Computational Mechanics (2021) 67:785–809
https://doi.org/10.1007/s00466-020-01963-6

ORIGINAL PAPER

Stabilized methods for high-speed compressible flows: toward hypersonic simulations

David Codoni[1] · Georgios Moutsanidis[2] · Ming-Chen Hsu[3] · Yuri Bazilevs[4] · Craig Johansen[1] · Artem Korobenko[1]

Received: 20 October 2020 / Accepted: 3 December 2020 / Published online: 20 January 2021

Abstract
A stabilized finite element framework for high-speed compressible flows is presented. The Streamline-Upwind/Petrov–Galerkin formulation augmented with discontinuity-capturing (DC) are the main constituents of the framework that enable accurate, efficient, and stable simulations in this flow regime. Full- and reduced-energy formulations are employed for this class of flow problems and their relative accuracy is assessed. In addition, a recently developed DC formulation is presented and is shown to be particularly well suited for hypersonic flows. Several verification and validation cases, ranging from 1D to 3D flows and supersonic to the hypersonic regimes, show the excellent performance of the proposed framework and set the stage for its deployment on more advanced applications.

Keywords Compressible flows · Stabilized methods · Shock-capturing · Finite elements · Supersonic flows · Hypersonic flows

1 Introduction

In computational fluid dynamics several approaches have been used in order to find the numerical solution of the Navier–Stokes equations: finite difference (FDM), finite volume (FVM) and finite element (FEM) methods. While FDM requires structured grids, even though curved cells are allowed, the FVM offers good geometric flexibility and no constraints on the mesh type are required. Similar to the FVM, the FEM works for both structured and unstructured grids with curved or rectilinear cells (or elements). For these reasons, both methods, FVM and FEM, can easily handle very complex geometries [75]. The superiority of FEM with respect to FVM is in the ease of implementation of the boundary conditions and the ease in obtaining higher-order accuracy, which are not trivial in FVM [18].

In this work the focus will be exclusively on the FEM numerical technique. The FEM has its origin in the structural mechanics field. It was then applied to the solution of fluid dynamics problems, not without any difficulties. Indeed, a lot of research has been dedicated to the method since the end of the 1970s [59] to the development of stabilized formulations for compressible [75] and incompressible flows [8]. The subject of the stabilized formulations for FEM in fluid dynamic is still an active research field and it is thanks to these formulations that FEM became a valid numerical technique for the study of fluid dynamics problems (turbulent and laminar flows, incompressible and compressible flows over a wide range of flow regimes).

The first finite element stabilized formulation, based on the streamline upwind concept, was developed as a Petrov–Galerkin formulation by [8] for the linear advection–diffusion equation and the incompressible Navier–Stokes equations. In [8], the authors showed the accuracy and robustness of the Streamline-Upwind/Petrov–Galerkin (SUPG) formulation for several numerical cases. In that period, other researchers developed upwind finite elements approximations in order to enhance the stability of convection dominated flows, such as in [5,58,59]. A mathematical analysis assessing the stability and order of convergence of the

✉ Artem Korobenko
artem.korobenko@ucalgary.ca

1 Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, AB T2L1Y6, Canada

2 Department of Civil Engineering, Stony Brook University, Stony Brook, NY 11794, USA

3 Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA

4 School of Engineering, Brown University, Providence, RI 02912, USA

FEM for convection–diffusion problems and linear hyperbolic problems can be found in [38]. The first step towards the generalization of the SUPG formulation to the compressible flows was made in [34,75,76]. In these works, the class of Petrov–Galerkin finite element formulation based on the streamline upwind methods was developed for the compressible Euler equations. The main idea was to write the system of conservation laws in quasi-linear form using the Jacobian matrix of the flux vector in the discontinuous part of the weighting function [34]. In [76] the definition of the stabilization parameter $\boldsymbol{\tau}$ was identified as subject of further research, since the accuracy of the solution depends on it. In order to select the optimal value of the parameter τ for every equation of the system describing the problem, it was proposed in [31] to define $\boldsymbol{\tau}$ as a matrix and by doing so separate $\boldsymbol{\tau}$ could be defined for each degree of freedom. The work presented in [31] was generalized to the space-time finite element formulations for compressible flows by [54], where a new design for $\boldsymbol{\tau}$ parameter was proposed. Up to this point it was established that the SUPG finite element formulations for convection-dominated flows produce oscillation-free solutions, unlike the classical Galerkin methods. In this regard, interesting results have been obtained when the conservation laws were solved in a particular set of entropy variables. A discussion on the finite element formulations derived in entropy variables can be found in [29,30]. In [29], it was shown that symmetric forms of the compressible Euler and Navier–Stokes equations can be obtained by selecting a particular set of entropy variables. The classical Galerkin weighted residuals formulation was adopted to discretize the symmetric Navier–Stokes equations resulting in an intrinsically stable discrete solution which always satisfies the second law of thermodynamic [29]. The authors of [29] concluded that whenever a discontinuity in the solution field is present, the Galerkin method was not effective and a Petrov–Galerkin formulation was necessary to solve this class of problems. A convergence analysis of FEM, based on streamline diffusion techniques with and without shock-capturing term, for hyperbolic conservation laws, was carried out in [39,40,57]. The authors showed the convergence of the finite elements solution to the entropy solution (physical solution) of the conservation laws stating that the results could be extended to a multidimensional system of conservation laws derived in entropy variables. The stabilized finite element formulations supplemented with shock-capturing terms in two different sets of variables, conservation and entropy variables, have been tested and compared in [45,46]. The results showed no superiority of one set of variables with respect to the other. Both choices in variables resulted in accurate solutions of the compressible flows and almost no differences in the two discrete solutions were observed [45]. From an analysis of finite element formulations in entropy variables, it was clear that the streamline upwind formulations could not perform well whenever a discontinuity was present in the solution field. Although the SUPG term ensured an oscillation-free discrete solution when the solution was smooth, it was found to be rather ineffective with cases involving discontinuos exact solutions. In those scenarios, a discrete solution with spurious oscillations was produced. In order to solve the problem the discontinuity-capturing (DC) term (or shock-capturing) was required. The DC term was first added to the SUPG formulation for the linear scalar convection–diffusion equation in [33] and to the convection–diffusion–reaction equation in [78]. It was generalized to multidimensional systems of advective–diffusion equations in [32]. From the DC term in entropy variables defined in [33], a DC parameter in conservation variables was defined in [45,46].

The Galerkin Least Squares (GLS) method represents another stabilized formulation which originates from a generalization of the SUPG method. The method was first developed for advection–diffusion equations in [28]. The GLS formulation for compressible Navier–Stokes equations was extended to any set of variables in [23–25]. The extension of the method to any choice of variables was derived starting from the GLS formulation in entropy variables. The performance comparison among the different sets of variables was described in [24,25]. It was stated that only a set of entropy variables could satisfy the second law of thermodynamics regardless of the use of any dissipation terms, while for all other sets of variables (conservation and primitive variables) dissipation terms were necessary, such as the GLS and the DC operator. Moreover, it was observed that both entropy and pressure primitive variables perform well in the incompressible limit of the compressible Navier–Stokes equations and for this reason they represent a good choice in a unified compressible-incompressible numerical framework [24,25]. The space-time GLS formulation in entropy variables supplemented with a DC term was applied to the solution of the compressible Euler and Navier–Stokes equations in [54]. An adaption of the GLS formulation to hypersonic flows computation with the entropy variables was described in [13], where instead of the perfect gas model an equilibrium chemistry algorithm was implemented. The initial formulation for thermochemical nonequilibrium with the entropy variables was also presented in [14]. First moving-mesh computations with the compressible-flow SUPG method were reported in 1990s, using the space-time compressible-flow SUPG method, first in 2D (axi-symmetric), for air intake of a jet engine at $M = 1.4$ in [2,68] and then in 3D, for two high-speed trains passing each other in a tunnel, in [66]. Most recently, the SUPG-based finite element formulation was presented in [42] using conservative variables to model the hypersonic atmospheric entry of large vehicles with an ablative thermal protection system.

A lot of research has been dedicated to find the correct expression for the stabilization parameters (for both

SUPG/GLS and DC terms). The stabilization parameters are indeed very important, since they affect the accuracy of the method and they should be selected in a way that a good compromise between accuracy and stability is achieved [23]. An approach was developed by [77], where the stabilization parameters were computed based on element-level matrices and vectors. The parameters developed in [77] took into account the element length scales, advection and element Reynolds number and were applied to the unsteady advection–diffusion equation and to the Navier–Stokes equations of incompressible flows. The stabilization parameters developed for each element using the advection-dominated and diffusion-dominated limits are described in [70,71,74]. The element level stabilization parameters were applied to the classic SUPG formulation developed in [8] supplemented by the shock-capturing of [46] for the inviscid compressible flows in [11,12]. In particular, in [12] the parameters were computed for each degree of freedom of each element. Further discussion on the computation of the time parameters of the stabilized formulation for compressible flows can be found in [72,73]. Following the first introduction of the DC term for the scalar advection–diffusion equation [33], another DC term was proposed in [78] in the context of the SUPG formulation for the convection–diffusion–reaction equations. The DC term introduced in [78] contains a second element length scale in the direction of the solution gradient (the first element length scale in the advection direction being the one used in the SUPG term definition). Using this element length scale in the direction of the solution gradient, the DC directional dissipation (DCDD) was developed and introduced in [69,71]. The DCDD parameter is selected in a way that the method is not over diffusive in the particular case of advection parallel to the solution strong gradient. Simpler and less computationally expensive DC parameters with respect to the one introduced in [46], were developed in [73,79] in the context of stabilized finite element formulation for compressible flows. The new shock-capturing parameters, categorized as YZβ Shock-Capturing [79–81], were derived from the DCDD idea for incompressible flows and unlike the DC developed in [46], their definition was based on conservation variables instead of entropy variables. In [80,81], the SUPG formulation supplemented with the YZβ Shock-Capturing was applied to the solution of inviscid supersonic flows and it proved to yield better shock quality than the SUPG formulation [8] with the DC developed in [46]. A number of new developments happened in the last few years in stabilization parameters and element lengths, including those for isogeometric discretization and those with node-numbering invariance. Some of them were summarized in [7]. We refer the interested reader to [44,50,61–64,82].

Some of the earliest compressible-flow computations with the SUPG plus DC method in complex engineering problems were reported in 1990s, for a delta-wing in [67,68], for a commercial aircraft in [68], for a missile in [56], for two high-speed trains in a tunnel in [66], and for a fighter aircraft in [66]. Most recently, the SUPG formulation was also successfully applied to several complex engineering problems in compressible flow regime such as gas turbines [7,43,84], rotorcraft [83], full vehicle aerodynamics [1], spacecraft parachute aerodynamics [41,60], and related applications in incompressible regime (e.g. analysis of the turbocharger [51,52]), however, little research has been dedicated to investigating the method in hypersonic flow regimes. It is the aim of the current work to investigate in details the performance of the SUPG-based finite element formulation for hypersonic flow modeling, compare two formulations with the different forms of the energy conservation equation and examine new DC operator.

The rest of the paper is structured as follows. In Sect. 2 the governing equations of viscous compressible flows are described as well as the steps to obtain the finite element SUPG formulation supplemented with a DC operator. The validation and verification of the stabilized finite element formulation is shown in Sect. 3. The results of numerical test cases for high speed flows, including the 1D Sod's problem, the 2D Mach 6 flat plate, the 2D Mach 10 oblique shock, the 2D Mach 17 cylinder case, and the 3D Mach 6 Viking Lander Capsule, are also presented in Sect. 3. Concluding remarks and future research are summarized in Sect. 4.

2 Methodology

2.1 Governing equations

The set of the governing equations solved in this work is the system of Navier–Stokes equations for compressible flows involving the conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

momentum

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \nabla \cdot \boldsymbol{\tau}, \tag{2}$$

and energy

$$\frac{\partial (\rho e_{tot})}{\partial t} + \nabla \cdot (\rho \mathbf{u} e_{tot}) + \nabla \cdot (p\mathbf{u}) = \nabla \cdot (\boldsymbol{\tau}\mathbf{u} - \mathbf{q}), \tag{3}$$

assuming no source terms. In the above equations, ρ is the density, $\mathbf{u}$ is the velocity vector, p is the pressure, $\boldsymbol{\tau}$ is the second-order viscous stress tensor

$$\boldsymbol{\tau} = \lambda \nabla \cdot \mathbf{u} + \mu \left((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T \right) \tag{4}$$

and $\mathbf{q}$ is the heat conduction vector

$$\mathbf{q} = -\kappa \nabla T, \tag{5}$$

where λ is the bulk viscosity, μ is the dynamic viscosity and κ is the thermal conductivity. In Eq. (3) e_{tot} is the total energy defined as the sum of the internal and kinetic energies

$$e_{tot} = e + \frac{\|\mathbf{u}\|^2}{2}, \tag{6}$$

where the internal energy is computed assuming a calorically perfect gas:

$$e = c_v T. \tag{7}$$

Pressure, density, and internal energy are related through the ideal gas equation of state

$$p = \rho(\gamma - 1)e, \tag{8}$$

where γ is the adiabatic index.

The Navier–Stokes equations of compressible flows can be compactly written as

$$\tilde{\mathbf{U}}_{,t} + \tilde{\mathbf{F}}^{adv}_{i,i} = \tilde{\mathbf{F}}^{diff}_{i,i}, \tag{9}$$

where $\tilde{\mathbf{U}}$ is a vector of conservation variables,

$$\tilde{\mathbf{U}} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e_{tot} \end{bmatrix}, \tag{10}$$

index $i = 1, ..., d$, where $d = 2, 3$ is the space dimension, and $\tilde{\mathbf{F}}^{adv}_i$ and $\tilde{\mathbf{F}}^{diff}_i$ are the ith component of the advective and diffusive fluxes, respectively, given by

$$\tilde{\mathbf{F}}^{adv}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 + p\delta_{1i} \\ \rho u_i u_2 + p\delta_{2i} \\ \rho u_i u_3 + p\delta_{3i} \\ \rho u_i e_{tot} + p u_i \end{bmatrix} \quad \tilde{\mathbf{F}}^{diff}_i = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} u_j - q_i \end{bmatrix}. \tag{11}$$

The advective flux vector can be further split in two terms as

$$\tilde{\mathbf{F}}^{adv}_i = \tilde{\mathbf{F}}^{adv/p}_i + \tilde{\mathbf{F}}^{p}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 \\ \rho u_i u_2 \\ \rho u_i u_3 \\ \rho u_i e_{tot} \end{bmatrix} + \begin{bmatrix} 0 \\ p\delta_{1i} \\ p\delta_{2i} \\ p\delta_{3i} \\ p u_i \end{bmatrix}. \tag{12}$$

We can simplify the compressible-flow equation system by introducing the mass and momentum balance in the energy equation leading to

$$\mathbf{U}_{,t} + \mathbf{F}^{adv/p}_{i,i} + \mathbf{F}^{p}_{i,i} + \mathbf{F}^{sp} = \mathbf{F}^{diff}_{i,i}, \tag{13}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho e \end{bmatrix} \tag{14}$$

$$\mathbf{F}^{adv}_i = \mathbf{F}^{adv/p}_i + \mathbf{F}^{p}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_1 \\ \rho u_i u_2 \\ \rho u_i u_3 \\ \rho u_i e \end{bmatrix} + \begin{bmatrix} 0 \\ p\delta_{1i} \\ p\delta_{2i} \\ p\delta_{3i} \\ 0 \end{bmatrix} \quad \mathbf{F}^{diff}_i = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ -q_i \end{bmatrix} \tag{15}$$

$$\mathbf{F}^{sp} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ p u_{i,i} - \tau_{ij} u_{j,i} \end{bmatrix} \tag{16}$$

The flux vector $\mathbf{F}^{sp}$ is the contribution of stress power in the energy equation.

As a result, we have two formulations which will be investigated in this work: The full-energy formulation defined by Eqs. (9)–(12) and the reduced-energy formulation defined by Eqs. (13)–(16). The latter formulation was presented in [84].

