

# Computers and Fluids

## An embedded shock-fitting technique on unstructured dynamic grids

--Manuscript Draft--

Manuscript Number:	CAF-D-20-00341R2
Article Type:	Research paper
Keywords:	shock-fitting; unstructured dynamic grid; Shock wave; Numerical simulation
Corresponding Author:	Aldo Bonfiglioli, Ph.D. University of Basilicata Potenza, ITALY
First Author:	Dongyang Zou, PhD
Order of Authors:	Dongyang Zou, PhD Aldo Bonfiglioli, Associate professor Renato Paciorri, Associate professor Jun Liu, Professor
Abstract:	<p>In this paper, a new shock-fitting technique based on unstructured dynamic grids is proposed to improve the performances of the unstructured ``boundary" shock-fitting technique developed by Liu and co-workers.</p> <p>The main feature of this new technique, which we call the ``embedded" shock-fitting technique, is its capability to insert or remove shocks or parts thereof during the calculation.</p> <p>This capability is enabled by defining subsets of grid-points (mutually connected by lines) which behave as either ``common"- or ``shock"-points,</p> <p>shock-waves being made of an ordered collection of shock-points.</p> <p>Two different sets of flow variables, corresponding to the upstream and downstream sides of the shocks, are assigned to the shock-points, which</p> <p>may be switched to common- and back to shock-points,</p> <p>a feature that allows to vary the length of the existing shocks and/or make new shock-branches appear.</p> <p>This paper illustrates the algorithmic features of this new technique and presents the results obtained when simulating both steady and un-steady, two-dimensional flows.</p>

1. As suggested by the reviewer, we have used the discretization error to measure the order-of-accuracy, rather than Richardson Extrapolation. The number of grid levels has also been raised to four. The revised text has been highlighted in red.
2. As indicated by the same reviewer, some text had been accidentally erased in what is now page 27 and has now been replaced.

Shock-waves are ``fitted'' on unstructured meshes using an unstructured Finite Volume method. Certain sub-sets of grid-points behave as either shock-points or ordinary grid-points. By switching on and off selected shock-points, it is possible to make shock waves appear, or dis-appear. It is then possible to simulate simple time-dependent shocked-flows involving topological changes in the shock-pattern.

# An embedded shock-fitting technique on unstructured dynamic grids

Dongyang Zou<sup>a</sup>, Aldo Bonfiglioli<sup>b</sup>, Renato Paciorri<sup>c</sup>, Jun Liu<sup>d,e</sup>

<sup>a</sup>*China Aerodynamics Research and Development Center, Mianyang 621000, China*

<sup>b</sup>*Scuola di Ingegneria, Università della Basilicata, Viale dell' Ateneo Lucano 10, 85100 Potenza, Italy*

<sup>c</sup>*Dipartimento di Ingegneria Meccanica e Aerospaziale, Università di Roma, La Sapienza, Via Eudossiana 18, 00184 Roma, Italy*

<sup>d</sup>*State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116024, P.R. China*

<sup>e</sup>*School of Aeronautics and Astronautics, Dalian University of Technology, Dalian 116024, P. R. China*

---

## Abstract

In this paper, a new shock-fitting technique based on unstructured dynamic grids is proposed to improve the performances of the unstructured “boundary” shock-fitting technique developed by Liu and co-workers in [1, 2]. The main feature of this new technique, which we call the “embedded” shock-fitting technique, is its capability to insert or remove shocks or parts thereof during the calculation. This capability is enabled by defining subsets of grid-points (mutually connected by lines) which behave as either “common”- or “shock”-points, shock-waves being made of an ordered collection of shock-points. Two different sets of flow variables, corresponding to the upstream and downstream sides of the shocks, are assigned to the shock-points, which may be switched to common- and back to shock-points, a feature that allows to vary the length of the existing shocks and/or make new shock-branches appear. This paper illustrates the algorithmic features of this new technique and presents the results obtained when simulating both steady and un-steady, two-dimensional flows.

*Key words:*

shock-fitting, unstructured dynamic grid, shock wave, numerical simulation

---

## 1. Introduction

The interest towards shock-fitting (or front-tracking) techniques has seen a sharp rise within the CFD community over the last few years. This is both because of the difficulties encountered by CFD

---

*Email address:* `aldo.bonfiglioli@unibas.it` (Aldo Bonfiglioli)

developers, see e.g. [3, 4], when trying to overcome some of the intrinsic limitations of the shock-capturing paradigm, but also because of the promising results obtained by the authors [1, 2, 5–8] over the last 10 years by coupling shock-fitting algorithms and unstructured meshes. In the mid 1990s Parpia and Parikh [9], van Rosendale [10] and Camarero and co-workers [11] developed different approaches all aimed at aligning some of the edges of an unstructured, adaptive triangulation to the various discontinuities that may occur within a flow-field, but without making explicit use of the Rankine-Hugoniot (R-H) jump relations. Similar ideas have re-surfaced very recently in the context of Discontinuous Galerkin Finite Element methods (DG-FEM). The DG-FEMs independently developed by two different research teams, [12–15] and [16–20], simultaneously solve for the location of the grid-points, in addition to the flow-variables, so as to constrain certain edges to be aligned with the discontinuities. It is noted that numerical methods that employ a data representation which is discontinuous across the element interfaces, which is the case with DG-FEM, but also with most Finite Volume (FV) methods, naturally lend themselves to be used to fit discontinuities as a collection of edges of the geometrical tessellation of the computational domain. Indeed, both the aforementioned DG-FEMs avoid the addition of artificial viscosity along the discontinuities (i.e. their capture) by ensuring that some of the edges of the mesh are aligned with the discontinuities that are present in the flow-field.

In contrast to the aforementioned approaches, the unstructured, shock-fitting algorithms developed by the authors bear closer resemblance with those developed, starting in the early 1970s, by Gino Moretti and his collaborators, using structured meshes. These *traditional* shock-fitting techniques on structured meshes evolved following two different routes.

“Boundary” shock-fitting, whereby the fitted-shock is made to coincide with one of the *boundaries* of the computational domain [21–24] is algorithmically simple, but only allows to deal with shock-patterns that exhibit a simple topology.

“Floating” shock-fitting [25, 26], whereby the fitted-shock is made up of a collection of shock-points (the intersections of the “fitted” shock-front with the grid-lines) which can *float* over a fixed background mesh. “Floating” shock-fitting allows to deal with relatively complex shock-topologies, but, when implemented on structured grids, it requires special provisions to avoid taking differences across the discontinuities and it is therefore somewhat harder to code; see [27] for a recent example.

The unstructured-grid approach developed by Paciorri and Bonfiglioli in [5–8] has features of

both the “boundary” and “floating” versions of Moretti’s technique: the discontinuities are double-sided *boundaries* of zero thickness which are free to *float* over a background triangulation and are treated as internal *boundaries* by the vertex-centered CFD code used to discretize the governing PDEs.

The focus of this paper is however on recent developments of a similar unstructured shock-fitting approach which has been recently proposed by Liu and co-workers [1, 2], who combined the traditional “boundary” shock-fitting with unstructured dynamic grids. Similarly to its structured-grid counterpart, the various shock-fronts are made to coincide with some of the boundaries of an unstructured, multi-block mesh. The shock-speed and shock-boundary conditions are obtained by solving the R-H jump relations and an unstructured, dynamic-grid technique is used not only to move the s to the new locations dictated by the shock-displacement velocity, but also to adjust the position of the internal grid-points so as to guarantee the quality of the entire computational mesh. From a coding view-point, the technique proposed by Liu and co-workers [1, 2], which we call unstructured “boundary” shock-fitting, requires little changes to be made to the existing CFD solver and it is significantly more versatile than its structured-grid counterpart because unstructured, dynamic-grids allow to handle fairly large deformations of the computational domain. Moreover, and unlike Paciorri and Bonfiglioli’s approach [5–8], the unstructured boundary shock-fitting technique described in [1, 2] has been coupled with cell-centered, rather than vertex-centered solvers.

The capability of the unstructured “boundary” shock-fitting technique to solve steady problems featuring fairly complex embedded shock waves has been demonstrated using various examples in [1, 2]. Even so, the unstructured “boundary” shock-fitting technique turns out to be unsuited to deal with un-steady flows featuring topological changes of the shock pattern.

In this paper we describe an alternative approach, which we call unstructured “embedded” shock-fitting technique. In contrast to the “boundary” shock-fitting technique described in [1, 2], where the shocks and the corresponding shock-points remain such for the entire duration of the simulation, the technique proposed here allows to define lines (i.e. sets of mutually connected edges of the triangulation) made of “special” grid-points which may switch from “shock” to “common” and vice-versa. These “special” grid-points can be selectively activated and de-activated, based on the local flow features, so as to allow the insertion of a new shock or the removal of an existing

one. This feature allows to follow the temporal evolution of the shock topologies that may occur in several simple unsteady shocked-flows, but it is not capable of dealing with all different kinds of topological changes that may take place in a compressible, unsteady flow. More precisely, the shock topologies that can be dealt with using the technique described here are all those that can be derived from the activation (or de-activation) of grid-points belonging to lines (as previously defined) pre-set at the beginning of the simulation. Moreover, to be able to cope with flow-fields featuring time-dependent changes of the shock topology, it is also necessary to supply a criterion for activating and de-activating selected shock-points, i.e. it is necessary to detect when and where the topological change takes place. This issue is also discussed in Sect. 2.4 and a simple solution capable of handling some, but not all, cases is described.

The paper is organized as follows. The algorithm is described in Sect. 2, where a comparison with previous work by the authors is also made. Numerical results are presented in Sect. 3 and include simulations of both steady and un-steady, two-dimensional flows.

## 2. Shock-fitting algorithms for unstructured grids

Shock-fitting techniques consist in two basic ingredients: computing and tracking.

The term “tracking” refers to the task of keeping track of the motion of the various discontinuities (either shocks or slip-lines) and the interaction points that are present within the flow-field.

The term “computing” refers to the task of approximating the governing PDEs by means of a gas-dynamics (CFD) solver within those regions of the computational domain that are bounded by the discontinuities.

Regardless of the CFD solver being used, in the unstructured-grid framework the fitted shock-waves<sup>1</sup> are represented as a sequence of shock-edges that are mutually connected at the shock-points and the displacement of the fitted discontinuities is computed by applying the R-H jump relations either within the shock-points or along the shock-edges.

Depending on whether the CFD solver uses a cell-centered or vertex-centered storage of the unknowns, the implementation of the “tracking” ingredient can be significantly different and so is

---

<sup>1</sup>for simplicity, we shall hereafter refer to shock-waves in the two-dimensional space, but the technique is also capable of dealing with other kinds of discontinuities as well as in the three-dimensional space.

the amount of coding that is required to enable the CFD solver to work in conjunction with the tracking algorithm. A couple of representative examples will be given hereafter.

In the unstructured shock-fitting algorithm proposed by Paciorri and Bonfiglioli [5], which has been used in conjunction with three different vertex-centered, Residual Distribution CFD solvers [28–30], the shock-points are inserted into the mesh by locally re-meshing the computational domain around the moving discontinuity. This is obtained by: *i*) splitting every shock-point into two distinct grid-points, corresponding to the upstream and downstream states, *ii*) cutting the grid along the shock-edges that connect the shock-points and *iii*) overlapping the upstream nodes on the corresponding downstream nodes. The R-H relations are applied within each shock-point to update its downstream state and compute the local shock-speed. In this approach, the fitted-discontinuity is treated by the CFD code as a boundary of zero thickness that moves throughout the computational domain.

The shock-fitting implementation proposed by Liu and co-workers [1, 2] and further improved in the present paper has been developed for being used in conjunction with cell-centered, FV solvers. In the tracking phase, the R-H jump relations are either applied along the shock-edges, as done in [1, 2], or within the shock-points, which is the case of the present contribution. In either case, the R-H relations provide the shock-speed, whereas the flux through the shock-edge is computed using the upstream state and shock-speed. Once the flux through the shock-edges is available, the update of the (cell-averaged) states of those cells that abut on the discontinuity is done in exactly the same way as for any other cell. The fitted-shocks are then moved in a Lagrangean manner, using the shock-speed available within all shock-points.

It is noted that the use of either a vertex- or cell-centered CFD solver has also an implication on the way the tracking algorithm modifies the mesh to account for the motion of the discontinuity. Indeed, in the tracking algorithm proposed for vertex-centered solvers, it is possible to change the grid connectivity without having to interpolate the flow states, because the dependent variables are stored within the grid-points of the mesh. On the contrary, in the tracking algorithm developed for cell-centered solvers, a change in the mesh-connectivity requires interpolating the dependent variables between the two grids, because the geometric location of the cell centers (where the dependent-variables are stored) changes when changing the mesh connectivity. For this reason, in the shock-fitting algorithm for vertex-centered solvers the shock motion can be obtained by



means of a local re-meshing, whereas in the shock-fitting algorithm for cell-centered solvers the shock motion is obtained by means of a deformation of the entire mesh, without changing the connectivity. Actually, a complete re-meshing of the computational domain is possible and it is performed when the mesh quality deteriorates, but it should not be repeated at each time step to avoid introducing excessive interpolation errors.

