integrals is not uniquely defined since $\mathbf{u}_h$ is discontinuous on Σ_{ik}. The normal physical flux $\mathbf{n}\cdot\mathbf{F}_i$ on Σ_{ik} is therefore replaced by a numerical flux function $\hat{\mathbf{f}}(\mathbf{u}_i,\mathbf{u}_k,\mathbf{n})$ which depends on the states $\mathbf{u}_i$ and $\mathbf{u}_k$ of the two cells sharing Σ_{ik} and on the normal vector $\mathbf{n}$. If the discrete unknown is defined as

$$\mathbf{u}_i(t)=\frac{1}{|\mathscr{C}_i|}\int_{\mathscr{C}_i}\mathbf{u}_h\,\mathrm{d}\Omega$$

the semi-discrete equations with the physical flux replaced by the numerical flux can therefore be written as,

$$|\mathscr{C}_i|\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t}+\sum_{k\in\mathscr{K}_{i,\neq}}\hat{\mathbf{f}}(\mathbf{u}_i,\mathbf{u}_k,\mathbf{v}_{ik})+\mathbf{v}_i\cdot\mathbf{F}_i^{\mathrm{b}}=0,\quad \mathbf{v}_{ik}=\int_{\Sigma_{ik}}\mathbf{n}\,\mathrm{d}\sigma,\quad \mathbf{v}_i=\int_{\Sigma_i}\mathbf{n}\,\mathrm{d}\sigma \tag{7}$$

where the boundary state $\mathbf{u}^{\mathrm{b}}$ is assumed to be constant over Σ_i.

FE methods can be instead constructed starting from the weak formulation (5) by considering a finite-dimensional space V_h^1 of linear or bilinear test functions ϕ_h, i.e. of functions which are globally continuous and linear or bilinear inside the elements E of $\mathscr{T}_h$. A convenient basis for V_h^1 is given by the Lagrangian polynomials N_i, which are associated with the vertices of the triangulation and which are different from zero only inside Ω_i. A generic function $\phi_h(\mathbf{x})$ of $\mathbf{u}_h(\mathbf{x},t)$ can therefore be written as the expansions,

$$\phi_h(\mathbf{x})=\sum_{k\in\mathscr{K}}N_k(\mathbf{x})\phi_k,\quad \mathbf{u}_h(\mathbf{x},t)=\sum_{k\in\mathscr{K}}N_k(\mathbf{x})\mathbf{u}_k(t) \tag{8}$$

where ϕ_i and $\mathbf{u}_h(t)$ are expansion coefficients which in fact are equal to the nodal values of the functions $\phi_h(\mathbf{x}_i)$ or $\mathbf{u}_h(\mathbf{x}_i,t)$. The approximate solution can be searched among functions u_h which are in the same finite-dimensional space V_h^1 chosen for ϕ_h (Galerkin method). By substituting expansions (8) into (5) the semi-discrete equations for a generic node i of $\mathscr{T}_h$ are obtained, i.e.

$$\sum_{k\in\mathscr{K}_i}\left[\int_{\Omega_i}N_iN_k\,\mathrm{d}\Omega\right]\frac{\mathrm{d}\mathbf{u}_k(t)}{\mathrm{d}t}+\int_{\Gamma_i}N_i\mathbf{n}\cdot\mathbf{F}(\mathbf{u}^{\mathrm{b}})\,\mathrm{d}\sigma-\int_{\Omega_i}\boldsymbol{\nabla}_{\mathbf{x}}N_i\cdot\mathbf{F}\left(\sum_{k\in\mathscr{K}_i}N_k\mathbf{u}_k(t)\right)\mathrm{d}\Omega=0 \tag{9}$$

where the various integrals have been restricted to Ω_i since N_i vanishes outside Ω_i, $\Gamma_i=\partial\Omega_i\cap\partial\Omega$, and the summations have been restricted according to the support of N_i. By reinterpolating the flux function by means of the same basis functions N_i used to approximate the test function and the unknown, i.e.

$$\mathbf{F}(\mathbf{u}_h)=\mathbf{F}\left(\sum_{k\in\mathscr{K}}N_k\mathbf{u}_k(t)\right)\approx\sum_{k\in\mathscr{K}}N_k\mathbf{F}(\mathbf{u}_k(t))=\sum_{k\in\mathscr{K}}N_k\mathbf{F}_k$$

the discrete Galerkin formulation of the Euler equations can be rewritten as,

$$\begin{aligned}&\sum_{k\in\mathscr{K}_i}\left[\int_{\Omega_i}N_iN_k\,\mathrm{d}\Omega\right]\frac{\mathrm{d}\mathbf{u}_k(t)}{\mathrm{d}t}+\sum_{k\in\mathscr{K}_i^{\partial}}\left[\int_{\Gamma_{ik}}N_iN_k\mathbf{n}\,\mathrm{d}\sigma\right]\cdot\mathbf{F}_k^{\mathrm{b}}\\&\quad-\sum_{k\in\mathscr{K}_i}\left[\int_{\Omega_{ik}}N_k\boldsymbol{\nabla}_{\mathbf{x}}N_i\,\mathrm{d}\Omega\right]\cdot\mathbf{F}_k=0\end{aligned} \tag{10}$$

where $\Omega_{ik} = \Omega_i \cap \Omega_k$ and Γ_{ik} denotes $\partial\Omega_i \cap \partial\Omega_k \cap \partial\Omega$. This approximation, usually referred to as 'group approximation', makes the evaluation of the integrals much simpler and greatly improves the computational efficiency of the method since all the quantities appearing between square brackets do not depend on the solution and can therefore be evaluated and stored at the beginning of the computation. The saving is particularly relevant when the flux function evaluation is computationally expensive as in the case of fluids characterized by complex equations of state.

Following the ideas introduced by Selmin [26] and Selmin and Formaggia [27], and more recently extended in Reference [41], the summations appearing in the discrete Galerkin formulation (10) can be rearranged in a much more convenient form, i.e. as summation over pairs of 'interacting' nodes, called node-pairs. Two nodes i and k interact if the associated regions Ω_i and Ω_k have a non-empty intersection, i.e. if $\Omega_{ik} = \Omega_i \cap \Omega_k \neq 0$. The mass matrix M_{ik} and the metric vectors $\boldsymbol{\eta}_{ik}$, $\boldsymbol{\chi}^{\partial}_{ik}$ and $\boldsymbol{\xi}^{\partial}_{i}$ are defined as

$$M_{ik} = \int_{\Omega_i} N_i N_k \, \mathrm{d}\Omega$$

$$\boldsymbol{\eta}_{ik} = \int_{\Omega_{ik}} (N_i \boldsymbol{\nabla}_{\mathbf{x}} N_k - N_k \boldsymbol{\nabla}_{\mathbf{x}} N_i) \, \mathrm{d}\Omega, \quad \boldsymbol{\chi}^{\partial}_{ik} = \int_{\Gamma_{ik}} N_i N_k \mathbf{n} \, \mathrm{d}\sigma, \quad \boldsymbol{\xi}^{\partial}_{i} = \int_{\Gamma_i} N_i \mathbf{n} \, \mathrm{d}\sigma$$

therefore the Galerkin discrete equations (10) can be rewritten as,

$$\sum_{k \in \mathscr{K}_i} M_{ik} \frac{\mathrm{d}\mathbf{u}_k}{\mathrm{d}t} + \sum_{k \in \mathscr{K}_{i,\neq}} \boldsymbol{\eta}_{ik} \cdot \frac{\mathbf{F}_i + \mathbf{F}_k}{2} + \boldsymbol{\xi}^{\partial}_{i} \cdot \mathbf{F}^{\mathrm{b}}_{i} + \sum_{k \in \mathscr{K}^{\partial}_{i,\neq}} \boldsymbol{\chi}^{\partial}_{ik} \cdot \frac{\mathbf{F}^{\mathrm{b}}_{k} - \mathbf{F}^{\mathrm{b}}_{i}}{2} = 0 \tag{11}$$

For a complete discussion of the metric vectors and the derivation of (11) see Reference [27]. If time accuracy is not an issue (such as for example in steady state calculations), the consistent mass matrix M_{ik} which couples the time derivative terms can be 'lumped' to a diagonal matrix $D_i = \sum_{k \in \mathscr{K}_i} M_{ik}$, and the corresponding equations can therefore be written as,

$$D_i \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \sum_{k \in \mathscr{K}_{i,\neq}} \boldsymbol{\eta}_{ik} \cdot \frac{\mathbf{F}_i + \mathbf{F}_k}{2} + \boldsymbol{\xi}^{\partial}_{i} \cdot \mathbf{F}^{\mathrm{b}}_{i} + \sum_{k \in \mathscr{K}^{\partial}_{i,\neq}} \boldsymbol{\chi}^{\partial}_{ik} \cdot \frac{\mathbf{F}^{\mathrm{b}}_{k} - \mathbf{F}^{\mathrm{b}}_{i}}{2} = 0 \tag{12}$$

The metric vectors $\boldsymbol{\eta}_{ik}$, $\boldsymbol{\chi}^{\partial}_{ik}$ and $\boldsymbol{\xi}^{\partial}_{i}$ are characterized by the remarkable properties

$$\boldsymbol{\eta}_{ik} = -\boldsymbol{\eta}_{ki}, \quad \sum_{k \in \mathscr{K}_{i,\neq}} \boldsymbol{\eta}_{ik} + \boldsymbol{\xi}^{\partial}_{i} = \mathbf{0}, \quad \sum_{k \in \mathscr{K}^{\partial}_{i,\neq}} \boldsymbol{\chi}^{\partial}_{ik} = \boldsymbol{\xi}^{\partial}_{i}$$

which are satisfied for any node i or k (notice that $\boldsymbol{\chi}^{\partial}_{ik} = \mathbf{0}$ and $\boldsymbol{\xi}^{\partial}_{i} = \mathbf{0}$ if i or k are internal nodes). These properties guarantee that the metric vectors behave similar to the integrated normal vectors $\mathbf{v}_{ik}$ previously defined in the context of finite volume methods and that, apart from the second-order boundary terms corresponding to the last summations appearing in Equation (11) or (12), the FE discretization (11) can be interpreted as a node-centred FV discretization in which the cell shapes are defined so that,

$$\mathscr{C}_i = \sum_{k \in \mathscr{K}_i} M_{ik} = D_i, \quad \mathbf{v}_{ik} = \boldsymbol{\eta}_{ik}, \quad \mathbf{v}_i = \boldsymbol{\xi}^{\partial}_{i} \tag{13}$$

To avoid the stability problems encountered with centred flux formulations, the simple average $(\mathbf{F}_i + \mathbf{F}_k)/2$ appearing in (12) must be replaced with a normal numerical flux function, thus arriving at the so-called FE/FV formulation

$$D_i \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \sum_{k \in \mathscr{K}_{i,\not\partial}} \hat{\mathbf{f}}(\mathbf{u}_i, \mathbf{u}_k, \boldsymbol{\eta}_{ik}) + \boldsymbol{\xi}_i^{\partial} \cdot \mathbf{F}_i^{\mathrm{b}} + \sum_{k \in \mathscr{K}_{i,\not\partial}^{\partial}} \boldsymbol{\chi}_{ik}^{\partial} \cdot \frac{\mathbf{F}_k^{\mathrm{b}} - \mathbf{F}_i^{\mathrm{b}}}{2} = 0 \tag{14}$$

The performance of both the FV scheme (7) and FE/FV scheme (14) is crucially dependent on the choice of the numerical flux function. To achieve second-order accuracy and avoid the generation of spurious oscillations in the neighbourhood of the shocks, a total variation diminishing (TVD) numerical flux formula is used in *zFlow*. A complete discussion of the class of upwind TVD numerical flux functions for the Euler equations and fluids characterized by arbitrary equations of state is beyond the scope of this paper (see e.g. Reference [42]), but it is however important to observe that any second-order accurate TVD numerical flux formula depends not only on the interface values $\mathbf{u}_i$ and $\mathbf{u}_k$ previously considered but also on two additional values $\mathbf{u}_m$ and $\mathbf{u}_n$. In the one-dimensional case, the additional values correspond to unknown values at the nodes $m = i - 1$ and $n = k + 1$, i.e. m, i, k, n denote four consecutive aligned points.