2.2 Weak formulations

The weak form of the problem can be obtained by applying the method of weighted residuals. The strong form of the Navier–Stokes equations is multiplied by a vector-valued test function $\mathbf{W} \in \mathbf{V}$, where $\mathbf{V}$ is a suitably chosen space of test functions, and integrated over the spatial domain Ω with boundary Γ. The integration by parts is then performed on the pressure and diffusive fluxes resulting in

$$\int_\Omega \mathbf{W} \cdot \left(\tilde{\mathbf{U}}_{,t} + \tilde{\mathbf{F}}^{adv/p}_{i,i}\right) d\Omega - \int_\Omega \mathbf{W}_{,i} \cdot \left(\tilde{\mathbf{F}}^{p}_i - \tilde{\mathbf{F}}^{diff}_i\right) d\Omega + \int_{\Gamma_h} \mathbf{W} \cdot \left(\tilde{\mathbf{F}}^{p}_i - \tilde{\mathbf{F}}^{diff}_i\right) n_i \, d\Gamma = 0 \tag{17}$$

for the full-energy form and in

$$\int_\Omega \mathbf{W} \cdot \left(\mathbf{U}_{,t} + \mathbf{F}^{adv/p}_{i,i} + \mathbf{F}^{sp}\right) d\Omega - \int_\Omega \mathbf{W}_{,i} \cdot \left(\mathbf{F}^{p}_i - \mathbf{F}^{diff}_i\right) d\Omega + \int_{\Gamma_h} \mathbf{W} \cdot \left(\mathbf{F}^{p}_i - \mathbf{F}^{diff}_i\right) n_i \, d\Gamma = 0 \tag{18}$$

for the reduced-energy form. Here, $\tilde{\mathbf{U}}$, $\tilde{\mathbf{F}}_i^{adv/p}$, $\tilde{\mathbf{F}}_i^{p}$, $\tilde{\mathbf{F}}_i^{diff}$ and $\mathbf{U}$, $\mathbf{F}_i^{adv/p}$, $\mathbf{F}_i^{p}$, $\mathbf{F}_i^{diff}$ are assumed to be functions of $\mathbf{Y}$, n_i is the i^{th} component of the outward unit surface normal vector $\mathbf{n}$, and Γ_h is the portion of Γ where the natural boundary conditions are applied. The last term on the left-hand-side of Eqs. (17) and (18) represents the known values of the prescribed traction and heat-flux boundary conditions.

Equations (17) and (18) may be re-written in a quasi-linear form using the pressure-primitive variables $\mathbf{Y}$ defined as

$$\mathbf{Y} = \begin{bmatrix} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{bmatrix}. \tag{19}$$

Given the suitably chosen space of vector-valued trial functions $\mathbf{S}$ for the pressure-primitive variables, the weak form of the Navier–Stokes equations of compressible flows for the full-energy case may be stated as: Find $\mathbf{Y} \in \mathbf{S}$, such that for all $\mathbf{W} \in \mathbf{V}$,

$$\begin{aligned} &\int_\Omega \mathbf{W} \cdot \left(\tilde{\mathbf{A}}_0 \mathbf{Y}_{,t} + \tilde{\mathbf{A}}_i^{adv/p} \mathbf{Y}_{,i} \right) d\Omega \\ &- \int_\Omega \mathbf{W}_{,i} \cdot \left(\tilde{\mathbf{A}}_i^{p} \mathbf{Y} - \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j} \right) d\Omega \\ &+ \int_{\Gamma_h} \mathbf{W} \cdot \left(\tilde{\mathbf{F}}_i^{p} - \tilde{\mathbf{F}}_i^{diff} \right) n_i d\Gamma = 0, \end{aligned} \tag{20}$$

and the reduced-energy case may be stated as: Find $\mathbf{Y} \in \mathbf{S}$, such that for all $\mathbf{W} \in \mathbf{V}$,

$$\begin{aligned} &\int_\Omega \mathbf{W} \cdot \left(\mathbf{A}_0 \mathbf{Y}_{,t} + \mathbf{A}_i^{adv/p} \mathbf{Y}_{,i} + \mathbf{A}_i^{sp} \mathbf{Y}_{,i} \right) d\Omega \\ &- \int_\Omega \mathbf{W}_{,i} \cdot \left(\mathbf{A}_i^{p} \mathbf{Y} - \mathbf{K}_{ij} \mathbf{Y}_{,j} \right) d\Omega \\ &+ \int_{\Gamma_h} \mathbf{W} \cdot \left(\mathbf{F}_i^{p} - \mathbf{F}_i^{diff} \right) n_i \, d\Gamma = 0. \end{aligned} \tag{21}$$

Here, the matrices $\mathbf{A}_0$, $\mathbf{A}_i^{adv/p}$, $\mathbf{A}_i^{p}$, $\mathbf{A}_i^{sp}$, $\mathbf{K}_{ij}$, $\tilde{\mathbf{A}}_0$, $\tilde{\mathbf{A}}_i^{adv/p}$, $\tilde{\mathbf{A}}_i^{p}$, $\tilde{\mathbf{K}}_{ij}$ are defined as

$$\begin{aligned} &\mathbf{A}_0 = \frac{\partial \mathbf{U}}{\partial \mathbf{Y}}, \qquad \mathbf{A}_i^{adv/p} = \frac{\partial \mathbf{F}_i^{adv/p}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{Y}}, \\ &\mathbf{A}_i^{p} = \frac{\partial \mathbf{F}_i^{p}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{Y}}, \qquad \mathbf{A}_i^{sp} = \frac{\mathbf{F}^{sp}}{\mathbf{Y}_{,i}}, \qquad \mathbf{K}_{ij} = \frac{\mathbf{F}_i^{diff}}{\mathbf{Y}_{,j}}, \\ &\tilde{\mathbf{A}}_0 = \frac{\partial \tilde{\mathbf{U}}}{\partial \mathbf{Y}}, \tilde{\mathbf{A}}_i^{adv/p} = \frac{\partial \tilde{\mathbf{F}}_i^{adv/p}}{\partial \tilde{\mathbf{U}}} \frac{\partial \tilde{\mathbf{U}}}{\partial \mathbf{Y}}, \\ &\tilde{\mathbf{A}}_i^{p} = \frac{\partial \tilde{\mathbf{F}}_i^{p}}{\partial \tilde{\mathbf{U}}} \frac{\partial \tilde{\mathbf{U}}}{\partial \mathbf{Y}}, \qquad \tilde{\mathbf{K}}_{ij} = \frac{\tilde{\mathbf{F}}_i^{diff}}{\mathbf{Y}_{,j}}, \end{aligned} \tag{22}$$

and the explicit expressions for the above matrices are given in "Appendix A".

2.3 SUPG and DC operators

We assume the fluid domain Ω is divided into N_{el} elements each denoted by Ω^e and denote by $\mathbf{Res}(\mathbf{Y})$ the strong-form residual of the Navier–Stokes equations of compressible flows. We define the SUPG operator as follows:

$$B_{\text{SUPG}}(\mathbf{W}, \mathbf{Y}) = \sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\left(\mathbf{A}_i^* \right)^{\mathrm{T}} \mathbf{W}_{,i} \right) \cdot \boldsymbol{\tau}_{\text{SUPG}} \mathbf{Res}(\mathbf{Y}) \, d\Omega. \tag{23}$$

Here, $\boldsymbol{\tau}_{\text{SUPG}}$ is the stabilization matrix for the pressure-primitive variables given by [25,84]

$$\boldsymbol{\tau}_{\text{SUPG}} = \mathbf{A}_0^{-1} \hat{\boldsymbol{\tau}}_{\text{SUPG}}, \tag{24}$$

where $\hat{\boldsymbol{\tau}}_{\text{SUPG}}$ is the stabilization matrix for conservation variables defined as [54,84]

$$\hat{\boldsymbol{\tau}}_{\text{SUPG}} = \left(\frac{4}{\Delta t^2} \mathbf{I} + G_{ij} \hat{\mathbf{A}}_i^* \hat{\mathbf{A}}_j^* + C_I \, G_{ij} G_{kl} \hat{\mathbf{K}}_{ik}^* \hat{\mathbf{K}}_{jl}^* \right)^{-\frac{1}{2}}. \tag{25}$$

Here Δt is the time step size, C_I is a positive constant derived from an appropriate element-wise inverse estimate [37], and G_{ij} are the components of the element metric tensor $\mathbf{G}$ defined as

$$G_{ij} = \sum_{k=1}^{d} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}, \tag{26}$$

where $\mathbf{x}(\boldsymbol{\xi})$ is the element isoparametric mapping. In the above expressions the definitions of $\mathbf{A}_i^*$ and its conservation-variable counterpart $\hat{\mathbf{A}}_i^*$ depend on the formulation employed. In the full-energy case we set $\mathbf{A}_i^* = \tilde{\mathbf{A}}_i^{adv/p} + \tilde{\mathbf{A}}_i^{p}$, while in the reduced-energy case we set $\mathbf{A}_i^* = \mathbf{A}_i^{adv/p} + \mathbf{A}_i^{p} + \mathbf{A}_i^{sp}$. Likewise, $\mathbf{K}_{ij}^* = \tilde{\mathbf{K}}_{ij}$ for the full-energy case, $\mathbf{K}_{ij}^* = \mathbf{K}_{ij}$ for the reduced-energy case, and $\hat{\mathbf{K}}_{ij}^*$ is their conservation-variable counterpart. The definition of $\hat{\boldsymbol{\tau}}_{\text{SUPG}}$ in Eq. (25) requires computing the square-root-inverse of a 5×5 matrix in 3D, which is computed numerically using the Denman–Beavers algorithm [17,65,84].

Following the approach for the SUPG operator design, we first define the DC operator for conservation variables as

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}_{,i} \cdot \hat{\mathbf{K}}_{\text{DC}} \mathbf{U}_{,i} \, d\Omega, \tag{27}$$

where $\hat{\mathbf{K}}_{\mathrm{DC}}$ is the matrix-valued DC viscosity. Changing variables from $\mathbf{U}$ to $\mathbf{Y}$ gives

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}_{,i} \cdot \hat{\mathbf{K}}_{\mathrm{DC}} \mathbf{A}_0 \mathbf{Y}_{,i} \, d\Omega, \tag{28}$$

which, in turn, defines the DC viscosity matrix for the pressure-primitive variables:

$$\mathbf{K}_{\mathrm{DC}} = \hat{\mathbf{K}}_{\mathrm{DC}} \mathbf{A}_0. \tag{29}$$

We assume a diagonal form of the DC viscosity matrix for the conservation variables, namely,

$$\hat{\mathbf{K}}_{\mathrm{DC}} = \mathrm{diag}\left(\hat{\kappa}_{\mathrm{C}}, \hat{\kappa}_{\mathrm{M}}, \hat{\kappa}_{\mathrm{M}}, \hat{\kappa}_{\mathrm{M}}, \hat{\kappa}_{\mathrm{E}}\right), \tag{30}$$

and provide two definitions of the diagonal entries in what follows.

- We refer to this version as DC_1, which was proposed in [84]:

$$\hat{\kappa}_{\mathrm{C}} = C_{\mathrm{C}} \frac{h \, |\mathbf{Res}_1|}{|\nabla \mathbf{U}_1|}, \tag{31}$$

$$\hat{\kappa}_{\mathrm{M}} = C_{\mathrm{M}} \frac{h \, |\mathbf{Res}_{2:d+1}|}{|\nabla \mathbf{U}_{2:d+1}|}, \tag{32}$$

$$\hat{\kappa}_{\mathrm{E}} = C_{\mathrm{E}} \frac{h \, |\mathbf{Res}_{d+2}|}{|\nabla \mathbf{U}_{d+2}|}. \tag{33}$$

- We refer to this version as DC_2, which was proposed in [7]:

$$\hat{\kappa}_{\mathrm{C}} = \min\left(C_{\mathrm{C}} \frac{h \, |\mathbf{Res}|_w}{|\nabla \mathbf{U}|_w}, \hat{\kappa}_{\mathrm{cap}}\right), \tag{34}$$

$$\hat{\kappa}_{\mathrm{M}} = \min\left(C_{\mathrm{M}} \frac{h \, |\mathbf{Res}|_w}{|\nabla \mathbf{U}|_w}, \hat{\kappa}_{\mathrm{cap}}\right), \tag{35}$$

$$\hat{\kappa}_{\mathrm{E}} = \min\left(C_{\mathrm{E}} \frac{h \, |\mathbf{Res}|_w}{|\nabla \mathbf{U}|_w}, \hat{\kappa}_{\mathrm{cap}}\right). \tag{36}$$

In the above definitions, h is the element size and C_{C}, C_{M}, and C_{E} are the $\mathcal{O}(1)$ positive constants corresponding to the continuity, momentum, and energy equations, respectively. Furthermore, in the definition of DC_2, $|\mathbf{Res}|_w$ is a weighted norm of the compressible-flow equation residual

$$|\mathbf{Res}|_w = c^2 \, |\mathbf{Res}_1| + \|\mathbf{u}\| \, |\mathbf{Res}_{2:d+1}| + |\mathbf{Res}_{d+2}|, \tag{37}$$

$|\nabla \mathbf{U}|_w$ is a weighted norm of the conservation-variable solution gradient

$$|\nabla \mathbf{U}|_w = c^2 \, |\nabla \mathbf{U}_1| + \|\mathbf{u}\| \, |\nabla \mathbf{U}_{2:d+1}| + |\nabla \mathbf{U}_{d+2}|, \tag{38}$$

$\hat{\kappa}_{\mathrm{cap}}$ is the maximum allowable value of the DC viscosity

$$\hat{\kappa}_{\mathrm{cap}} = \left(\left(\mathbf{u} - \hat{\mathbf{u}}\right) \cdot \mathbf{G}^{-1} \left(\mathbf{u} - \hat{\mathbf{u}}\right) + c^2 \mathrm{tr}(\mathbf{G}^{-1}) \right)^{\frac{1}{2}}, \tag{39}$$

c is the sound speed, and $\mathbf{G}^{-1}$ is the inverse of the element metric tensor.

Remark 1 The DC viscosity definitions originate from the so-called CAU DC technique [3]. It may also be viewed as YZβ DC with $\beta = 1$. We also note that the CAU DC technique is an extension of the "δ_{91}" shock-capturing technique [45,46] to unsteady flows. The CAU is residual-based in the context of unsteady problems, while "δ_{91}" was residual-based in the context of steady, inviscid flows.

Remark 2 Note that in the definition of DC_2 the weighted norm appropriately scales the components of the residual and solution gradient in order to measure the size of these vectors that have entries of different dimensions.

Remark 3 Note the introduction of a DC viscosity upper bound or "cap", $\hat{\kappa}_{\mathrm{cap}}$, which is a multi-dimensional generalization of the upwind viscosity $h(u + c)/2$. While it is expected that, on average, the residual-based definition of the DC viscosity will stay well below the upwind limit, division by the gradient norm can lead to local spikes in this quantity, which are mitigated by the cap. The introduction of the cap reduces the degree of nonlinearity associated with the DC terms and, as a result, improves convergence of the Newton–Rhapson iterations. The cap idea was introduced and successfully employed in [6] for shock-capturing for solids.

2.4 Complete formulation and solution procedure

The final semi-discrete SUPG formulation for compressible flows supplemented by a DC operator is obtained by simply adding the Galerkin, SUPG, and DC terms. The formulation for the full-energy case becomes: Find $\mathbf{Y}^h \in \mathbf{S}^h$, such that for all $\mathbf{W}^h \in \mathbf{V}^h$,

$$\begin{aligned}
&\int_{\Omega} \mathbf{W}^h \cdot \left(\tilde{\mathbf{A}}_0 \mathbf{Y}^h_{,t} + \tilde{\mathbf{A}}_i^{adv/p} \mathbf{Y}^h_{,i} \right) d\Omega \\
&- \int_{\Omega} \mathbf{W}^h_{,i} \cdot \left(\tilde{\mathbf{A}}_i^{p} \mathbf{Y}^h - \tilde{\mathbf{K}}_{ij} \mathbf{Y}^h_{,j} \right) d\Omega \\
&+ \sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\left(\mathbf{A}_i^*\right)^{\mathrm{T}} \mathbf{W}^h_{,i} \right) \cdot \boldsymbol{\tau}_{\mathrm{SUPG}} \mathbf{Res}(\mathbf{Y}^h) \, d\Omega \\
&+ \sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}^h_{,i} \cdot \mathbf{K}_{\mathrm{DC}} \mathbf{Y}^h_{,i} \, d\Omega \\
&+ \int_{\Gamma_h} \mathbf{W}^h \cdot \left(\tilde{\mathbf{F}}_i^{p} - \tilde{\mathbf{F}}_i^{diff} \right) n_i d\Gamma = 0,
\end{aligned} \tag{40}$$

and for the reduced-energy case becomes: Find $\mathbf{Y}^h \in \mathbf{S}^h$, such that for all $\mathbf{W}^h \in \mathbf{V}^h$,

$$\begin{aligned}
&\int_\Omega \mathbf{W}^h \cdot \left(\mathbf{A}_0 \mathbf{Y}^h_{,t} + \mathbf{A}_i^{adv/p} \mathbf{Y}^h_{,i} + \mathbf{A}_i^{sp} \mathbf{Y}^h_{,i}\right) d\Omega \\
&- \int_\Omega \mathbf{W}^h_{,i} \cdot \left(\mathbf{A}_i^p \mathbf{Y}^h - \mathbf{K}_{ij} \mathbf{Y}^h_{,j}\right) d\Omega \\
&+ \sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\left(\mathbf{A}_i^*\right)^{\mathrm{T}} \mathbf{W}^h_{,i}\right) \cdot \boldsymbol{\tau}_{\mathrm{SUPG}} \mathbf{Res}(\mathbf{Y}^h)\, d\Omega \\
&+ \sum_{e=1}^{N_{el}} \int_{\Omega^e} \mathbf{W}^h_{,i} \cdot \mathbf{K}_{\mathrm{DC}} \mathbf{Y}^h_{,i}\, d\Omega \\
&+ \int_{\Gamma_h} \mathbf{W}^h \cdot \left(\mathbf{F}_i^p - \mathbf{F}_i^{diff}\right) n_i\, d\Gamma = 0.
\end{aligned} \tag{41}$$

The generalized-α method in [36] is used to integrate the resulting systems in time. At each time step a non-linear system of equations is solved using the Newton–Raphson method.