The aforementioned difficulty in changing the grid connectivity at each time step considerably limits the possibility to deal with transient flows which experience changes in the number of shocks and/or interaction points, i.e. changes in the flow topology. Indeed, in all test-cases reported in [1, 2] the flow topology remains unchanged. In the present paper this restriction can be overcome in some cases: the “embedded” shock-fitting which is described here is capable of dealing with changes in the number of shocks and/or interaction points without introducing changes in the mesh-connectivity at each time step.

### *2.1. Grid layout*

In this paragraph we describe how we have defined the fitted discontinuities using an unstructured mesh. To this end, let us consider the rectangular region shown in Fig. 1 which has been tessellated using unstructured triangular cells. The solid and dashed edges describe the triangulation at different time instants and the dashed-dotted red lines are the fitted shocks that connect some of the grid-points. Two different types of grid-points, respectively shown in Fig. 1 using green and red circles, have been defined: common grid-points, labeled with the letter **C**, and shock-points, labeled with the letter **S**. Despite the use of a cell-centered storage in our FV solver, nodal (i.e. within the grid-points of the mesh) values of the dependent variables are also needed both by the CFD solver, see Sect. 2.3, and the shock-fitting algorithm. The difference between the **C** and **S** points is that a single set of dependent variables is computed within each **C** point using data of all cells that share that **C** point, whereas two different sets of variables, corresponding to the upstream and downstream sides of the discontinuity are computed within an **S** point using data of only those cells that are located on the upstream, resp. downstream side of the discontinuity; further details will be given in Sect. 2.2.2. A grid-velocity vector is also defined within every grid-point: the velocity of a shock-point is computed by enforcing the R-H relations, as will be described in Sect. 2.2.2, whereas the velocity of the common grid-points are computed using the grid-deformation algorithm described in Sect. 2.2.4. It is important to underline that in the

unstructured “boundary” shock-fitting technique described in [1, 2] the **S**-points remain such for the entire the simulation, whereas in the “embedded” technique presented herein each **S**-point can change its “nature”, i.e. be “downgraded” to a **C**-point and then switched back to an **S**-point. This capability only applies to grid-points that have been initially set as **S**-points or to grid-points that belong to lines (i.e. sets of mutually connected edges of the triangulation) also pre-set at the beginning of the simulation.

The dual nature of the shock-points provides a certain flexibility in dealing with time-dependent flow topologies; this particular aspect will be covered in more depth in Sect. 2.4.

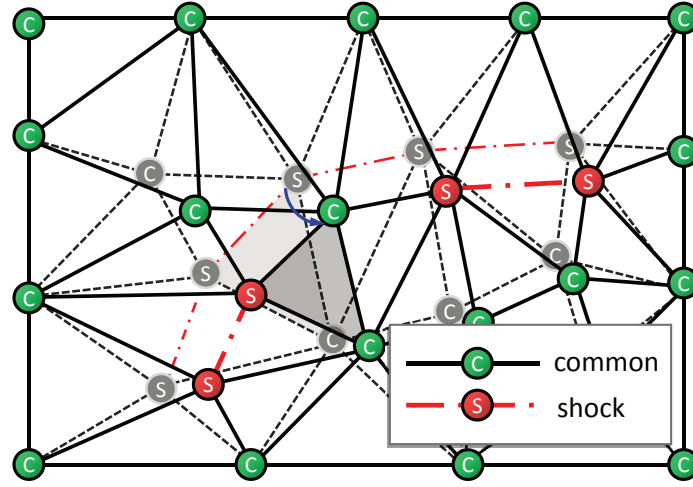


Figure 1: Embedded shock-fitting algorithm: the blue arrow shows a shock-point whose status changes from **S** to **C**.

## 2.2. Motion of the shock-points

Suppose that the CFD solver has just completed the task of computing (i.e. updating) the solution at time level  $n + 1$  within all cells of the mesh, using the solution, mesh and grid velocities available at time level  $n$ . At this stage, tracking comes into play which amounts to enforce the R-H relations within each shock-point, thus obtaining the new shock-speed. The new shock-speed allows to displace the shock-points and obtain a new mesh at time  $n + 1$ . These various steps will be described in detail in the following paragraphs.



tangent to the  $k^{th}$  shock-edge:

$$M_{\tau k} = \frac{\mathbf{u}_{dk} \cdot \boldsymbol{\tau}_k}{a_{dk}} \quad \text{where} \quad \boldsymbol{\tau}_k = \frac{(A - O_k)}{|(A - O_k)|} \quad (3)$$

In Eq. (3)  $\mathbf{u}_{dk}$  and  $a_{dk}$  respectively denote the shock-downstream values of the fluid velocity and sound speed at the centroid  $O_k$  of the  $k^{th}$  edge; these two quantities are extrapolated from the adjacent cell located on the downstream side of the shock.

Depending on the value of  $M_{\tau k}$ , three different situations may arise:

- (i):  $-1 < M_{\tau k} < 1$ : a disturbance originating in  $O_k$  propagates along the shock line as a disturbance in a subsonic flow. In this situation, the disturbance will be able to reach shock-point  $A$  and, therefore, we set  $\chi_k = 1$ .
- (ii):  $M_{\tau k} \geq 1$ : a disturbance originating in  $O_k$  propagates along the shock line as a disturbance moving in a supersonic flow in the same direction. Therefore, shock-point  $A$  belongs to the domain of influence of the  $k$ -th shock edge, so that we set  $\chi_k = 1$ .
- (iii):  $M_{\tau k} \leq -1$ : the disturbance originating in  $O_k$  moves in the direction opposite to  $A$ . Therefore, shock-point  $A$  is outside the domain of influence of the  $k^{th}$  shock-edge and, therefore, we set  $\chi_k = 0$ .

All three aforementioned scenarios can be accounted for by defining  $\chi_k$  as follows:

$$\chi_k = \begin{cases} 0, & \text{if } M_{\tau k} \leq -1 \\ 1, & \text{if } M_{\tau k} > -1 \end{cases}. \quad (4)$$

### 2.2.2. Calculation of the shock-upstream and shock-downstream states within a shock-point

In order to apply the R-H relations at shock-point  $A$ , a pair of shock-upstream  $\mathbf{U}_u = (\rho_u, \mathbf{u}_u, p_u)^T$  and shock-downstream  $\mathbf{U}_d = (\rho_d, \mathbf{u}_d, p_d)^T$  states is needed in addition to the shock-normal unit vector which has already been computed as described in Sect. 2.2.1.

Since the CFD solver used in this work is cell-centered, the shock-upstream and shock-downstream nodal values at shock-point  $A$  have to be recovered from the cell-averaged states of the cells that share  $A$ . Clearly, this must be done while avoiding interpolating across the discontinuity.

By reference to Fig. 2, one sees that the shock edges meeting at shock-point  $A$  split the neighboring cells into an upstream (subscript  $u$ ) and a downstream (subscript  $d$ ) region. The shock-upstream state  $\mathbf{U}_u$  in  $A$  should only be computed using data averaged from the neighboring cells

that fall within the upstream region and similarly for the downstream states  $\mathbf{U}_d$ . This can be accomplished using the following weighted averages:

$$\mathbf{U}_u = \frac{\sum_{k=1}^{N_d} c_k \mathbf{U}_k}{\sum_{k=1}^{N_d} c_k}, \quad \mathbf{U}_d = \frac{\sum_{k=1}^{N_u} c_k \mathbf{U}_k}{\sum_{k=1}^{N_u} c_k} \quad (5)$$

where the summations in Eq. (5) range over the  $N_d$ , resp.  $N_u$ , triangles that belong to the shock-downstream, resp. shock-upstream, region and the weighting coefficient:

$$c_k = \frac{\chi_k}{|(A - G_k)|}, \quad (6)$$

is similar to the  $\alpha_k$  weight of Eq. (2), but it is defined on a cell-basis, rather than per shock-edge. In Eq. (6)  $G_k$  denotes the centroid of the  $k^{th}$  triangular cell.

### 2.2.3. Enforcement of the jump relations

As already mentioned, each shock-point is characterized by a pair of upstream and downstream states and a shock-speed:  $w = \mathbf{w} \cdot \mathbf{n}$ . It can be safely assumed that the shock-upstream states have been correctly computed using Eq. (5) since the upstream side of the shock behaves like a supersonic outflow boundary which only requires information propagating from the upstream side of the shock.

The situation is however different on the shock-downstream side. Here we assume that the following Riemann variable:

$$R_d = \frac{a_d}{\delta} - \mathbf{u}_d \cdot \mathbf{n}, \quad \text{where} \quad \delta = \frac{\gamma - 1}{2} \quad (7)$$

which is conveyed from the shock-downstream region towards the shock, is the only quantity that has been correctly computed using the available shock-downstream state  $\mathbf{U}_d$ .

The shock-speed  $w$  within the shock-point  $A$  is computed by combining the four (in 2D) R-H relations and Eq. (7).

Using the upstream, shock-normal component of the Mach number:

$$M_{u,rel} = \frac{\mathbf{u}_u \cdot \mathbf{n} - w}{a_u} \quad (8)$$

it is possible to re-write Eq. (7) and three of the four jump relations as follows:

$$\begin{aligned}
p_d' &= \frac{p_u (2\gamma M_{u,rel}^2 - (\gamma - 1))}{\gamma + 1} \\
\rho_d' &= \frac{\rho_u (\gamma + 1) M_{u,rel}^2}{(\gamma - 1) M_{u,rel}^2 + 2} \\
\frac{\rho_d' (\mathbf{u}_d' \cdot \mathbf{n} - \mathbf{u}_u \cdot \mathbf{n} + a_u M_{u,rel})^2}{\gamma p_d'} &= \frac{(\gamma - 1) M_{u,rel}^2 + 2}{2\gamma M_{u,rel}^2 - (\gamma - 1)} \\
\frac{a_d'}{\delta} - \mathbf{u}_d' \cdot \mathbf{n} &= \frac{a_d}{\delta} - \mathbf{u}_d \cdot \mathbf{n}
\end{aligned} \tag{9}$$

The unknowns that appear in Eq. (9) are the shock-speed  $w$  and the updated values of pressure  $p_d'$ , density  $\rho_d'$  and shock-normal velocity component  $\mathbf{u}_d' \cdot \mathbf{n}$ , since  $a_d' = \gamma p_d' / \rho_d'$ . These four equations can be combined into a single non-linear, algebraic equation in the unknown  $M_{u,rel}$ :

$$\frac{1}{\delta} \sqrt{(\gamma M_{u,rel}^2 - \delta)(\delta M_{u,rel}^2 + 1)/M_{u,rel}^2 + (M_{u,rel}^2 - 1)/M_{u,rel}} = \frac{\gamma + 1}{2a_u} (R_d + \mathbf{u}_u \cdot \mathbf{n}). \tag{10}$$

The left-hand side of Eq. (10) varies almost linearly with  $M_{u,rel}$  so that the root of Eq. (10) can be readily obtained by means of Newton's root-finding algorithm using the value of  $M_{u,rel}$  at the current time-level  $t$  as the initial guess. Once  $M_{u,rel}$  is available, the shock-speed magnitude  $w$  follows from Eq. (8) and will be used to move the discontinuity. The updated set of shock-downstream values  $\mathbf{V}_d'$  is computed from Eq. (9) and will only be used to compute the cell-gradient, as described in Sect. 2.3.2.