The extension of high-resolution schemes to the multidimensional case is therefore not immediate since, for general unstructured grids, the points are not aligned as in the one-dimensional case (or as in the case of multidimensional structured grids).

In *zFlow*, following the simple but very effective technique originally proposed in Reference [43], $\mathbf{u}_m$ and $\mathbf{u}_n$ are chosen as the values pertaining to the 'better aligned' nodes with respect to the triangulation edge associated to the node-pair ik as shown in Figure 3. Notice that the 'extended node pair' m, i, k, n can be predetermined on a purely geometrical basis at the beginning of the computation, thus improving the computational efficiency of the methods.

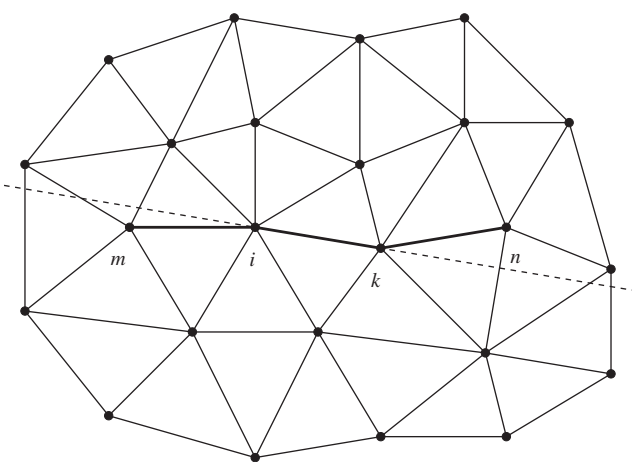

Figure 3. Simple 2D triangulations showing the nodes m, i, j, n of an extended node-pair (thick line).

In conclusion the discrete Galerkin formulation of the Euler equation with a high-resolution TVD numerical flux exploiting the extended node pair concept can be written as,

$$D_i \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}t} + \sum_{k \in \mathscr{K}_{i,\neq}} \hat{\mathbf{f}}(\mathbf{u}_m, \mathbf{u}_i, \mathbf{u}_k, \mathbf{u}_n, \boldsymbol{\eta}_{ik}) + \boldsymbol{\xi}_i^{\partial} \cdot \mathbf{F}_i^{\mathrm{b}} + \sum_{k \in \mathscr{K}_{i,\neq}^{\partial}} \boldsymbol{\chi}_{ik}^{\partial} \cdot \frac{\mathbf{F}_k^{\mathrm{b}} - \mathbf{F}_i^{\mathrm{b}}}{2} = 0 \tag{15}$$

zFlow employs the total variation diminishing (TVD) flux formula introduced by Davis [44] extended to the case of real gases following the Vinokur–Montagné approach [31] and using the Van Albada [45] 'smooth' limiter function together with the Harten Entropy fix [46].

The flux can be written as,

$$\hat{\mathbf{f}} = \frac{1}{2}[\mathbf{F}(\mathbf{u}_i) + \mathbf{F}(\mathbf{u}_k)] \cdot \boldsymbol{\eta}_{ik} - \frac{1}{2} \mathscr{R}_{\boldsymbol{\eta}_{ik}}^{(R)} \langle \boldsymbol{\Lambda}_{\boldsymbol{\eta}_{ik}}^{(R)} \rangle \boldsymbol{\Psi}(\delta \mathbf{w}_{ik}, \delta \mathbf{w}_{\bigstar})$$

where $\mathscr{R}_{\boldsymbol{\eta}_{ik}}^{(R)} = \mathscr{R}_{\boldsymbol{\eta}_{ik}}(\mathbf{u}^{(R)})$ and $\boldsymbol{\Lambda}_{\boldsymbol{\eta}_{ik}}^{(R)} = \boldsymbol{\Lambda}_{\boldsymbol{\eta}_{ik}}(\mathbf{u}^{(R)})$ denote the right eigenvector and the (diagonal) eigenvalue matrices of the Jacobian matrix in the direction $\boldsymbol{\eta}_{ik}$ evaluated at the Vinokur–Montagné generalized Roe average state $\mathbf{u}^{(R)}(\mathbf{u}_i, \mathbf{u}_k)$. The 'wavestrength' vector $\delta \mathbf{w}_{ik}$ for a generic node pair ik is defined as

$$\delta \mathbf{w}_{ik} = \mathscr{L}_{\boldsymbol{\eta}_{ik}}^{(R)}(\mathbf{u}_k - \mathbf{u}_i), \qquad \mathscr{L}_{\boldsymbol{\eta}_{ik}}^{(R)} = (\mathscr{R}_{\boldsymbol{\eta}_{ik}}^{(R)})^{-1}$$

while $\delta \mathbf{w}_{\bigstar}$ denotes a vector of upwind biased wavestrengths given by

$$\delta \mathbf{w}_{\bigstar}^{(\ell)} = \begin{cases} \delta \mathbf{w}_{mi}^{(\ell)} & \text{if } \lambda^{(\ell)} > 0 \\ \delta \mathbf{w}_{kn}^{(\ell)} & \text{if } \lambda^{(\ell)} < 0 \end{cases}$$

The expression $\langle \boldsymbol{\Lambda}_{\boldsymbol{\eta}_{ik}}^{(R)} \rangle$ denotes a 'smoothed' absolute value and is defined as

$$\langle \boldsymbol{\Lambda}_{\beta_{ik}}^{(R)} \rangle = \mathrm{diag}\,\{\langle \lambda_k \rangle\}, \quad \langle \lambda_k \rangle = \begin{cases} |\lambda_k| & \text{if } |\lambda_k| > h, \\ (\lambda_k^2 + h^2)/(2h) & \text{if } |\lambda_k| \leqslant h, \end{cases} \qquad h = \varepsilon(M+1)$$

where $M = |\mathbf{v}|/c$ is the Mach number and ε is a free parameter. In the results hereby presented the value $\varepsilon = 1/10$ has always been used. The function $\boldsymbol{\Psi} \in \mathbb{R}^{d+2}$ is a vector function whose ℓth component is given by a scalar function $\Psi(a,b) \in \mathbb{R}$, where $a \in \mathbb{R}$ and $b \in \mathbb{R}$, which operates on the ℓth components $\delta \mathbf{w}_{ik}^{(\ell)}$ and $\delta \mathbf{w}_{\bigstar}^{(\ell)}$ of the wavestrength vectors, i.e.

$$\boldsymbol{\Psi}^{(\ell)}(\delta \mathbf{w}_{ik}, \delta \mathbf{w}_{\bigstar}) = \Psi(\delta \mathbf{w}_{ik}^{(\ell)}, \delta \mathbf{w}_{\bigstar}^{(\ell)})$$

The scalar function $\Psi(a,b)$ can be written as

$$\Psi(a,b) = a\left[1 - \Theta\left(\frac{b}{a}\right)\right]$$

in which the function $\Theta(b/a)$ can be selected as any of the 'flux limiter' functions available in the literature. In this work the Van Albada limiter has been used, so that

$$\Theta(r) = \frac{r^2 + r}{r^2 + 1}, \quad r = \frac{b}{a}, \quad \Psi(a,b) = a - \frac{ab(a+b)}{a^2 + b^2}$$

2.3. Boundary treatment for general fluids

The boundary condition treatment for the Euler equations is rather involved since the quantities that must be prescribed are in general different from the vector of the conservative variables $\mathbf{u}$ or from some of its components. The quantities known at the boundary are instead non-linear functions of the conservative variables, say $\psi^k(\mathbf{u})$, $k = 1, \ldots, p$ where $p \leqslant d + 2$, such as for example the pressure $P = \Pi(\mathbf{u})$ or the flow angle $\alpha = \alpha(\mathbf{u})$. Both the number p of quantities to be prescribed and the quantities themselves, i.e. the actual expression of the functions $\psi(\mathbf{u})$, depend on the local flow conditions (for more details see e.g. Reference [42]).

The number of conditions and the quantities that can be prescribed can be determined by considering the projection of the Euler equations in the direction normal to the boundary, which reveals that the solution can be regarded as the combination of $d + 2$ inward and outward travelling waves of non-linear functions $\Gamma^k(\mathbf{u})$ called 'Riemann invariants', $k = 1, \ldots, d + 2$. The propagation speed of the functions $\Gamma^k(\mathbf{u})$ is given by the Eigenvalues $\lambda_n^k(\mathbf{u})$ of the Jacobian matrix $\mathscr{A}_n(\mathbf{u})$. The solution at the boundary is therefore a combination of information travelling inside-out from the domain—the Riemann invariants associated to characteristic lines with space-time slope $\lambda_n^k > 0$ if $\mathbf{n}$ is the outward unit normal vector—and information travelling outside-in the domain, and it is only the latter information that must be prescribed. As a consequence, the number of quantities that must be prescribed is equal to the number of negative eigenvalues $\lambda_n^k < 0$.

The boundary state $\mathbf{u}^{\mathrm{b}}$ is therefore computed by combining the p prescribed data, where p is the number of negative eigenvalues, with the $d + 2 - p$ Riemann invariant associated to the positive eigenvalues, i.e. as the solution of the system

$$\begin{cases} \psi^k(\mathbf{u}^{\mathrm{b}}) = \psi_{\mathrm{b}}^k & k = 1, \ldots, p \\ \Gamma^k(\mathbf{u}^{\mathrm{b}}) = \Gamma^k(\mathbf{u}) & k = p+1, \ldots, d+2 \end{cases} \tag{16}$$

where ψ_{b}^k denote the prescribed value of the quantity ψ^k, and $\mathbf{u}$ is the solution at the boundary (which is different from $\mathbf{u}^{\mathrm{b}}$ because of the weak enforcement of the boundary conditions). Notice that, in order for the system to be solvable, the functions $\psi^k(\mathbf{u})$ must not be a combination of the outward-travelling Riemann invariant.

In order to work with simpler expressions (and, in particular, to avoid the analytical computation of the Riemann invariants in finite form which may prove a very involved and lengthy matter for a fluid with a complex EOS) the linearized version of system (16) for the primitive variables $\mathbf{v} = [\rho, P, \mathbf{q}]$ is considered. For simplicity the same notation $\Gamma^k(\mathbf{v})$ and $\psi^k(\mathbf{v})$ is used to indicate the Riemann invariants and the prescribed data even if it obviously denotes func-

tions which are different from $\Gamma(\mathbf{u})$ and $\psi^k(\mathbf{u})$. The functions $\psi^k(\mathbf{v}^b)$ and $\Gamma^k(\mathbf{v}^b)$ are therefore approximated as,

$$\psi^k(\mathbf{v}^b) \approx \left(\frac{\partial \psi^k}{\partial \mathbf{v}}\right)_{\mathbf{v}} (\mathbf{v}^b - \mathbf{v}) + \psi^k(\mathbf{v}), \quad \Gamma^k(\mathbf{v}^b) \approx \left(\frac{\partial \Gamma^k}{\partial \mathbf{v}}\right)_{\mathbf{v}} (\mathbf{v}^b - \mathbf{v}) + \Gamma^k(\mathbf{v})$$

so that the system to be solved in order to evaluate the boundary state can be written as,

$$\begin{cases} \left(\dfrac{\partial \psi^k}{\partial \mathbf{v}}\right)_{\mathbf{v}} \delta\mathbf{v} = \psi_b^k - \psi^k(\mathbf{v}) \\ \left(\dfrac{\partial \Gamma^k}{\partial \mathbf{v}}\right)_{\mathbf{v}} \delta\mathbf{v} = 0 \end{cases}$$

where the unknown is $\delta\mathbf{v} = \mathbf{v}^b - \mathbf{v}$. The analytical solution of the above system is now presented for an inflow and an outflow boundary. The expression of the linearized Riemann invariants in the 1D, 2D and 3D Euler equations is given in Appendix A.