Remark 4 The resulting linear equation systems at each non-linear iteration is solved to a tolerance using the Generalized Minimal Residual (GMRES) technique [53] with block-diagonal preconditioning. Nodal blocks (4×4 in 2D and 5×5 in 3D) are extracted from the left-hand-side matrix, **LU** factorization is performed for each nodal block, and the linear equation system is left- and right-preconditioned using the block-diagonal $\mathbf{L}^{-1}$ and $\mathbf{U}^{-1}$ matrices, respectively. This simple choice leads to an efficient linear solver with excellent parallel scaling.

3 Numerical results

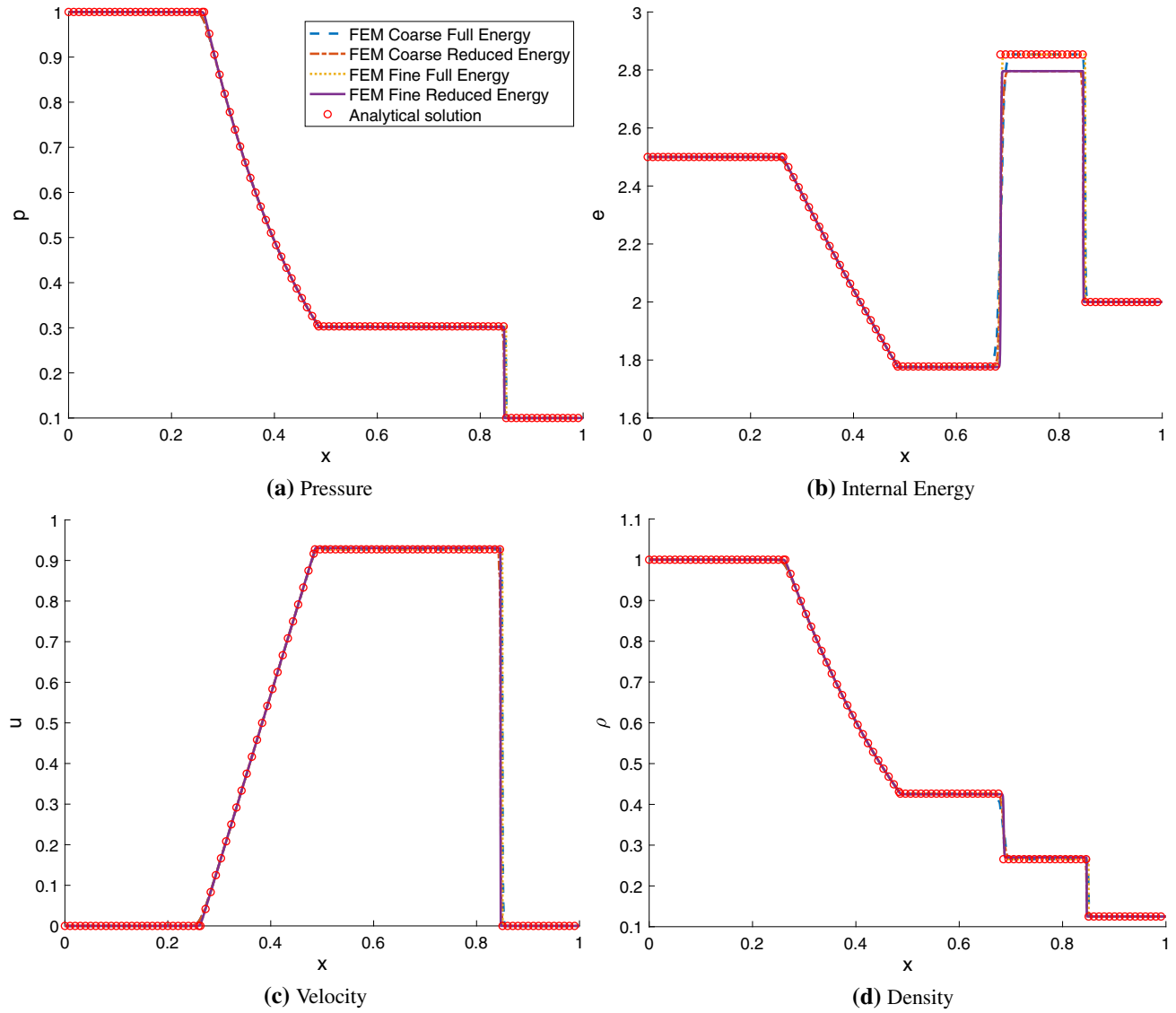
In this section numerical examples of hypersonic flow simulations are presented. The main purpose of these numerical test cases is to show the robustness and accuracy of the formulation in applications of hypersonic flow regimes. The 1D Sod shock tube problem [55] is the first case selected to perform a detailed comparison between the full and reduced energy formulations. Next, the results of a 2D Mach 6 flat plate [10] problem with detailed mesh convergence study, are presented. A Mach 10 NASA wedge test case [4] is shown, followed by the Mach 17 flow over a cylinder problem [48] to investigate the instabilities, namely "Carbuncle problem" [21], encountered when traditional CAU DC operator, DC_1, is used. An alternative definition of DC operator, DC_2, that alleviates this problem, is proposed. The Section concludes with 3D simulation of Viking Lander Capsule [20] and results comparisons to NASA LAURA code and wind tunnel experiment. In the following examples the DC_1 operator is used, unless stated otherwise.

3.1 1D Sod shock tube case

The shock tube case is a 1D, inviscid and unsteady case [55]. An initial diaphragm separates two regions of the same fluid at rest and at different pressure and density. After the instantaneous removal of the diaphragm a shock wave propagates in the low pressure region while an expansion wave propagates in the opposite direction. The contact surface across which pressure and velocity stay the same, propagates in the same direction of the shock wave. In order to simulate this case, the following initial conditions have been adopted [55]: $\rho_L = 1\,[\mathrm{kg/m^3}]$, $p_L = 1\,[\mathrm{Pa}]$, $u_L = 0\,[\mathrm{m/s}]$, $T_L = 3.484\mathrm{e}^{-3}\,[\mathrm{K}]$ and $\rho_R = 0.125\,[\mathrm{kg/m^3}]$, $p_R = 0.1\,[\mathrm{Pa}]$, $u_R = 0\,[\mathrm{m/s}]$, $T_R = 2.787\mathrm{e}^{-3}\,[\mathrm{K}]$, where subscript L and R represent left and right regions with respect to the diaphragm. The specific heat ratio is $\gamma = 1.4$ and the gas constant is $R = 287\,[\mathrm{J/(kg\,K)}]$. The length of the domain is $L = 1\,[\mathrm{m}]$ and two sets of meshes are investigated, i.e. the coarse mesh of 10^3 elements and the fine one of 10^4 elements. For each mesh, the simulations are carried out using both the full and the reduced energy formulations. The results obtained from the four simulations at the time instant $t = 0.1\,[\mathrm{s}]$, are compared with the analytical solution and shown in Fig. 1. In Fig. 1 the internal energy is computed as $e = c_v T$ where c_v is the specific heat coefficient at constant volume.

Even though it cannot be fully appreciated in Fig. 1 because of the scale of the axis, the full energy formulation gives more accurate estimates for all flow quantities in all the regions. This is more evident in Fig. 1b, where the reduced energy formulation underestimates the value of the internal energy, e, in the post-shock region of the domain. Figure 2 illustrates in detail the main features of the density distribution, in particular the front and the tail of the expansion fan, shown in Fig. 2a, b respectively, the contact surface in Fig. 2c and the shock wave in Fig. 2d.

Both formulations perform very well in predicting the location of the front wave of the expansion fan, showing an excellent agreement with the analytical solution for the fine mesh. Figure 2b shows that the reduced energy formulation underpredicts the density value in the post-expansion region, while the full energy formulation agrees with the analytic solution. The location of the contact surface is correctly predicted by both formulations, but it can be seen in Fig. 2c that the full energy formulation is more diffusive than the reduced energy formulation. In Fig. 2c, d, it is evident that the reduced energy formulation overpredicts the density value in the region between the contact surface and the shock wave, while once again the full energy formulation predictions are in excellent agreement with the analytical solution. Finally, Fig. 2d shows that the shock wave location is correctly predicted by the full energy formulation, while the reduced energy underpredicts the shock location with a relative error with respect to the analytical value of 0.4%. In



(a) Pressure **(b)** Internal Energy

(c) Velocity **(d)** Density

Fig. 1 Comparison of the numerical results for pressure (**a**), internal energy (**b**), velocity (**c**) and density (**d**) distributions with the analytical solution of the Sod shock tube case [55]

general, the fine mesh results show less diffusive discontinuities (contact surface and shock wave), with respect to the coarse mesh results.

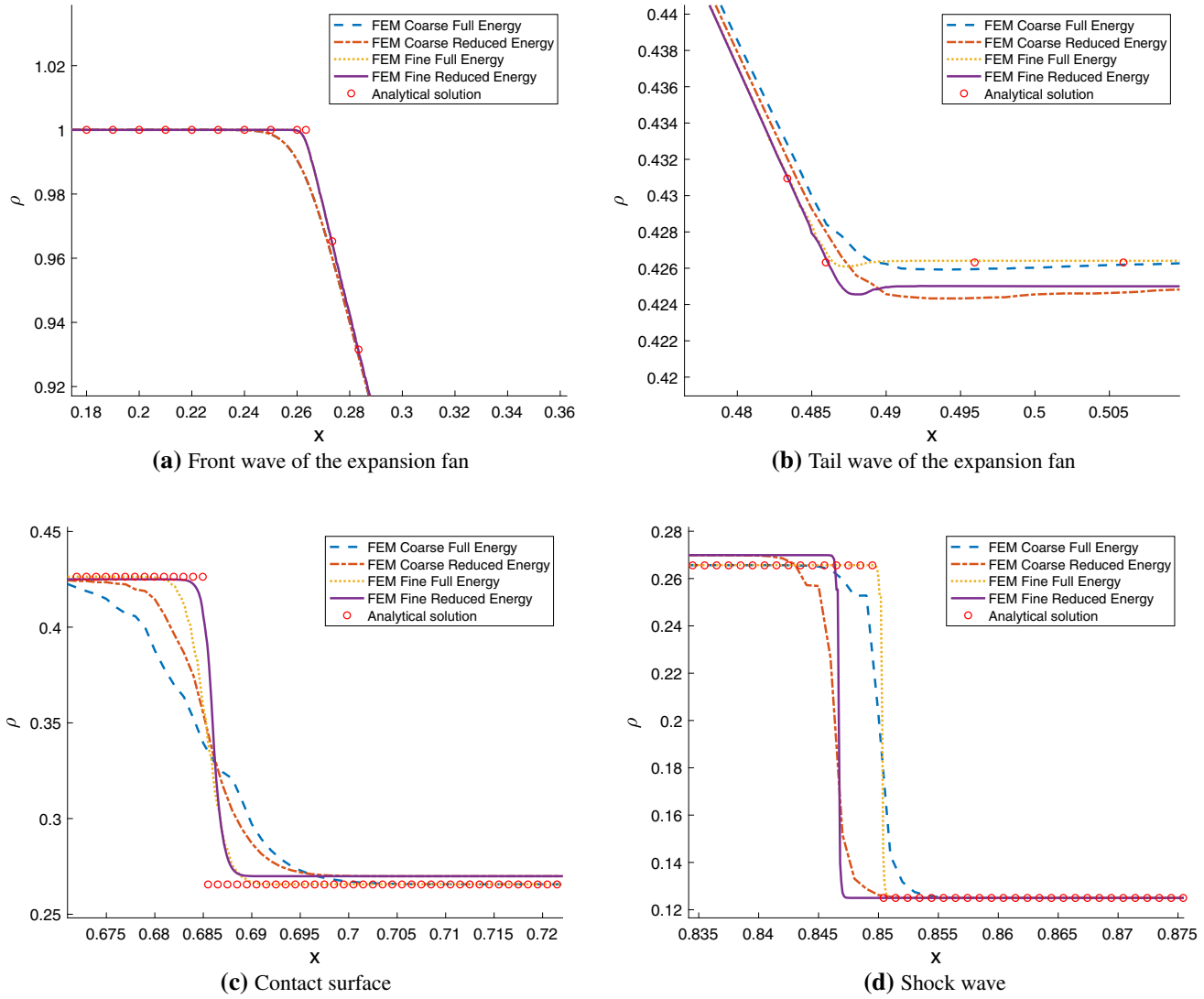
The stabilized formulations, described in the previous section, can handle the unsteady discontinuities in the field, such as moving shock waves and contact surfaces. In terms of accuracy the full energy formulation is superior to the reduced energy one. Moreover, the smoothness of the solution in the entire field shows the effectiveness of the SUPG and DC stabilization operators.

3.2 2D Mach 6 flat plate

The flat plate example is one of the fundamental benchmark cases for code validation. The physics behind this case is well understood. When a viscous flow interacts with a flat plate at zero incidence angle, a boundary layer develops from the leading edge and grows in thickness downstream. The displacement effect of the boundary layer deflects the outer inviscid flow away from the wall. When the flow is supersonic, a weak oblique shock wave forms at the leading edge as a result of the displacement.

This case is based on the work presented in [10]. The computational domain, illustrated in Fig. 3, extends for 1 [m] in the x direction (free stream direction) and for 0.3 [m] in the vertical direction.

At the inflow and top boundaries, the free stream conditions are prescribed as $p_\infty = 5.17$ [Pa], $u_\infty = 849.34$ [m/s] (parallel to the wall) and $T_\infty = 48.89$ [K], that correspond to $M_\infty = 6$ and $\mathrm{Re}_{\infty,L} = 10^4$, which is the local Reynolds



(a) Front wave of the expansion fan

(b) Tail wave of the expansion fan

(c) Contact surface

(d) Shock wave

Fig. 2 Details of the main features of the density distribution for the shocktube case [55]: front wave of the expansion fan (**a**), tail wave of the expansion fan (**b**), contact surface (**c**) and shock wave (**d**). The analytical solution is represented by red circles and is shown for comparison

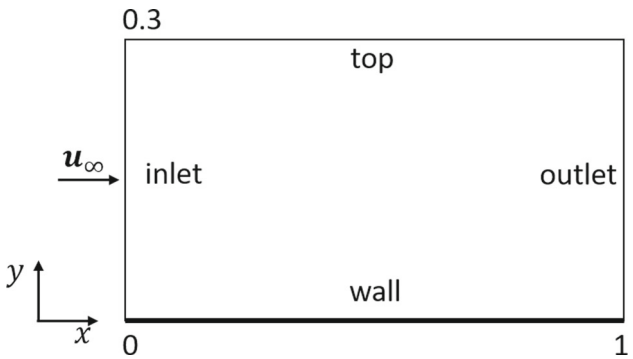


Fig. 3 Computational domain for the Mach 6 flat plate case with free stream velocity at zero angle of attack

number at the reference length of 0.1 [m]. At the wall the no slip condition for the velocity and a constant temperature $T_w = T_{aw} = 353.57$ [K] are prescribed, where

$$T_{aw} = T_\infty \left(1 + \frac{\gamma - 1}{2}\sqrt{Pr}M_\infty^2\right) \tag{42}$$

is the adiabatic temperature computed with a recovery factor equal the square root of Prandtl number. At the outlet, zero traction and zero heat transfer are prescribed. The following Sutherland law is used for the viscosity computation:

$$\mu = \mu_r \frac{T_r + S}{T_\infty + S}\left(\frac{T_\infty}{T_r}\right)^{1.5} \tag{43}$$

where $\mu_r = 1.716 \cdot 10^{-5}$ [Pa s], $T_r = 273.15$ [K] and $S = 110.4$ [K]. The thermal conductivity is $k = \mu c_p / Pr$ with $Pr = 0.72$, while the specific heat ratio is $\gamma = 1.4$. The

simulation was carried out using a constant time step $\Delta t = 5 \cdot 10^{-6}$ [s], which corresponds to a maximum CFL of 4.