#### 2.2.4. Unstructured dynamic mesh technique

The position of the shock-front at time level  $n + 1$  can be easily computed by displacing all shock-points using the shock-speed  $\mathbf{w} = w\mathbf{n}$  and shock-normal computed as described in Sect. 2.2.1 and 2.2.3. However, if only the shock-points were moved, invalid cell-shapes would quickly appear within the computational mesh. This can be avoided by moving all grid-points in the mesh using an unstructured, dynamic-mesh technique which was originally developed for problems involving the motion of a body within a fluid. In the present framework, the technique is employed to track the motion of the fitted shock waves which behave like moving bodies, so that the entire computational mesh deforms, without changing the mesh connectivity, to conform to the instantaneous position of the body or the fitted-shock. As shown in Fig. 3, all edges in the mesh are modeled using linear springs and Hooke's law determines the force at every grid-point  $i$  exerted by the surrounding

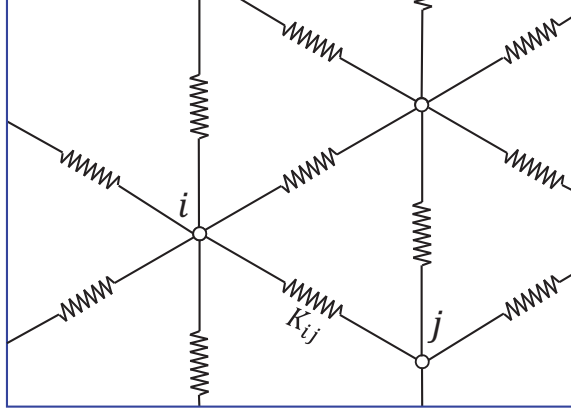


Figure 3: Mesh deformation algorithm: sketch of the spring analogy.

grid-point  $j$ :

$$\mathbf{R}_i = \sum_{j=1}^{N_i} K_{ij}(\mathbf{O}_j - \mathbf{O}_i). \quad (11)$$

In Eq. (11)  $N_i$  is the number of distance-1 neighbors of grid-point  $i$ , and  $K_{ij}$  is the stiffness of the spring connecting grid-points  $i$  and  $j$ . When the initial computational grids are generated, the forces at every grid-point can be computed, by setting  $K_{ij}$  inversely proportional the length of the edge connecting  $i$  and  $j$ . As time elapses, grid-points on the outer boundary of the mesh are held fixed, whereas the instantaneous locations of the grid-points located on the inner boundaries (either bodies or fitted discontinuities) are prescribed. In order to hold the force  $\mathbf{R}_i$  at every grid-point  $i$  constant during the deformation process, the static equilibrium equations are iteratively solved at each interior grid-point  $i$  to provide the nodal displacements at each time step.

### 2.3. Cell-centered FV CFD solver

The MCFS (Mixture of Capturing and Fitting Shocks) code relies on a Finite Volume (FV) discretization of the Arbitrary Lagrangian Eulerian (ALE) formulation of the Euler equations. Details concerning the FV method can be found in [32], whereas the specific ALE formulation, which satisfies the Geometric Conservation Law, is described in [33]. The MCFS solver uses a cell-centered storage of the dependent variables, i.e. the array  $\mathbf{Q}_i^n$  represents the cell-averaged state within each triangular control-volume  $\Omega_i$ :

$$\mathbf{Q}_i^n = \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{Q}(x, y, t) \, dx \, dy$$

at time  $t = n \Delta t$ .

In FV methods, the solution  $\mathbf{Q}(x, y, t)$  experiences a jump along the interface between two adjacent cells, such as edge  $k$  in Fig. 4, which is shared by the control-volumes  $i$  and  $j$ . In order to get a single-valued approximation of the inviscid flux across the edge common to cells  $i$  and  $j$ , so-called numerical flux functions are used. Two different approaches have been used, depending on whether the edge is a shock-edge, i.e. its end-points are both shock-points, or it is a “common” edge; the two approaches will be described in the next two paragraphs.

### 2.3.1. Fluxes across shock edges

A method inspired by the idea of flux-vector splitting methods has been used to compute the inviscid flux  $\mathbf{F}_n$  through the shock-edge of normal  $\mathbf{n}_k$ :

$$\mathbf{F}_n = \begin{pmatrix} \varphi_1 \\ \varphi_1 \mathbf{u} + p \mathbf{n}_k \\ \varphi_1 E + p \varphi_2 \end{pmatrix} \quad (12)$$

In Eq. (12) standard notation has been used for pressure,  $p$ , velocity  $\mathbf{u}$  and total internal energy  $E$ . The grid velocity has also been accounted for by defining  $\varphi_1 = \rho(\mathbf{u} - \dot{\mathbf{x}}_c) \cdot \mathbf{n}_k$  and  $\varphi_2 = \dot{\mathbf{x}}_c \cdot \mathbf{n}_k$  where the velocity  $\dot{\mathbf{x}}_c$  of the shock-edge  $k$  is the arithmetic average of the shock-speed of its end-points:

$$\dot{\mathbf{x}}_c = \frac{1}{2} \sum_{i=1}^2 \mathbf{w}_i \quad (13)$$

In flux-vector splitting methods, the flux is split into a forward flux  $\mathbf{F}_n^+$  and a backward flux  $\mathbf{F}_n^-$ , such that:

$$\mathbf{F}_n = \mathbf{F}_n^+ + \mathbf{F}_n^-$$

When the shock-normal Mach number:

$$M_n = \frac{\mathbf{u} \cdot \mathbf{n}}{a}$$

is larger than one, then  $\mathbf{F}_n^+ \equiv \mathbf{F}_n$ ; conversely, when  $M_n \leq -1$ , then  $\mathbf{F}_n^- \equiv \mathbf{F}_n$ . Since the shock-normal Mach number is larger than one along the shock-upstream side of a fitted shock, it follows that the flux through a shock-edge can be completely determined using values located along its upstream side

$$\mathbf{F}_n = \mathbf{F}(\mathbf{U}_u)$$



where  $\mathbf{U}_u$  is the extrapolated value of the dependent variables at the centroid of the shock-upstream edge.

### 2.3.2. Fluxes across common edges

When dealing with common edges, any numerical flux function can be used to compute an approximation to the inviscid flux through that edge. Regardless of the choice being made, the numerical flux function depends upon the unit vector,  $\mathbf{n}_k$ , normal to edge  $k$ , and the two different states, labeled  $\mathbf{Q}_L$  and  $\mathbf{Q}_R$  in Fig. 4, that can be reconstructed at the centroid  $O_k$  of edge  $k$  from cell  $i$ , resp.  $j$ :

$$\mathbf{Q}_L = \mathbf{Q}_i + \nabla \mathbf{Q}_i \cdot (O_k - O_i) \quad (14a)$$

$$\mathbf{Q}_R = \mathbf{Q}_j + \nabla \mathbf{Q}_j \cdot (O_k - O_j) \quad (14b)$$

The reconstruction defined by Eq. (14) can be either piecewise-constant, whenever the term involving the gradient in Eq. (14) is neglected, or piecewise-linear, resp. leading to first- and second-order-accurate reconstructions. In this latter case, the (cell-wise constant) gradient  $\nabla \mathbf{Q}_i$  that appears in Eq. (14a), and similarly for  $\nabla \mathbf{Q}_j$ , can be computed using Green-Gauss' theorem:

$$\nabla \mathbf{Q}_i = -\frac{1}{2|\Omega_i|} \sum_{k=1}^3 \hat{\mathbf{Q}}_k \mathbf{n}_k l_k. \quad (15)$$

In Eq. (15) the subscript  $k$  spans the 3 cell-edges of the  $i^{th}$  triangular control volume, whose area is  $|\Omega_i|$ ;  $l_k$  is the length of the  $k^{th}$  cell-edge and  $\mathbf{n}_k$  is the outward unit vector normal to the edge and  $\hat{\mathbf{Q}}_k$  denotes the set of dependent variables within the grid-point opposite edge  $k$ .

Nodal values in the grid-points are computed as the inverse-distance weighted-average of the cell-averaged values within all the  $N_c$  cells surrounding grid-point  $n$ :

$$\hat{\mathbf{Q}}_n = \frac{\sum_{m=1}^{N_c} c_m \mathbf{Q}_m}{\sum_{m=1}^{N_c} c_m} \quad \text{where} \quad c_m = |(G_m - O_n)|^{-1}. \quad (16)$$

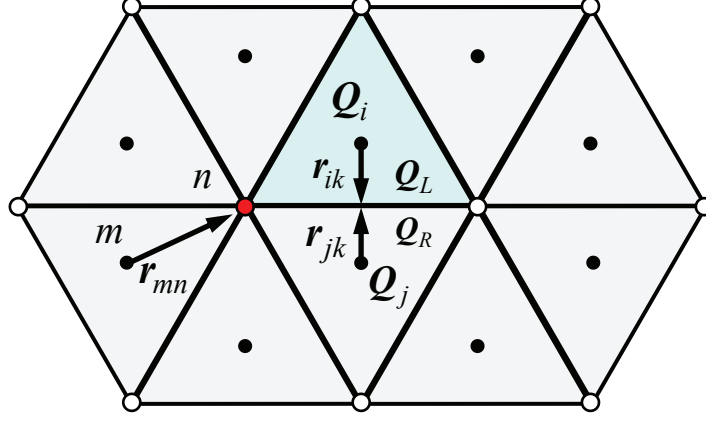


Figure 4: FV algorithm: reconstruction of the dependent variables within the grid-points of the mesh.

#### 2.4. Topological changes

This step represents the most innovative part of the algorithm presented in this paper. Indeed, in the present implementation of the “embedded” technique, the nature of some specific grid-points may change in time. This functionality has been implemented using a simple trick. The lines made up of grid-points that can behave as either **S**- or **C**-points are treated as (inner) boundaries during the entire calculation. Two different kinds of boundary conditions can be applied withing these grid-points: either by enforcing the R-H relations (**S**-point), as described in Sect. 2.2.3, or using a simple through-flow condition whereby the downstream and upstream states are set equal (**C**-point)<sup>3</sup>. By doing so, it is possible to treat several topological variations that involve changes in the number of shocks. This can be illustrated by reference to Fig. 1, which shows a flow-field at time  $t$  with a shock wave made up of five shock-points. The cell-edges formed by these shock-points are automatically defined as shock-edges so that the shock wave has been discretized into four shock-edges, shown using dash-dotted lines in Fig. 1. Suppose that at a later time  $t' > t$  (solid line), a portion of the shock wave has weakened to the point that the shock wave must be split into two parts. To deal with this topological change it is only needed to change the status of the shock-point that has disappeared from **S** into **C**. By doing so, the split up of a single shock-wave into two distinct shock-waves can be easily dealt with without changing the topology of the

---

<sup>3</sup>The through-flow condition is a boundary condition widely used in structured multi-block solvers to mutually connect different grid-blocks so as to allow un-perturbed flow across adjacent block-boundaries.

computational mesh.

To allow **S**-points to become **C**-points and vice-versa, it is necessary to introduce a criterion that allows to detect the formation of a shock or its disappearance. This is because **S**-point are “downgraded” to **C**-points whenever the shock becomes weak enough, but they may regain their double-valued **S**-state if the shock strength increases again at a later time. Following [25], the local shock-strength is measured within all shock-points using the shock-upstream, relative Mach number ( $M_{u,rel}$ ), defined in Eq. (8) and computed by solving Eq. (10). More precisely, an **S**-point is converted into a **C**-point whenever the following condition is met:

$$M_{u,rel} < 1 + \epsilon. \quad (17)$$

The parameter  $\epsilon$  in Eq. (17) is an arbitrary threshold, which has been set equal to 0.03.

The criterion (17) is of wide applicability, because it is capable of detecting the weakening of a shock, i.e. that an **S**-point should become a **C**-point. In the opposite situation, i.e. when a **C**-point becomes a shock-point again, we have also used the same criterion (17) with the direction of the inequality reversed. By doing so, it has been possible to deal with the last two test-cases described in this paper. In general, however, shock-formation is the result of the coalescence of compression waves and, therefore, the construction of a general criterion should be based upon the analysis of the slopes of the characteristic lines in the region where coalescence occurs. An example of this kind of detection procedure has been described by Moretti in [25], where it is applied to unsteady one-dimensional flows. Of course, its extention to two- or three-dimensional flows is not trivial, but Moretti’s paper is certainly going to be a useful guideline for future developments.

### 3. Applications

The current capabilities of our “embedded” shock-fitting algorithm will be hereafter demonstrated by reference to a number of test-cases, including steady and unsteady two-dimensional flows, governed by the Euler equations. More precisely, the first two test-cases involve steady flows and demonstrate the basic features of the methodology, without accounting for changes in shock-wave topology. On the contrary, the last two test-cases, which involve unsteady shocked flows, show the ability of the new technique to handle simple changes of the shock-topology, thus also demonstrating its current limitations.

### 3.1. NACA 0012 airfoil

Results are presented first for a well documented external flow test-case, namely the two dimensional, inviscid, transonic flow past the NACA 0012 airfoil at  $\alpha_\infty = 0^\circ$  angle of attack and free-stream Mach number equal to  $M_\infty = 0.8$ .

The computational mesh, which is shown in Fig. 5, is made of 5294 triangular cells and 2728 grid-points, 102 of these have been placed along the airfoil's profile.

For the given free-stream Mach number a nearly normal shock-wave forms on both sides of the profile at about mid-chord, see Figs. 6a and 6b. To be able to draw a comparison between shock-fitting and shock-capturing simulations on nearly identical grids, the shock has been fitted on the upper side of the airfoil and captured on its lower side, as can be clearly seen in Fig. 6a. Some further details concerning the shock-fitting calculation will now be given.

The shock-fitting flow-field, as well as the shape of the fitted-shock, have been initialized using a shock-capturing calculation of the entire flow-field, which is not shown here for brevity. Figure 5 shows the initial and final shape of the fitted shock. At the beginning of the simulation, six adjacent grid-points, shown using filled circles in Fig. 5, have been marked as shock-points; their initial location has been set by picking up grid-points located inside the thickness of the captured shock and mutually connected by edges of the triangulation. These shock-edges have been marked in blue in Fig. 5.