2.3.1. Inflow boundary. An inflow boundary is first considered. It is defined as the subset of $\partial\Omega$ where the flow enters into the domain Ω, i.e. such that $q_n = \mathbf{q}\cdot\mathbf{n} < 0$ ($\mathbf{n}$ points out from Ω).

In the subsonic case, where $M_n = |q_n|/c < 1$, there is only one positive eigenvalue $(q_n + c)$ and consequently $d + 2 - 1$ quantities are to be prescribed. These are the thermodynamic stagnation state (here given in terms of the entropy s and the stagnation enthalpy h^t) and the flow direction. These are in fact the quantities typically known or measured at the inflow of a duct or, for what concerns the stagnation thermodynamic state, at an upstream reservoir. The missing quantity to completely define the boundary state is the Riemann invariant associated to the eigenvalue $q_n + c$ evaluated from inside Ω. The equations for the computation of $\mathbf{v}^b$ are therefore

$$\begin{aligned} \delta h^t &= \delta h^t_{\alpha} \\ \delta s &= \delta s_{\alpha} \\ \delta P + \rho c \delta q_n &= 0 \end{aligned} \tag{17}$$

Notice that the direction of the boundary velocity vector $\mathbf{q}^b$ is a prescribed datum and that only its magnitude q^b is to be determined. It is therefore convenient to rewrite the third equation of the previous system in terms of the unknown δq. If $\boldsymbol{\beta}$ denotes the prescribed unit normal vector in the direction of $\mathbf{q}^b$, this implies that $\mathbf{q}^b = \boldsymbol{\beta} q^b$. A simple calculation reveals that

$$\delta q_n = \beta_n \delta q - \delta\xi, \qquad \delta\xi = q_n - \beta_n q$$

where $\beta_n = \boldsymbol{\beta}\cdot\mathbf{n}$. System (17) can therefore be rewritten as

$$\begin{aligned} \delta h^t &= \delta h^t_{\alpha} \\ \delta s &= \delta s_{\alpha} \\ \delta P + \rho c \beta_n \delta q &= \rho c \, \delta\xi \end{aligned} \tag{18}$$

which is a system of three algebraic equations in the four unknown δh^t, δs, δP and δq ($\delta h^t_{\boldsymbol{\alpha}}$ and $\delta s_{\boldsymbol{\alpha}}$ are the boundary data). The missing relation is provided by the (linearized) Gibbs–Duhem thermodynamic relation

$$\delta h = T\delta s + \frac{\delta P}{\rho} + \sum_{i=1}^{n_c} \left(\frac{\partial h}{\partial n_i}\right)_{P,s,n_{j\neq i}} \delta n_i \tag{19}$$

In (19), n_i is the mole fraction of component i and n_c is the number of components in the mixture. Imposing that there is no phase change implies that the composition does not change, therefore the last term in (19) is zero and it reduces to

$$\delta h = T\delta s + \frac{\delta P}{\rho} \tag{20}$$

Equation (20) can be used to express δP as a function of δh^t and δs, as shown in the following. Summing the term δq^2 on the left- and right-hand sides of the previous equation gives

$$\begin{aligned}
\delta h^t &= T\delta s + \frac{\delta P}{\rho} + \frac{1}{2}\,\delta q^2 \\
&= T\delta s + \frac{\delta P}{\rho} + \mu q \delta q \\
&= T\delta s + \frac{\delta P}{\rho} + q\delta q + \frac{1}{2}\,(\delta q)^2
\end{aligned}$$

where $\mu(\,\cdot\,)$ is the average operator $[(\,\cdot\,)^{\mathrm{b}} + (\,\cdot\,)]/2$. The pressure difference is therefore given by

$$\delta P = \rho\left(\delta h^t - T\delta s - q\delta q - \frac{1}{2}(\delta q)^2\right)$$

Inserting the above expression in the last equation (18) yields,

$$\rho\left(\delta h^t_{\boldsymbol{\alpha}} - T\delta s_{\boldsymbol{\alpha}} + (c\beta_n - q)\delta q - \frac{1}{2}(\delta q)^2 - c\delta\xi\right) = 0$$

which can be rewritten as the second-order algebraic equation

$$\frac{1}{2}(\delta q)^2 - (c\beta_n - q)\delta q + (c\delta\xi + T\delta s_{\boldsymbol{\alpha}} - \delta h^t_{\boldsymbol{\alpha}}) = 0$$

having roots $(\delta q)^{\pm}$ given by

$$(\delta q)^{\pm} = (c\beta_n - q) \pm \sqrt{(c\beta_n - q)^2 - 2(c\delta\xi + T\delta s_{\boldsymbol{\alpha}} - \delta h^t_{\boldsymbol{\alpha}})}$$

The physical relevant solution is that which yields $\delta q = 0$ when the prescribed data are equal to the internal state $\mathbf{u}$. In this case $\delta h^t_{\boldsymbol{\alpha}} = 0$, $\delta s_{\boldsymbol{\alpha}} = 0$, and $\delta\xi = 0$, and the two roots can be written as

$$(\delta q)^{\pm} = (c\beta_n - q) \pm |c\beta_n - q|$$

Applying the conditions of inflow boundary and subsonic flow in the normal direction gives $\beta_n < 0$ and $c\beta_n - q < 0$. In order to get $\delta q = 0$, $(\delta q)^+$ must be selected in the above expression. In conclusion the velocity difference is given by

$$(\delta q) = (c\beta_n - q) + \sqrt{(c\beta_n - q)^2 - 2(c\delta\xi + T\delta s_\alpha - \delta h_\alpha^t)}$$

The calculation of the boundary state for an inflow boundary which is supersonic in the normal direction is trivial since all the eigenvalues are negative and $d + 2$ boundary data must be prescribed (the entire vector $\mathbf{v}^b$ is the prescribed data in this case).

2.3.2. Outflow boundary. An outflow boundary is a subset of $\partial\Omega$ such that $\mathbf{q} \cdot \mathbf{n} > 0$. Also in this case the number of conditions to be prescribed depends on the flow being subsonic or supersonic in the normal direction. If the flow in the normal direction is subsonic there is only one eigenvalue $\lambda = q_n - c < 0$, while all the eigenvalues are positive in the supersonic case.

The subsonic case is analysed first. The equations which allow for the computation of the boundary state are

$$\begin{aligned} \delta P + \rho c\, \delta q_n &= 0 \\ c^2 \delta\rho + \delta P &= 0 \\ \delta P &= \delta P_\alpha \end{aligned} \tag{21}$$

where in the three-dimensional case, the second equation can be obtained as a linear combination of the second, third and fourth characteristic equations. The solution of the system is in this case almost trivial, namely

$$\begin{aligned} \delta\rho &= -\frac{\delta P_\alpha}{c^2} \\ \delta\mathbf{q} &= \frac{\delta P_\alpha}{\rho c^2}\mathbf{n} \\ \delta P &= \delta P_\alpha \end{aligned} \tag{22}$$

In the supersonic case no condition is to be imposed and the boundary state is simply equal to the internal state, i.e. $\mathbf{v}^b = \mathbf{v}$.

2.4. Time discretization

The semi-discretized FE formulation (15) is a system of coupled non-linear ordinary differential equations (ODE) which can be written as,

$$M\frac{\mathrm{d}U}{\mathrm{d}t} + R(U) = 0 \quad \text{or} \quad D\frac{\mathrm{d}U}{\mathrm{d}t} + R(U) = 0$$

where M and D denote the consistent and lumped (or diagonalized) mass matrices, respectively, U is the global vector of the unknown solution and $R(U)$ the global residual vector of components R_i given by

$$R_i = \sum_{k \in \mathscr{K}_{i,\neq}} \hat{\mathbf{f}}(\mathbf{u}_m, \mathbf{u}_i, \mathbf{u}_k, \mathbf{u}_n, \boldsymbol{\eta}_{ik}) + \boldsymbol{\xi}_i^{\partial} \cdot \mathbf{F}_i^{\mathrm{b}} + \sum_{k \in \mathscr{K}_{i,\neq}^{\partial}} \boldsymbol{\chi}_{ik}^{\partial} \cdot \frac{\mathbf{F}_k^{\mathrm{b}} - \mathbf{F}_i^{\mathrm{b}}}{2}$$

The equations can be integrated in time with both explicit or implicit time marching schemes. A step of an explicit time integration method is usually much cheaper both in terms of CPU time and of memory usage than a step of an implicit backward Euler scheme. However, the time step Δt which can be used in an explicit scheme is limited by the CFL condition

$$\Delta t \leqslant \min_{i \in \mathscr{K}} \mu \frac{\Delta x_i}{\lambda_i^{\max}}$$

where $\lambda_i^{\max}$ denotes the maximum eigenvalue of the Jacobian matrix of the Euler equations evaluated at the node i, i.e. $\lambda_i^{\max} = |\mathbf{v}_i| + c_i$, and Δx_i is a characteristic mesh size at node i computed as

$$\Delta x_i = \min_{k \in \mathscr{K}_{i,\neq}} |\mathbf{x}_i - \mathbf{x}_k|$$

The value of the CFL number μ depends on the type of time integration scheme considered but is always $O(1)$ in the explicit case. Conversely, an implicit scheme such as Backward Euler does not have any limit on the allowable Δt (at least in the linear case). This means that an implicit scheme may be very convenient in the computation of steady state solutions or of unsteady solution where the CFL condition dictates a time step Δt which is much smaller than that which would be used because of accuracy considerations alone. Moreover, an implicit time integration scheme may be convenient when the computational resources required to solve the system are of the same order of magnitude of those required to compute its right-hand side alone, as may be the case for the solution of the Euler equations for fluids governed by complex equations of state.

In *zFlow* both the explicit total variation stable Runge–Kutta scheme suggested by Shu [47] and the implicit backward Euler scheme have been implemented and, as expected, the backward Euler scheme proved to be much more efficient than the explicit scheme for steady state computations, especially for fluids characterized by complex EOS (see Sections 4 and 6).

A single time step of the Runge–Kutta scheme can be written as,

$$\begin{aligned} U_0 &= U^n \\ U_k &= (1-\psi)U_0 + \psi(U_{k-1} - \Delta t D^{-1} R(U_{k-1}))^n, \quad k = 1, \ldots, m \\ U^{n+1} &= U_m \end{aligned}$$

where n denotes the time level and m the number of Runge–Kutta stages. The values of the coefficients ψ, $k = 1, \ldots, m$ can be found for example in Reference [47].