3.2.1 Grid convergence analysis

Together with the validation, the verification of a numerical framework is essential. It is important to verify that the partial differential equations of the model describing the physical problem are solved correctly. The verification process involves getting estimates of the numerical uncertainties or numerical errors related to the discretization methods. Indeed, the solution should be independent of the grid and theoretically the exact solution (solution of the continuum equations) should be obtained when the grid spacing approaches zero value. In this section numerical uncertainties, in terms of grid convergence index (GCI), are presented following the guidelines in [9]. Three different uniform structured grids are considered, a coarse grid of 107×32 elements, a medium grid of 214×64 elements and a fine one of 428×128 elements. Since the meshes are uniform with equal grid spacing in both directions x and y, it is easy to define the grid size h which is simply the length of one cell. The grid refinement factor for the coarse-medium meshes is computed as $r_{32} = h_3/h_2$ and for the medium-fine meshes $r_{21} = h_2/h_1$, being $h_1 < h_2 < h_3$. In the case presented here $r_{32} = r_{21} = 2$. The apparent order of convergence p can be calculated as:

$$p = \frac{1}{\log(r_{21})} |\log|\epsilon_{32}/\epsilon_{21}| + q(p)| \tag{44}$$

$$q(p) = \log\left(\frac{r_{21}^p - s}{r_{32}^p - s}\right) \tag{45}$$

$$s = sign\left(\frac{\epsilon_{32}}{\epsilon_{21}}\right) \tag{46}$$

where $\epsilon_{32} = \phi_3 - \phi_2$ and $\epsilon_{21} = \phi_2 - \phi_1$, and ϕ_n refers to the solution relative to the nth grid. The solution selected to carry out the grid convergence analysis is the wall pressure p_w. It is possible to calculate the extrapolated value from the solutions of the fine and medium grids, ϕ_{21}^{ext}, and from solutions of medium and coarse grids, ϕ_{32}^{ext}, as follows:

$$\phi_{21}^{ext} = \frac{r_{21}^p \phi_1 - \phi_2}{r_{21}^p - 1} \tag{47}$$

$$\phi_{32}^{ext} = \frac{r_{32}^p \phi_2 - \phi_3}{r_{32}^p - 1} \tag{48}$$

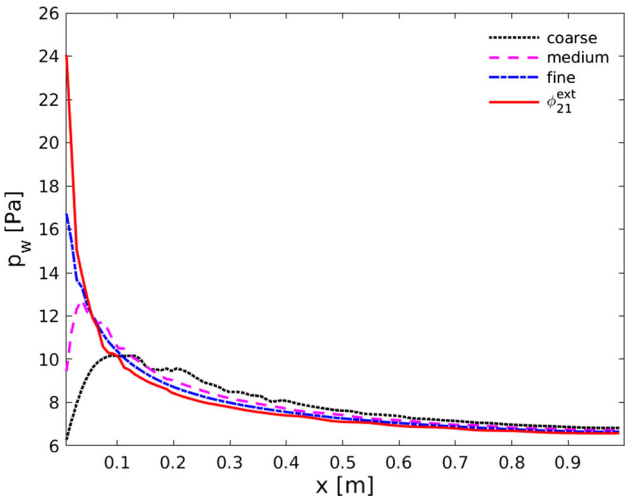


Fig. 4 Wall pressure distribution for the coarse, medium and fine grid together with the extrapolated value ϕ_{21}^{ext}

Finally from the approximate relative errors:

$$e_a^{21} = \left|\frac{\phi_1 - \phi_2}{\phi_1}\right| \tag{49}$$

$$e_a^{32} = \left|\frac{\phi_2 - \phi_3}{\phi_2}\right| \tag{50}$$

The fine and medium grid convergence indexes can be computed as below:

$$GCI_{fine}^{21} = \frac{F_s e_a^{21}}{r_{21}^p - 1} \tag{51}$$

$$GCI_{medium}^{32} = \frac{F_s e_a^{32}}{r_{32}^p - 1} \tag{52}$$

where $F_s = 1.25$ is the safety factor. The grid convergence indexes are estimates of the numerical uncertainties. The average (global) GCI^g values and the average order of convergence p^g are obtained by averaging over all the local values. Applying the procedure just described to the Mach 6 flat plate case, the following values are calculated: $p^g \sim 1$, $GCI_{fine} = 3.94\%$ and $GCI_{medium} = 6.28\%$. The wall pressure distribution for the three grids is presented in Fig. 4. The local GCI_{fine}^{21} (computed for every node of the plate) is used to represent the numerical uncertainties as error bars on the plot of the wall pressure distribution for the fine grid. The result is shown in Fig. 5. It can be observed that the approximate numerical error is large in the leading edge region, close to the leading edge which is therefore a critical point in the numerical analysis of this case.

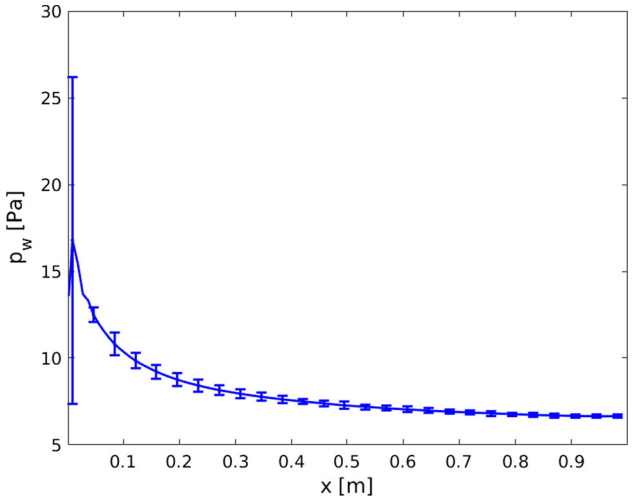


Fig. 5 Wall pressure distribution for the fine grid with numerical uncertainties reported as error bars

3.2.2 Numerical results

Following the grid convergence study, the fine mesh is selected for simulations (see Fig. 6). The grid spacing $\Delta x \sim 0.0023$ [m] is uniform in the x direction, while a wall refinement is adopted in the y direction with a minimum grid spacing $\Delta y = 5{\cdot}10^{-4}$ [m] at the wall. First, the Mach number contours are observed. The compression wave due to the viscous boundary layer at the plate obtained from the full energy formulation can be seen in Fig. 7. The reduced energy formulation produces similar qualitative results, showing smooth steady-state solution throughout the whole domain. The normalized Mach profile at the station $x = 0.75$ [m] is shown for both formulations in Fig. 8. The Mach profile is normalized by the Mach number at the edge of the boundary layer, M_e. The Mach number at the edge of the boundary layer predicted by the full energy formulation is $M_e = 5.65$, while the one predicted by the reduced energy formulation is $M_e = 5.7$. The spatial coordinate normal to the wall, y, is normalized by the boundary layer thickness, which is $\delta = 0.056$ [m] and $\delta = 0.055$ [m] for the full and reduced energy formulations respectively. The results in Fig. 8 show good agreement with the experiment data obtained by [47] except very close to the wall due to technical difficulties in the velocity measurements in the wind tunnel. The comparison of the skin friction coefficient, c_f, distribution along the plate with the numerical results obtained using the finite difference method by [10], is presented in Fig. 9. The results from both formulations show an excellent agreement. Finally the non dimensional wall pressure distribution, p_w/p_∞, is presented in Fig. 10. Again the comparison with the numerical results in [10] shows the robustness and accuracy of the two stabilized formulations presented in this work at high Mach numbers.

3.3 2D Mach 10 oblique shock problem

The oblique shock case presented in this section involves a flat plate and a flow hitting the plate with 5° incidence. The experimental test was performed at NASA Langley's 31-inch Mach 10 facility [16]. The free stream flow is characterized by Mach number $M_\infty = 9.7$ and a Reynolds number based on free stream values of $Re_\infty = 3.8 \cdot 10^5$. A weak oblique shock wave forms at the leading edge of the wedge. The computational domain corresponds to the flow region along the top surface of the wedge and is shown in Fig. 11.

The domain is 0.2 [m] long and 0.03 [m] high and it is discretized with a structured mesh of 1000 × 150 elements shown in Fig. 12.

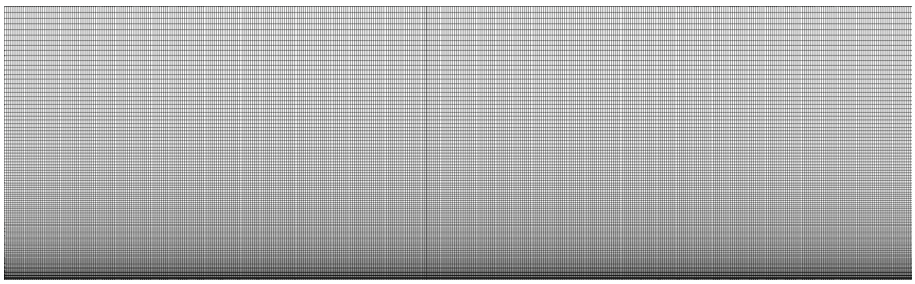


Fig. 6 Discretization of the computational domain for the flat plate case at Mach 6 with a structure mesh

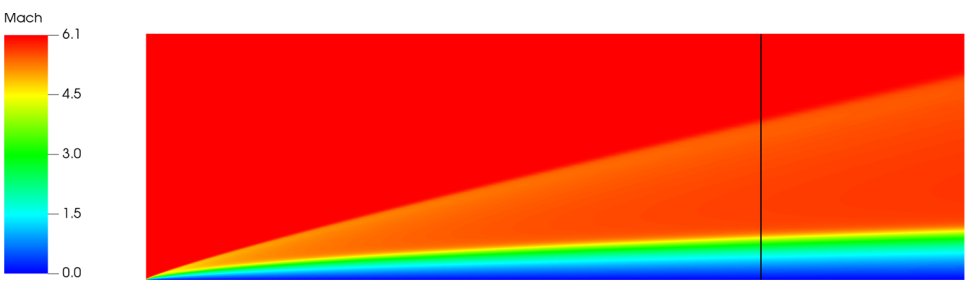


Fig. 7 Mach number contours for the Mach 6 flat plate case computed using the full energy formulation. The black line indicates the station $x = 0.75$ [m]

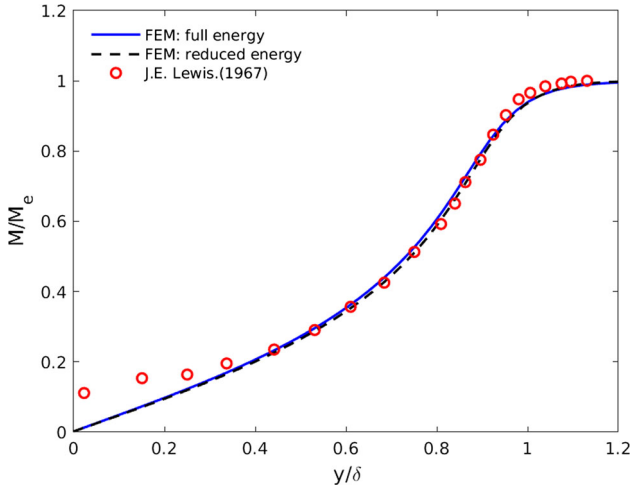


Fig. 8 The normalized Mach profile at $x = 0.75$ [m] compared to the experiment data obtained by [47]

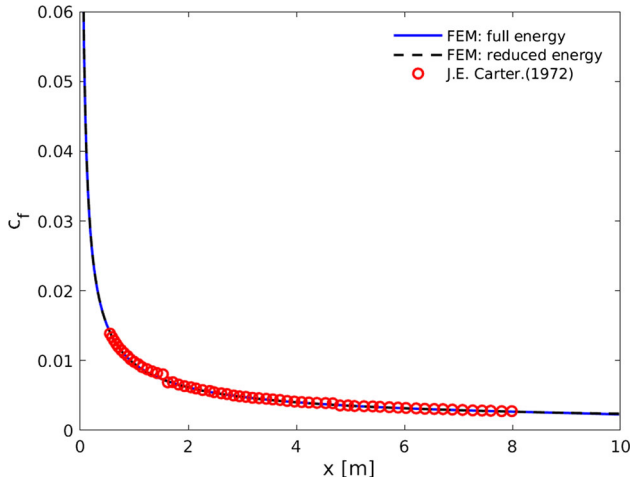


Fig. 9 The skin friction coefficient distribution along the wall is compared to numerical results of [10]. The skin friction coefficient is defined as: $c_f = 2\tau_w / (\rho_\infty \|u_\infty\|)$

The grid is refined in both directions, x and y, at the wall and at the leading edge of the plate. The minimum grid spacing in x and y is $\Delta x = \Delta y = 2.5 \cdot 10^{-5}$ [m]. Defining u the velocity component parallel to the plate and v the velocity component normal to the wall, the free stream conditions prescribed at the inlet and top boundaries of the domain are: $u_\infty = 1401.94$ [m/s], $v_\infty = -122.65$ [m/s], $p_\infty = 68.4$ [Pa] and $T_\infty = 52.3$ [K]. The no slip conditions and a constant temperature $T_w = 314$ [K] are prescribed at the wall. Zero traction and zero heat flux are prescribed at the outlet. For this case thermodynamic properties of real gas are adopted (see [27]), while the transport properties are calculated from the kinetic theory models. In particular the viscosity is computed from the Chapman-Cowling relation [15]:

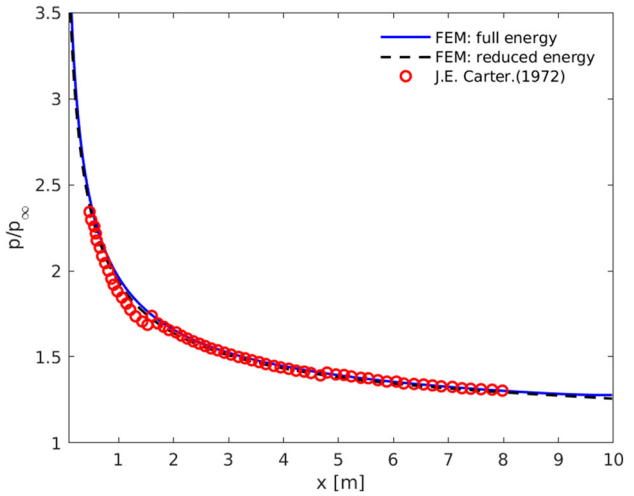


Fig. 10 The wall pressure distribution non dimensionalized by the free stream pressure (p_w / p_∞) is compared to the numerical results obtained by [10]

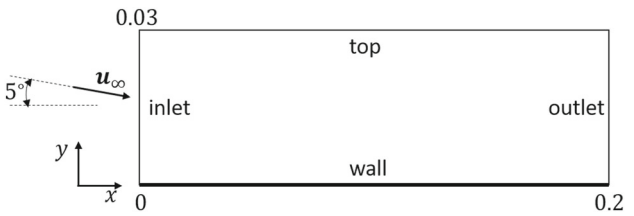


Fig. 11 Computational domain for the Mach 9.7 oblique shock case

$$\mu = 2.6693 \cdot 10^{-6} \frac{\sqrt{MT}}{\sigma^2 \Omega} \, [Pa \cdot s] \tag{53}$$

where $\sigma = 3.689$ [Å] [27] is collision diameter for air and Ω are the collision integrals computed using empirical equations for the Lennard–Jones (12-6) potential [49]. The thermal conductivity is calculated from the kinetic theory models in [26] with the modified Eucken correction as below:

$$k = \mu \left(\frac{15}{4} + 1.32 \left(\frac{c_p}{R} - \frac{5}{2} \right) \right) R \tag{54}$$

The Mach contours for full and reduced energy formulations are shown in Fig. 13a, b respectively.

Figure 14 shows the Mach profile at $x = 0.106$ [m] where the results are compared with simulation results using OpenFOAM [4].