As the calculation is advanced in time, the shock-shape changes due to the displacement of the shock-points which, in turn, induce the deformation of the grid. It is important to stress that in the present test-case the initially selected shock-points remain **S**-points during the entire simulation. Figure 5 shows both the initial mesh, which is symmetric with respect to the airfoil's chord, and a zoom around the foot of the shock of the final mesh, i.e. the mesh obtained once the steady solution has been reached and, therefore, all shock-velocities have vanished. It is worth mentioning that the grid has only been deformed without changing its topology.

An ad-hoc treatment is required for the end-point (marked EP in Fig. 5) where the fitted-shock terminates. EP is a shock-point, but in contrast to all other shock-points (including the one on the airfoil's wall at the foot of the shock) it is impossible to unambiguously assign the triangles surrounding EP to either the shock-upstream or shock-downstream regions; this can be done for the only two triangles (see Fig. 5) that are attached to the pair of shock-edges bounded by EP and

its neighboring shock-point. This difficulty has been circumvented by setting the upstream, resp. downstream states in EP equal to the cell-averaged states of the only two triangles that can be uniquely identified as being shock-upstream and shock-downstream. Similarly, the shock-normal direction in EP has been set equal to the unit vector normal to the only shock-edge grid-point EP belongs to.

Concerning the shock-point at the foot of the shock, its shock-normal has to be tangential to the airfoil's profile.

When comparing shock-fitting against shock-capturing solution using Fig. 6a, it is clear that the shock-fitting approach on a coarse and un-adapted mesh returns a much more realistic shock-thickness than shock-capturing. Not only the shock-capturing calculation predicts an un-physically large shock-thickness, it also completely misses the so-called Zierep [34] singularity, visible in Fig. 6b, that occurs on the downstream foot of the shock which, in contrast, is picked up when the shock is fitted.

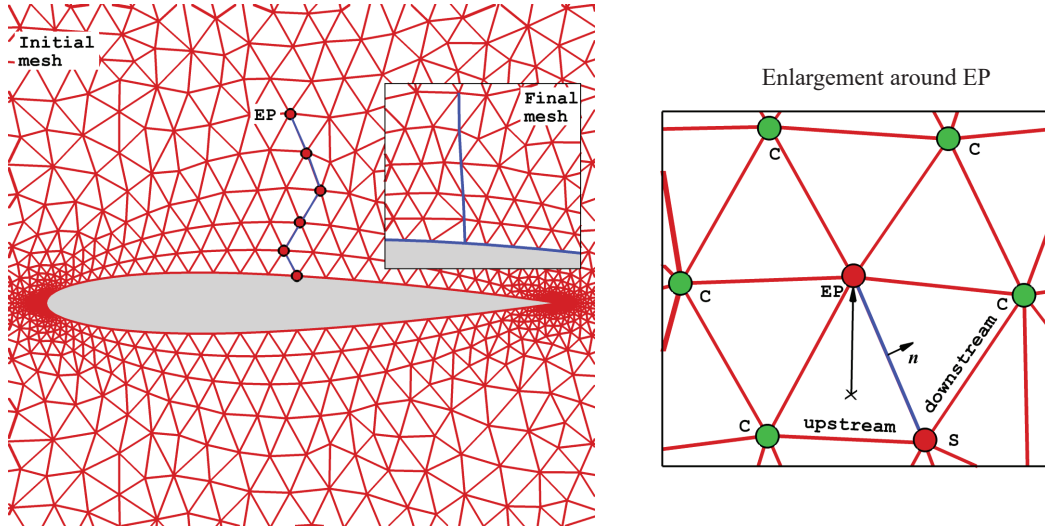
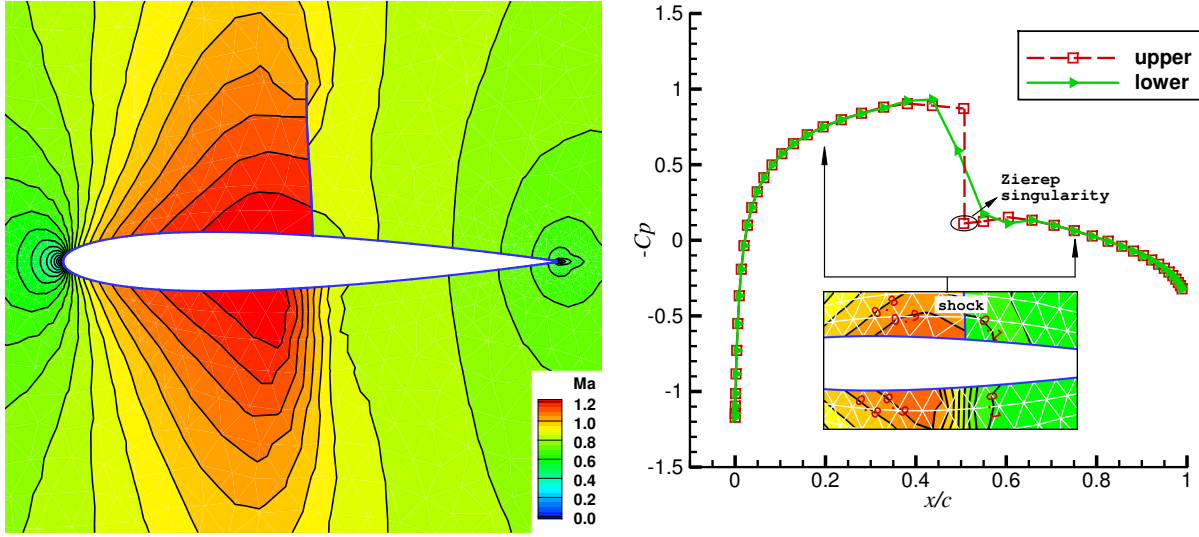


Figure 5: NACA 0012: Computational mesh and sketch of the grid layout around shock-point EP.



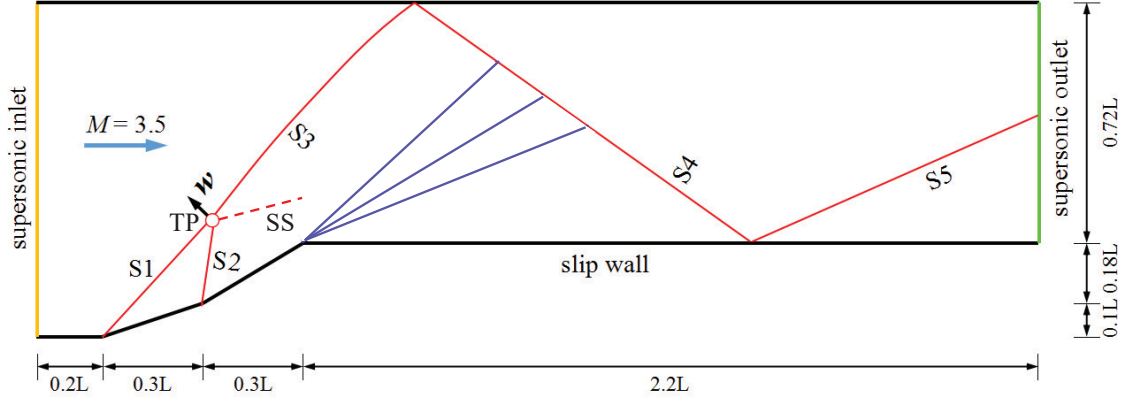
(a) Mach iso-contour lines.

(b) Pressure coefficient distributions along the profile: shock-fitting on the upper side, shock-capturing on the lower.

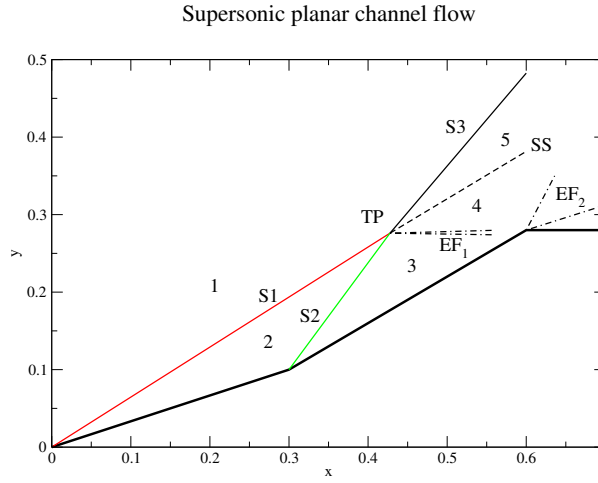
Figure 6: NACA 0012,  $M_\infty = 0.8$ ,  $\alpha_\infty = 0^\circ$ .

### 3.2. Supersonic planar channel flow

The second test-case that has been analyzed consists in a supersonic flow (the inlet Mach number being  $M_1 = 3.5$ ) through a planar, variable area duct. The geometry of the duct is shown in Fig. 7a: the two constant-area portions of the duct are joined through a double ramp, giving an outlet-to-inlet area ratio equal to 0.72. Two oblique, straight shocks of the same family, hereafter labeled S1 and S2, originate at the two concave corners of the lower wall; their interaction gives rise to a new shock, S3, a slip-stream, SS, and an expansion fan,  $EF_1$ , centered at the triple-point, TP, where the interaction takes place. A sketch of the flow pattern in the neighborhood of the interaction point TP is shown in Fig. 7; as long as no further interaction takes place, the various flow states can be analytically computed and are reported in Tab. 1a, where  $\theta$  denotes the angle between the local streamline and the horizontal axis. A second, stronger expansion fan,  $EF_2$  takes place at the convex corner of the lower wall and gives rise to a complex interaction (not shown in Fig. 7) involving the two expansion fans, the slip-stream and shock S3, which bends before being reflected from the upper wall. The reflected shock S4 is further reflected from the lower wall and



(a) computational domain and boundary conditions.



(b) Sketch of the area where the interaction between two shocks of the same family takes place.

Figure 7: Supersonic planar channel flow.

leaves the duct as shock S5, see Fig. 7a.

When simulating this problem with the shock-fitting technique, suitable models have to be used to account for the motion of the triple-point TP and the two shock-points (hereafter called reflection-points  $RP_{3-4}$  and  $RP_{4-5}$ ) where the regular reflection of shocks S3-S4 and S4-S5 takes place. The velocities of these two reflection-points, which are constrained to move along the upper, resp. lower wall, have been set equal to the component in the direction tangential to the wall of

region	M	$p/p_1$	$\theta$
1	3.500	1.000	$0^\circ$
2	2.395	4.037	$18.4^\circ$
3	1.892	8.489	$31.0^\circ$
4	1.910	8.256	$31.5^\circ$
5	1.551	8.256	$31.5^\circ$

(a) Flow states in the neighborhood of the interaction point TP.

	grid-points	triangles
shock-capturing	7,174	13,953
shock-fitting (at steady-state)	7,203	14,011

(b) Features of the two grids; the shock-capturing and initial shock-fitting grids are identical.

Table 1: Supersonic planar channel flow.

the incident-shock velocity within the two reflection-points:

$$\mathbf{w}(RP_{3-4}) = (\mathbf{w}_3 \cdot \boldsymbol{\tau}_w) \boldsymbol{\tau}_w \quad (18a)$$

$$\mathbf{w}(RP_{4-5}) = (\mathbf{w}_4 \cdot \boldsymbol{\tau}_w) \boldsymbol{\tau}_w \quad (18b)$$

In Eq. (18)  $\boldsymbol{\tau}_w$  is the unit vector tangent to the wall;  $\mathbf{w}_3 = w_3 \mathbf{n}_3$  and  $\mathbf{w}_4 = w_4 \mathbf{n}_4$  are the shock velocities computed within the reflection point  $RP_{3-4}$ , resp.  $RP_{4-5}$ , using the normals to the corresponding incident shocks, S3 and S4. Figure 8 schematically shows how the displacement velocity of the reflection point  $RP_{3-4}$  is calculated.



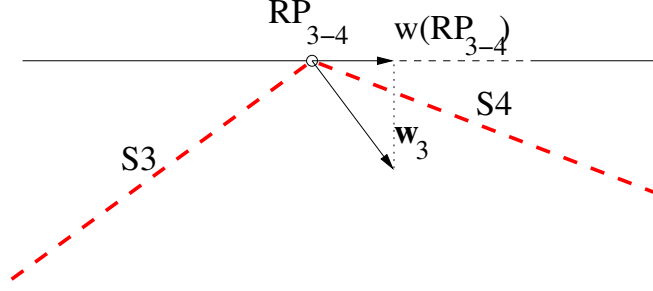


Figure 8: Supersonic planar channel flow: calculation of the displacement velocity,  $\mathbf{w}_{RP_{3-4}}$ , of the point where the regular reflection of shocks S3-S4 occurs.