A step of the implicit backward Euler scheme can instead be written as,

$$\left[\frac{D}{\Delta t} + \frac{\partial R(U)}{\partial U}\right]^n \Delta U^n = A(u)^n \Delta U^n = -R(U)^n$$

where ΔU^n denotes the solution increment $U^{n+1}-U^n$. For the sake of computational efficiency, however, the linearization of the residual appearing in the expression of the coefficient matrix A is evaluated in an approximate manner by using the first-order Vinokur–Montagné-generalized Roe scheme

$$\hat{\mathbf{f}}(\mathbf{u}_i,\mathbf{u}_k,\boldsymbol{\eta}_{ik})=\frac{1}{2}[\mathbf{F}(\mathbf{u}_i)+\mathbf{F}(\mathbf{u}_k)]\cdot\boldsymbol{\eta}_{ik}-\frac{1}{2}\langle\mathscr{A}^{(R)}_{\boldsymbol{\eta}_{ik}}(\mathbf{u}_i,\mathbf{u}_k)\rangle(\mathbf{u}_k-\mathbf{u}_i)$$

This means that, for a generic internal or boundary node i, the linearized increment of the ith component of the residual vector $R(U)$ is evaluated as

$$\begin{aligned}\frac{\partial R_i(U)}{\partial U}\Delta U^n \approx{}& \frac{1}{2}\sum_{k\in\mathscr{K}_{i,\neq}}[(\mathscr{A}^{(i)}_{\boldsymbol{\eta}_{ik}}+\langle\mathscr{A}^{(R)}_{\boldsymbol{\eta}_{ik}}\rangle)\Delta\mathbf{u}_i+(\mathscr{A}^{(k)}_{\boldsymbol{\eta}_{ik}}-\langle\mathscr{A}^{(R)}_{\boldsymbol{\eta}_{ik}}\rangle)\Delta\mathbf{u}_k]\\ &+\mathscr{A}^{(i)}_{\boldsymbol{\xi}^{\partial}_i}\mathscr{B}^{(i)}\Delta\mathbf{u}_i+\frac{1}{2}\sum_{k\in\mathscr{K}^{\partial}_{i,\neq}}[\mathscr{A}^{(i)}_{\boldsymbol{\chi}^{\partial}_{ik}}\mathscr{B}^{(i)}\Delta\mathbf{u}_i-\mathscr{A}^{(k)}_{\boldsymbol{\chi}^{\partial}_{ik}}\mathscr{B}^{(k)}\Delta\mathbf{u}_k]\end{aligned}\tag{23}$$

where the matrix $\mathscr{B}$ is an approximation of the Jacobian of the relation which gives the boundary state $\mathbf{u}^{\mathrm{b}}$ as a function of the solution at boundary $\mathbf{u}$, so that

$$\Delta\mathbf{u}^{\mathrm{b}}_i=\mathscr{B}^{(i)}\Delta\mathbf{u}_i$$

The matrix $\mathscr{B}^{(i)}$ is computed as follows. The time linearization of system (16) gives

$$\begin{cases}\left(\dfrac{\partial\psi^k}{\partial\mathbf{u}}\right)_{\mathbf{u}^{\mathrm{b}}}\Delta\mathbf{u}^{\mathrm{b}}_i=0 & k=1,\ldots,p\\ \left(\dfrac{\partial\Gamma^k}{\partial\mathbf{u}}\right)_{\mathbf{u}^{\mathrm{b}}}\Delta\mathbf{u}^{\mathrm{b}}_i=\left(\dfrac{\partial\Gamma^k}{\partial\mathbf{u}}\right)_{\mathbf{u}}\Delta\mathbf{u}_i & k=p+1,\ldots,d+2\end{cases}\tag{24}$$

where it is assumed that the boundary conditions are time independent and that system (16) is satisfied at time level n. By denoting with $\mathscr{G}^{\mathrm{b}}_i$ the matrix having the first p rows equal to $(\partial\psi^k/\partial\mathbf{u})_{\mathbf{u}^{\mathrm{b}}}$ and the last $d+2-p$ rows equal to $(\partial\Gamma^k/\partial\mathbf{u})_{\mathbf{u}^{\mathrm{b}}}$, and with $\mathscr{G}_i$ the matrix having the first p rows equal to zero and the remaining $d+2-p$ rows equal to $(\partial\Gamma^k/\partial\mathbf{u})_{\mathbf{u}}$, system (24) can be written in the compact form as

$$\mathscr{G}^{\mathrm{b}}_i\Delta\mathbf{u}^{\mathrm{b}}_i=\mathscr{G}_i\Delta\mathbf{u}_i$$

which shows that

$$\mathscr{B}_i=(\mathscr{G}^{\mathrm{b}}_i)^{-1}\mathscr{G}_i$$

The solution of the linear systems occurring at each iteration is performed by means of the incomplete LU factorization preconditioned generalized minimum residual (GMRES) method implemented in the SLATEC common mathematical library available on http://www.netlib.org.

Very large *CFL* numbers can be used with the backward Euler time integration scheme. However, in order to avoid the computation blowup, relatively small values of the *CFL* number are required in the initial steps of the computation. In order to fully exploit the advantages offered by an implicit time integration scheme in the computation of steady state

solution the CFL number must therefore be increased as the solution converges. This can be obtained in an automatic fashion by computing the value of the CFL number CFL_i at the ith iteration as,

$$CFL_i = \min\left\{\max\left[\frac{CFL_{\min}}{(e_i/e_{\max})^K},\ CFL_{\min}\right] CFL_{\max}\right\} \tag{25}$$

where e_i denotes the L_2 norm of the residual at the ith iteration, $e_{\max}$ the maximum value of the residual L_2 norm occurred in all the iterations preceding the current one, i.e. $e_{\max} = \max_{1 \leqslant i \leqslant i-1} e_i$, and $CFL_{\min}$, $CFL_{\max}$ and K are user defined parameters which control the evolution of the CFL value during the computation. The parameters $CFL_{\min}$ and $CFL_{\max}$ are used to impose the minimum and maximum possible CFL number, while the parameter K controls how rapidly the CFL number increases as the solution converges. Typical values of K range from 0.5 to 2.

3. UNCONVENTIONAL THERMODYNAMIC PROPERTIES FOR CFD

Numerical schemes for the solution of the Euler equations require the calculation of the speed of sound and of some other partial derivatives of standard thermodynamic functions (see Appendix A). The convective speed of sound is defined as

$$c^2 = \left(\frac{\partial P}{\partial \rho}\right)_s = -v^2 \left(\frac{\partial P}{\partial v}\right)_s \tag{26}$$

The pressure partial derivatives mentioned, defined as

$$\alpha = \left(\frac{\partial P}{\partial \rho}\right)_e, \quad \beta = \left(\frac{\partial P}{\partial e}\right)_\rho \tag{27}$$

are necessary for the computation of the Jacobian matrices, while the entropy partial derivatives

$$\chi = \left(\frac{\partial s}{\partial \rho}\right)_e \quad \text{and} \quad \phi = \left(\frac{\partial s}{\partial e}\right)_\rho \tag{28}$$

are necessary for the calculation of the inflow boundary if an implicit numerical scheme is employed. If the code is used to investigate non-classical effects, the Γ function, defined as

$$\Gamma = 1 + \frac{\rho}{c}\left(\frac{\partial c}{\partial \rho}\right)_s = \frac{v^3}{2c^2}\left(\frac{\partial^2 P}{\partial v^2}\right)_s \tag{29}$$

must also be evaluated. If the EOS is in the common form $P = P(v, T)$, where v is the specific volume and T the temperature, the above expressions can be conveniently expressed as:

$$c = \sqrt{-v^2 \gamma \left(\frac{\partial P}{\partial v}\right)_T} \tag{30}$$

$$\beta = \frac{1}{C_v}\left(\frac{\partial P}{\partial T}\right)_v \tag{31}$$

$$\alpha = c^2 - \beta P v^2 \tag{32}$$

$$\chi = -v^2\frac{P}{T} \tag{33}$$

$$\phi = \frac{1}{T} \tag{34}$$

$$\Gamma = \frac{v^3}{2c^2}\left\{\left(\frac{\partial^2 P}{\partial v^2}\right)_T - 3\,\frac{T}{C_v}\left(\frac{\partial P}{\partial T}\right)_v\left(\frac{\partial^2 P}{\partial v \partial T}\right)_{v,T} + \left[\frac{T}{C_v}\left(\frac{\partial P}{\partial T}\right)_v\right]^2\left\{3\left(\frac{\partial^2 P}{\partial T^2}\right)_v + \frac{1}{T}\left(\frac{\partial P}{\partial T}\right)_v\left[1 - \frac{T}{C_v}\left(\frac{\partial C_v}{\partial T}\right)_v\right]\right\}\right\} \tag{35}$$

It can be noted that the calculation of the functions (30)–(35) involves only the following terms:

$$\left(\frac{\partial P}{\partial v}\right)_T, \quad \left(\frac{\partial P}{\partial T}\right)_v, \quad \left(\frac{\partial^2 P}{\partial v^2}\right)_T, \quad \left(\frac{\partial^2 P}{\partial T^2}\right)_v, \quad \left(\frac{\partial^2 P}{\partial T \partial v}\right)_{v,T}, \quad \left(\frac{\partial C_v}{\partial T}\right)_v$$

and the integral

$$\int_\infty^v \left(\frac{\partial^2 P}{\partial T^2}\right)_v \mathrm{d}v$$

If complex EOS or EOS for mixtures are adopted, their estimation is not trivial: the problem is extensively addressed in Reference [36].

Presently, the estimation of thermodynamic properties for CFD calculations can only be implemented by means of EOS; statistical mechanics methods are computationally too demanding and require in many cases molecular data which are unavailable. To avoid numerical problems, it is important that all thermodynamic properties are derived from a consistent thermodynamic model. A thermodynamic model is consistent if all the properties are derived from the minimum set of information: an equation of state in the form $P = P(v, T)$ and an expression for the specific heat at constant pressure (or specific volume) in the ideal gas state (C_p^0 or C_v^0). All properties can be derived using exact thermodynamic transformations (see e.g. Reference [37]). Modern thermodynamic models are often based on the so-called fundamental relations: $e = e(v, s)$, $g = g(T, P)$, $a = a(\rho, T)$, $h = h(P, s)$, where e is the internal energy, g is the Gibbs energy and a is the Helmholtz energy, h is the enthalpy (see e.g. Reference [48]). An expression for $C_P^0 = C_P^0(T)$ is however necessary. Models based on pressure explicit EOS or on fundamental relations are equivalent from the perspective of thermodynamic consistency. Accuracy, especially in the close to critical region, is strongly dependent on the complexity of the equation of state (see e.g. Reference [49]), which in turn requires a greater effort if properties to be implemented in the code are derived analytically. If mixtures are taken into consideration, mixing rules which are thermodynamically correct currently exist only for EOS which are cubic in the specific volume (see e.g. References [50, 51]).

As shown in Reference [36] the computational time required to evaluate derived thermodynamic functions containing partial derivatives by applying an adaptive numerical algorithm is far greater if compared to the evaluation of their analytical expressions. To obtain analytical expression for various thermodynamic models as those included in *zFlow*, a software for symbolical manipulation of algebraic expressions and analytical derivation is very helpful [52]. Details of the derivation and expressions, of the thermodynamic functions cited, for various thermodynamic models can be found in Reference [36].

A last remark is devoted to the calculation of thermodynamic properties in the critical region: EOS based on the classical mean-field theory fails to predict the correct trends of thermodynamic properties close to the critical point (see e.g. Reference [53]). The peculiar behaviour of substances in the critical region strongly affects both volumetric and calorimetric properties, thus if CFD simulations involve the critical region, the so-called crossover functions should be integrated into the thermodynamic model [54–56]. Crossover functions for cubic equations of state are available [57, 58] and were recently extended to mixtures [59].

4. VALIDATION: RECOVERY OF CLASSICAL RESULTS

To the knowledge of the authors, reference simulations of fluid flows in the so-called real gas thermodynamic region are not available. As a validation of the numerical scheme implemented in *zFlow*, results from a two-dimensional calculation using nitrogen as the fluid are compared to a well known and thoroughly documented reference simulation. The test case is a transonic air flow around a RAE2822 airfoil [60]. Far field conditions are $M = 0.75$, angle of attack $\alpha = 3°$, $P = 1$ bar and $T = 25°\mathrm{C}$. As it can be noted the thermodynamic state is in a region in which no real gas effects are to be expected and the ideal gas model was used in Reference [60].