The results from the full energy formulation show very good agreement. Some discrepancies, however, can be observed in terms of the shock wave angle. The shock wave angle predicted by the full energy formulation is $\beta_s \sim 7°$, while the angle predicted by the reduced energy formulation is $\beta_s \sim 6.3°$. The angle prediction of the full energy formu-

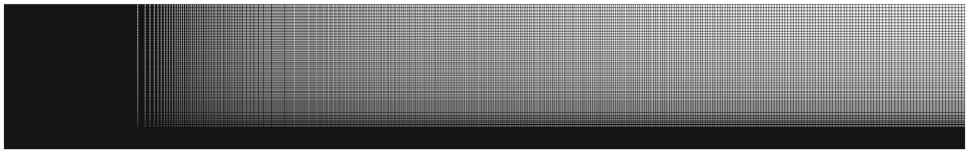
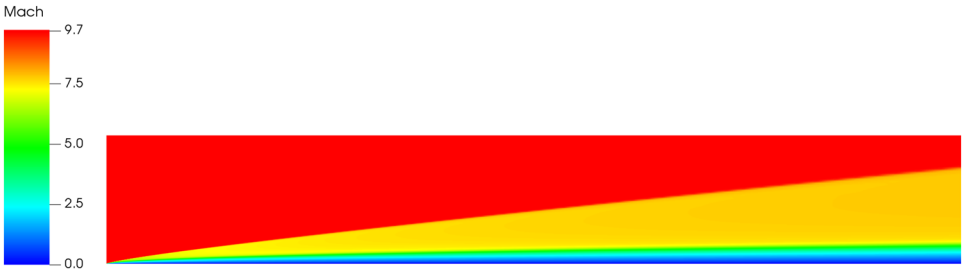
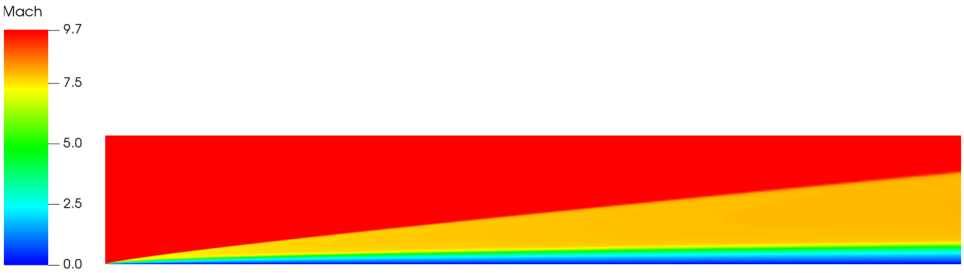


Fig. 12 Discretization of the computational domain for the oblique shock example at Mach 9.7 with a structured mesh



(a) Full energy formulation.



(b) Reduced energy formulation.

Fig. 13 Mach number contours for the Mach 9.7 oblique shock case predicted by the full (**a**) and the reduced (**b**) energy formulation

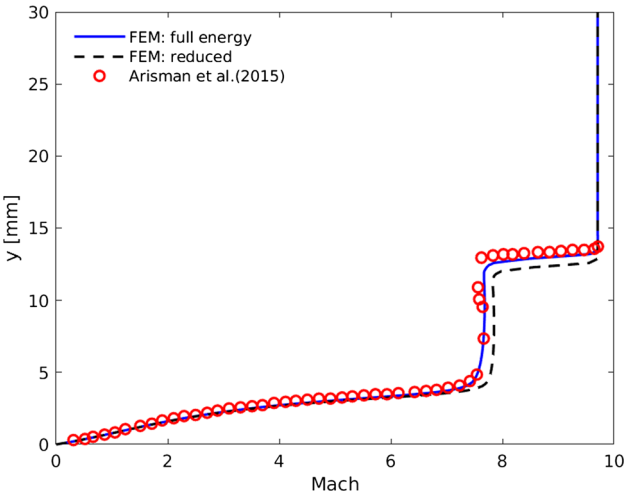


Fig. 14 Mach profile at $x = 0.106$ [m] obtained using both formulations is compared with numerical results of [4] for the Mach 9.7 oblique shock

lation is very close to the one predicted by OpenFOAM [4], which is $\beta_a \sim 7.2°$, while the reduced energy formulation underpredicts this value. The smaller shock wave angle predicted by the reduced energy formulation is the reason behind the higher Mach number predicted in the post-shock region. The velocity profiles in the boundary layer at four different locations, $x = 0.588,\ 0.784,\ 0.91,\ 1.068$ [m] are compared with the experiment data [16] in Fig. 15. The results obtained by both formulations are in very good agreement with the experiment data.

3.4 2D Mach 17.6 cylinder case

The two-dimensional cylinder case [48] described in this section is a hypersonic benchmark case that has the goal to show the robustness of the numerical framework in handling a strong bow shock and the accuracy of the predictions under such extreme flow conditions. The free stream flow conditions are defined by a Mach number $M = 17.6$ and a Reynolds number $Re = 3.769 \cdot 10^6$. For hypersonic flow past a cylinder, which is a blunt body, a strong, detached shock wave develops and once the steady state is reached, the shock stand-off distance can be measured. The inflow of the computational domain is represented by an arc centered at the cylinder's center. The domain is 10 [m] long and 3.5 [m] high. The computational domain is illustrated in Fig. 16.

The cylinder radius is 1 [m]. The domain is discretized with a structured mesh of 60×64 elements in the azimuthal and radial directions respectively. The mesh is uniform in the azimuthal direction, while it is refined close to the wall with the first wall element's height being 6 [mm]. The mesh is shown in Fig. 17.

The free stream velocity, pressure and temperature are $v_\infty = -4.9906 \cdot 10^3$ [m/s], $p_\infty = 576.2$ [Pa] and $T_\infty =$

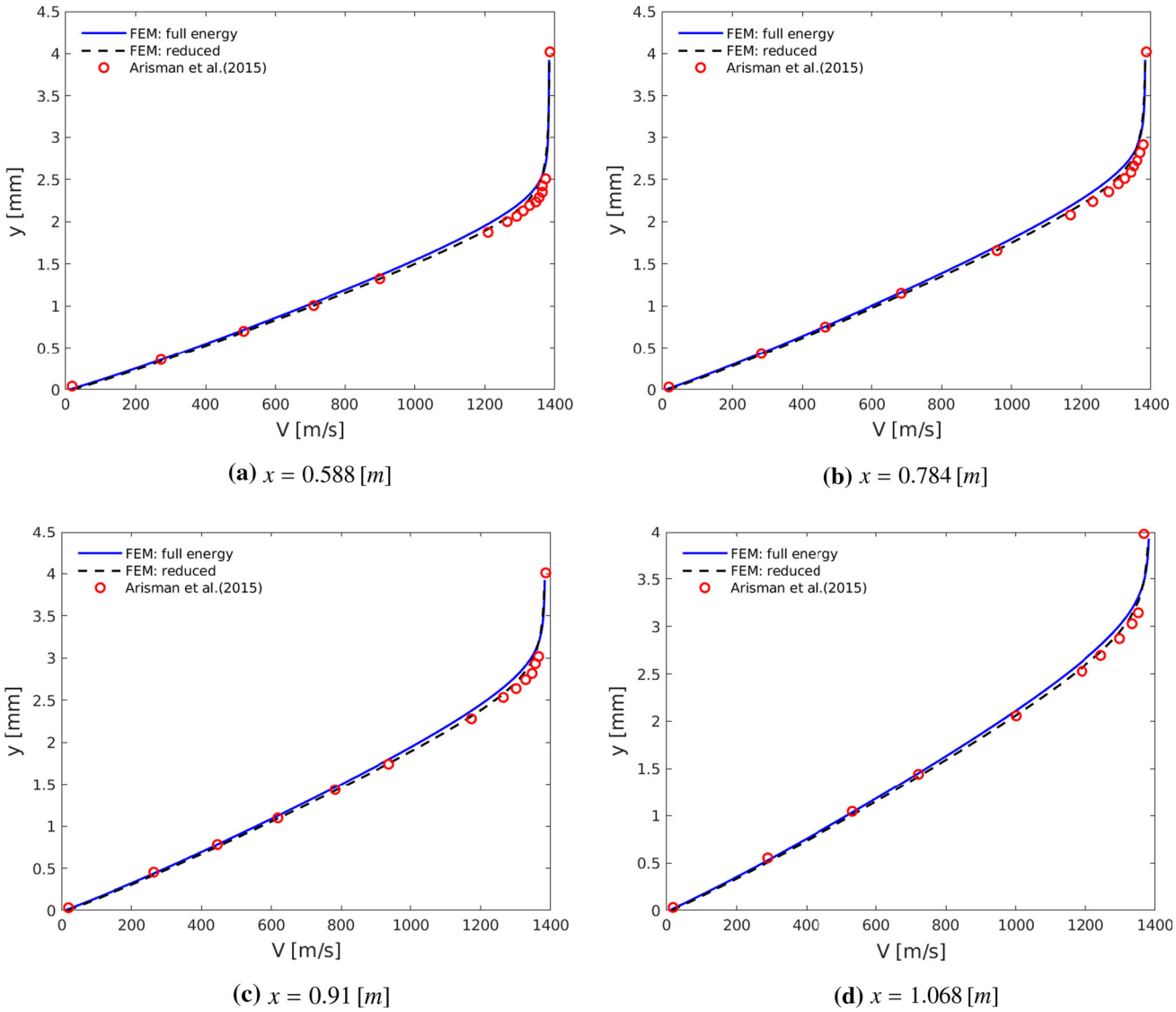


Fig. 15 Velocity profiles in the boundary layer at different locations along the plate for the Mach 9.7 oblique shock case. The profiles predicted by the two formulations (full and reduced energy) are compared to the experiment data [16]

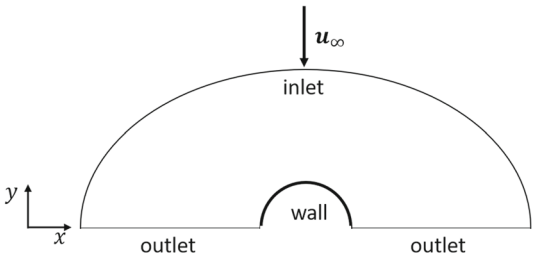


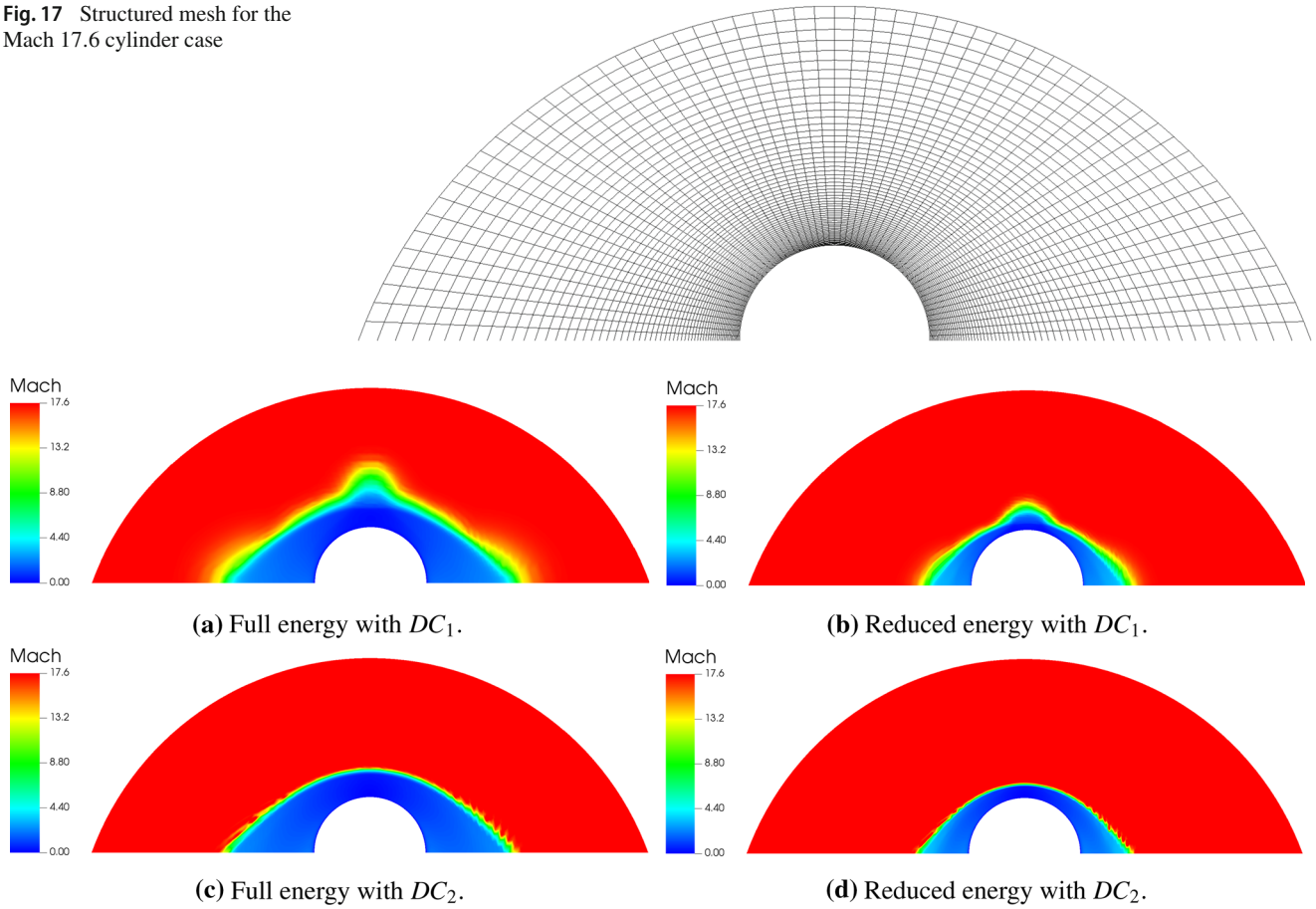
Fig. 16 Computational domain for the Mach 17.6 cylinder case

200 [K] respectively. The free stream conditions are prescribed at the domain's inlet. The no-slip condition and constant temperature $T_w = 500\,[\mathrm{K}]$ are prescribed at the cylinder's surface. Zero traction and zero heat flux are prescribed at the outlet of the domain. The perfect gas relation is used with the gas constant $R = 287\,[\mathrm{J/(kg\,K)}]$ and specific heat ratio $\gamma = 1.4$. The Prandtl number is $Pr = 0.71$. The temperature dependent viscosity is determined by the Sutherland law:

$$\mu = \mu_r \frac{T_r + S}{T_\infty + S} \left(\frac{T_\infty}{T_r} \right)^{1.5} \tag{55}$$

where $\mu_r = 1.716 \cdot 10^{-5}\,[\mathrm{Pa\,s}]$, $T_r = 273.15\,[\mathrm{K}]$ and $S = 110.4\,[\mathrm{K}]$. The thermal conductivity is computed from the Prandtl number definition as $k = \mu c_p / Pr$.

The simulations are carried out using both reduced and full energy formulations and using two different DC operators, DC_1 and DC_2. The Mach contours visualizations in Fig. 18 show that both formulations using the DC_1 operator



(a) Full energy with DC_1. **(b)** Reduced energy with DC_1.

(c) Full energy with DC_2. **(d)** Reduced energy with DC_2.

Fig. 17 Structured mesh for the Mach 17.6 cylinder case

Fig. 18 Mach number contours for the Mach 17.6 cylinder case: **a** full energy with DC_1, **b** reduced energy with DC_1, **c** full energy with DC_2, **d** reduced energy with DC_2

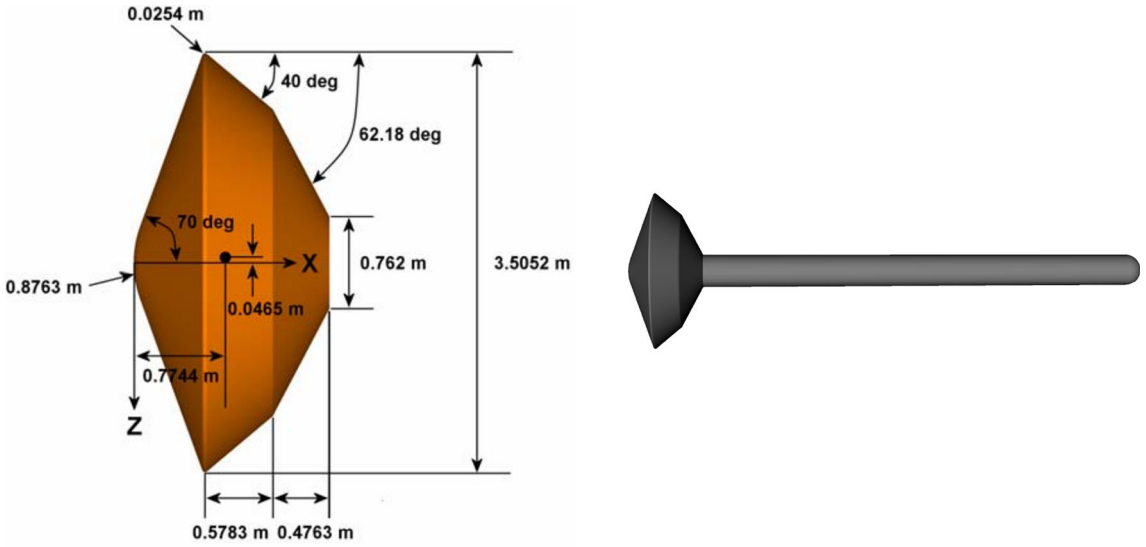
are affected by the carbuncle problem [21]. The carbuncle phenomenon is a shock instability that appears when numerical low-dissipative shock-capturing techniques are used [19] to approximate multi-dimensional shock waves [21]. It is not clear what causes the carbuncle phenomenon to appear and there is not a universal solution for this numerical problem [21]. In [19], it was observed that the upstream Mach number, the shock structure, and the computational grid affect the carbuncle phenomenon. In this work the carbuncle problem is solved when the second DC operator, DC_2, is adopted instead of the operator DC_1. Using the DC_1 operator, numerical instabilities arise at the shock wave for both formulation as shown in Fig. 18a, b. The shock wave instability does not appear when the DC_2 operator is used (see Fig. 18c, d). Comparing qualitatively Fig. 18a, c with Fig. 18b, d, it can be observed that the full energy formulation predicts a greater shock stand-off distance than the one predicted by the reduced energy formulation. The shock stand-off distance predicted by both formulations is compared with the data presented in [48] and is shown in Table 1. Table 1 shows that the full energy formulation is able to predict the shock stand-off distance with good accuracy, while the reduced energy formulation strongly underpredicts the distance value with a relative error of 46% with respect to the predictions by [48].

Table 1 Shock stand-off distance value predicted by the full and reduced energy formulations using the DC_2 operator. Comparison with the predictions by [48]

Predictions	Stand-off [m]	Relative error [%]
Full energy	0.49	9%
Reduced energy	0.24	46%
Mazaheri and Kleb [48]	0.45	

3.5 Viking Lander capsule

The Viking Lander Capsule is an atmospheric re-entry vehicle (see Fig. 19a). The availaibility of the capsule's geometry, numerical, experimental, and flight data make the Viking Lander Capsule an excellent three-dimensional, hypersonic benchmark case. The simulated model is shown in Fig. 19b. The model dimensions are 3.48% of the cap-



(a) Geometry [24]. **(b)** Computational model.

Fig. 19 The figure on the left **a** shows the real geometric dimensions of the Viking lander capsule [22]. The figure on the right **b** shows the simulation model consisting of the capsule and the sting

Table 2 Tunnel flow conditions [20]

M	Re_D	**u** [m/s]	p [Pa]	T [K]	ρ [kg/m^3]
6	$1.24 \cdot 10^6$	946.4	718.1	62.8	$3.961 \cdot 10^{-2}$

sule real dimensions illustrated in Fig. 19a, which is similar to the wind-tunnel experiment [35]. The model includes the sting to match the experimental set-up [35]. The free stream flow is characterized by a Mach number $M = 6$ and a Reynolds number based on the scaled diameter of the capsule $D = 1.22 \cdot 10^{-1}$ [m], $Re_D = 1.24 \times 10^6$. The tunnel undisturbed flow conditions are computed in [20] and shown in Table 2. At the inlet the free stream conditions shown in Table 2 are prescribed. The no slip boundary conditions and a constant temperature $T_w = 300$ [K] are prescribed at the capsule and sting surfaces. At the outlet and laterals, zero traction and zero heat transfer boundary conditions are imposed. The fluid is assumed as a perfect gas with the specific heat ratio $\gamma = 1.4$, the gas constant $R = 287$ [J/(kg K)] and Prandtl number $Pr = 0.72$. The temperature dependent viscosity is determined by the Sutherland law:

$$\mu = \mu_r \frac{T_r + S}{T_\infty + S} \left(\frac{T_\infty}{T_r} \right)^{1.5} \tag{56}$$

where $\mu_r = 1.716 \cdot 10^{-5}$ [Pa s], $T_r = 273.15$ [K] and $S = 110.4$ [K]. The thermal conductivity is computed from the Prandtl number definition as $k = \mu c_p / Pr$.

The sketch of the computational domain is shown in Fig. 20.

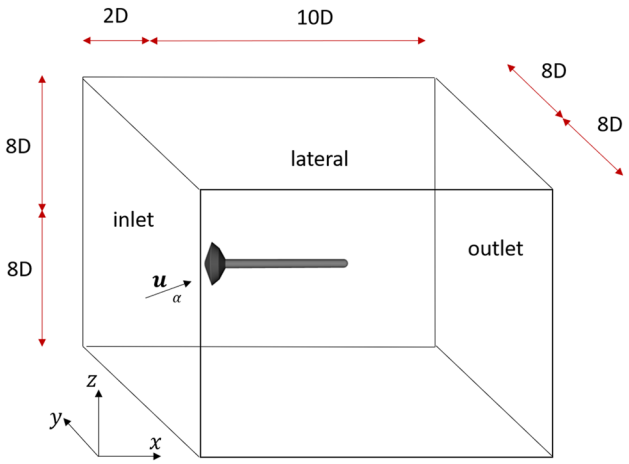


Fig. 20 Computational domain for the Viking Lander Capsule case

The model's surfaces are discretized with linear triangular elements, while linear tetrahedron and prism elements are used to discretize the volume and the boundary layer respectively. The mesh used in the simulation involves approximately 8 millions volume elements. In order to assess the resolution of the present mesh with respect to the one used in [20], it is useful to define the wall Reynolds number as:

$$Re_w = \frac{\rho a \Delta\eta}{\mu} \tag{57}$$

where $\Delta\eta$ is the height of the first element at the wall and the density ρ, the speed of sound a, and the viscosity μ are computed at the wall. The boundary layer is resolved using

10 normal elements and $\Delta\eta = 10^{-4}$ [m], giving a maximum wall Reynolds number of around 700, which is relatively high if compared with the wall grid spacing adopted by [20], where they kept the values $Re_w = 1, 0.5$ for the coarse and fine meshes respectively. The mesh used for the present simulations is illustrated in Fig. 21.

The simulations are carried out using both reduced and full energy formulations and using DC_2 operator. The simulations are carried out for two different angles of attack, in particular $\alpha = 10°, 20°$. For the case $\alpha = 20°$, in Fig. 22, the Mach contours and the wall pressure coefficient predicted by the full (Fig. 22a) and reduced (Fig. 22b) energy formulations are compared with the ones computed using NASA LAURA code [20] (see Fig. 22c). The flow topology, such as the shock stand-off distance and the bow shock shape, simulated by the full energy formulation qualitatively agrees with the LAURA predictions, while the reduced energy predicts a smaller shock stand-off distance. The maximum value of the pressure coefficient predicted by the reduced energy formulation ($C_p = 1.86$) agrees very well with LAURA predictions ($C_p = 1.85$), while it is slightly underestimated by the full energy formulation ($C_p = 1.8$). The aerodynamic forces are computed by integrating the Cauchy stress tensor acting on the wall surface, $\boldsymbol{\sigma} \cdot \mathbf{n}$, all over the capsule's surface and neglecting the sting contribution. Knowing that the axial direction coincides with the x-axis and the normal direction coincides with z-axis, the axial and normal forces are defined as $F_A = F_x$ and $F_N = F_z$ respectively. The axial C_A, normal C_N, lift C_L and drag C_D coefficients are calculated as below:

$$C_A = \frac{F_A}{q_\infty S_{ref}} \tag{58}$$

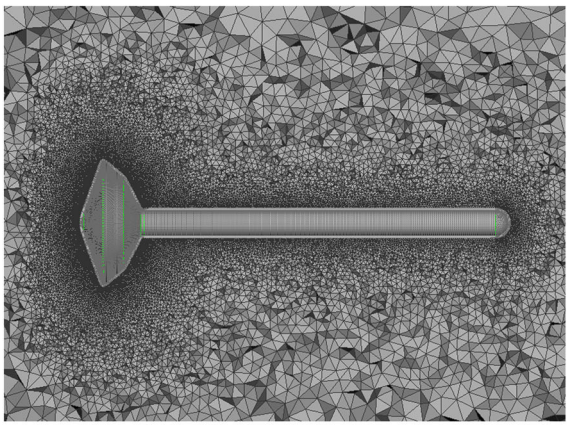
$$C_N = \frac{F_N}{q_\infty S_{ref}} \tag{59}$$

$$C_L = -C_A \sin\alpha + C_N \cos\alpha \tag{60}$$

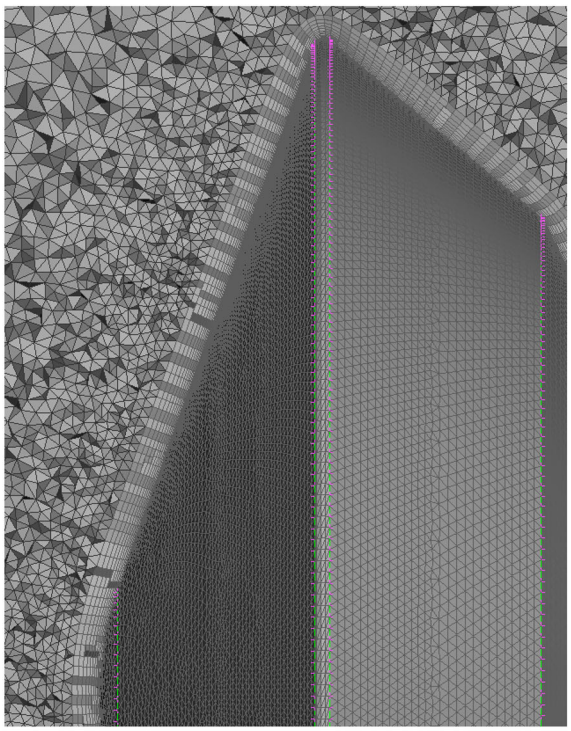
$$C_D = C_A \cos\alpha + C_N \sin\alpha \tag{61}$$

where $q_\infty = 1/2\rho|\mathbf{u}|^2$ is the free stream dynamic pressure and $S_{ref} = \pi D^2/4$ is the reference area. The values of the aerodynamic coefficients predicted by both reduced and full energy formulations are shown in Fig. 23 and tabulated in Table 3 together with numerical and experiment data [20].

In Table 3 the relative error is computed with respect to the experiment values. Table 3 and Fig. 23 show that both formulations predict aerodynamic forces with good accuracy even though the boundary layer is poorly resolved if compared with the one used in [20]. However, it is worth to notice that the relative errors, with which the forces coefficients are predicted by the full energy formulation, in percentage remain almost constant for the two angles of attack α, simulated. This error consistency does not hold for the predictions of

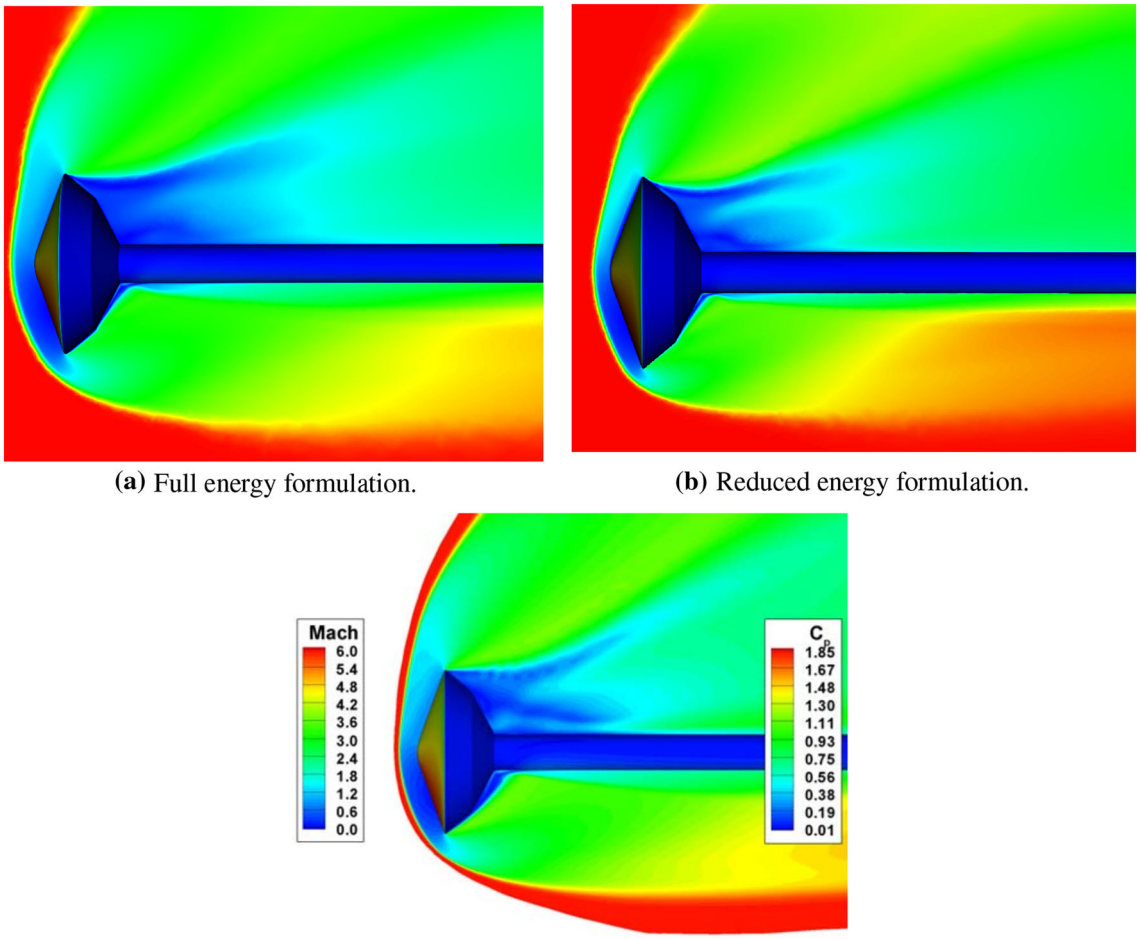


(a) Mesh.



(b) Zoom on the boundary layer.

Fig. 21 The computational mesh used for the Viking Lander Capsule case is illustrated in **a**, while the boundary layer resolution can be appreciated in the mesh detail shown in **b**



(a) Full energy formulation.

(b) Reduced energy formulation.

(c) LAURA simulation [22].

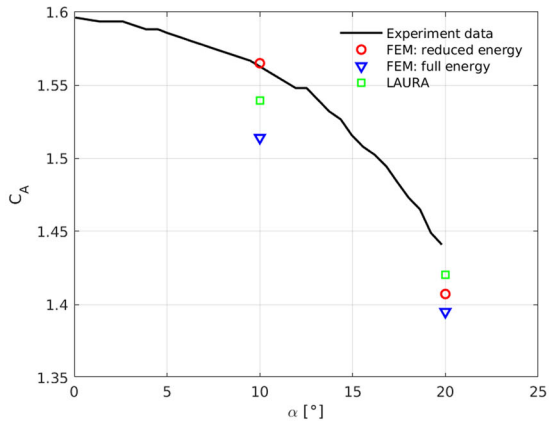
Fig. 22 Mach contours and wall pressure coefficient distribution for the case $\alpha = 20°$. The full energy predictions are shown in **a**, the reduced energy predictions are in **b**, while results predicted by LAURA code [20] are shown in **c**

the reduced energy formulation, where the relative error for the case $\alpha = 20°$ is greater than the one for the case $\alpha = 10°$.

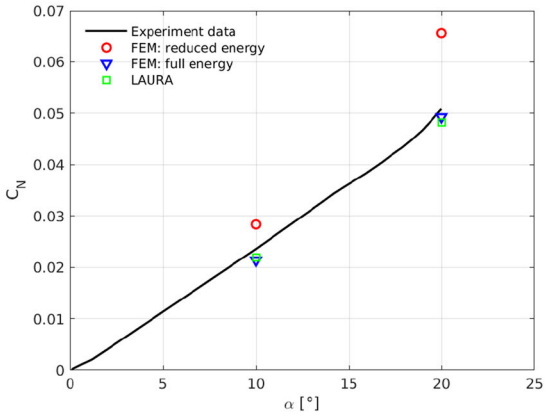
4 Conclusion

In this work two SUPG finite element formulations with DC for the solution of compressible flows are presented. In particular, the full-energy and reduced-energy formulations are tested with the pressure-based primitive variable set in the context of hypersonic flows and their relative accuracy is assessed for this flow regime. Note that the reduced-energy formulation is more convenient for fluid-structure interaction modeling involving thermally-coupled solids, because only the heat flux appears in the energy-equation slot of the traction vector. Likewise, the pressure-based primitive variables (which is not the most common for hypersonic regime) are more convenient for setting boundary conditions and implementing FSI coupling. Moreover, pressure-primitive variables lead to a natural extension of the weakly enforced essential boundary condition and sliding-interface formulations, originally defined for incompressible flows, to the compressible flow regime (see e.g. [7,84]). The robustness, stability, and accuracy of the two formulations are shown by solving several benchmark cases for a wide range of Mach numbers and cases complexity, such as the 1D Sod's problem, the 2D Mach 6 flat plate, the 2D Mach 10 oblique shock case, the 2D Mach 17.6 cylinder case, and the 3D Viking Lander Capsule. The numerical results presented in Sect. 3 show that the proposed formulation behaves very well in the context of high-speed flows maintaining good accuracy and stability in both steady and unsteady cases. While stable in the presence of strong shocks, the method is not over-diffusive because the shock-capturing viscosity is residual based and vanishes quickly in the part of the domain where the solution is smooth.