Concerning the triple-point TP, only the three shock S1, S2 and S3 have been fitted in the current shock-fitting calculation, whereas the expansion fan,  $EF_1$ , and the slip-stream, SS, have been captured. This is not a limitation of the shock-fitting technique described herein, which also allows to treat slip-streams as fitted discontinuities, see [2, 35], but rather a deliberate choice aimed at demonstrating the use of the hybrid approach, whereby some of the discontinuities are fitted and the remaining ones are captured. Even though there are clear indications, see [6, 35], that the more discontinuities are fitted the more accurate is the solution, the hybrid approach may prove useful, for instance, in situations where a very complex interaction takes place or an automatic shock detector initially fails to identify the time-dependent formation of a discontinuity.

As far as the calculation of the TP displacement velocity is concerned, it is noted that in the interaction between two shocks of the same family, the location and the speed of the TP that is formed due to this interaction only depends upon the incident shocks, S1 and S2; therefore, the velocity,  $\mathbf{w}_{TP}$ , of the TP has been written as a function of the two different shock speeds,  $\mathbf{w}_1 = w_1 \mathbf{n}_1$  and  $\mathbf{w}_2 = w_2 \mathbf{n}_2$ , that can be separately computed at the TP using the unit vectors normal to the S1 and S2 shocks:

$$\mathbf{w}_{TP} = \alpha \mathbf{n}_1 + \beta \mathbf{n}_2. \quad (19)$$

The coefficients  $\alpha$  and  $\beta$  in Eq. (19) have been determined by requiring that the components of the TP velocity vector along the normals to the S1 and S2 shocks equals the corresponding shock speeds, i.e.

$$\mathbf{w}_{TP} \cdot \mathbf{n}_1 = w_1 \quad (20a)$$

$$\mathbf{w}_{TP} \cdot \mathbf{n}_2 = w_2 \quad (20b)$$

Solving Eq. (20) gives:

$$\alpha = \frac{w_1 - w_2 (\mathbf{n}_1 \cdot \mathbf{n}_2)}{1 - (\mathbf{n}_1 \cdot \mathbf{n}_2)^2} \quad (21a)$$

$$\beta = \frac{w_2 - w_1 (\mathbf{n}_1 \cdot \mathbf{n}_2)}{1 - (\mathbf{n}_1 \cdot \mathbf{n}_2)^2} \quad (21b)$$

The use of Eqs. (19) and (21) to compute the triple-point velocity applies to the interaction between two shocks of the same family. The reader is referred to [35] for a more comprehensive review of the various shock-interactions that may take place in a 2D flow, along with their numerical treatment. Note, also, that in addition to the S1, S2 and S3 shocks also the S4 and S5 shocks have been fitted.

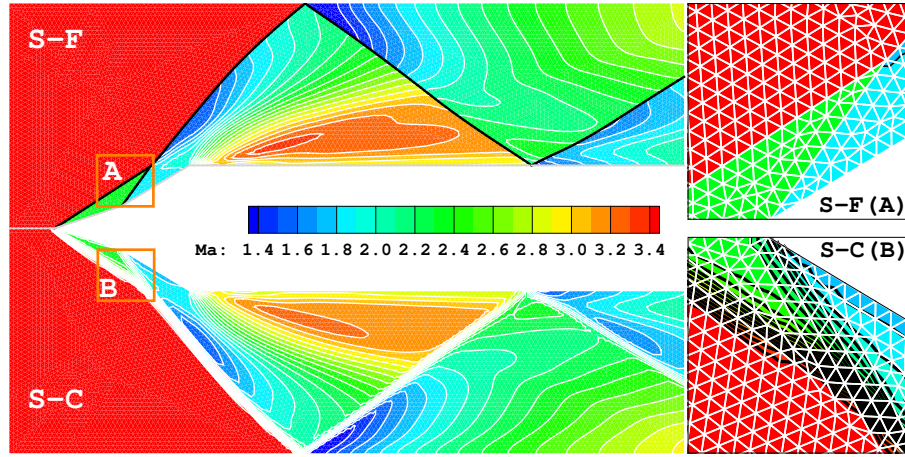


Figure 9: Supersonic planar channel flow: Mach iso-contour lines computed with shock-fitting (S-F, upper frames) and shock-capturing (S-C, lower frames).

The present flow configuration has been numerically simulated in both shock-fitting and shock-capturing mode. The shock-capturing solution has been computed using the shock-capturing option of the MCFS code; this is the same solver used in the smooth regions of the flow-field in the shock-fitting calculation. Similarly to the NACA0012 test-case of Sect. 3.1, the initial flow-field and shock-point location have been initialized using the shock-capturing solution and the **S**-points do not change their “nature” during the simulation.

The two unstructured triangular grids used in these calculations, whose characteristics are summarized in Tab. 1b, have both been generated by setting a characteristic length-scale equal to  $0.02L$ . The shock-capturing mesh has also been used as the initial mesh for the shock-fitting calculation. As shown in Tab. 1b, the initial and steady-state shock-fitting meshes feature a

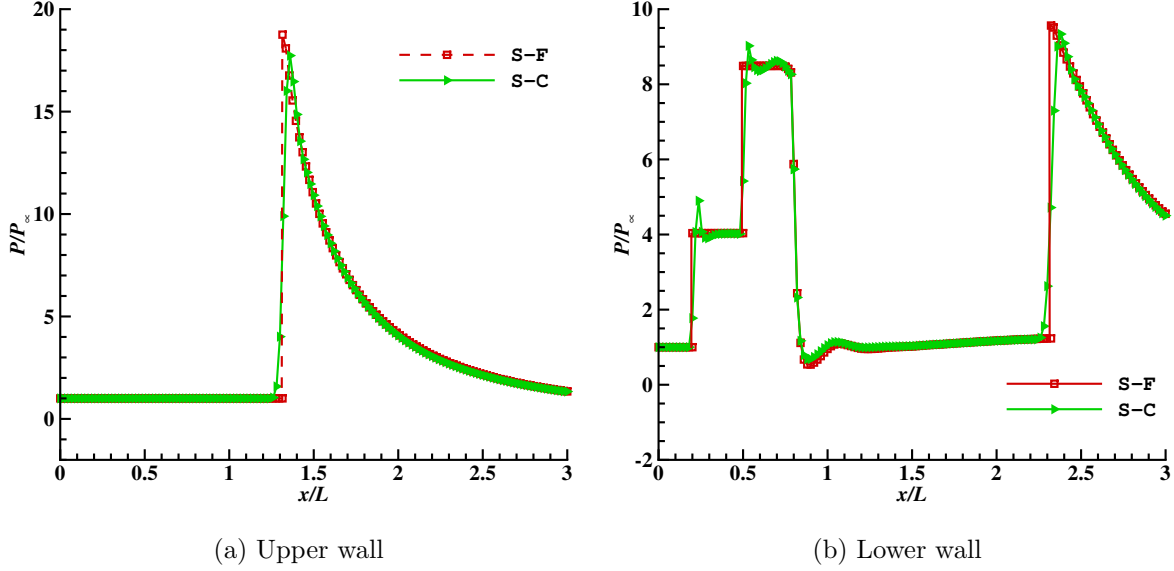


Figure 10: Supersonic planar channel flow: dimensionless pressure distributions along the channel's walls computed by means of shock-capturing (S-C) and shock-fitting (S-F).

slightly different number of grid-points and triangles. The shock-fitting technique moves the shock waves, along with the triple-point and reflection-points, towards their steady-state location, thus deforming the initial grid. In order to maintain a reasonable mesh quality, re-meshing was used and this explains the different number of grid-points and triangles of the initial and steady-state shock-fitting meshes.

Figure 9 shows the Mach number field obtained in the shock-fitting (S-F, upper half of the frame) and shock-capturing (S-C, lower half of the frame) simulations. The larger frames on the left show the entire computational domain, whereas the smaller ones on the right (frames A and B) show a zoom of the Mach number distribution superimposed on the computational mesh in the neighborhood of the triple-point.

Frame B of Fig. 9 clearly reveals that the interaction between the captured shocks S1 and S2 takes place over a finite region whose size is comparable, for the given mesh resolution, to that of the entire uniform flow region 2. This is in contrast with the shock-fitting solution shown in frame A of Fig. 9: by fitting not only the shocks, but also the TP, it is possible to exactly predict the uniform flow states downstream of shocks S2 and S3 (regions 2 and 3 of Fig. 7b) even on the relatively coarse mesh that has been used.

This observation is further supported by Fig. 10, which shows the dimensionless pressure distributions along the upper and lower walls of the channel. Even though both modeling practices agree on the location of the discontinuities along both walls, significant differences exist between the two. In particular, the shock-capturing calculation not only overshoots the pressure levels downstream of shocks S1 and S2 (see Fig 10b), but also fails to predict the uniform flow states in regions 2 and 3, which are clearly visible in the pressure distribution computed by means of shock-fitting.

These results clearly show the better quality of the solutions obtained using shock-fitting with respect to those obtained using shock-capturing. However, a fairer comparison between the two different shock-modeling options should make use of the most appropriate mesh for each solver, i.e. the shock-capturing solutions should be computed on anisotropically adapted meshes. Such a comparison has been made by two of the authors in [36, 37]: from a qualitative point-of-view it confirms that shock-fitting delivers superior accuracy compared to shock-capturing. A quantitative comparison between shock-fitting and shock-capturing on anisotropically adapted meshes based, for instance, on the order-of-convergence measured within the shock-downstream region is still lacking (to the best of the authors’ knowledge) in the literature and is certainly worth being investigated.

### 3.3. Hypersonic blunt-body flow

We consider here the hypersonic flow ( $M_\infty = 20$ ) past the fore-body of a circular cylinder. This kind of hypersonic flows, which features the presence of an isolated shock-wave standing ahead of the body, has been dealt with very effectively in the literature [21, 22, 24, 38, 39] by means of the so-called “boundary” shock-fitting technique, whereby the bow shock is made to coincide with one of the boundaries of the computational domain and second-, or higher-order-accurate, discretization schemes are used to approximate the governing PDEs within the shock-layer, i.e. the region bounded by the bow shock and the body. The mesh, which only covers the shock-layer, is deformed under the action of the shock-speed computed along the fitted bow-shock until the steady solution, corresponding to vanishing shock-speeds, is reached. Since the final (steady) shape of the bow-shock is not known a priori, a reasonable initial condition and shock-shape have to be supplied to the “boundary” shock-fitting algorithm. This is typically done by running a

preliminary shock-capturing calculation, which supplies the steady (captured) flow-field. A shock-detection algorithm, see [40] for a comprehensive review, is then used to extract an approximate shock-shape (which is going to be one of the boundaries of the initial shock-fitting mesh) and the solution within the shock-layer.

Using the “embedded” shock-fitting technique described in this paper, however, it is no longer necessary to run a preliminary shock-capturing calculation, for reasons that will be described hereafter.

Figure 11 shows a detail of the initial computational mesh and the initial solution supplied to our “embedded” shock-fitting technique: the overall mesh is made of 8051 triangular cells and 4177 grid-points and all flow variables have been initialized to their free-stream values throughout the entire computational domain. The yellow curve in Fig. 11 is the initial shock-front which we have “arbitrarily” defined by connecting the 84 grid-points that are located along the circular arc of equation:  $x^2 + y^2 = (1.3 R)^2$ ,  $R$  being the cylinder’s radius.

As mentioned in Sect. 2.4, the shock-points may have to be “downgraded” to **C** (common) points whenever the shock becomes weak enough and they may regain their **S** (shock) state if the shock strength builds up again at a later time.

We shall now examine how the steady solution is approached by following the temporal evolution of the solution, which is advanced in time using Runge-Kutta time-stepping.

Figure 12 shows the computed density iso-contours at different stages of the iterative process, as indicated by the iteration counter shown on top of each frame. The dashed-dotted yellow line shown in all frames of Fig. 12 indicates the initial location of the shock-front, which is also shown in Fig. 11.

At the initial time (time-step 0), both the shock-upstream and shock-downstream dependent variables within the shock-points (which are located along the yellow line of Fig. 12b) are equal to the corresponding free-stream values. It follows that right hand side of Eq. (10) reduces to  $(\gamma + 1) / (\gamma - 1)$ , which corresponds to a shock-upstream relative Mach number,  $M_{u,rel} = 1$ , as can be easily verified from Eq. (10). The criterion (17) is therefore met at all shock-points which are set into the **C** state; this is precisely what is expected to happen, because there are no shocks in the initial solution.

As time elapses, see Fig. 12b), the enforcement of the impermeability condition along the solid

body gives rise to a captured shock-wave which travels upstream towards the pre-set initial shock-front. Since the disturbance moves in the wall-normal direction with a spatially non-uniform speed, not all the shock-points are reached by the captured shock-wave at the same time. At time-step 185, the disturbance has reached only the shock-points that are farther away from the centerline, thus triggering their state from **C** to **S**; two disjoint fitted-shocks build up, as shown in Fig. 12b). Close inspection of Fig. 12b) reveals that once (some of) the shock-points have been activated, i.e. set into **S** state, they also start moving with the local shock speed. Therefore, the fitted shock, shown using a solid red line in frames b) to e) of Fig. 12, starts moving away from its initial location.