The mesh employed in the test case documented in Reference [60] is a Type-O grid with 320×64 nodes. The *zFlow* calculation was performed over 192 nodes along the airfoil and 49 along the outward direction.

Due to the fact that the *zFlow* calculation employs the PRSV equation of state, which of course tends asymptotically to the ideal gas behaviour at low temperature and pressure, the recovery of this classical results is intended as a check of the correct implementation of the numerical scheme and its link to the thermodynamic model. Plate 1(a) shows the iso-Mach lines in the flow field around the RAE2822 airfoil as calculated by *zFlow* using a real gas thermodynamic model. As a quantitative comparison, in Plate 1(b) the pressure coefficient C_p as obtained by the *zFlow* calculation is displayed together with the C_p reported in Reference [60]. As it can be noted the *zFlow* calculation compares very well with the reference data.

5. COMPUTATIONAL EFFICIENCY

In order to show the effectiveness offered by the implicit backward Euler time integration scheme in relation with the computation of steady state flows of fluids modelled by complex EOS, the CPU time was evaluated in some test cases. The benchmark is the calculation of the flow around the RAE2822 airfoil described in the preceding section.

Table I. CPU time required for the computation of the residual, linear system entries, solution of the linear system, and thermodynamics computations.

Thrm. model	Residual	System eval.	System sol.	EOS eval.
Ideal	0.18	0.22	2.70	0.01
PRSV	0.18	0.22	2.70	11.06

The figures indicates the measured timings in seconds for an AMD Athlon MP 1500 processor.

Table I shows the CPU time required for (1) the evaluation of the residual vector (required by the explicit Runge–Kutta time stepping method), (2) the evaluation and (3) the solution of the system of linear algebraic equations (required by the implicit backward Euler scheme), and (4) the EOS evaluations to be performed at each iteration and for all the nodes of the grid. The timings are given for both the ideal gas and the PRSV equation of state. Notice that, in the latter case, most of the CPU time is spent in the evaluation of the equation of state and that the time required to (approximately) solve the system of linear algebraic equations is only about one fourth of that required by the EOS evaluations.

The very large computational resources required for the EOS evaluation are largely due to the saturation state computations required to check whether the thermodynamic point falls in the single phase or in the vapour–liquid mixture region. This very time-consuming computation is however unavoidable if meaningful quantities are to be computed, especially when the flow thermodynamic states are in the neighbourhood of the saturation curve. Notice also that this is indeed the case if BZT effects are to be investigated—one of the primary motivations for the development of the *zFlow* code.

All the computations previously described (see Table I) are to be performed at each time step for both the explicit forward or the implicit backward Euler time integration scheme. As a consequence the CPU time required for a single time step advancement of the solution with the PRSV equation of state is roughly the same with both these methods. For the commonly considered explicit multistage Runge–Kutta time stepping schemes, however, the residuals and the EOS evaluations are to be performed at each stage of the time integration method, i.e. several times (typically four or five times) for each time step. This means that the CPU time required at each time step by such a method is much greater than that required by the simpler forward or backward Euler scheme. In the ideal gas case, instead, the most expensive operation is the linear system solution and consequently an implicit step is much more demanding in terms of computational resources than an explicit one (both single or multistage).

The number of time steps required to reach the steady state solution by an explicit methods is however much greater than that required by an implicit method since in the latter case a much greater CFL number can be employed. The number of iterations required by a Runge–Kutta scheme is generally at least two orders of magnitude larger than that required by the implicit backward Euler method, even if special convergence acceleration techniques such as residual averaging, enthalpy damping and local time stepping are employed. This means that, in the ideal gas case, there is a tradeoff between the number of steps required to reach the steady state solution and the computational cost of a single time step. However this is not the

case for the PRSV computation, where the implicit backward Euler scheme is clearly much more efficient than the explicit Runge–Kutta scheme.

In the RAE2822 airfoil computation previously considered, for example, a reduction of five orders of magnitude of the L_2 norm of the density residuals was obtained after only 95 time steps, with values of $CFL_{\min}$, $CFL_{\max}$ and K appearing in Equation (25) equal to 1, 10^{10} and 2, respectively. The final value of the CFL number was equal to 4.78×10^9. In view of the very large CPU time needed, no attempt was made to compute the solution with the explicit multistage Runge–Kutta method.

6. NON-CLASSICAL GAS DYNAMICS EFFECTS IN A SILOXANE FLOW

As stated in the Introduction, one of the main driving forces for the development of *zFlow* is the investigation of the so-called non-classical fluid dynamic effects occurring in transonic and supersonic flows of high molecular weight fluids. As a first test of the *zFlow* capabilities of simulating unconventional phenomena which might occur in the dense gas thermodynamic region for BZT fluids, one of the examples presented in Reference [25] is calculated using MD_6M (octadecamethyloctasiloxane, $C_{18}H_{54}O_7Si_8$). MD_6M is a complex linear siloxanes molecule and the model used to compute its thermodynamic properties is described in References [36, 61]. In the example mentioned, in order to simplify the calculation and enhance the disappearance of the shock waves, a blade is approximated by two staggered sinusoidal bumps of length C. In the paper by Monaco *et al.* a supersonic flow of steam is compared to a flow of FC-71 ($C_{18}F_{39}N$). The same absolute free stream conditions are chosen so that the steam flow is in the ideal gas region while for FC-71 the undisturbed state is in the negative Γ region.

In order to obtain a qualitative comparison with the results published in Reference [25], the free stream parameters are chosen so that the reduced temperature and pressure are approximately the same as for FC-71 ($T_r = 0.995, P_r = 0.940$). As it can be noted in Plate 2, the flow fields calculated by *zFlow* is very similar to the one reported in Reference [25] and this can be regarded as a further indirect validation of the code.

Moreover the interesting phenomena of the formation of compression fans in the MD_6M flow, around the trailing and leading edge is observed. As it can be noted in Plates 2(a) and 3(a), in the usual case of a fluid obeying to conventional fluid dynamic laws, a compression shock wave emanates from both the leading and the trailing edge. The unconventional phenomena depicted in Plates 2(b) and 3(b) is in agreement with the theory of non-classical gas dynamics which prohibits the formation of compression shock waves for fluid flows in the thermodynamic region where Γ is negative ($\Gamma = -0.11$ for the free stream conditions of Plates 2(b) and 3(b)). In Plate 3(b) the different dependence of the Mach number from the density is evident if compared to the ideal gas case of Plate 3(a).

Figure 4 shows the values of C_P and Γ along the upper wall calculated by *zFlow* and also highlights a comparison with the data reported in Reference [25]. It can be noted that the curves almost superimpose in the case of the ideal gas solution while in the non-classical cases the trends are similar but the values are different as it is to be expected being the fluids and the employed EOSs different.

Even if the MD_6M flow simulation gives indication about the possibility of obtaining positive results for the application to ORC turbines for which siloxanes are indeed already

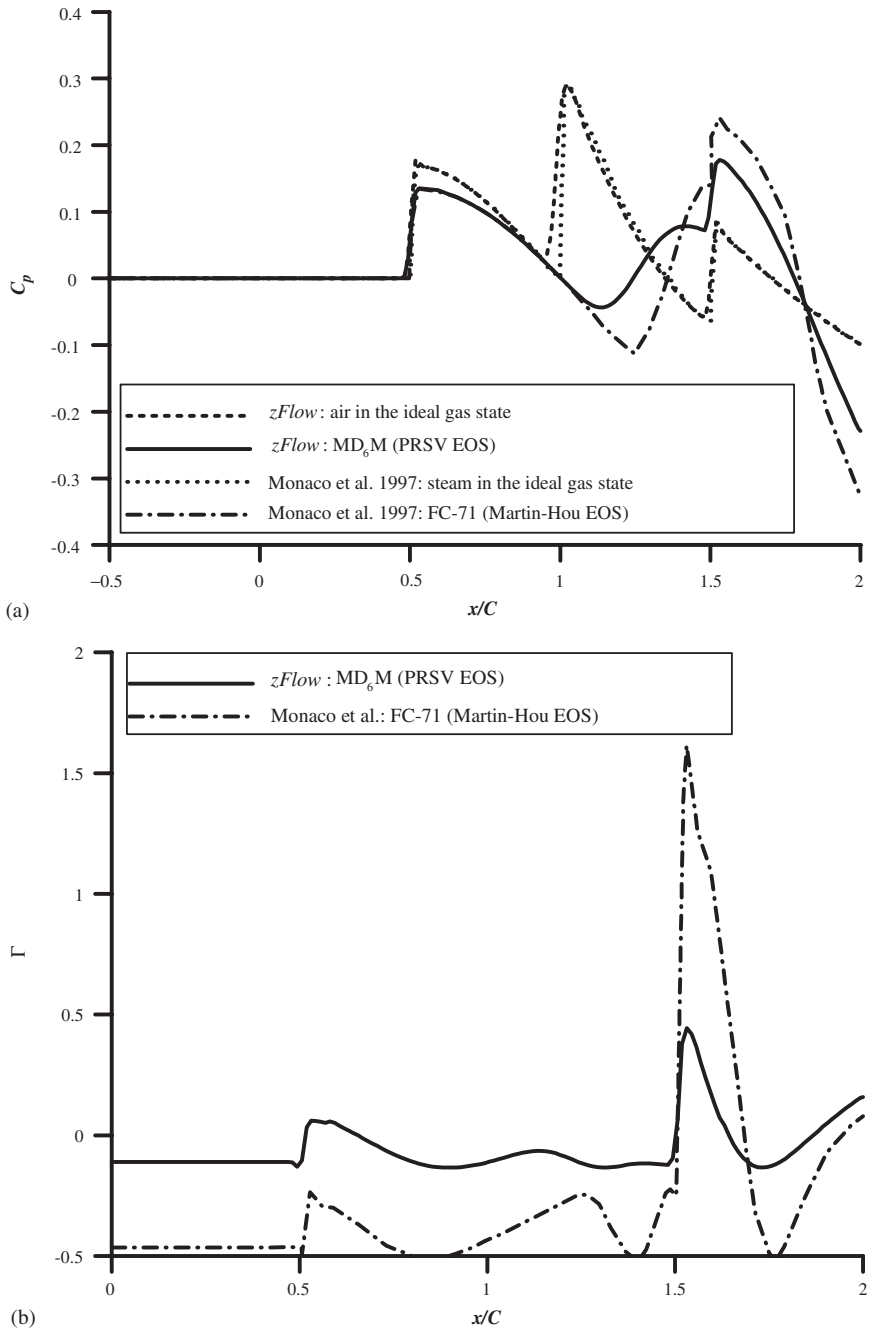

Figure 4. Variation of C_P (a) and Γ (b) along the upper wall for the channel of Plate 2. *zFlow* results for air (ideal gas) and MD_6M are reported together with data of similar calculations by Monaco *et al.* [25] for comparison.

employed, a word of caution is mandatory: the working fluid currently employed in some ORC plants is a simpler molecule for which non-conventional effects might not occur or be less significant. For more complex molecules as MD_6M or FC-71, it is very likely that the predicted negative Γ region lies at temperatures which are well above their thermal stability limit. Furthermore the presence and extension of the negative Γ region is tightly connected to the adopted thermodynamic model. Even if the PRSV EOS was found to be conservative with respect to non-classical fluid dynamics effects [36], an effort is planned in order to obtain a more accurate thermodynamic model.

7. CONCLUSIONS

This paper presents a numerical method for the solution of the Euler equations for fluids characterized by complex equations of state which has been implemented in a computer code named *zFlow*.

The distinguishing features of this code are (1) the usage of a high resolution upwind space discretization method for general unstructured and hybrid grids which are very well suited for the accurate computation of high Mach number flows and allows for the greatest geometrical flexibility and versatility, (2) the use of an implicit time integration scheme which proved to be crucial for the effective computations of fluids characterized by complex and therefore very computationally demanding EOS, and (3) the integration with an extensive library of thermodynamic models for both pure fluids and mixtures which allows for the realistic simulation of fluid flow phenomena which cannot be predicted by means of the simple polytropic ideal gas model commonly considered in most CFD codes.

The effectiveness of *zFlow* has been investigated by performing several calculations. Validation results concern the simulation of the transonic flow around the RAE2822 airfoil for a simple but very well documented test case. The results obtained with *zFlow* compare very well with the data available in the literature. The second computation is aimed at testing the performance of the method in the simulation of non-classical phenomena which may occur in the dense gas thermodynamic regime for BZT fluids. Also in this case the results are very satisfactory and in perfect agreement with the theory of non-classical gas dynamics which prohibits the formation of compression shock waves for fluids in the thermodynamic region where Γ is negative.

The plan for further development of *zFlow* includes the improvement in computational efficiency by avoiding the very time-consuming saturation calculations whenever possible, the extension to three dimensions, and the extension to the complete Reynolds-averaged Navier–Stokes equations. Much of the work in this direction has already been completed and will be the subject of a forthcoming paper.

APPENDIX A

A.1. Jacobian matrices and eigenstructure of the Euler equations in conservation form

The eigenstructure of the Jacobian matrices is of fundamental importance for the construction of effective discretization schemes of the Euler equations and is therefore briefly reviewed in

the following. Some notation is introduced first. A square matrix A with a complete set of eigenvectors can be factorized in terms of the left and right eigenvector matrices L and R and of the diagonal eigenvalue matrix Λ. The left, right, and eigenvalue matrices satisfy to the relations

$$AR = R\Lambda, \quad LA = \Lambda L, \quad LR = RL = I$$

from which the spectral decomposition

$$A = R\Lambda L, \quad \Lambda = LAR$$

can be easily deduced.

Since the flux function $\mathbf{F}(\mathbf{u})$ depends on the pressure P and the Jacobian matrices are the derivatives of the Euler flux function with respect to the conservative variables $\mathbf{u}$, it is convenient to introduce a relation which expresses the pressure as a function of the conservative variable vector $\mathbf{u}$. To distinguish this function from the EOS function $P = P(\rho, e)$, it is denoted by $\Pi = \Pi(\rho, \rho e^t, \rho \mathbf{v})$, a function which is simply related to $P(\rho, e)$ by the relation

$$\Pi(\mathbf{u}) = \Pi(\rho, \rho e^t, \rho \mathbf{v}) = P(\rho, \rho e^t/\rho - |\mathbf{v}|^2/(2\rho^2))$$

The partial derivatives of $\Pi(\mathbf{u})$ can be written in compact form as

$$\nabla_{\mathbf{u}} \Pi(\mathbf{u}) = \left[\frac{\partial \Pi}{\partial \rho}, \frac{\partial \Pi}{\partial \rho e^t}, \frac{\partial \Pi}{\partial \rho v_j} \right]^{\mathrm{T}} = [\Pi_\rho, \Pi_{\rho e^t}, \Pi_{\rho v_j}]^{\mathrm{T}}$$

where $j = 1, \ldots, d$. Notice that only 2 of the $d + 2$ components of the partial derivatives of $\Pi(\mathbf{u})$ are independent from one another since the underlying function $P(\rho, e)$ has only two independent variables (and partial derivatives). The application of the chain rule yields,

$$\begin{aligned} \Pi_\rho &= \frac{\partial P(\rho, e)}{\partial \rho} - \frac{e}{\rho} \frac{\partial P(\rho, e)}{\partial e} \\ \Pi_{\rho e^t} &= \frac{1}{\rho} \frac{\partial P(\rho, e)}{\partial e} \\ \Pi_{\rho v_j} &= \frac{v_j}{\rho} \frac{\partial P(\rho, e)}{\partial e} \end{aligned} \tag{A1}$$

The determination of the functions $P(\rho, e)$, $\alpha(\rho, e) = (\partial P/\partial \rho)_e$ and $\beta(\rho, e) = (\partial P/\partial e)_\rho$—which requires in general very involved calculations since the thermodynamic equation of state directly available is usually in the form $P = P(\rho, T)$, where T denotes the fluid temperature—is discussed in Section 3.

The Jacobian matrices, the left and right eigenvector matrices, and the eigenvalue matrices of the Euler equations are given in the following. For compactness of notation, two additional auxiliary variables are introduced. They appear over and over in the expression of the eigenvector matrices of the Euler equations, namely

$$\phi = \Pi_\rho - c^2, \quad \zeta = |\mathbf{v}|^2 - \Pi_\rho / \Pi_{\rho e^t} \tag{A2}$$

where c denotes the speed of sound. It can be shown that the speed of sound is related to the partial derivatives of the pressure function $\Pi(\mathbf{u})$ by the expression

$$c^2 = \Pi_\rho + (h^t - |\mathbf{v}|^2)\Pi_{\rho e^t} \tag{A3}$$

where h^t is the total enthalpy $h^t = e^t + p/\rho$. In addition, the partial derivatives $\Pi_{\rho v_j}$ can be written as

$$\Pi_{\rho v_j} = v_j \Pi_{\rho e^t} \tag{A4}$$

by combining the second and third relations (A1). The entries of the Jacobian matrices and the entries of their eigendecomposition in one, two and three spatial dimensions can therefore be explicitly written as described hereafter. It is worthwhile anticipating that, by virtue of relations (A2)–(A4), the Jacobian matrices in the direction $\mathbf{n}$ and their eigendecomposition can be expressed entirely in terms of the values of the thermodynamic variables h, Π_ρ, $\Pi_{\rho e^t}$ and on the velocity vector $\mathbf{v}$ (and, obviously, on the unit normal vector $\mathbf{n}$).

A.1.1. One-dimensional Euler equations. In the following $q_n = v_x, n_x = 1$

$$A = \begin{bmatrix} 0 & 0 & n_x \\ -q_n(h^t - \Pi_\rho) & q_n(1 + \Pi_{\rho e^t}) & h^t n_x - v_x q_n \Pi_{\rho e^t} \\ -v_x q_n + \Pi_\rho x & \Pi_{\rho e^t} n_x & q_n(2 - \Pi_{\rho e^t}) \end{bmatrix} \tag{A5}$$

$$\Lambda = \begin{bmatrix} v_x - c & 0 & 0 \\ 0 & v_x & 0 \\ 0 & 0 & v_x + c \end{bmatrix} \tag{A6}$$

$$R = \begin{bmatrix} 1 & 1 & 1 \\ h^t - c v_x & v^2 - \Pi_\rho / \Pi_{\rho e^t} & h^t + c v_x \\ v_x - c & v_x & v_x + c \end{bmatrix} \tag{A7}$$

$$L = \frac{1}{c^2} \begin{bmatrix} (\Pi_\rho + c v_x)/2 & \Pi_{\rho e^t}/2 & (\Pi_{\rho v_x} - c)/2 \\ \phi & \Pi_{\rho e^t} & \Pi_{\rho v_x} \\ (\Pi_\rho - c v_x)/2 & \Pi_{\rho e^t}/2 & (\Pi_{\rho v_x} + c)/2 \end{bmatrix} \tag{A8}$$

A.1.2. Two-dimensional Euler equations. In the following q_n and q_t denote the normal and tangential components of the velocity vector $\mathbf{v}$, i.e. $q_n = v_x n_x + v_y n_y$, $q_t = v_y n_x - v_x n_y$.

$$A = \begin{bmatrix} 0 & 0 & n_x & n_y \\ -q_n(h^t - \Pi_\rho) & q_n(1 + \Pi_{\rho e^t}) & h^t n_x - v_x q_n \Pi_{\rho e^t} & h^t n_y - v_y q_n \Pi_{\rho e^t} \\ -u q_n + \Pi_\rho n_x & \Pi_{\rho e^t} n_x & q_n + v_x(1 - \Pi_{\rho e^t}) n_x & v_x n_y - v_y \Pi_{\rho e^t} n_x \\ -v_y q_n + \Pi_\rho n_y & \Pi_{\rho e^t} n_y & v_y n_x - v_x \Pi_{\rho e^t} n_y & q_n + v_y(1 - \Pi_{\rho e^t}) n_y \end{bmatrix} \tag{A9}$$

$$\Lambda = \begin{bmatrix} q_n - c & 0 & 0 & 0 \\ 0 & q_n & 0 & 0 \\ 0 & 0 & q_n & 0 \\ 0 & 0 & 0 & q_n + c \end{bmatrix} \tag{A10}$$

$$R = \begin{bmatrix} 1 & 1 & 0 & 1 \\ h^t - cq_n & \zeta & -q_t & h^t + cq_n \\ u - cn_x & u & n_y & u + cn_x \\ v - cn_y & v & -n_x & v + cn_y \end{bmatrix} \tag{A11}$$

$$L = \frac{1}{c^2} \begin{bmatrix} (\Pi_\rho + cq_n)/2 & \Pi_{\rho e^t}/2 & (\Pi_{\rho v_x} - cn_x)/2 & (\Pi_{\rho v_y} - cn_y)/2 \\ -\phi & -\Pi_{\rho e^t} & -\Pi_{\rho v_x} & -\Pi_{\rho v_y} \\ c^2 q_t & 0 & c^2 n_y & -c^2 n_x \\ (\Pi_\rho - cq_n)/2 & \Pi_{\rho e^t}/2 & (\Pi_{\rho v_x} + cn_x)/2 & (\Pi_{\rho v_y} + cn_y)/2 \end{bmatrix} \tag{A12}$$

A.1.3. Three-dimensional Euler equations. In the following q_n denote the normal component of the velocity vector **v**, i.e. $q_n = v_x n_x + v_y n_y + v_z n_z$.

$$A = \begin{bmatrix} 0 & 0 & n_x & n_y & n_z \\ -q_n(h^t - \Pi_\rho) & q_n(1 + \Pi_{\rho e^t}) & h^t n_x - v_x q_n \Pi_{\rho e^t} & h^t n_y - v_y q_n \Pi_{\rho e^t} & h^t n_z - v_z q_n \Pi_{\rho e^t} \\ -v_x q_n + \Pi_\rho n_x & \Pi_{\rho e^t} n_x & q_n + v_x(1 - \Pi_{\rho e^t}) n_x & v_x n_y - v_y \Pi_{\rho e^t} n_x & v_x n_z - v_z \Pi_{\rho e^t} n_x \\ -v_y q_n + \Pi_\rho n_y & \Pi_{\rho e^t} n_y & v_y n_x - v_x \Pi_{\rho e^t} n_y & q_n + v_y(1 - \Pi_{\rho e^t}) n_y & v_y n_z - v_z \Pi_{\rho e^t} n_y \\ -v_z q_n + \Pi_\rho n_z & \Pi_{\rho e^t} n_z & v_z n_x - v_x \Pi_{\rho e^t} n_z & v_z n_y - v_y \Pi_{\rho e^t} n_z & q_n + v_z(1 - \Pi_{\rho e^t}) n_z \end{bmatrix} \tag{A13}$$

$$\Lambda = \begin{bmatrix} q_n - c & 0 & 0 & 0 & 0 \\ 0 & q_n & 0 & 0 & 0 \\ 0 & 0 & q_n & 0 & 0 \\ 0 & 0 & 0 & q_n & 0 \\ 0 & 0 & 0 & 0 & q_n + c \end{bmatrix} \tag{A14}$$

$$R = \begin{bmatrix} 1 & n_x & n_y & n_z & 1 \\ h^t - cq_n & \zeta n_x + vn_z - wn_y & \zeta n_y + wn_x - un_z & \zeta n_z + un_y - vn_x & h^t + cq_n \\ u - cn_x & un_x & un_y - n_z & un_z + n_y & u + cn_x \\ v - cn_y & vn_x + n_z & vn_y & vn_z - n_x & v + cn_y \\ w - cn_z & wn_x - n_y & wn_y + n_x & wn_z & w + cn_z \end{bmatrix} \tag{A15}$$

$$L = \frac{1}{c^2} \begin{bmatrix} (\Pi_\rho + cq_n)/2 & \Pi_{\rho e^t}/2 & (\Pi_{\rho v_x} - cn_x)/2 & (\Pi_{\rho v_y} - cn_y)/2 & (\Pi_{\rho v_z} - cn_z)/2 \\ -\phi n_x - c^2(vn_z - wn_y) & -\Pi_{\rho e^t} n_x & -\Pi_{\rho v_x} n_x & -\Pi_{\rho v_y} n_x + c^2 n_z & -\Pi_{\rho v_z} n_x - c^2 n_y \\ -\phi n_y - c^2(wn_x - un_z) & -\Pi_{\rho e^t} n_y & -\Pi_{\rho v_x} n_y - c^2 n_z & -\Pi_{\rho v_y} n_y & -\Pi_{\rho v_z} n_y + c^2 n_x \\ -\phi n_z - c^2(un_y - vn_x) & -\Pi_{\rho e^t} n_z & -\Pi_{\rho v_x} n_z + c^2 n_y & -\Pi_{\rho v_y} n_z - c^2 n_x & -\Pi_{\rho v_z} n_z \\ (\Pi_\rho - cq_n)/2 & \Pi_{\rho e^t}/2 & (\Pi_{\rho v_x} + cn_x)/2 & (\Pi_{\rho v_y} + cn_y)/2 & (\Pi_{\rho v_z} + cn_z)/2 \end{bmatrix} \tag{A16}$$

A.2. Riemann invariants and characteristic equations

The Riemann invariants and the characteristic equations for the one-, two- and three-dimensional Euler equations (projected in an arbitrary direction **n** in the multidimensional case) are hereby presented. Since the computation of the Riemann invariants using the conservative form of the Euler equations is in fact very involved, another formulation is described, i.e. the Riemann invariants and characteristic equations which can be obtained by considering the Euler equations in the 'primitive' form which governs the evolution in space and time of the primitive variables ρ, P and $\mathbf{v}$.

A.2.1. One-dimensional Euler equations. In the 1D case, the Riemann invariants in the differential form can be written as,

$$\begin{aligned} \mathrm{d}\Gamma_1 &= \frac{1}{2}\left(\frac{\mathrm{d}P}{c^2} - \frac{\rho}{c}\,\mathrm{d}u\right) \\ \mathrm{d}\Gamma_2 &= \mathrm{d}\rho - \frac{\mathrm{d}P}{c^2} \\ \mathrm{d}\Gamma_3 &= \frac{1}{2}\left(\frac{\mathrm{d}P}{c^2} + \frac{\rho}{c}\,\mathrm{d}u\right) \end{aligned} \tag{A17}$$

and the characteristic equations are therefore

$$\mathrm{d}P - \rho c\,\mathrm{d}u = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u - c \tag{A18}$$

$$c^2\mathrm{d}\rho - \mathrm{d}P = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u \tag{A19}$$

$$\mathrm{d}P + \rho c\,\mathrm{d}u = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u + c \tag{A20}$$

A.2.2. Two-dimensional Euler equations. In the 2D case, the Riemann invariants in the differential form can be written as,

$$\begin{aligned} \mathrm{d}\Gamma_1 &= \frac{1}{2}\left(\frac{\mathrm{d}P}{c^2} - \frac{\rho}{c}\,\mathrm{d}q_n\right) \\ \mathrm{d}\Gamma_2 &= \mathrm{d}\rho - \frac{\mathrm{d}P}{c^2} \\ \mathrm{d}\Gamma_3 &= \rho\,\mathrm{d}q_t \\ \mathrm{d}\Gamma_4 &= \frac{1}{2}\left(\frac{\mathrm{d}P}{c^2} + \frac{\rho}{c}\mathrm{d}q_n\right) \end{aligned} \tag{A21}$$

and the characteristic equations are therefore

$$\mathrm{d}P - \rho c\,\mathrm{d}q_n = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u - c \tag{A22}$$

$$c^2\,\mathrm{d}\rho - \mathrm{d}P = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u \tag{A23}$$

$$\mathrm{d}q_t = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u \tag{A24}$$

$$\mathrm{d}P + \rho c\,\mathrm{d}q_n = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u + c \tag{A25}$$

A.2.3. Three-dimensional Euler equations. In the 3D case, the Riemann invariants in the differential form can be written as,

$$\begin{aligned} \mathrm{d}\Gamma_1 &= \frac{1}{2}\left(\frac{\mathrm{d}P}{c^2} - \frac{\rho}{c}\,\mathrm{d}q_n\right) \\ \mathrm{d}\Gamma_2 &= \left(\mathrm{d}\rho - \frac{\mathrm{d}P}{c^2}\right) n_x + \rho(n_z\mathrm{d}v_y - n_y\mathrm{d}v_z) \\ \mathrm{d}\Gamma_3 &= \left(\mathrm{d}\rho - \frac{\mathrm{d}P}{c^2}\right) n_y + \rho(n_x\mathrm{d}v_z - n_z\mathrm{d}v_x) \\ \mathrm{d}\Gamma_4 &= \left(\mathrm{d}\rho - \frac{\mathrm{d}P}{c^2}\right) n_z + \rho(n_y\mathrm{d}v_x - n_x\mathrm{d}v_y) \\ \mathrm{d}\Gamma_5 &= \frac{1}{2}\left(\frac{\mathrm{d}P}{c^2} + \frac{\rho}{c}\mathrm{d}q_n\right) \end{aligned} \tag{A26}$$

and the characteristic equations are therefore

$$\mathrm{d}P - \rho c\,\mathrm{d}q_n = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u - c \tag{A27}$$

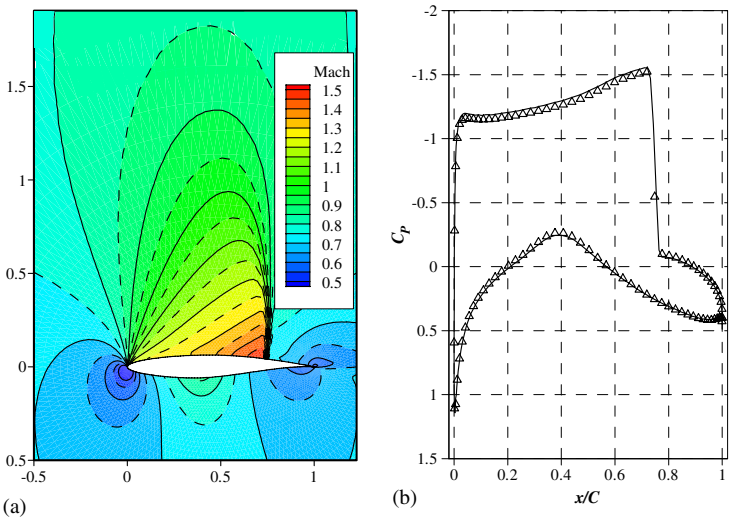

Plate 1. (a) Iso-Mach lines calculated by *zFlow* (nitrogen and PRSV equation of state) for comparison with the reference case in Reference [60]. Far field conditions: $M = 0.75$, angle of attack $\alpha = 3°$, $P = 1$bar and $T = 25°$C. Grid: Type-O, 192×49 nodes. (b) Comparison between the pressure coefficient calculated by *zFlow* ($\triangle$) and the reference data reported in Reference [60] ($-$).

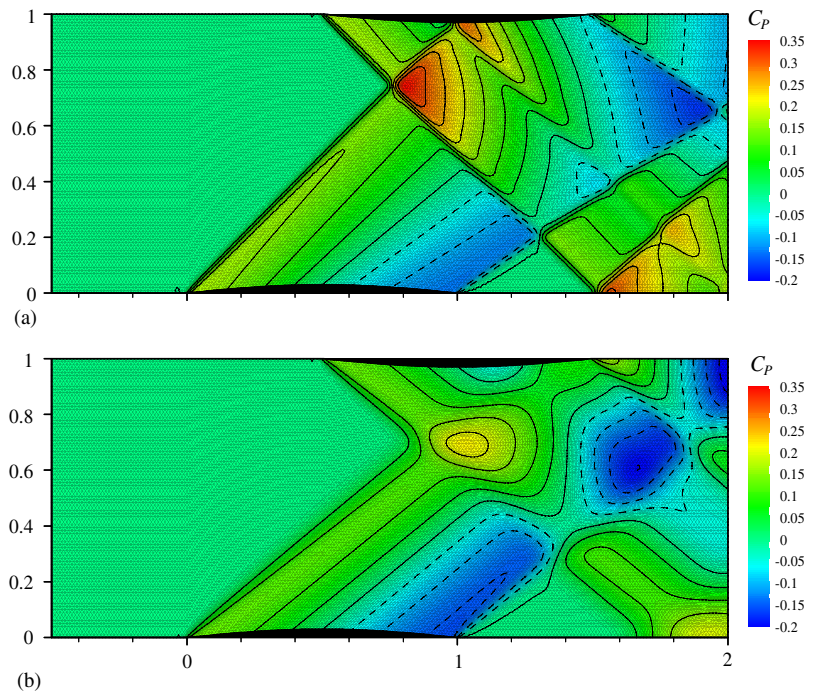

Plate 2. Constant C_p contours for the flow of: (a) air (ideal gas); and (b) MD_6M (PRSV EOS) between two staggered sinusoidal blades. Free stream conditions for MD_6M: $M_\infty = 1.6$, $P_\infty = 6.362$ bar, $T_\infty = 686.82$ K.

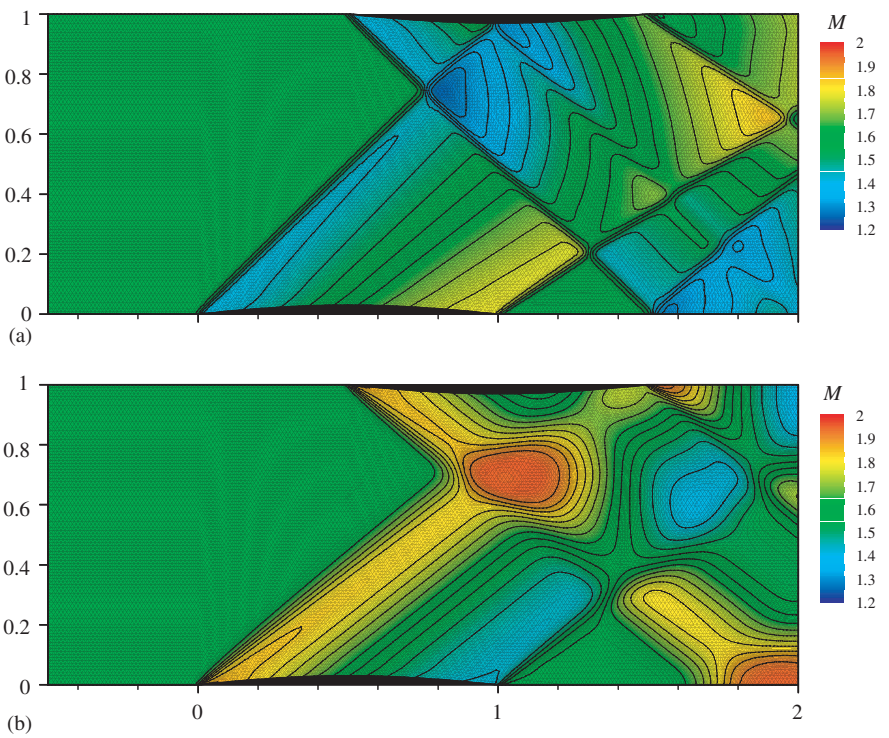

Plate 3. Constant Mach contours for the flow of: (a) air (ideal gas); and (b) MD_6M (PRSV EOS) between two staggered sinusoidal blades. Free stream conditions for MD_6M: $M_\infty = 1.6$, $P_\infty = 6.362$ bar, $T_\infty = 686.82$ K.

$$(c^2\,\mathrm{d}\rho - \mathrm{d}P)n_x + \rho c^2(n_z\mathrm{d}v_y - n_y\mathrm{d}v_z) = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u \tag{A28}$$

$$(c^2\,\mathrm{d}\rho - \mathrm{d}P)n_y + \rho c^2(n_x\mathrm{d}v_z - n_z\mathrm{d}v_x) = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u \tag{A29}$$

$$(c^2\,\mathrm{d}\rho - \mathrm{d}P)n_z + \rho c^2(n_y\mathrm{d}v_x - n_x\mathrm{d}v_y) = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u \tag{A30}$$

$$\mathrm{d}P + \rho c\,\mathrm{d}q_n = 0 \quad \text{along} \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = u + c \tag{A31}$$

REFERENCES

1. Cramer MS. Nonclassical dynamics of classical gases. *Nonlinear Waves in Real Fluids of International Center for Mechanical Sciences, Courses and Lectures*. Springer: Berlin, 1991.
2. Cramer MS, Kluwick A. On the propagation of waves exhibiting both positive and negative nonlinearity. *Journal of Fluid Mechanics* 1984; **142**:9–37.
3. Cramer MS, Kluwick A, Watson LT, Pelz W. Dissipative waves in fluids having both positive and negative nonlinearity. *Journal of Fluid Mechanics* 1986; **169**:323–336.
4. Cramer MS. Negative nonlinearity in selected fluorocarbons. *Physics of Fluids A* 1989; **11**(1):1894–1897.
5. Chandrasekar D, Prasad P. Transonic flow of a fluid with positive and negative nonlinearity through a nozzle. *Physics of Fluids A* 1991; **3**(3):427–438.
6. Schnerr G, Leidner P. Diabatic supersonic flows of dense gases. *Physics of Fluids* 1991; **3**(10):2445–2458.
7. Aldo AC, Argrow BM. Dense gas flows in minimum length nozzles. *Journal of Fluids Engineering* 1994; **117**:270–276.
8. Brown BP, Argrow BM. Nonclassical dense gas flows for simple geometries. *AIAA Journal* 1998; **36**(10): 1842–1847.
9. Brown BP, Argrow BM. Application of Bethe–Zel'dovic–Thompson fluids in organic Rankine cycle engines. *Journal of Propulsion and Power* 2000; **16**(6):1118–1123.
10. Drikakis D, Tsangaris S. Real gas effects for compressible nozzle flows. *Journal of Fluids Engineering* 1993; **115**:115–120.
11. Wagner B, Schmidt W. Theoretical investigation of real gas effects in cryogenic wind tunnels. *AIAA Journal* 1978; **16**(6):580–586.
12. Anderson WK. Numerical study of the aerodynamic effects of using sulfur hexafluoride as a test gas in wind tunnels. *Technical Paper* 3086, NASA, NASA Langley Research Center, Hampton, VA, May 1991.
13. Anders JB, Anderson WK, Murthy AV. Transonic similarity theory applied to a supercritical airfoil in heavy gases. *Journal of Aircraft* 1999; **36**(6):957–964.
14. Cravero C, Satta A. A CFD model for real gas flows. In *ASME Turbo Expo*, Munich, Germany. ASME: New York, 2000:1–10.
15. Hoffren J, Talonpoika T, Larjola J, Siikonen T. Numerical simulation of real-gas flow in a supersonic turbine nozzle ring. *Journal of Engineering for Gas Turbines and Power* 2002; **124**:395–403.
16. Schnerr GH, Leidner P. Two-dimensional nozzle flow of dense gases. In *Fluids Engineering Conference*. ASME: New York, 1993, 93–FE–8.
17. Morren S. Transonic aerodynamics of dense gases. *Technical Memorandum TM* 103722, NASA, 1991.
18. Cramer MS, Tarkenton GM. Transonic flows of Bethe–Zel'dovich–Thompson fluids. *Journal of Fluid Mechanics* 1992; **240**:197–228.
19. Brown BP, Argrow BM. Two-dimensional shock tube flow for dense gases. *Journal of Fluid Mechanics* 1997; **349**:95–115.
20. Fergason SH, Argrow BM. Navier–Stokes analysis of dense gas shock tube flow fields. In *International Workshop on Dense Gas Dynamics*, 1999.
21. Fergason SH, Argrow BM. Simulations of nonclassical dense gas dynamics. In *35th AIAA Thermophysics Conference*, 2001; 2001–2752.
22. Guardone A. Nonclassical gasdynamics: thermodynamic modeling and numerical simulation of multidimensional flows of BZT fluids. *Ph.D. Thesis*, Politecnico di Milano, Italy, 2001.
23. Guardone A, Vigevano L. Roe linearization for the van der Waals gas. *Journal of Computational Physics* 2002; **175**:50–78.
24. Schnerr GH, Leidner P. Numerical investigation of axial cascades for dense gases. In *PICAST'1—Pacific International Conference on Aerospace Science Technology*, Chin EL (ed.), vol. 2. National Cheng Kung University Publications, Taiwan, Republic of China, 1993; 818–825.

25. Monaco JF, Cramer MS, Watson LT. Supersonic flows of dense gases in cascade configurations. *Journal of Fluid Mechanics* 1997; **330**:31–59.
26. Selmin V. The node-centered finite volume approach: bridge between finite differences and finite elements. *Computer Methods in Applied Mechanics and Engineering* 1993; **102**:107–138.
27. Selmin V, Formaggia L. Unified construction of finite element and finite volume discretizations for compressible flows. *International Journal for Numerical Methods in Engineering* 1996; **39**:1–32.
28. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
29. Roe PL, Pike J. Efficient construction and utilization of approximate Riemann solution. In *Computing Methods in Applied Science and Engineering*. North-Holland: Amsterdam, 1984.
30. Roe PL. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics* 1986; **18**:337–365.
31. Vinokur M, Montagné JL. Generalized flux-vector splitting and Roe average for an equilibrium real gas. *Journal of Computational Physics* 1990; **89**(2):276–300.
32. Mottura L, Vigevano L, Zaccanti M. An evaluation of Roe's scheme generalizations for equilibrium real gas flows. *Journal of Computational Physics* 1997; **138**:354–399.
33. Glaister P. An approximate linearized Riemann solver for the Euler equations for real gases. *Journal of Computational Physics* 1988; **74**:382–408.
34. Abgrall R. An extension of Roe's upwind scheme to algebraic equilibrium real gas models. *Computers & Fluids* 1991; **19**(2):171.
35. Guardone A, Selmin V, Vigevano L. An investigation of Roe's linearization and average for ideal and real gases. *Scientific Report DIA-SR* 99-01, Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, January 1999.
36. Colonna P, Silva P. Dense gas thermodynamic properties of single and multi-component fluids for fluid dynamics simulations. *ASME Journal of Fluids Engineering* 2003; **125**(3):414–427.
37. Reynolds WC. *Thermodynamic properties in S.I.* Department of Mechanical Engineering, Stanford University: Stanford, CA, 1979.
38. Keenan JH, Keyes FG *et al. Steam Tables*. Wiley: New York, 1969.
39. Haar L, Gallagher JS. Thermodynamics properties of ammonia. *Journal of Physical and Chemical Reference Data* 1978; **7**(30):635–791.
40. Starling KE. Equation of state and computer prediction. *Fluid Thermodynamic Properties for Light Petroleum Substances*. Gulf Publishing Co.: Huston, TX, 1973.
41. Guardone A, Quartapelle L. Spatially factorized Galerkin and Taylor–Galerkin schemes for multidimensional conservation laws. *Scientific Report DIA-SR* 00-18, Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, October 2000.
42. Godlewski E, Raviart PA. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer: New York, NY, USA, 1995.
43. Weatherill N, Hassan O, Marchant M, Marcum D. Adaptive inviscid flow solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes. *AIAA Report AIAA*-93-3390, AIAA, 1993.
44. Davis S. TVD finite difference schemes and artificial viscosity. *ICASE Report* 84-20 *NASA CR*-172373, NASA Langley Research Center, 1984.
45. Van Albada G, Van Leer B, Roberts W. A comparative study of computational methods in cosmic gas dynamics. *Astronomy & Astrophysics* 1982; **108**:76–84.
46. Harten A. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* 1983; **49**:357–393.
47. Shu C-H. Total-variation-diminishing time discretizations. *SIAM Journal on Scientific and Statistical Computing* 1988; **9**(6):1073–1084.
48. Lemmon EW, Jacobsen RT. An international standard formulation for the thermodynamic properties of 1,1,1-trifluoroethane (hfc-143a) for temperatures from 161 to 450 K and pressures to 50 mP. *Journal of Physical and Chemical Reference Data* 2000; **29**(4):521–552.
49. Span R, Wagner W, Lemmon EW, Jacbosen RT. Multiparameter equations of state—recent trends and future challanges. *Fluid Phase Equilibria* 2001; **183–184**:1–20.
50. Wong DSH, Sandler SI. A theoretically correct mixing rule for cubic equations of state. *AIChE Journal* 1992; **38**:671–680.
51. Valderrama JO. The state of the cubic equations of state. *Industrial & Engineering Chemistry Research* 2003; **42**(8):1603–1618.
52. Taylor R. Automatic derivation of thermodynamic property functions using computer algebra. *Fluid Phase Equilibria* 1997; **129**:37–47.
53. Levelt Sengers JMH, Morrison G, Chang RF. Critical behavior in fluids and fluid mixtures. *Fluid Phase Equilibria* 1983; **14**:19–44.
54. Tang S, Sengers JV. Thermodynamic behavior of fluids in the supercritical region. *Journal of Supercritical Fluids* 1991; **4**:209–214.

55. Luettmer-Strathmann J, Tang S, Sengers JV. A parametric crossover model for fluids in the critical region. *Fluid Phase Equilibria* 1992; **75**:39–52.
56. Anisimov MA, Povodyrev AA, Sengers JV. Crossover critical phenomena in complex fluids. *Fluid Phase Equilibria* 1999; **158–160**:537–547.
57. Smith DH, Ferer M. Improved phase boundary for one-component vapor–liquid equilibrium incorporating critical behavior and cubic equations of state. *Fluid Phase Equilibria* 1995; **113**:103–115.
58. Kiselev SB. Cubic crossover equation of state. *Fluid Phase Equilibria* 1998; **147**:7–23.
59. Kiselev SB. Cubic crossover equation of state for mixtures. *Fluid Phase Equilibria* 1999; **162**:51–82.
60. Viviand H. Numerical solution of two-dimensional reference test cases. *Advisory Report* 211-6 *on Test Cases for Inviscid Flow Field*, AGARD, 1985.
61. Angelino G, Colonna P. Multicomponent working fluids for organic Rankine cycles (ORCs). *Energy* 1998; **23**(6):449–463.