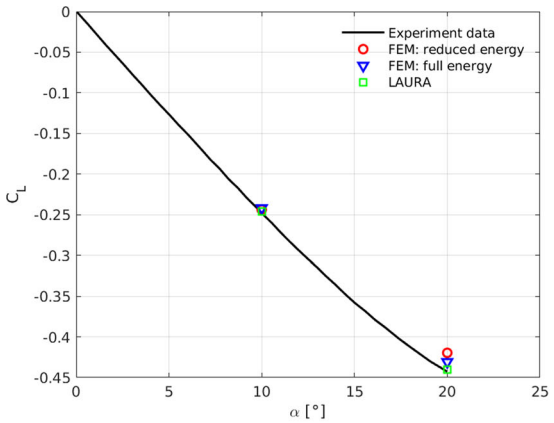
The present paper clearly demonstrates that the full-energy formulation is superior in accuracy relative to the



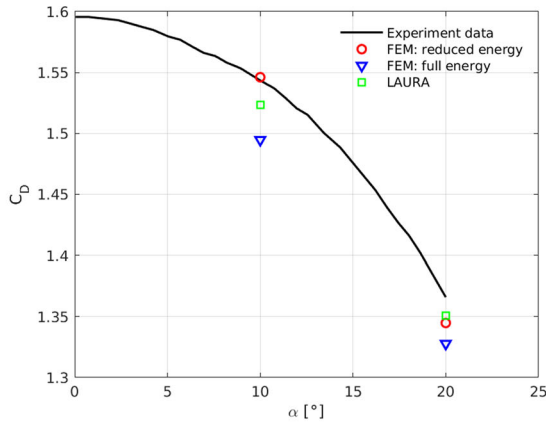
(a) Axial force coefficient C_A.



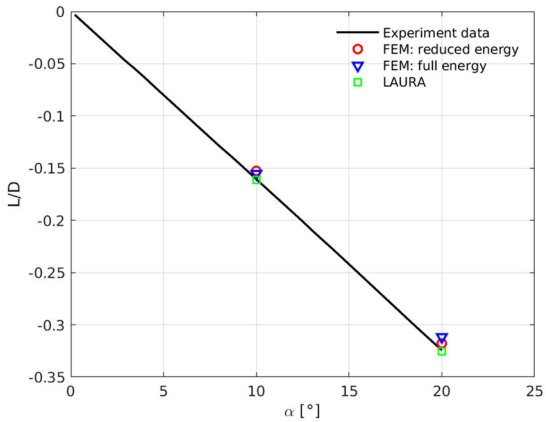
(b) Normal force coefficient C_N.



(c) Lift coefficient C_L.



(d) Drag coefficient C_D.



(e) Lift to Drag ratio L/D.

Fig. 23 Forces coefficients as function of the free stream angle of attack α. The axial force (**a**), normal force (**b**), lift (**c**) and drag (**d**) coefficients and the lift to drag ratio (**e**) predicted by the full and reduced energy formulations of the present simulations are compared to the numerical results by [20] and to the experiment data by [35]

Table 3 Forces coefficients values for the two angles of attack simulated using the reduced and full energy formulations

α	C_A	C_N	C_L	C_D	L/D
10°					
Experiment	1.5628	0.02357	−0.24828	1.54341	−0.16063
LAURA	1.53969	0.02197	−0.2458	1.52348	−0.16132
Full energy	1.51404	0.02139	−0.24185	1.49475	−0.1618
	(3.12%)	(9.26%)	(2.59%)	(3.15%)	(0.73%)
Reduced energy	1.56499	0.02835	−0.24384	1.54614	−0.15771
	(0.14%)	(20.31%)	(1.79%)	(0.18%)	(1.82%)
20°					
Experiment	1.44095	0.05085	−0.44276	1.36593	−0.32438
LAURA	1.42042	0.0482	−0.44049	1.35059	−0.32566
Full energy	1.39502	0.04928	−0.43081	1.32775	−0.32447
	(3.19%)	(3.09%)	(2.7%)	(2.8%)	(0.03%)
Reduced energy	1.40717	0.06559	−0.41965	1.34474	−0.31207
	(2.34%)	(28.97%)	(5.22%)	(1.55%)	(3.8%)

Experiment and numerical (LAURA) data are extracted from [20]

reduced-energy formulation for hypersonic flows. There appears to be a bias error associated with the reduced energy formulation near shock waves, resulting in weaker shocks than would otherwise occur. In addition, the DC technique that makes use of a single viscosity parameter based on the weighted norms of the residual and solution gradient, and capped at the maximum allowable value corresponding to upwind viscosity, is able to overcome the well-known carbuncle instability without smearing the shock.

The successful performance of the proposed formulation for high-speed flows sets the stage for the deployment of the techniques developed to more advanced applications such as fluid–structure interaction modeling of hypersonic aircraft. Moreover, the framework will be augmented with weak imposition of the Dirichlet boundary conditions to relax the requirement on a boundary layer resolution. It should also be noted that we have not considered the reacting flows in the present work and will augment the formulation with the chemistry model in the future work.

Acknowledgements This research was enabled in part by support provided by WestGrid (www.westgrid.ca) and Compute Canada Calcul Canada (www.computecanada.ca). Financial support was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). Y. Bazilevs was partially supported by the NSF Award No. 1854436.

Appendix A

The matrices used for Navier–Stokes equations of compressible flows with full energy equation are given by

$$\tilde{\mathbf{A}}_0 = \begin{bmatrix} \rho\beta_T & 0 & 0 & 0 & -\rho\alpha_p \\ \rho\beta_T u_1 & \rho & 0 & 0 & -\rho\alpha_p u_1 \\ \rho\beta_T u_2 & 0 & \rho & 0 & -\rho\alpha_p u_2 \\ \rho\beta_T u_3 & 0 & 0 & \rho & -\rho\alpha_p u_3 \\ \rho\beta_T e_{\text{tot}} & \rho u_1 & \rho u_2 & \rho u_3 & \rho\left(-\alpha_p e_{\text{tot}} + c_{\text{v}}\right) \end{bmatrix}, \tag{A.1}$$

where $\beta_T = 1/p, \alpha_p = 1/T$.

Its inverse $\tilde{\mathbf{A}}_0^{-1} = \mathbf{Y}_{,\tilde{\mathbf{U}}}$ is given by

$$\tilde{\mathbf{A}}_0^{-1} = \begin{bmatrix} \dfrac{-\alpha_p e_{\text{tot}} + \alpha_p \|\mathbf{u}\|^2 + c_{\text{v}}}{\rho\beta_T c_{\text{v}}} & -\dfrac{\alpha_p u_1}{\rho\beta_T c_{\text{v}}} & -\dfrac{\alpha_p u_2}{\rho\beta_T c_{\text{v}}} & -\dfrac{\alpha_p u_3}{\rho\beta_T c_{\text{v}}} & \dfrac{\alpha_p}{\rho\beta_T c_{\text{v}}} \\ -\dfrac{u_1}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0 \\ -\dfrac{u_2}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{u_3}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ \dfrac{\|\mathbf{u}\|^2 - e_{\text{tot}}}{\rho c_{\text{v}}} & -\dfrac{u_1}{\rho c_{\text{v}}} & -\dfrac{u_2}{\rho c_{\text{v}}} & -\dfrac{u_3}{\rho c_{\text{v}}} & \dfrac{1}{\rho c_{\text{v}}} \end{bmatrix}. \tag{A.2}$$

We then give the details of the Euler Jacobian matrices by

$$\tilde{\mathbf{A}}_1^{\text{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_1 & \rho & 0 & 0 & -\rho\alpha_p u_1 \\ \rho\beta_T u_1^2 & 2\rho u_1 & 0 & 0 & -\rho\alpha_p u_1^2 \\ \rho\beta_T u_1 u_2 & \rho u_2 & \rho u_1 & 0 & -\rho\alpha_p u_1 u_2 \\ \rho\beta_T u_1 u_3 & \rho u_3 & 0 & \rho u_1 & -\rho\alpha_p u_1 u_3 \\ (\rho\beta_T e_{\text{tot}} + 1)\, u_1 & \rho\left(e_{\text{tot}} + u_1^2\right) + p & \rho u_1 u_2 & \rho u_1 u_3 & \rho\left(-\alpha_p e_{\text{tot}} + c_{\text{v}}\right) u_1 \end{bmatrix}, \tag{A.3}$$

$$\tilde{\mathbf{A}}_2^{\text{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_2 & 0 & \rho & 0 & -\rho\alpha_p u_2 \\ \rho\beta_T u_1 u_2 & \rho u_2 & \rho u_1 & 0 & -\rho\alpha_p u_1 u_2 \\ \rho\beta_T u_2^2 & 0 & 2\rho u_2 & 0 & -\rho\alpha_p u_2^2 \\ \rho\beta_T u_2 u_3 & 0 & \rho u_3 & \rho u_2 & -\rho\alpha_p u_2 u_3 \\ (\rho\beta_T e_{\text{tot}} + 1)\, u_2 & \rho u_1 u_2 & \rho\left(e_{\text{tot}} + u_2^2\right) + p & \rho u_2 u_3 & \rho\left(-\alpha_p e_{\text{tot}} + c_{\text{v}}\right) u_2 \end{bmatrix}, \tag{A.4}$$

$$\tilde{\mathbf{A}}_3^{\text{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_3 & 0 & 0 & \rho & -\rho\alpha_p u_3 \\ \rho\beta_T u_1 u_3 & \rho u_3 & 0 & \rho u_1 & -\rho\alpha_p u_1 u_3 \\ \rho\beta_T u_2 u_3 & 0 & \rho u_3 & \rho u_2 & -\rho\alpha_p u_2 u_3 \\ \rho\beta_T u_3^2 & 0 & 0 & 2\rho u_3 & -\rho\alpha_p u_3^2 \\ (\rho\beta_T e_{\text{tot}} + 1)\, u_3 & \rho u_1 u_3 & \rho u_2 u_3 & \rho\left(e_{\text{tot}} + u_3^2\right) + p & \rho\left(-\alpha_p e_{\text{tot}} + c_{\text{v}}\right) u_3 \end{bmatrix}, \tag{A.5}$$

$$\tilde{\mathbf{A}}_1^{p} = \begin{bmatrix} 0&0&0&0&0 \\ 1&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{bmatrix}, \tag{A.6}$$

$$\tilde{\mathbf{A}}_2^{p} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ 1&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{bmatrix}, \tag{A.7}$$

$$\tilde{\mathbf{A}}_3^{p} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 1&0&0&0&0 \\ 0&0&0&0&0 \end{bmatrix}. \tag{A.8}$$

Note that $\tilde{\mathbf{A}}_i = \tilde{\mathbf{A}}_i^{\text{adv}\backslash p} + \tilde{\mathbf{A}}_i^{p}$.

Finally, we give the diffusive matrices by

$$\tilde{\mathbf{K}}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & (2\mu+\lambda)\, u_1 & \mu u_2 & \mu u_3 & \kappa \end{bmatrix}, \tag{A.9}$$

$$\tilde{\mathbf{K}}_{12} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&\lambda&0&0 \\ 0&\mu&0&0&0 \\ 0&0&0&0&0 \\ 0&\mu u_2&\lambda u_1&0&0 \end{bmatrix}, \tag{A.10}$$

$$\tilde{\mathbf{K}}_{13} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&\lambda&0 \\ 0&0&0&0&0 \\ 0&\mu&0&0&0 \\ 0&\mu u_3&0&\lambda u_1&0 \end{bmatrix}, \tag{A.11}$$

$$\tilde{\mathbf{K}}_{21} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&\mu&0&0 \\ 0&\lambda&0&0&0 \\ 0&0&0&0&0 \\ 0&\lambda u_2&\mu u_1&0&0 \end{bmatrix}, \tag{A.12}$$

$$\tilde{\mathbf{K}}_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 2\mu+\lambda & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & \mu u_1 & (2\mu+\lambda)\, u_2 & \mu u_3 & \kappa \end{bmatrix}, \tag{A.13}$$

$$\tilde{\mathbf{K}}_{23} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&\lambda&0 \\ 0&0&\mu&0&0 \\ 0&0&\mu u_3&\lambda u_2&0 \end{bmatrix}, \tag{A.14}$$

$$\tilde{\mathbf{K}}_{31} = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&\mu&0 \\ 0&0&0&0&0 \\ 0&\lambda&0&0&0 \\ 0&\lambda u_3&0&\mu u_1&0 \end{bmatrix}, \tag{A.15}$$

$$\tilde{\mathbf{K}}_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda u_3 & \mu u_2 & 0 \end{bmatrix}, \tag{A.16}$$

$$\tilde{\mathbf{K}}_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 2\mu+\lambda & 0 \\ 0 & \mu u_1 & \mu u_2 & (2\mu+\lambda)\, u_3 & \kappa \end{bmatrix}. \tag{A.17}$$

For the Navier–Stokes equations with reduced energy formulation, the matrices corresponding to pressure-primitive variables are as follows:

The matrix $\mathbf{A}_0 = \mathbf{U}_{,\mathbf{Y}}$ is given by

$$\mathbf{A}_0 = \begin{bmatrix} \rho\beta_T & 0 & 0 & 0 & -\rho\alpha_p \\ \rho\beta_T u_1 & \rho & 0 & 0 & -\rho\alpha_p u_1 \\ \rho\beta_T u_2 & 0 & \rho & 0 & -\rho\alpha_p u_2 \\ \rho\beta_T u_3 & 0 & 0 & \rho & -\rho\alpha_p u_3 \\ \rho\beta_T e & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.18}$$

It's inverse $\mathbf{A}_0^{-1} = \mathbf{Y}_{,\mathbf{U}}$ is given by

$$\mathbf{A}_0^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{\alpha_p}{\rho\beta_T c_\mathrm{v}} \\ -\dfrac{u_1}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0 \\ -\dfrac{u_2}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{u_3}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ -\dfrac{T}{\rho} & 0 & 0 & 0 & \dfrac{1}{\rho c_\mathrm{v}} \end{bmatrix}. \tag{A.19}$$

The Euler–Jacobian matrices are given by

$$\mathbf{A}_1^{\mathrm{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_1 & \rho & 0 & 0 & -\rho\alpha_p u_1 \\ \rho\beta_T u_1^2 & 2\rho u_1 & 0 & 0 & -\rho\alpha_p u_1^2 \\ \rho\beta_T u_1 u_2 & \rho u_2 & \rho u_1 & 0 & -\rho\alpha_p u_1 u_2 \\ \rho\beta_T u_1 u_3 & \rho u_3 & 0 & \rho u_1 & -\rho\alpha_p u_1 u_3 \\ \rho\beta_T e u_1 & \rho e & 0 & 0 & 0 \end{bmatrix}, \tag{A.20}$$

$$\mathbf{A}_2^{\mathrm{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_2 & 0 & \rho & 0 & -\rho\alpha_p u_2 \\ \rho\beta_T u_1 u_2 & \rho u_2 & \rho u_1 & 0 & -\rho\alpha_p u_1 u_2 \\ \rho\beta_T u_2^2 & 0 & 2\rho u_2 & 0 & -\rho\alpha_p u_2^2 \\ \rho\beta_T u_2 u_3 & 0 & \rho u_3 & \rho u_2 & -\rho\alpha_p u_2 u_3 \\ \rho\beta_T e u_2 & 0 & \rho e & 0 & 0 \end{bmatrix}, \tag{A.21}$$

$$\mathbf{A}_3^{\mathrm{adv}\backslash p} = \begin{bmatrix} \rho\beta_T u_3 & 0 & 0 & \rho & -\rho\alpha_p u_3 \\ \rho\beta_T u_1 u_3 & \rho u_3 & 0 & \rho u_1 & -\rho\alpha_p u_1 u_3 \\ \rho\beta_T u_2 u_3 & 0 & \rho u_3 & \rho u_2 & -\rho\alpha_p u_2 u_3 \\ \rho\beta_T u_3^2 & 0 & 0 & 2\rho u_3 & -\rho\alpha_p u_3^2 \\ \rho\beta_T e u_3 & 0 & 0 & \rho e & 0 \end{bmatrix}, \tag{A.22}$$

$$\mathbf{A}_1^p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.23}$$

$$\mathbf{A}_2^p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.24}$$

$$\mathbf{A}_3^p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.25}$$

$$\mathbf{A}_1^{\mathrm{sp}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p-\tau_{11} & -\tau_{12} & -\tau_{13} & 0 \end{bmatrix}. \tag{A.26}$$

$$\mathbf{A}_2^{\mathrm{sp}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\tau_{21} & p-\tau_{22} & -\tau_{23} & 0 \end{bmatrix}. \tag{A.27}$$

$$\mathbf{A}_3^{\mathrm{sp}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\tau_{31} & -\tau_{32} & p-\tau_{33} & 0 \end{bmatrix}. \tag{A.28}$$

The diffusivity matrices are given by

$$\mathbf{K}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & \kappa \end{bmatrix}, \tag{A.29}$$

$$\mathbf{K}_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.30}$$

$$\mathbf{K}_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.31}$$

$$\mathbf{K}_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.32}$$

$$\mathbf{K}_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 2\mu+\lambda & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & \kappa \end{bmatrix}, \tag{A.33}$$

$$\mathbf{K}_{23} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.34}$$

$$\mathbf{K}_{31} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.35}$$

$$\mathbf{K}_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.36}$$

$$\mathbf{K}_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 2\mu+\lambda & 0 \\ 0 & 0 & 0 & 0 & \kappa \end{bmatrix}. \tag{A.37}$$

The matrices for the conservation variables may be obtained from the corresponding matrices for the pressure-primitive variables using the following transformations: $\hat{\mathbf{A}}_i = \mathbf{A}_i\mathbf{A}_0^{-1}$, $\hat{\mathbf{A}}_i^{\text{adv}\backslash p} = \mathbf{A}_i^{\text{adv}\backslash p}\mathbf{A}_0^{-1}$, $\hat{\mathbf{A}}_i^{p} = \mathbf{A}_i^{p}\mathbf{A}_0^{-1}$, $\hat{\mathbf{A}}_i^{\text{sp}} = \mathbf{A}_i^{\text{sp}}\mathbf{A}_0^{-1}$, and $\hat{\mathbf{K}}_{ij} = \mathbf{K}_{ij}\mathbf{A}_0^{-1}$

References

1. Ahrabi BR, Mavriplis DJ (2020) An implicit block ilu smoother for preconditioning of Newton–Krylov solvers with application in high-order stabilized finite-element methods. Comput Methods Appl Mech Eng 358:112637
2. Aliabadi SK, Tezduyar TE (1993) Space-time finite element computation of compressible flows involving moving boundaries and interfaces. Comput Methods Appl Mech Eng 107(1–2):209–223
3. Almeida RC, Galeão AC (1996) An adaptive Petrov–Galerkin formulation for the compressible Euler and Navier–Stokes equations. Comput Methods Appl Mech Eng 129(1):157–176
4. Arisman CJ, Johansen CT, Bathel BF, Danehy PM (2015) Investigation of gas seeding for planar laser-induced fluorescence in hypersonic boundary layer. AIAA J 53(12):3637–3651
5. Baba K, Tabata M (1981) On a conservative upwind finite element scheme for the convective diffusion equations. RAIRO Analyse Numerique 15(1):3–25
6. Bazilevs Y, Kamensky D, Moutsanidis G, Shende S (2020) Residual-based shock capturing in solids. Comput Methods Appl Mech Eng 358:112638
7. Bazilevs Y, Takizawa K, Wu MCH, Kuraishi T, Avsar R, Xu Z, Tezduyar TE (2020) Gas turbine computational flow and structure analysis with isogeometric discretization and a complex-geometry mesh generation method. Comput Mech. https://doi.org/10.1007/s00466-020-01919-w
8. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput Methods Appl Mech Eng 32:199–259
9. Celik IB et al (2008) Procedure for estimation and reporting of uncertainty due to discretization in CFD applications. J Fluids Eng 130(7):07
10. Carter JE (1972) Numerical solutions of the Navier–Stokes equations for the supersonic laminar flow over a two-dimensional compression corner. Technical report, nasa-tr-r-385, NASA Langley Research Center; Hampton, VA, United States
11. Catabriga L, Coutinho ALGA, Tezduyar TE (2005) Compressible flow SUPG parameters computed from element matrices. Commun Numer Methods Eng 21:465–476
12. Catabriga L, Coutinho ALGA, Tezduyar TE (2006) Compressible flow SUPG stabilization parameters computed from degree-of-freedom submatrices. Comput Mech 38:334–343
13. Chalot F, Hughes TJR (1994) A consistent equilibrium chemistry algorithm for hypersonic flows. Comput Methods Appl Mech Eng 112:25–40
14. Chalot F, Hughes TJR, Shakib F (1990) Symmetrization of conservation laws with entropy for high-temperature hypersonic computations. Comput Syst Eng 1(2–4):495–521
15. Chapman S, Cowling TG (1970) The mathematical theory of nonuniform gases. Cambridge University Press, Cambridge
16. Danehy P, Bathel B, Ivey C, Inman J, Jones S (2009) NO PLIF study of hypersonic transition over a discrete hemispherical roughness element. In: 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition
17. Denman ED, Beavers AN (1976) The matrix sign function and computations in systems. Appl Math Comput 2(1):63–94
18. Dick E (2009) Introduction to finite element methods in computational fluid dynamics. In: Wendt JF (ed) Computational fluid dynamics, chapter 10. Springer, Berlin
19. Dumbser M, Moschetta JM, Gressier J (2003) A matrix stability analysis of the carbuncle phenomenon. J Comput Phys 197:647–670
20. Edquist KT (2006) Computations of Viking Lander capsule hypersonic aerodynamics with comparisons to ground and flight data. In: AIAA atmospheric flight mechanics conference and exhibit
21. Elling V (2009) The carbuncle phenomenon is incurable. Acta Mathematica Scientia 29(6):1647–1656
22. Flaherty T (1972) Aerodynamics data book, ver-10. TR-3709014, Martin Marietta Corporation, Denver Division
23. Hauke G (2001) Simple stabilizing matrices for the computation of compressible flows in primitive variables. Comput Methods Appl Mech Eng 190:6881–6893

24. Hauke G, Hughes TJR (1994) A unified approach to compressible and incompressible flows. Comput Methods Appl Mech Eng 113:389–395
25. Hauke G, Hughes TJR (1998) A comparative study of different sets of variables for solving compressible and incompressible flows. Comput Methods Appl Mech Eng 153:1–44
26. Hirschfelder JO, Curtiss CF, Bird RB (1954) Molecular theory of gases and liquids. Wiley, Hoboken
27. Hollis BR (1996) Real-gas flow properties for NASA Langley research center aerothermodynamic facilities complex wind tunnels. Nasa contractor report 4755, NASA Langley Research Center; Hampton, VA, United States
28. Hughes TJR, Franca LP, Hulbert GM (1989) A new finite element formulation for computational fluid dynamics: VIII the Galerkin/least-squares method for advective-diffusive equations. Comput Methods Appl Mech Eng 73:173–189
29. Hughes TJR, Franca LP, Mallet M (1986) A new finite element formulation for computational fluid dynamics: I symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics. Comput Methods Appl Mech Eng 54:223–234
30. Hughes TJR, Franca LP, Mallet M (1987) A new finite element formulation for computational fluid dynamics: VI convergence anaysis of the generalized SUPG formulation for linear time-dependent multidimensional advective–diffusive systems. Comput Methods Appl Mech Eng 63:97–112
31. Hughes TJR, Mallet M (1986) A new finite element formulation for computational fluid dynamics: III the generalized streamline operator for multidimensional advective–diffusive systems. Comput Methods Appl Mech Eng 58:305–328
32. Hughes TJR, Mallet M (1986) A new finite element formulation for computational fluid dynamics: IV a discontinuity-capturing operator for multidimensional advective–diffusive systems. Comput Methods Appl Mech Eng 58:329–336
33. Hughes TJR, Mallet M, Mizukami A (1986) A new finite element formulation for computational fluid dynamics: II beyond SUPG. Comput Methods Appl Mech Eng 54:341–355
34. Hughes TJR, Tezduyar TE (1984) Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. Comput Methods Appl Mech Eng 45:217–284
35. Ingoldby RN, Michel FC, Flaherty TM, Doryand MG, Preston B, Villyard KW, Steele RD (1976) Entry data analysis for viking landers 1 and 2 final report. NASA CR-159388, Martin Marietta Corporation, Denver Division
36. Jansen KE, Whiting CH, Hulbert GM (2000) A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. Comput Methods Appl Mech Eng 190:305–319
37. Johnson C (1987) Numerical solution of partial differential equations by the finite element method. Cambridge University Press, Cambridge
38. Johnson C, Navert U, Pitkaranta J (1984) Finite element methods for linear hyperbolic problems. Comput Methods Appl Mech Eng 45:285–312
39. Johnson C, Szepessy A (1987) On the convergence of a finite element method for a nonlinear hyperbolic conservation law. Math Comput 49(180):427–444
40. Johnson C, Szepessy A, Hansbo P (1990) On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. Math Comput 54(189):107–129
41. Kanai T, Takizawa K, Tezduyar TE, Tanaka T, Hartmann A (2019) Compressible-flow geometric-porosity modeling and spacecraft parachute computation with isogeometric discretization. Comput Mech 63:301–321
42. Kirk BS, Stogner RH, Bauman PT, Oliver TA (2014) Modeling hypersonic entry with the fully-implicit Navier–Stokes (fin-s) stabilized finite element flow solver. Comput Fluids 92:281–292
43. Kozak N, Rajanna MR, Wu MCH, Murugan M, Bravo L, Ghoshal A, Hsu MC, Bazilevs Y (2020) Optimizing gas turbine performance using the surrogate management framework and high-fidelity flow modeling. Energies 13(17):4283
44. Kuraishi T, Takizawa K, Tezduyar TE (2019) Tire aerodynamics with actual tire geometry, road contact and tire deformation. Comput Mech 63:1165–1185
45. Le Beau GJ, Ray SE, Aliabadi SK, Tezduyar TE (1993) SUPG finite element computation of compressible flows with entropy and conservation variables formulations. Comput Methods Appl Mech Eng 104:397–422
46. Le Beau GJ, Tezduyar TE (1991) Finite element computation of compressible flows with the SUPG formulation. Am Soc Mech Eng Fluids Eng Div Pub FED 123:21–27
47. Lewis JE (1967) Experimental investigation of supersonic laminar, two-dimensional boundary layer separation in a compression corner with and without cooling. Ph.D. thesis, California Institute of Technology
48. Mazaheri A, Kleb B (2007) Exploring hypersonic, unstructured-grid issues through structured grids. In: 18th AIAA computational fluid dynamics conference
49. Neufeld PD, Janzen AR, Aziz RA (1972) Empirical equations to calculate 16 of the transport collision integral ω for the Lennard–Jones (12–6) potential. J Chem Phys 57(3):1100–1102
50. Otoguro Y, Takizawa K, Tezduyar TE (2020) Element length calculation in B-spline meshes for complex geometries. Comput Mech 65:1085–1103
51. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Avsar R, Zhang Y (2019) Space-time VMS flow analysis of a turbocharger turbine with isogeometric discretization: computations with time-dependent and steady-inflow representations of the intake/exhaust cycle. Comput Mech 64:1403–1419
52. Otoguro Y, Takizawa K, Tezduyar TE, Nagaoka K, Mei S (2019) Turbocharger turbine and exhaust manifold flow computation with the space-time variational multiscale method and isogeometric analysis. Comput Fluids 179:764–776
53. Saad Y, Schultz MH (1986) Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7(3):856–869
54. Shakib F, Hughes TJR, Johan Z (1991) A new finite element formulation for a computational fluid dyamics: X. The compressible Euler and Navier–Stokes equations. Comput Methods Appl Mech Eng 89:141–219
55. Sod GA (1978) A survey of several finite difference methods for systems of nonlinear hyperbolic conservations laws. J Comput Phys 27:1–31
56. Sturek WB, Ray S, Aliabadi S, Waters C, Tezduyar TE (1997) Parallel finite element computation of missile aerodynamics. Int J Numer Meth Fluids 24:1417–1432
57. Szepessy A (1989) Convergence of a shock-capturing streamline diffusion finite element method for a scalar conservation law in two space dimensions. Math Comput 53(188):527–545
58. Tabata M (1977) A finite element approximation corresponding to the upwind finite differencing. Mem Numer Math 4:47–63
59. Tabata M (1978) Uniform convergence of the upwind finite element approximation for semilinear parabolic problems. J Math Kyoto Univ 18(2):327–351
60. Takizawa K, Tezduyar TE, Kanai T (2017) Porosity models and computational methods for compressible-flow aerodynamics of parachutes with geometric porosity. Math Models Methods Appl Sci 27:771–806

61. Takizawa K, Tezduyar TE, Kuraishi T (2015) Multiscale ST methods for thermo-fluid analysis of a ground vehicle and its tires. Math Models Methods Appl Sci 25:2227–2255
62. Takizawa K, Tezduyar TE, McIntyre S, Kostov N, Kolesar R, Habluetzel C (2014) Space-time VMS computation of wind-turbine rotor and tower aerodynamics. Comput Mech 53:1–15
63. Takizawa K, Tezduyar TE, Otoguro Y (2018) Stabilization and discontinuity-capturing parameters for space-time flow computations with finite element and isogeometric discretizations. Comput Mech 62:1169–1186
64. Takizawa K, Ueda Y, Tezduyar TE (2019) A node-numbering-invariant directional length scale for simplex elements. Math Models Methods Appl Sci 29:2719–2753
65. Tejada-Martínez AE, Akkerman I, Bazilevs Y (2012) Large-eddy simulation of shallow water Langmuir turbulence using isogeometric analysis and the residual-based variational multiscale method. J Appl Mech 79(1):010909
66. Tezduyar T, Aliabadi S, Behr M, Johnson A, Kalro V, Litke M (1996) Flow simulation and high performance computing. Comput Mech 18:397–412
67. Tezduyar T, Aliabadi S, Behr M, Johnson A, Mittal S (1993) Parallel finite-element computation of 3D flows. Computer 26(10):27–36
68. Tezduyar TE, Aliabadi SK, Behr M, Mittal S (1994) Massively parallel finite element simulation of compressible and incompressible flows. Comput Methods Appl Mech Eng 119:157–177
69. Tezduyar TE (2001) Adaptive determination of the finite element stabilization parameters. In: Proceedings of the ECCOMAS computational fluid dynamics conference 2001
70. Tezduyar TE (2002) Calculation of the stabilization parameters in SUPG and PSPG formulations. In: Proceedings of the first South-American congress on computational mechanics
71. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. Int J Numer Methods Fluids 43:555–575
72. Tezduyar TE (2004) Determination of the stabilization and shock-capturing parameters in SUPG formulation of compressible flows. In: Proceedings of European congress on computational methods in applied sciences and engineering ECCOMAS 2004
73. Tezduyar TE (2004) Finite element methods for fluid dynamics with moving boundaries and interfaces. In: Stein E, De Borst R, Hughes TJR (eds) Encyclopedia of computational mechanics, volume 3: fluids, chapter 17. Wiley, New York
74. Tezduyar TE (2005) Calculation of the stabilization parameters in finite element formulations of flow problems. In: Idelsohn SR, Sonzogni V (eds) Applications of computational mechanics in structures and fluids. CIMNE, Barcelona, pp 1–19
75. Tezduyar TE, Hughes TJR (1982) Development of time-accurate finite element techniques for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. NASA technical report NASA-CR-204772
76. Tezduyar TE, Hughes TJR (1983) Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations. In: 21st aerospace sciences meeting, AIAA paper 83-0125
77. Tezduyar TE, Osawa Y (2000) Finite element stabilization parameters computed from element matrices and vectors. Comput Methods Appl Mech Eng 190:411–430
78. Tezduyar TE, Park YJ (1986) Discontinuity capturing finite element formulations for nonlinear convection–diffusion–reaction equations. Comput Methods Appl Mech Eng 59:307–325
79. Tezduyar TE, Senga M (2006) Stabilization and shock-capturing parameters in SUPG formulation of compressible flows. Comput Methods Appl Mech Eng 195:1621–1632
80. Tezduyar TE, Senga M (2007) SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing. Comput Fluids 36:147–159
81. Tezduyar TE, Senga M, Vicker D (2006) Computation of inviscid supersonic flows around cylinders and spheres with the SUPG formulation and YZβ shock-capturing. Comput Mech 38:469–481
82. Ueda Y, Otoguro Y, Takizawa K, Tezduyar TE (2020) Element-splitting-invariant local-length-scale calculation in B-spline meshes for complex geometries. Math Models Methods Appl Sci. https://doi.org/10.1142/S0218202520500402
83. Xu F, Bazilevs Y, Hsu MC (2019) Immersogeometric analysis of compressible flows with application to aerodynamic simulation of rotorcraft. Math Models Methods Appl Sci 29(5):905–938
84. Xu F, Moutsanidis G, Kamensky D, Hsu MC, Murugan M, Ghoshal A, Bazilevs Y (2017) Compressible flows on moving domains: stabilized methods, weakly enforced essential boundary conditions, sliding interfaces, and application to gas-turbine modeling. Comput Fluids 158:201–220