As time further elapses, see Fig. 12c), more and more shock-points are reached by the captured shock-wave and switch their state from **C** to **S**, so that the two fitted-shocks increase their length. At step 240, two distinct fitted-shocks can still be seen, because those shock-points that are close to the centerline have not yet been reached by the disturbance and, therefore, are still being treated as **C** points. Consequently, close to the centerline, the bow shock is still being captured, rather than fitted. This can be clearly seen by looking at the density iso-contour lines of Fig. 12c).

At time-step 295 the two fitted-shocks have merged into a single bow shock; its location is still close, but different from the initial (circular) shock-shape.

After 10,000 steps, the fitted-shock has moved further upstream to its final (steady) location: the shock speed, and consequently all grid velocities, have vanished, the grid no longer changes and the steady solution has finally been reached.

Figure 13 shows the computational mesh at different time steps: at first the fitted **shock moves** towards the wall and then outwards. Re-meshing of the computational domain has been performed twice (at time step 600 and 5800) as soon as the mesh quality was seen to deteriorate.

In the present test-case it has been easy to preset a special mesh line whose grid-points can be switched into **S**-points during the simulation, because the shock-topology at steady-state is known. In general, however, the time-dependent evolution of the flow topology is not known “a priori” or it is so complex that it may not possible to describe it using lines of grid-points preset at the beginning of the simulation.

A quantitative analysis of the shock-fitting solution has been carried out by comparing the pressure distribution with the reference solution [41]. Figure 14 plots pressure at the wall (**nor-**

malized using the free-stream static pressure) against the azimuthal angle,  $\Theta$ . The comparison shows that the solution computed using the proposed “embedded” shock-fitting technique is in good agreement with the reference solution.

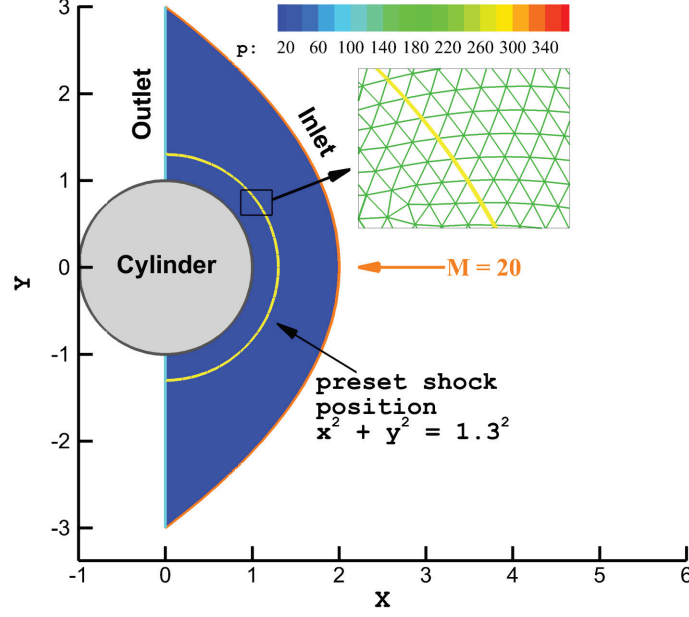


Figure 11: Hypersonic blunt-body flow: the initial flow-field and shock position.

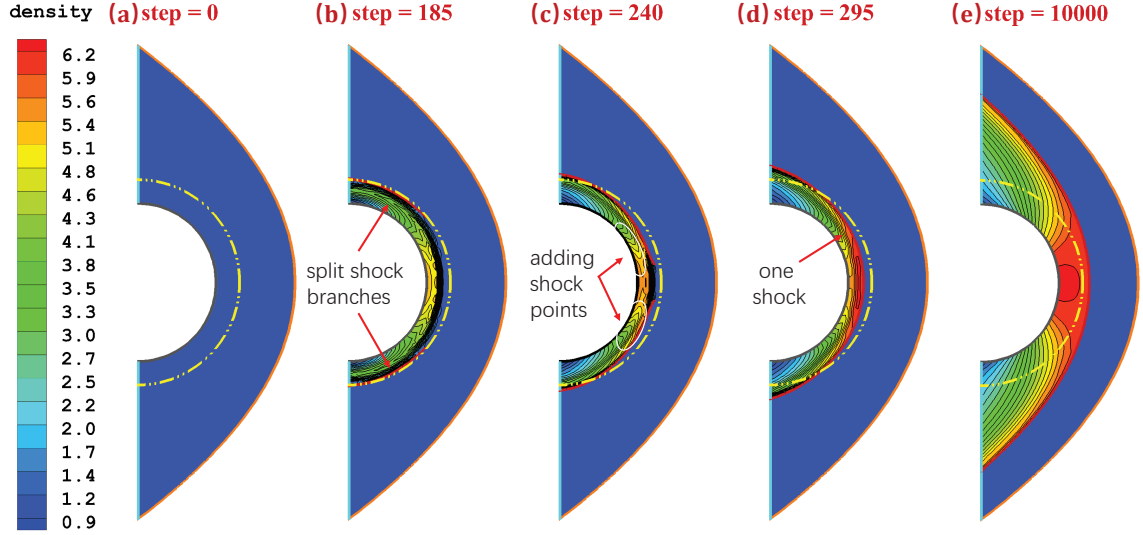


Figure 12: Hypersonic blunt-body flow: convergence process of shock position (density iso-contours).

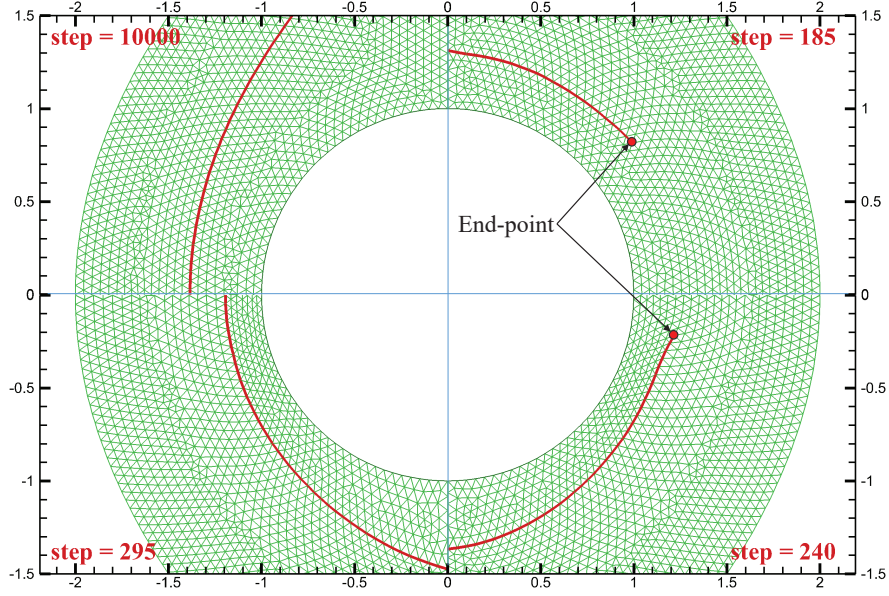


Figure 13: Hypersonic blunt-body flow: computational meshes at different time steps.

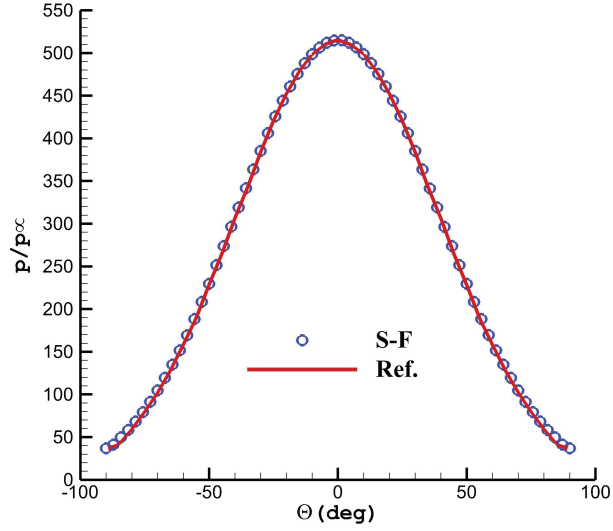


Figure 14: Hypersonic blunt-body flow: pressure distribution along the wall.

Finally, a grid convergence analysis has been made by measuring the rate at which the discretization error, i.e. the difference between the analytical and computed solutions, decreases when the mesh spacing is reduced. For the present test-case, the two quantities that can be analytically



computed are the stagnation enthalpy<sup>4</sup>, which should be constant throughout the entire flow-field, and equal to its free-stream value:

$$\frac{H_\infty}{a_\infty^2} = \frac{1 + \delta M_\infty^2}{\gamma - 1}$$

and the pressure<sup>5</sup> at the stagnation point, which should be equal to the post-shock stagnation pressure and can be computed using the so-called Rayleigh pitot formula [42, Eq. (100)]:

$$\frac{p^0}{p_\infty} = \left[ \frac{(\gamma + 1) M_\infty^2}{2} \right]^{\frac{\gamma}{\gamma-1}} \left[ \frac{(\gamma + 1)}{2\gamma M_\infty^2 - (\gamma - 1)} \right]^{\frac{1}{\gamma-1}}$$

The geometrical features of the four grid-levels (labeled 0 to 3, level 0 being the finest) have been summarized in Tab. 2: the spacing parameter  $h$  is the distance between two adjacent grid-points on the domain boundaries and it has been recursively halved when generating the finer grid-levels. All four grids, despite being not nested, feature a grid-point located at the stagnation point.

Table 2: Hypersonic blunt-body flow: features of the four grid-levels used to perform the grid-convergence analysis.

Grid level	$h/R$	Grid features at convergence	
		Triangles	Grid-points
3	0.2	446	292
2	0.1	1645	957
1	0.05	6300	3439
0	0.025	24741	12894

Figure 15, which shows the pseudo-temporal evolution of the density update, confirms that the iterative convergence error, which might adversely affect the stagnation enthalpy distribution if non-zero, has indeed been driven to machine accuracy on all grid-levels.

---

<sup>4</sup>made dimensionless using the free-stream speed of sound squared

<sup>5</sup>made dimensionless using the free-stream static pressure

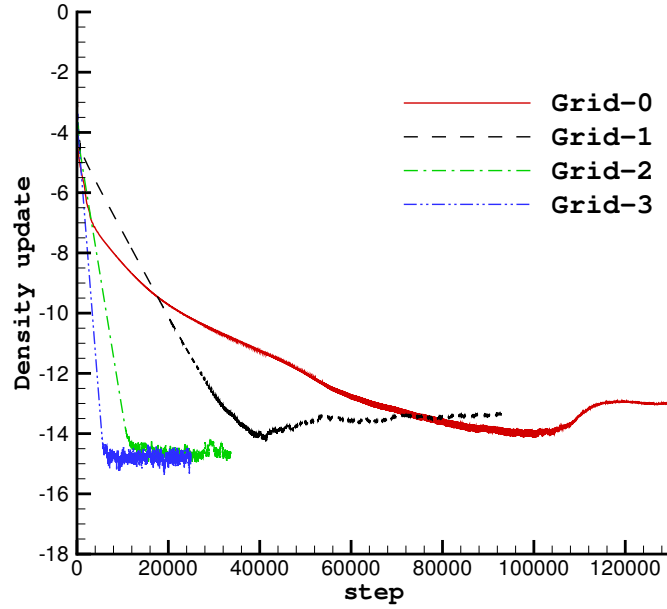


Figure 15: Hypersonic blunt-body flow: iterative convergence of the density update on all grid-levels.

The stagnation enthalpy error has been computed *globally* over the entire computational domain ( $|\Omega|$  denotes its total area) using the  $L_1$ -norm:

$$L_1(\epsilon_H) = \frac{1}{|\Omega|} \int_{\Omega} \frac{|H(\mathbf{x}) - H_{\infty}|}{a_{\infty}^2} d\Omega \quad (22)$$

whereas the stagnation pressure error has been computed *locally* at the stagnation point. For each pair of grid-levels the *observed* order-of-convergence,  $\tilde{n}$ , is computed as follows [43–45]:

$$\tilde{n}_{i,i+1} = \frac{\log(\epsilon_{i+1}/\epsilon_i)}{\log(h_{i+1}/h_i)} \quad (23)$$

where  $\epsilon_i$  denotes either the global or local measure of the discretization error on grid-level  $i$  and the ratio  $h_{i+1}/h_i$  is equal to 2, see Tab. 2.

Figure 16 shows, using a double logarithmic scale, how the discretization error decays as the mesh is refined; the two straight lines corresponding to first and second order slopes have also been drawn. It can be seen that the stagnation pressure error exhibits second-order convergence. As far as the stagnation enthalpy error is concerned, the observed order is about 1.2 on the coarsest pair of grid levels, but then monotonically increases up to 1.7 on the finest pair.

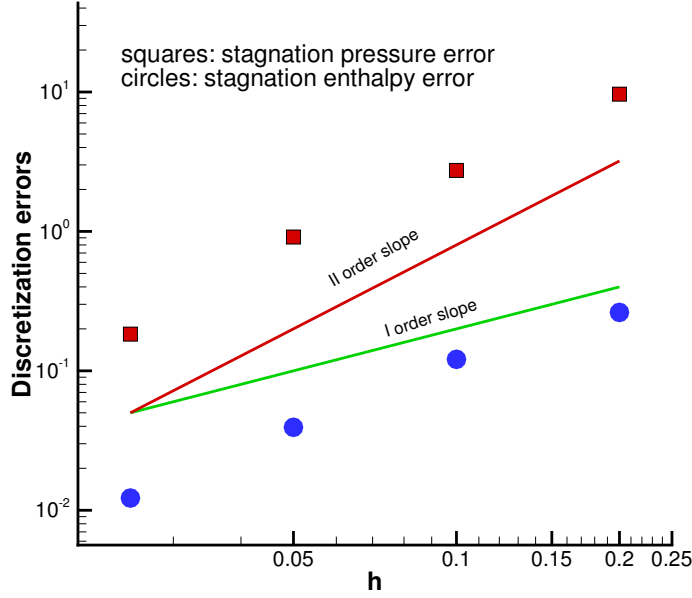


Figure 16: Hypersonic blunt-body flow: discretization error decay as the mesh is refined.

### 3.4. Shock-vortex interaction

Also this latest test-case, which consists in an unsteady problem involving a shock-vortex interaction, allows to demonstrate the capability of the “embedded” shock-fitting technique to handle changing shock-topologies. Figure 17 shows the computational domain and the boundary and initial ( $t = 0$ ) conditions: a straight, stationary shock, whose shock-upstream Mach number is  $M_s = 1.21$  is standing  $0.7 L$  downstream of the supersonic inlet boundary. The supersonic upstream flow carries a vortex, from the left to the right of Fig. 17, towards the stationary shock. At  $t = 0$ , the center of the vortex is located  $0.2 L$  ahead of the shock. The computational mesh is initially made up of 62,462 triangular cells and 31,532 grid nodes, 101 grid-points are located at  $x = 0.7 L$  and are treated as shock-points. The initial shape of the vortex can be described using a polar coordinate system with the origin attached to the center of the vortex. The perturbation velocity field,  $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_\infty$ , consists of a clockwise vortex characterized by a purely tangential velocity component:

$$\begin{cases} \tilde{u}_\theta = -\tilde{\epsilon} |\mathbf{u}_\infty| \tau e^{\alpha(1-\tau^2)} \\ \tilde{u}_r = 0 \end{cases}, \quad (24)$$

where  $\tau = r/r_c$  is the dimensionless radial distance from the center of the vortex and  $\tilde{\epsilon}$ ,  $\alpha$  and  $r_c$  are two parameters that control the shape and magnitude of the perturbation.

In the present simulation, we choose  $\tilde{\epsilon} = 1.11$ ,  $\alpha = 0.5$  and  $r_c = 0.05L$ , so the vortex Mach number

$$M_v = \frac{\max(\tilde{u}_\theta)}{a_\infty} = M_s \tilde{\epsilon} (2\alpha)^{-\frac{1}{2}} e^{(\alpha - \frac{1}{2})} \approx 1.244, \quad (25)$$

where  $a_\infty$  is the sound speed of the free-stream, shock-upstream flow.

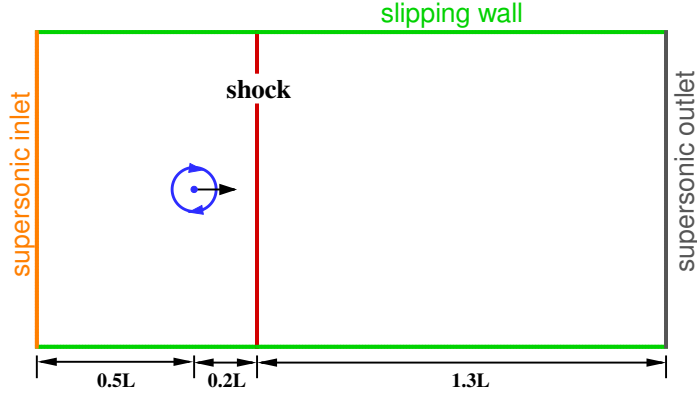


Figure 17: Shock vortex interaction: computational domain and boundary conditions.

The temporal evolution of the shock-vortex interaction is shown in Fig. 18, where density iso-contour lines corresponding to subsequent time instants have been reported. Once the vortex has come sufficiently close to the shock, see Fig. 18a, close to the vortex core the shock-wave ceases to be straight and becomes a weak oblique shock. Being the vortex clock-wise, the relative, shock-upstream Mach number,  $M_{u,rel}$  in Eq. (8), increases above its initial value,  $M_s$ , within those shock-points that are located above the vortex core, whereas  $M_{u,rel}$  decreases within the shock-points below. Whenever the  $M_{u,rel}$  value computed within a shock-point verifies the inequality (17), the **S**-point is converted into a **C**-point so that the fitted shock-wave is split into two distinct branches. This is clearly visible in Fig. 18b. Figure 18c shows that the shock-wave splits up to give way to the vortex, as more and more **S**-points are converted into **C**-points. The reverse process can be seen in Fig. 18d: once the vortex has moved sufficiently far downstream, the shock strength builds up again and the lower shock-branch is seen to increase its length. From an algorithmic viewpoint, this corresponds to the fact that grid-points located along the fitted shock-front regain their **S** state.

The same observation applies to Fig. 18e, which refers to a later time. At time  $t a_\infty/L = 0.55$ , see Fig. 18f, the two shock branches have again merged into a single shock.

In order to confirm that the fitted shocks are in the right location, a shock-capturing calculation has been run using the same solver on a fine grid made of half a million quadrilaterals ( $1,000 \times 500$ ). The comparison between the shock positions computed by means of shock-fitting and the pressure field computed by means of shock-capturing at different time instants is presented in Fig. 19.

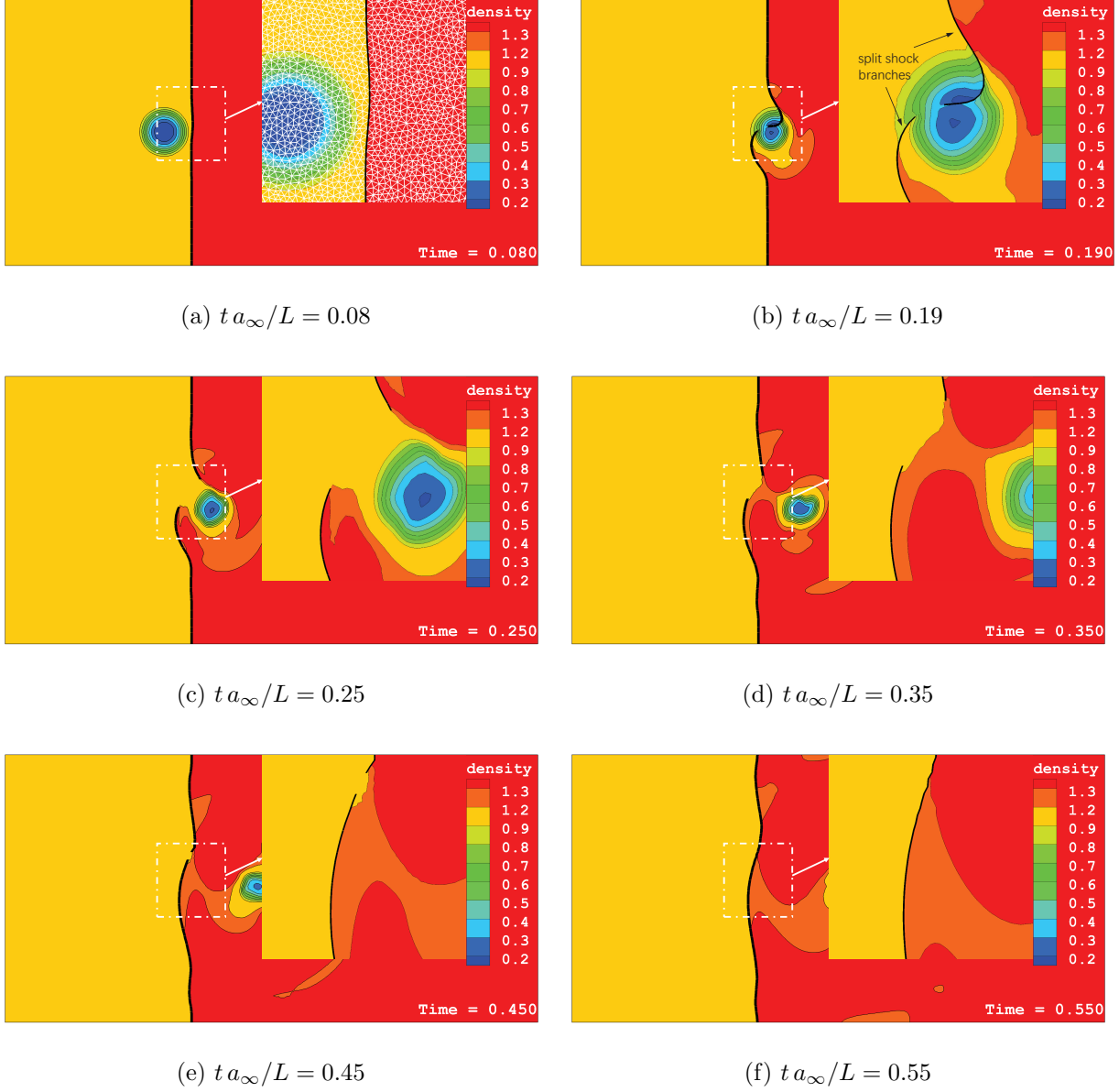


Figure 18: Shock-vortex interaction: density iso-contours at different time instants.

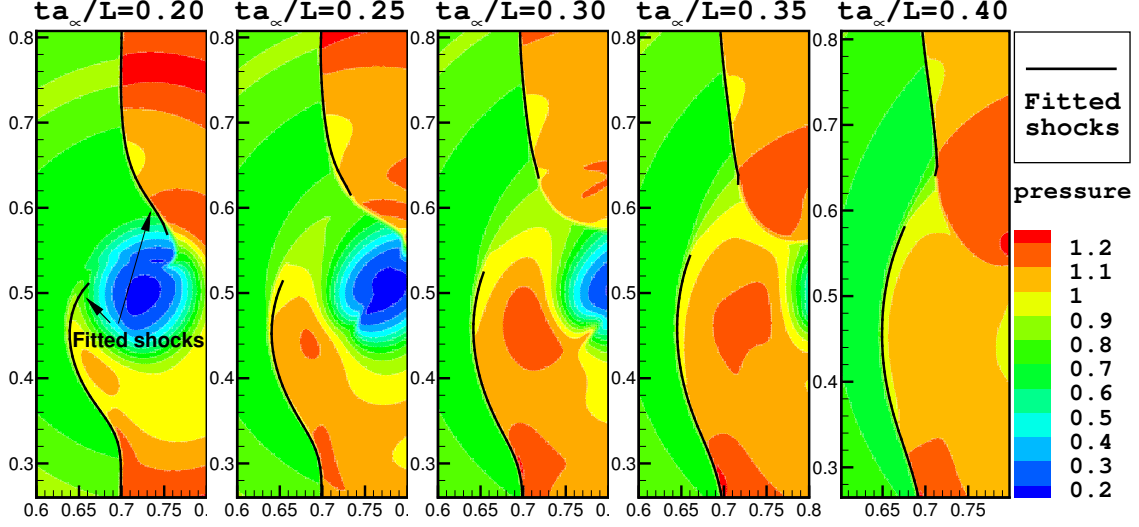


Figure 19: Shock vortex interaction: fitted-shocks (shown using solid black lines) are superimposed on top of the pressure field captured on a much finer mesh.

Figure 19 also allows to understand the limitations of the present technique. For instance, the flow-field computed by the shock-capturing solver at  $ta_\infty/L = 0.35$  and  $ta_\infty/L = 0.40$  clearly reveals the presence of two shock branches that originate close to the end-point of the upper fitted shock-wave. These two shock branches cannot be fitted because they are in regions where there are no “special” points that can switch their status from **C** to **S**.

It is therefore clear that the technique proposed in this paper is not capable of handling all kind of topological changes in the shock wave pattern that may occur in an un-steady flow. Some knowledge (even if approximate) of the different shock-patterns that will develop in time, eventually gained by running an hybrid simulation first, is required to preset a sufficient number of “special” points, which will be “automatically” activated, or de-activated during the simulation using the criterion (17).

#### 4. Conclusion

The “boundary” shock-fitting technique for unstructured grids originally proposed by some of the authors in [1, 2] has been modified to make it capable of dealing with several simple topological changes that may occur in the unsteady simulation of compressible flows characterized by the

presence of shock-waves.

In this improved version of the algorithm, which has been called “embedded” shock-fitting, the fitted shock-fronts are made up of a connected sequence of “shock”-points; in contrast to all other “common” grid-points, where a single set of flow variables is stored, shock-points are characterized by a pair of flow states, corresponding to the upstream and downstream sides of the discontinuity.

Moreover, in this improved version of the algorithm, the shock-points are allowed to change their status, which can be switched to common and then back to shock, depending upon the local flow conditions.

Thanks to this algorithmic improvement, the “embedded” shock-fitting technique is capable of handling some simple topological changes that may take place during the unsteady simulations of a shocked flow.

The simulation of a shock-vortex interaction and the flow past the fore-body of circular cylinder which is impulsively set into motion at hypersonic speed have been used to highlight the capabilities and the limits of the proposed “embedded” technique.

The capability to handle whatsoever topological change may occur during an unsteady flow calculation certainly requires further and non-trivial algorithmic efforts, including the development of a reliable shock-detection technique; we believe, however, that the proposed “embedded” shock-fitting technique is an important step in this direction.

## Acknowledgments

This work is partially supported by National Natural Sciences Foundation of China under grant 11872144.

## References

## References

- [1] J. Liu, D. Zou, A Shock-Fitting Technique for ALE Finite Volume Methods on Unstructured Dynamic Meshes, Springer International Publishing, Cham, 2017, pp. 131–149. doi:10.1007/978-3-319-68427-7\_6.

- [2] D. Zou, C. Xu, H. Dong, J. Liu, A shock-fitting technique for cell-centered finite volume methods on unstructured dynamic meshes, *Journal of Computational Physics* 345 (2017) 866 – 882. doi:<https://doi.org/10.1016/j.jcp.2017.05.047>.
- [3] D. Zaide, P. Roe, Shock capturing anomalies and the jump conditions in one dimension, in: *Fluid Dynamics and Co-located Conferences*, American Institute of Aeronautics and Astronautics, 2011. doi:10.2514/6.2011-3686.
- [4] D. W. Zaide, P. L. Roe, A second-order finite volume method that reduces numerical shock-wave anomalies in one dimension, in: *Fluid Dynamics and Co-located Conferences*, American Institute of Aeronautics and Astronautics, 2013. doi:10.2514/6.2013-2699.
- [5] R. Paciorri, A. Bonfiglioli, A shock-fitting technique for 2d unstructured grids, *Computers & Fluids* 38 (3) (2009) 715 – 726. doi:10.1016/j.compfluid.2008.07.007.
- [6] R. Paciorri, A. Bonfiglioli, Shock interaction computations on unstructured, two-dimensional grids using a shock-fitting technique, *Journal of Computational Physics* 230 (8) (2011) 3155 – 3177. doi:<http://dx.doi.org/10.1016/j.jcp.2011.01.018>.
- [7] A. Bonfiglioli, M. Grottadaurea, R. Paciorri, F. Sabetta, An unstructured, three-dimensional, shock-fitting solver for hypersonic flows, *Computers & Fluids* 73 (0) (2013) 162–174. doi:10.1016/j.compfluid.2012.12.022.
- [8] A. Bonfiglioli, R. Paciorri, L. Campoli, Unsteady shock-fitting for unstructured grids, *International Journal for Numerical Methods in Fluids* 81 (4) (2016) 245–261, fld.4183. doi:10.1002/fld.4183.
- [9] I. H. Parpia, P. Parikh, A solution-adaptive mesh generation method with cell-face orientation control, in: *AIAA, Aerospace Sciences Meeting and Exhibit*, 32nd, Reno, NV; 10-13 Jan., no. AIAA-1994-416, 1994. doi:10.2514/6.1994-416.
- [10] J. V. Rosendale, Floating shock fitting via lagrangian adaptive meshes, Tech. rep., Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681-0001, NASA Contractor Report 194997; ICASE Report No. 94-89 (1994). URL <https://apps.dtic.mil/dtic/tr/fulltext/u2/a289758.pdf>



- [11] J. Trepanier, M. Paraschivoiu, M. Reggio, R. Camarero, A Conservative Shock Fitting Method on Unstructured Grids, *Journal of Computational Physics* 126 (2) (1996) 421–433. doi:10.1006/jcph.1996.0147.
- [12] M. J. Zahr, P.-O. Persson, An optimization based Discontinuous Galerkin approach for high-order accurate shock tracking, *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-0063.
- [13] M. Zahr, P.-O. Persson, An optimization-based approach for high-order accurate discretization of conservation laws with discontinuous solutions, *Journal of Computational Physics* 365 (2018) 105 – 134. doi:https://doi.org/10.1016/j.jcp.2018.03.029.
- [14] A. Shi, P.-O. Persson, M. J. Zahr, An optimization-based discontinuous galerkin approach for high-order accurate shock tracking with guaranteed mesh quality, in: *AIAA Scitech 2019 Forum*, 2019, p. 1151. doi:10.2514/6.2019-1151.
- [15] M. J. Zahr, A. Shi, P.-O. Persson, Implicit shock tracking using an optimization-based high-order discontinuous galerkin method, *Journal of Computational Physics* (2020) 109385doi:10.1016/j.jcp.2020.109385.
- [16] A. Corrigan, A. D. Kercher, D. A. Kessler, A moving discontinuous Galerkin finite element method for flows with interfaces, *International Journal for Numerical Methods in Fluids* 89 (9) (2019) 362–406. doi:10.1002/fld.4697.
- [17] A. T. Corrigan, A. D. Kercher, D. A. Kessler, D. Wood-Thomas, Application of the moving Discontinuous Galerkin method with interface condition enforcement to shocked compressible flows, *AIAA AVIATION Forum*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-4272.
- [18] A. T. Corrigan, A. Kercher, D. A. Kessler, The moving Discontinuous Galerkin method with interface condition enforcement for unsteady three-dimensional flows, *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-0642.
- [19] A. D. Kercher, A. Corrigan, D. A. Kessler, The moving discontinuous galerkin finite element

method with interface condition enforcement for compressible viscous flows, arXiv preprint arXiv:2002.12740.

- [20] A. D. Kercher, A. Corrigan, A least-squares formulation of the moving discontinuous galerkin finite element method with interface condition enforcement, arXiv preprint arXiv:2003.01044.
- [21] G. Moretti, M. Abbett, A time-dependent computational method for blunt body flows., AIAA Journal 4 (1966) 2136–41. doi:10.2514/2.6898.
- [22] D. A. Kopriva, T. A. Zang, M. Y. Hussaini, Spectral methods for the Euler equations: The blunt body problem revised, AIAA Journal 29 (1991) 1458–1462. doi:10.2514/3.10760.
- [23] X. Zhong, Leading-edge receptivity to free-stream disturbance waves for hypersonic flow over a parabola, Journal of Fluid Mechanics 441 (-1) (2001) 315–367. doi:10.1017/S0022112001004918.
- [24] M. Najafi, K. Hejranfar, V. Esfahanian, Application of a shock-fitted spectral collocation method for computing transient high-speed inviscid flows over a blunt nose, Journal of Computational Physics 257, Part A (2014) 954 – 980. doi:http://dx.doi.org/10.1016/j.jcp.2013.09.037.
- [25] G. Moretti, Shock-Fitting Analysis, Springer International Publishing, Cham, 2017, pp. 3–31. doi:10.1007/978-3-319-68427-7\_1.
- [26] F. Nasuti, M. Onofri, Steady and Unsteady Shock Interactions by Shock Fitting Approach, Springer International Publishing, Cham, 2017, pp. 33–55. doi:10.1007/978-3-319-68427-7\_2.
- [27] P. S. Rawat, X. Zhong, On high-order shock-fitting and front-tracking schemes for numerical simulation of shock-disturbance interactions, Journal of Computational Physics 229 (19) (2010) 6744 – 6780. doi:DOI: 10.1016/j.jcp.2010.05.021.
- [28] R. Paciorri, A. Bonfiglioli, Basic Elements of Unstructured Shock-Fitting: Results Achieved and Future Developments, Springer International Publishing, Cham, 2017, pp. 59–84. doi:10.1007/978-3-319-68427-7\_3.

- [29] L. Campoli, P. Quemar, A. Bonfiglioli, M. Ricchiuto, Shock-Fitting and Predictor-Corrector Explicit ALE Residual Distribution, Springer International Publishing, Cham, 2017, pp. 113–129. doi:10.1007/978-3-319-68427-7\_5.
- [30] A. Lani, V. De Amicis, SF: An Open Source Object-Oriented Platform for Unstructured Shock-Fitting Methods, Springer International Publishing, Cham, 2017, pp. 85–112. doi:10.1007/978-3-319-68427-7\_4.
- [31] M. D. Salas, A shock-fitting primer, Crc Press Boca Raton Fl (2009) xvi+400.
- [32] G. Zheng, L. Jun, Q. Zhang-hua, New computational method to determine trim angle of attack for reentry vehicles, Journal of Spacecraft and Rockets 41 (6) (2004) 1063–1065. doi:10.2514/1.6211.
- [33] J. Liu, X. Bai, H. Zhang, Z. Guo, Discussion about GCL for deforming grids, Aeronautical Computing Technique 39 (4) (2009) 1–5.
- [34] J. Zierep, New results for the normal shock in inviscid flow at a curved surface, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik 83 (9) (2003) 603–610. doi:10.1002/zamm.200310051.
- [35] S. Chang, X. Bai, D. Zou, Z. Chen, J. Liu, An adaptive discontinuity fitting technique on unstructured dynamic grids, Shock Waves 29 (8) (2019) 1103–1115. doi:10.1007/s00193-019-00913-3.
- [36] M. S. Ivanov, A. Bonfiglioli, R. Paciorri, F. Sabetta, Computation of weak steady shock reflections by means of an unstructured shock-fitting solver, Shock Waves 20 (4) (2010) 271–284. doi:10.1007/s00193-010-0266-y.
- [37] A. Bonfiglioli, R. Paciorri, Hypersonic flow computations on unstructured grids: Shock-capturing vs. shock-fitting approach, in: 40th Fluid Dynamics Conference and Exhibit, AIAA CP 2010-449. doi:10.2514/6.2010-4449.
- [38] D. A. Kopriva, J. H. Kolas, A conservative staggered-grid Chebyshev multidomain method for compressible flows, Journal of Computational Physics 125 (1) (1996) 244 – 261. doi:10.1006/jcph.1996.0091.

- [39] D. A. Kopriva, Shock-fitted multidomain solution of supersonic flows, *Computer Methods in Applied Mechanics and Engineering* 175 (3) (1999) 383 – 394. doi:[https://doi.org/10.1016/S0045-7825\(98\)00362-4](https://doi.org/10.1016/S0045-7825(98)00362-4).
- [40] Z. Wu, Y. Xu, W. Wang, R. Hu, Review of shock wave detection method in CFD post-processing, *Chinese Journal of Aeronautics* 26 (3) (2013) 501–513. doi:[10.1016/j.cja.2013.05.001](https://doi.org/10.1016/j.cja.2013.05.001).
- [41] A. Lyubimov, V. Rusanov, Gas flows past blunt bodies, part II: table of the gasdynamic function, NASA TT F-715, 1973.
- [42] A. R. Staff, Equations, tables, and charts for compressible flow, Tech. rep., NASA Ames Research Centre, NACA Report 1135 (1953).  
URL [https://www.nasa.gov/sites/default/files/734673main\\_Equations-Tables-Charts-Compre](https://www.nasa.gov/sites/default/files/734673main_Equations-Tables-Charts-Compre)
- [43] A. Di Mascio, R. Paciorri, B. Favini, Truncation error analysis in turbulent boundary layers, *J. Fluids Eng.* 124 (3) (2002) 657–663. doi:[10.1115/1.1478564](https://doi.org/10.1115/1.1478564).
- [44] C. J. Roy, Grid Convergence Error Analysis for Mixed-Order Numerical Schemes, *AIAA Journal* 41 (4) (2003) 595–604. doi:[10.2514/2.2013](https://doi.org/10.2514/2.2013).
- [45] A. Bonfiglioli, R. Paciorri, Convergence analysis of shock-capturing and shock-fitting solutions on unstructured grids, *AIAA journal* 52 (7) (2014) 1404–1416. doi:[10.2514/1.J052567](https://doi.org/10.2514/1.J052567).

Dongyang Zou: Methodology, Software, Validation, Visualization, Writing - Original Draft; Aldo Bonfiglioli: Writing - Review & Editing, Supervision; Renato Paciorri: Formal analysis, Supervision; Jun Liu: Conceptualization, Supervision, Funding acquisition.

**Declaration of interests**

✓ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: