Extrapolated DIscontinuity Tracking for complex 2D shock interactions

Mirco Ciallella[a], Mario Ricchiuto[a], Renato Paciorri[b], Aldo Bonfiglioli[c]

[a]*Team CARDAMOM, INRIA, Univ. Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, 200 Avenue de la Vieille Tour, 33405 Talence cedex, France*
[b]*Dip. di Ingegneria Meccanica e Aerospaziale, Università di Roma "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy*
[c]*Scuola di Ingeneria - Università degli Studi della Basilicata, Viale dell'Ateneo Lucano 10, 85100 Potenza, Italy*

Abstract

A new shock-tracking technique that avoids re-meshing the computational grid around the moving shock-front was recently proposed by the authors [1]. This paper describes further algorithmic improvements which make the extrapolated Discontinuity Tracking Technique (*eDIT*) capable of dealing with complex shock-topologies featuring shock-shock and shock-wall interactions. Various test-cases are included to describe the key features of the methodology and prove its order-of-convergence properties.

Keywords: Shock-tracking, shock-fitting, unstructured grids, embedded boundary, shock interactions, Taylor expansions

1. Introduction

The numerical models nowadays used in the simulation of compressible high-speed flows within the continuum framework can be cast into two main categories: shock-capturing (SC) and shock-fitting (SF). SC can be traced back to the 1950s paper by VonNeumann and Richtmyer [2] and lays its foundations in the mathematical theory of weak solutions [3]. The key ingredients of SC discretizations are very well known in the CFD community and need not to be recalled here. SF techniques, on the contrary, rely on a very different standpoint, which consists in identifying the shocks as lines within a two-dimensional (2D) flow-field (surfaces in 3D) and computing their motion, upstream and downstream states according to the Rankine-Hugoniot jump relations. Fitted-shocks made their first appearance in two NACA technical reports written by Emmons [4, 5] in the 1940s, but it is between the late 1960s and the end of the 1980s that the technique has been extensively developed by Gino Moretti and collaborators under the name "Shock-Fitting" and, starting in the early 1980s, by James Glimm and collaborators under the name "Front-Tracking". Moretti's SF techniques (both the "boundary" and "floating" variants) were designed for structured-grid solvers, the only kind of CFD codes in use at that time. The development of structured-grid SF codes turned out to be algorithmically fairly complex, especially when dealing with flows featuring shock-shock and shock-boundary interactions [6]. Algorithmic complexity contributed to make CFD practitioners "shy away from shock-fitting" [7] to embrace the myth of "one code for all flows" that has so much contributed to the popularity of SC discretizations.

Starting in the late 1980s, however, the CFD community has shown increasing interest towards unstructured meshes because of their ability to easily describe complex geometries and the possibility of locally changing the mesh size to adapt it to the local flow features. Taking advantage of the greater flexibility offered by unstructured meshes, Paciorri and Bonfiglioli [8] developed a new unstructured SF technique suitable for vertex-centered solvers operating with simplicial elements (triangles in 2D and tetrahedra in 3D). Their approach alleviated many of the algorithmic difficulties incurred by

the SF techniques within the structured-grid setting. In recent years, the unstructured SF technique originally proposed in [8] has been further improved making it capable of dealing with multiple interacting discontinuities [9] and unsteady flows [10, 11] in 2D, as well as 3D flows [12], thus opening a new route for simulating compressible "shocked" flows. In particular, not only shocks and contact discontinuities are fitted, but also their interaction points, for example the triple point that arises in Mach reflections [13]. Most of the aforementioned developments have been included in an open-source platform [14] with the goal of fostering even more the interest in such methods.

Thanks to these new developments the interest in shock-fitting/tracking methods has seen a resurgence with the proposal of several implicit and explicit algorithms [15, 16, 17, 18, 19, 20, 21, 22, 23]. A limitation of this technique is that it heavily relies on the flexibility of triangular and tetrahedral grids to locally produce a fitted unstructured grid around the discontinuities.

The initial idea that motivated the present work comes from the similarity between the constraints arising from SF, as conceived by Paciorri and Bonfiglioli in [8], and those related to the construction of boundary-fitted grids for simulating flows around complex geometries. More precisely, the SF technique [8] requires that at each time-step the computational grid is locally re-meshed to follow the moving shock-front. This approach turns out to be best suited to vertex-centered CFD solvers, but it can be problematic if a different type of data representation is used. The *eDIT* approach, which has been described by the authors in [1] and is further developed in this paper, allows to overcome this limitation, because it avoids re-meshing by using ideas borrowed from immersed and embedded boundary methods developed to allow a flexible management of complex geometries. These two approaches rely on different philosophies, however.

Immersed methods are based on an extension of the flow equations outside the physical domain (typically within solid bodies). This extension is formulated using some smooth approximation of the Dirac delta function to localize the boundary, as well as to impose the boundary condition. These methods are relatively old, and based on the original ideas of Peskin [24]. Finite element and unstructured mesh extensions for elliptic PDEs as well as for incompressible, and compressible flows have been discussed in [25, 26, 27].

Embedded methods, on the other hand, solve the PDEs only in the physical domain, while replacing the exact boundary with some less accurate approximation, combined with some weak enforcement of the boundary condition. There is a certain number of techniques to perform this task, which go from the combination of XFEM-type methods with penalization or Nitsche's type approaches [28], to several types of cut finite element methods with improved stability [29, 30], to approximate domain methods such as the well known ghost-fluid method [31, 32], and the more recent shifted boundary method (SBM) [33, 34].

As in the latter, we impose modified conditions on surrogate shock-manifolds, acting as boundaries between the shock-upstream and shock-downstream regions. The values of the flow variables imposed on these surrogate boundaries are extrapolated from the tracked shock-front, accounting for the non-linear jump and wave propagation conditions, as done in the unstructured SF approach of [8]. As in shock-fitting and front-tracking methods [35], the shock-front is discretized by an independent lower-dimensional mesh, and its position, as well as the position of the two surrogate boundaries, are themselves part of the computational result.

In this paper, we focus on the new features coded in the algorithm to enable the computation of shock-shock and shock-wall interactions. In particular, when several discontinuities are mutually interacting, or interact with a solid boundary, the following issues must be addressed:

1. identification of the various sub-domains obtained when the discontinuities cut through the computational domain;

2. calculation of the velocity of those points where different discontinuities mutually interact (for example the triple point in a Mach reflection) or reach the boundary;

3. the algorithm used to transfer the dependent variables between the shock-front and the surrogate boundaries must be applicable also when the shape of the surrogate boundaries becomes very complex, which typically occurs in the neighborhood of an interaction.

This paper is organized as follows: § 2 introduces the governing conservation equations; § 3 describes the structure of the algorithm; § 4 focuses on the various test-cases considered; finally in § 5 we draw some conclusions.

2. Generalities

We consider the numerical approximation of solutions of the steady limit of the Euler equations reading:

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F} = 0 \quad \text{in} \quad \Omega \subset \mathbb{R}^2 \tag{1}$$

with conserved variables and fluxes given by:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \mathbb{I} \\ \rho H \mathbf{u} \end{bmatrix} \tag{2}$$

having denoted by ρ the mass density, by $\mathbf{u}$ the velocity, by p the pressure, and with $E = e + \mathbf{u} \cdot \mathbf{u}/2$ the specific total energy, e being the specific internal energy. Finally, the total specific enthalpy is $H = h + \mathbf{u} \cdot \mathbf{u}/2$, with $h = e + p/\rho$ the specific enthalpy. For simplicity in this paper we work with the classical perfect gas equation of state:

$$p = (\gamma - 1)\rho e \tag{3}$$

with γ the constant (for a perfect gas) ratio of specific heats. However, note that the method discussed allows in principle to handle any other type of gas, see e.g. [36].

In all applications involving high-speed flows, solutions of (1) are only piecewise continuous. In d space dimensions, discontinuities are represented by $d - 1$ manifolds governed by the well known Rankine-Hugoniot jump conditions reading:

$$[\![\mathbf{F}]\!] \cdot \mathbf{n} = \omega [\![\mathbf{U}]\!] \tag{4}$$

having denoted by $\mathbf{n}$ the local unit vector normal to the shock, by $[\![\cdot]\!]$ the corresponding jump of a quantity across the discontinuity, and with ω the normal component of the shock speed.

As discussed in the introduction, the method proposed exploits ideas from two different approaches: the unstructured shock-fitting method [8] and subsequent works [9, 12, 10]; the shifted boundary method by [33] and subsequent works [34, 37].

Although the paper focuses on triangle based nodal solvers, the method proposed is also applicable to structured and unstructured cell-centered solvers [38].

3. Extrapolated DIscontinuity Tracking (*eDIT*)

3.1. Generalities

To illustrate the algorithmic features of the *eDIT* method, let us consider a two-dimensional domain and a discontinuity front crossing the domain at a given time t. As shown in Fig. 1a, the

front is described using a collection of edges whose endpoints are marked by squares. Throughout the paper these two entities will be referred to as *front-edges* or *discontinuity-edges*, and *front-points* or *discontinuity-points* respectively, and together they constitute what will be referred to as *front-mesh* or *discontinuity-mesh*. The computational domain itself is discretized by means of a background mesh. As already said, we consider here solvers based on a nodal variable arrangement on triangular grids, but the extrapolation proposed readily applies to cell-centered solvers. Figure 1a clearly shows that the position of the discontinuity-points is completely independent of the location of the grid-points of the *background mesh*. Regardless of the method used to solve the Euler equations on the background mesh, the representation of the flow variables across the discontinuity is discontinuous. In our case, two values of the flow variables are stored in each front-point. Other arrangements (edge based for example) are of course possible. The flow solvers considered here are based on a continuous nodal approximation, which amounts to store one solution value at each grid-point of the background mesh. As already said, this is not essential for the method developed.

Assume that at time $\underline{t}$ the solution is known at all grid- and discontinuity-points. The computation of the subsequent time level $\underline{t+\Delta t}$ using the *eDIT* method can be split into several steps that will be described in detail in the following sub-sections.

3.2. Geometrical setting

The first step consists in flagging the triangles crossed by the front. As shown on Fig. 1b, this leads to the creation of a cavity of flagged elements enclosing the discontinuity. Here, in contrast to the unstructured technique proposed in [8], or to more classical methods such as the boundary shock fitting [39], the mesh within or in proximity of this cavity is not modified to be conformal with the discontinuity. This cavity separates two sub-domains, and allows to define two surrogate boundaries, which are the intersections between the boundaries of the sub-domains and the boundaries of the front cavity. The mesh of the sub-domains separated by the front cavities will be referred to as the *computational mesh*. The computational mesh is identical to the background mesh, except for the removal of the flagged elements enclosing the discontinuity. In the simplest setting of a single discontinuity, the upstream and downstream surrogate boundaries, drawn using red lines in Fig. 1b, will be called $\tilde{\Gamma}_U$ and $\tilde{\Gamma}_D$. In the following we also refer to them as surrogate-discontinuities. The discontinuity-boundary, representing the actual front position, will be referred to as Γ and its upstream and downstream sides as Γ_U and Γ_D, respectively. Contrary to $\tilde{\Gamma}_U$ and $\tilde{\Gamma}_D$, Γ_U and Γ_D are superimposed and, of course, coincide with Γ.

Compared to its original version [1], the current algorithm is capable of dealing with several discontinuity-fronts which implies that the computational domain is split into several, disjoint, sub-domains.

To apply the Rankine-Hugoniot jump relations (4) within each pair of discontinuity-points, we need to define the tangent and normal directions, $\boldsymbol{\tau}$ and $\mathbf{n}$, in correspondence of each of these points. The computation of these vectors is carried out using finite-difference formulae which involve the coordinates of the front-point itself and those of the neighboring front-points that belong to its range of influence. Specifically, centred finite difference formulae will be used when the downstream state is subsonic whereas upwind formulae are employed whenever the downstream state is supersonic. Further details are given in [1, 40]. The unit vector $\boldsymbol{n}$ is conventionally assumed to be directed from the high entropy (downstream) side towards the low entropy (upstream) side of the discontinuity, see Fig. 2. It is also convenient to express the components of the velocity vector in the $(\boldsymbol{n}, \boldsymbol{\tau})$ reference frame which is locally attached to the discontinuity:

$$u_n = \mathbf{u} \cdot \mathbf{n} \quad \text{and} \quad u_\tau = \mathbf{u} \cdot \boldsymbol{\tau} \tag{5}$$
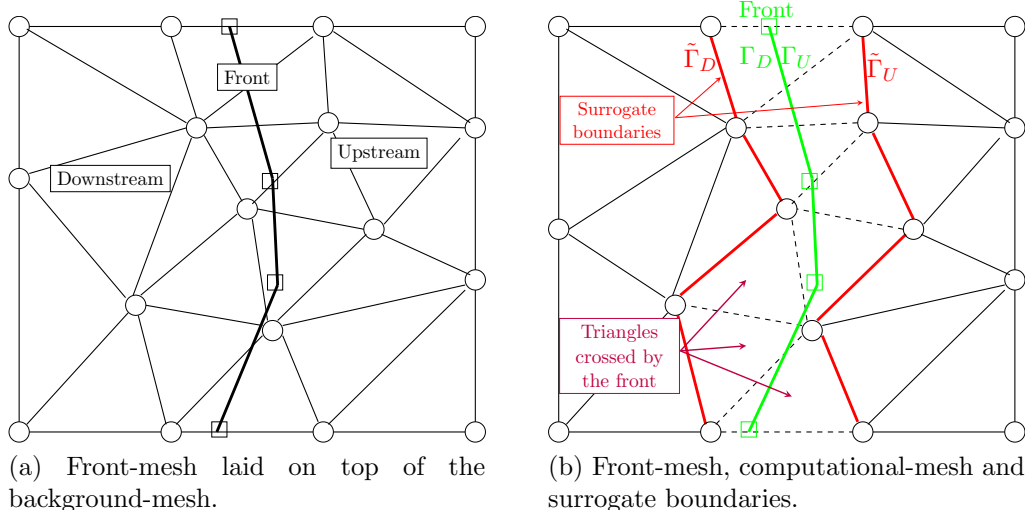
(a) Front-mesh laid on top of the background-mesh.

(b) Front-mesh, computational-mesh and surrogate boundaries.

Figure 1: The computational-mesh is obtained by removing those cells of the background mesh that are crossed by the front-mesh.

3.3. Definition of the sub-domains

In principle the flow may evolve n different fronts, which need to be handled numerically. The identification of these discontinuities is in itself a challenging problem (see e.g. [41, 42] and references therein), which is out of the scopes of this work. Here we assume to be given in advance a set $\{\mathcal{F}_j\}_{j=1}^n$ of n discontinuities, as well as their topology and nature (shocks and/or contact discontinuities), and the set of numbered sub-domains $\{\Omega_l\}_{l=1}^m$ separated by the discontinuities. We also start from a brute set of the ensemble of all points and edges of the surrogate discontinuities, which we denote by $\tilde{\Gamma}$. We process this ensemble as follows:

1. associate to each point in $\tilde{\Gamma}$ the index of the closest discontinuity (distance measured by orthogonal projection);
2. associate to each point in $\tilde{\Gamma}$ the Upstream/Downstream flag based on its position w.r.t. the orientation of the front normals;
3. for each discontinuity assemble the corresponding arrays of upstream and downstream surrogate boundaries (edge collection) as better shown in Fig. 2;
4. build a pointer providing the explicit mapping between actual and surrogate discontinuities.

The connectivity obtained allows to move easily from one surrogate discontinuity to the corresponding front-mesh, or to the other surrogate corresponding to the same discontinuity (e.g. move between the upstream and downstream surrogates $\tilde{\Gamma}_{Uj}$ and $\tilde{\Gamma}_{Dj}$ of the two shocks of Fig. 2), as well as between different surrogates bounding the same sub-domain (e.g. from $\tilde{\Gamma}_{D1}$ to $\tilde{\Gamma}_{U2}$ in Fig. 2).

3.4. Solution update using the CFD solver

The solution is updated to time level $\underline{t+\Delta t}$ using a shock-capturing code. The flow computations are performed on the non-communicating sub-domains separated by the front cavity, and including the surrogate discontinuities $\tilde{\Gamma}_U$ and $\tilde{\Gamma}_D$ as boundaries (see Fig. 3). As we will see in the next section, the boundary conditions imposed on the surrogate discontinuities are defined starting from the values of the flow variables on Γ_U and Γ_D which, as already said, are in general different, because Γ is a discontinuity.
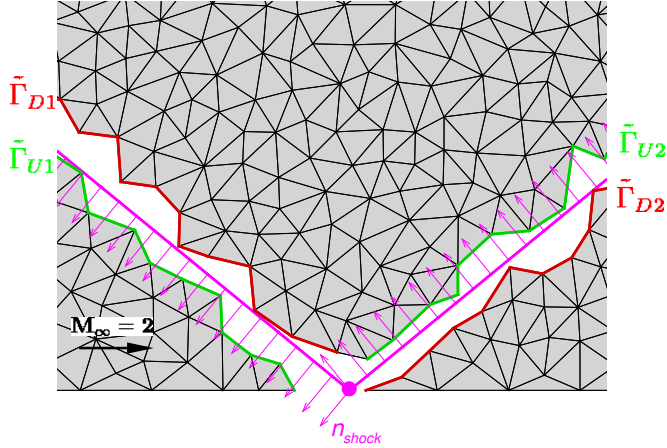
5

Figure 2: Definition of surrogate boundaries when multiple fronts interact; in the regular reflection shown here, the surrogate boundaries marked in green are located upstream w.r.t. the incident and reflected shocks, whereas those marked in red are located downstream.
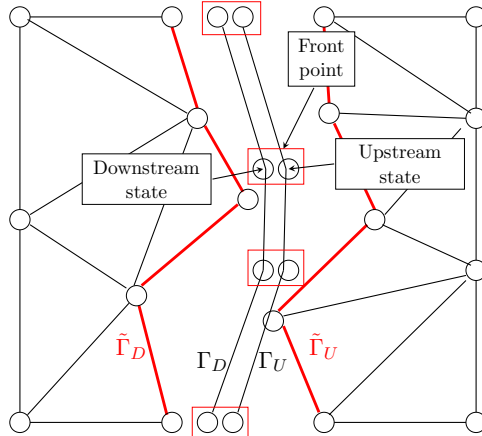

Figure 3: The solution update is performed using the computational mesh.

Concerning the solver used in this paper, it is based on a <u>Residual Distribution (RD)</u> method evolving in time approximations of the values of the flow variables in grid-points. The method has several appealing characteristics, including the possibility of defining genuine multidimensional upwind strategies for Euler flows, by means of a wave decoupling exploiting appropriately preconditioned forms of the equations [43]. By combining ideas from both the stabilized finite element and finite volume methods, these schemes allow to achieve second order of accuracy and monotonicity preservation with a compact stencil of nearest neighbors. The interested reader can refer to [44, 45] and references therein for an in-depth review of this family of methods, as well as to [43, 46] and references therein for some specific choices of the implementation used here.

3.5. Surrogate discontinuity conditions update

The flow solver provides updated nodal values at all grid-points of the computational mesh at time level $t + \Delta t$. When dealing with *shock waves*, the shock-upstream surrogate boundary, $\tilde{\Gamma}_U$ behaves like a supersonic outflow and, therefore, no boundary conditions should be applied. However,

along the shock-downstream surrogate $\tilde{\Gamma}_D$ the flow is subsonic in the shock-normal direction and in principle only the characteristic variable[1] conveyed by the slow acoustic wave has been correctly updated, while boundary conditions for the remaining ones (the fast acoustic, entropy and vorticity waves) should be imposed. The situation is similar on both sides of a *contact discontinuity*.

Updating correctly the interface conditions across the embedded discontinuity is very delicate, and it is the key of the method proposed. It involves several steps, which are detailed hereafter.

3.5.1. Computational mesh to discontinuity mesh transfer

The transfer of the flow data from the computational mesh to the front-mesh is necessary before imposing the interface conditions. In this work we use a CFD solver strongly relying on the use of Roe's parameter vector $\mathbf{Z} = \sqrt{\rho}\,(1, H, u, v)^t$ [47, 48] as the dependent variable, so the transfer is also done in terms of $\mathbf{Z}$. This is evidently not a necessary choice, and any other choice of state vector is acceptable. This first transfer is required to update the flow variables along Γ_U and Γ_D. Following [37, 1], this is achieved by first defining a map:

$$\tilde{\mathbf{x}} = M(\mathbf{x}) \tag{6}$$

where, as shown in Fig. 4, the point $\tilde{\mathbf{x}} = \mathbf{x}(A^i)$ belonging to the computational mesh is the projection in the direction of the front-normal $\mathbf{n}$ of a front-point $\mathbf{x} = \mathbf{x}(P_i)$ of the front-mesh. Then we use a forward Taylor series expansion to express the data along the discontinuity in terms of the values of the flow variables and of their derivatives on the computational mesh. A first-order transfer reads:

$$\mathbf{Z}(\mathbf{x}) = \mathbf{Z}(\tilde{\mathbf{x}}) + o(\|\mathbf{x} - \tilde{\mathbf{x}}\|), \tag{7}$$

while a second order extrapolation is written as:

$$\mathbf{Z}(\mathbf{x}) = \mathbf{Z}(\tilde{\mathbf{x}}) + \nabla\mathbf{Z}(\tilde{\mathbf{x}}) \cdot (\mathbf{x} - \tilde{\mathbf{x}}) + o(\|\mathbf{x} - \tilde{\mathbf{x}}\|^2). \tag{8}$$

With the exception of few methods based on a C^1 approximation (see e.g. [49]), in general, the gradients of the flow variables are undefined at mesh-points. To avoid handling specific singular cases, we have implemented (8) using interpolated reconstructed nodal gradients. Note that, in order to achieve an overall second order of accuracy in the calculation of $\mathbf{Z}(\mathbf{x})$, the approximation of the gradient in Eq. (8) only needs to be consistent, i.e. at least first-order-accurate.

In practice we proceed as shown in Fig. 4. When moving from the computational mesh to Γ_U (see Fig. 4a) a front-point P_i is mapped to a point A^i. The values of the dependent variables and of their gradients in A^i are interpolated from the neighbors A^i_1 and A^i_2:

$$\phi(A^i) = w_2\,\phi(A^i_1) + w_1\,\phi(A^i_2) \tag{9}$$

where ϕ is either $\mathbf{Z}$ or $\nabla\mathbf{Z}$, and w_1 and w_2 are the weights, equal to the normalized distances between A^i and grid-points A^i_1 and A^i_2. The evaluation of the gradient in the grid-points of the surrogate boundaries may be performed using different approaches, as reported in § 3.7. Once the value of $\mathbf{Z}$ and $\nabla\mathbf{Z}$ in point A^i has been computed using Eq. (9), $\mathbf{Z}$ in P_i is computed by means of Eq. (8), having set $\mathbf{x} = \mathbf{x}(P_i)$ and $\tilde{\mathbf{x}} = \mathbf{x}(A^i)$.
When moving from the computational mesh to Γ_D the procedure is identical: the front-point P_i is updated using its mapped point B^i (see Fig. 4b). Values of the relevant quantities are again linearly interpolated, and $\mathbf{Z}$ is computed by means of Eq. (8), having set $\mathbf{x} = \mathbf{x}(P_i)$ and $\tilde{\mathbf{x}} = \mathbf{x}(B^i)$.

---

[1]Hereafter, characteristic variables shall also be referred to as Riemann variables.

(a) From the surrogate boundary $\tilde{\Gamma}_U$ to the shock-upstream side of the front Γ_U.

(b) From the surrogate boundary $\tilde{\Gamma}_D$ to the shock-downstream side of the front Γ_D.
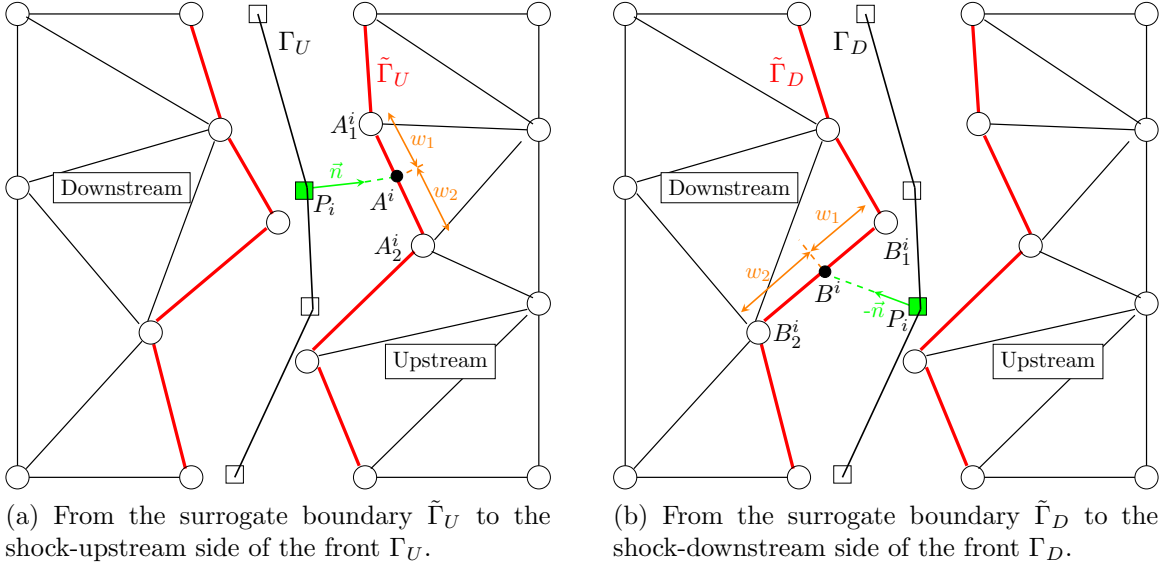
Figure 4: First transfer between the the surrogate boundaries and the front-mesh.

3.5.2. Riemann variables and jump conditions enforcement

The solution updated by the flow solver and extrapolated to the discontinuity mesh does not take into account the coupling between the different sub-domains. This coupling is done here based on the Rankine-Hugoniot jump relations. On the upstream side Γ_U of a *shock wave* all the Riemann variables are transported into the shock, so the data extrapolated from the computational mesh on this side is assumed to be correct. On the shock-downstream side, however, this holds true only for the Riemann variable associated with the slow acoustic wave that moves towards the shock (the subscript $_D$ denotes values along Γ_D):

$$R_D = a_D^{t+\Delta t} + \frac{\gamma - 1}{2} (u_n)_D^{t+\Delta t} \tag{10}$$

We need to provide relations to compute along Γ_D the values of the Riemann variables corresponding to the fast acoustic wave, as well as to the entropy and vorticity waves. These relations are supplied by the Ranking-Hugoniot jump conditions which, for a perfect gas, can be recast as:

$$
\begin{aligned}
\rho_D^{t+\Delta t}(u_n)_D^{t+\Delta t} - \omega_s^{t+\Delta t}\rho_D^{t+\Delta t} &= \rho_U^{t+\Delta t}(u_n)_U^{t+\Delta t} - \omega_s^{t+\Delta t}\rho_U^{t+\Delta t} \\
\rho_D^{t+\Delta t}((u_n)_D^{t+\Delta t} - \omega_s^{t+\Delta t})^2 + p_D^{t+\Delta t} &= \rho_u^{t+\Delta t}((u_n)_U^{t+\Delta t} - \omega_s^{t+\Delta t})^2 + p_U^{t+\Delta t} \\
\frac{\gamma}{\gamma - 1}\frac{p_D^{t+\Delta t}}{\rho_D^{t+\Delta t}} + \frac{1}{2}((u_n)_D^{t+\Delta t} - \omega_s^{t+\Delta t})^2 &= \frac{\gamma}{\gamma - 1}\frac{p_U^{t+\Delta t}}{\rho_U^{t+\Delta t}} + \frac{1}{2}((u_n)_U^{t+\Delta t} - \omega_s^{t+\Delta t})^2 \\
(u_\tau)_D^{t+\Delta t} &= (u_\tau)_U^{t+\Delta t}
\end{aligned}
\tag{11}
$$

with $\omega_s^{t+\Delta t}$ the shock speed. The algebraic non-linear system made up of Eqs. (10) and (11) allows to compute the five unknowns $(\rho_D, \mathbf{u}_D, p_D, \omega_s)$ given the known upstream state $(\rho_U, \mathbf{u}_U, p_U)$ and the slow acoustic Riemann variable R_D.

For a *contact discontinuity*, we use as input the following set of Riemann variables, which we assume to be correctly updated in the computational mesh:

$$R_D = a_D^{t+\Delta t} + \frac{\gamma - 1}{2}(u_n)_D^{t+\Delta t} \ , \ \ R_U = a_U^{t+\Delta t} - \frac{\gamma - 1}{2}(u_n)_U^{t+\Delta t}$$

$$s_D = \frac{p_D^{t+\Delta t}}{(\rho_D^{t+\Delta t})^\gamma} \ , \ \ s_U = \frac{p_U^{t+\Delta t}}{(\rho_U^{t+\Delta t})^\gamma} \tag{12}$$

$$V_D = (u_\tau)_D^{t+\Delta t} \ , \ \ V_U = (u_\tau)_U^{t+\Delta t}$$

Note that when dealing with a contact discontinuity the situation is essentially symmetric and the jump relations simplify somewhat, and can be shown to reduce to:

$$\begin{aligned} p_U^{t+\Delta t} &= p_D^{t+\Delta t} \\ (u_n)_U^{t+\Delta t} &= \omega_{cd}^{t+\Delta t} \\ (u_n)_D^{t+\Delta t} &= \omega_{cd}^{t+\Delta t} \end{aligned} \tag{13}$$

The algebraic non-linear system made up of Eqs. (12) and (13) allows to compute the nine unknowns $(\rho_U, \mathbf{u}_U, p_U, \rho_D, \mathbf{u}_D, p_D, \omega_{cd})$, given the six Riemann variables (R, s, V) on the two sides of the discontinuity and the jump relations (13).

For both shocks and contact discontinuities, a non-linear system of algebraic equations needs to be solved in each discontinuity-point. This is done using the Newton-Raphson root-finding algorithm, thus providing the correct states and ω at time level t+Δt.
Whenever two or more discontinuities interact special care is required for the front-points belonging to different discontinuities. The numerical treatment of these points is done in a case by case manner and detailed in the sections related to the specific tests in § 4.

3.5.3. Discontinuity mesh to computational mesh transfer

Once the value of the unknowns have been corrected along the discontinuity mesh to account for the jump conditions, the corrected values need to be transferred back to the surrogate discontinuities. To do this, we proceed in a similar manner as before by writing:

$$\mathbf{Z}(\tilde{\mathbf{x}}) = \mathbf{Z}(\mathbf{x}) - \nabla \mathbf{Z}(\tilde{\mathbf{x}}) \cdot (\mathbf{x} - \tilde{\mathbf{x}}) + o(\|\mathbf{x} - \tilde{\mathbf{x}}\|^2). \tag{14}$$

Note that there is a substantial difference in the direction in which the Taylor expansion is performed which is now a backward expansion from the arrival point (the surrogate discontinuity) to the point where the data is available (the discontinuity mesh). As in the previous case, several choices are possible to evaluate the gradient involved in this transfer. We shall focus on this in § 3.7. In the first-order accurate case, Eq. (14) simplifies to:

$$\mathbf{Z}(\tilde{\mathbf{x}}) = \mathbf{Z}(\mathbf{x}) + o(\|\mathbf{x} - \tilde{\mathbf{x}}\|) \tag{15}$$

Finally note that for shock-waves only grid-points on $\tilde{\Gamma}_D$ need to be updated.

3.6. Front displacement and nodal re-initialization

The new position of the front at time level t+Δt is computed by displacing all discontinuity-points using the following first-order-accurate (in time) formula:

$$\mathbf{x}\left(P^{t+\Delta t}\right) = \mathbf{x}\left(P^t\right) + \omega^{t+\Delta t}\mathbf{n}\,\Delta t \tag{16}$$
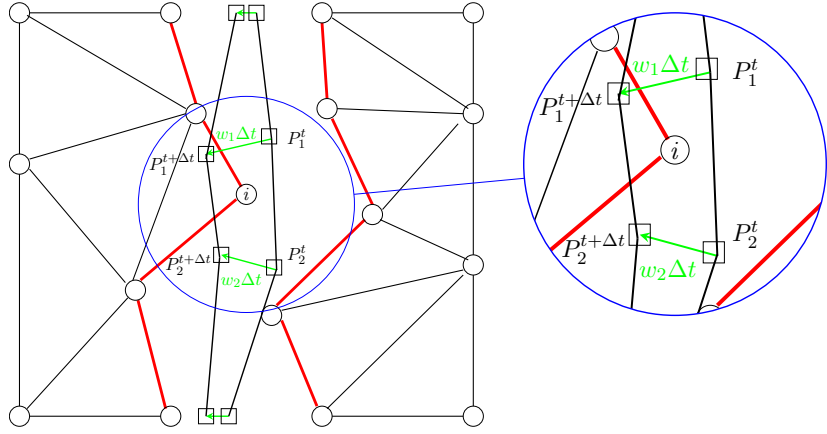
Figure 5: The front overtakes a grid-point of the background mesh during its motion.

where $\mathbf{x}(P)$ denotes the geometrical location of the shock-points, and $\omega^{t+\Delta t}$ may represent either $\omega_s^{t+\Delta t}$ or $\omega_{cd}^{t+\Delta t}$. The use of a first-order-accurate temporal integration formula has no impact as long as steady flows are of interest. For unsteady flows, second-order-accurate time integration formulae should be used, as done for example in [10, 11].

Since the discontinuity can freely float over the background triangulation, it may happen that it crosses the surrogate boundaries. This situation has been sketched in Fig. 5, where grid-point i has been overtaken by the moving front. Whenever this happens, the flow state within grid-point i has to be changed accordingly. In particular, the flow variables in these points are re-computed through an interpolation. To evaluate whether a grid-point i has been overtaken or not by the discontinuity a simple approach has been implemented. Since $\mathbf{n}$ points always upstream, the sign of the following scalar product allows us to distinguish the grid-points located upstream from those located downstream.

$$\alpha_i = (\mathbf{x}_p - \mathbf{x}_i) \cdot \mathbf{n} = \begin{cases} < 0 \text{ , } i \text{ is upstream} \\ > 0 \text{ , } i \text{ is downstream} \end{cases} \tag{17}$$

where $\mathbf{x}_i$ and $\mathbf{x}_p$ are, respectively, the coordinates of a grid-point i and its projection over the closest front-edge and $\mathbf{n}$ is the normal vector computed in the closest front-point to $\mathbf{x}_i$. If α_i changes sign when the front moves, this means that grid-point i has been overtaken and its state has to be updated.

3.7. Evaluation of the nodal gradients

The technique used to extrapolate the flow variables back and forth between the surrogate discontinuities and the discontinuity mesh involves the knowledge of nodal gradients which need to be reconstructed. The solvers used in this work are based on collocated nodal methods, so we will discuss the recovery of the nodal gradients in this specific case. The generalization to other cases is of course possible, but left out of this work.

As for the enforcement of the jump conditions we distinguish two main cases. For a shock wave, in the upstream domain we already said that all characteristic information runs into the shock. For this situation, it seems natural to evaluate the nodal gradients along the surrogate discontinuity based on the data pre-computed by the flow solver within the upstream domain. To this end we have tested two gradient recovery strategies:
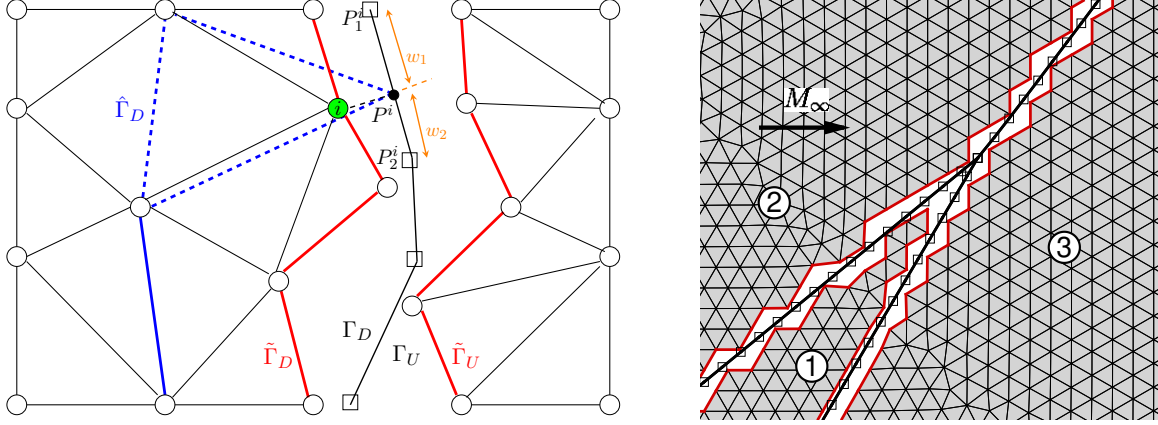
10

Figure 6: Left: auxiliary triangle defining the nodal gradient with corrected data on the front. Right: a problematic example of an interaction for which no auxiliary triangle can be built close to the interaction point in domain 1.

- a Green-Gauss (GG) gradient reconstruction, essentially boiling down to an area weighted average of the gradients in the cells surrounding a grid-point;

- an area-weighted version of Zienkiewicz-Zhu's patch super-convergent method (ZZ) [50].

Explicit formulas for both methods are provided in Appendix A.

Within the region downstream of a shock, as well as for contact discontinuities, as discussed in the previous sections, the data computed by the flow solver must be corrected to account for the jump conditions. This fact has led us in the past to include the corrected data in the evaluation of the nodal gradients in these cases [1]. In the aforementioned reference, which did not account for discontinuity interactions, the gradient term of Eq. (14) is replaced with the cell-wise gradient of an auxiliary triangle (marked using a dashed line in Fig. 6) containing the surrogate grid-point to be updated (grid-point i in Fig. 6), and defined by a point on the discontinuity-mesh (point P^i in Fig. 6), and two grid-points of the computational mesh, not belonging to the surrogate discontinuity. However, when interactions are present, singular topological situations may arise in which this approach cannot be used. For example, for region 1 in Fig. 6, a long strip of one row of elements is trapped within two discontinuities and the interaction point. For the grid-points along the boundary of this strip there are no neighboring inner grid-points to construct the auxiliary triangle. Several other gradient-reconstruction formulations have also been tested. The simplest involves using a one-sided recovered gradient not accounting for the Rankine-Hugoniot correction. The latter not only turned out to be very useful for computing interacting discontinuities, but much simpler, especially in view of possible extensions to three space dimensions. In the following, we will refer to the method of [1] as eST, while $eDIT_{GG}$ (GG reconstruction), and $eDIT_{ZZ}$ (ZZ reconstruction) will denote the extrapolated discontinuity tracking obtained using one sided reconstructions. Finally, $eDIT_{FO}$ (first order) refers to simulations run with a second-order-accurate discretization of the governing PDEs, but only first-order accurate data transfer between the surrogate and actual discontinuities, i.e. by using Eqs. (7) and (15).

4. Numerical Results

To illustrate the capabilities of our method, we provide here examples representative of several types of interactions which can occur in gas-dynamics. The solutions obtained with the extrapolated tracking method will be compared to i) full-fledged computations, ii) hybrid simulations in which only some of the discontinuities are explicitly tracked while the others are captured, and iii) solutions obtained with the standard shock-capturing method. To evaluate the gradient recovery strategies, a grid-convergence analysis is presented for two test-cases (transonic source flow and blunt body problem) to asses quantitative differences between the approach described in [1] and the more flexible ones proposed here to deal with interactions.

4.1. Transonic source flow

This test-case is very useful because of the availability of the analytical solution, which allows to perform grid-convergence studies [1, 51, 52]. Assuming that the analytical velocity field has a purely radial velocity component, it may be easily verified that the two-dimensional, compressible Euler equations, written in a polar coordinate system, become identical to those governing a compressible quasi-one-dimensional flow with a nozzle area variation linear w.r.t. the radial distance from the pole of the reference frame. The computational domain consists in the annulus sketched in Fig. 7a: the

(a) Sketch of the computational domain.

(b) Detail of the level 0 unstructured mesh inside the first gradient.

Figure 7: Transonic source flow.

ratio between the radii of the outer and inner circles ($L = r_{in}$) has been set equal to $r_{out}/r_{in} = 2$. A transonic (shocked) flow has been simulated by imposing a supersonic inlet flow at $M = 2$ on the inner circle and a ratio between the outlet static and inlet total pressures $p_{out}/p_{in}^0 = 0.47$ such that the shock forms at $r_{sh}/r_{in} = 1.5$. The Delaunay mesh shown in Fig. 7b, which contains 6,916 grid-points and 13,456 triangles, has been generated using the `gmsh` mesh generator [53] in such a way that no systematic alignment occurs between the edges of the triangulation and the circular iso-contour lines of the analytical solution. By doing so, the discrete problem is made truly two-dimensional. The sequence of nested triangulations that have been employed for SC and all $eDIT$ computations are summarized in Tab. 1 for both the background and the computational grids.

The discretization error for a mesh of size h, ϵ_h, is defined as the difference between the numerical solution, u_h, and the analytical one, u_0:

$$\epsilon_h(\mathbf{x}) = u_h(\mathbf{x}) - u_0(\mathbf{x}) \tag{18}$$

12

Table 1: Transonic source flow: characteristics of the background and computational meshes used to perform the grid-convergence tests.

Grid level	Background grid		Both grids	Computational grid	
	Grid-points	Triangles	h	Grid-points	Triangles
0	1,369	2,548	0.5286E-01	1,369	2,328
1	5,286	10,192	0.2641E-01	5,286	9,788
2	20,764	40,768	0.1320E-01	20,764	39,968
3	82,296	163,072	0.6602E-02	82,296	161,454

The availability of the exact solution allows to compute the discretization error *locally* using Eq. (18) or, *globally*, by computing the L_q norms of the discretization error over the entire computational domain Ω using Eq. (19):

$$L_q(\epsilon_h) = \left(\frac{1}{|\Omega|} \int_\Omega |\epsilon_h(\mathbf{x})|^q \, d\Omega \right)^{1/q}, \qquad q = 1, 2, \infty. \tag{19}$$

Herein, the L_1 norm has been employed and computed using Gaussian quadrature rules.

Different versions of the extrapolated shock/discontinuity tracking method will be compared to SC computations by doing a thorough grid-convergence analysis. Results obtained from the aforementioned approaches have been separately displayed in the grid-convergence plots for the shock-upstream (supersonic) and shock-downstream (subsonic) regions, i.e. $r < r_{sh}$ and $r > r_{sh}$. In all convergence plots the errors are computed in terms of the parameter vector $\mathbf{Z}$.

Firstly, the global errors obtained on the shock-upstream side of the domain have been displayed to corroborate the fact that, for the supersonic region, the results obtained with all methods are almost identical. Indeed, Fig. 8 shows that, on the shock-upstream side, all approaches are able to retain the formal order of accuracy of the method. Instead, when looking at the results shown in Fig. 9, which shows the global errors within the subsonic region, the situation is considerably different. An almost asymptotic convergence is observed for all second order versions of the extrapolated tracking method (eST, $eDIT_{GG}$ and $eDIT_{ZZ}$). Instead, for SC, the convergence rate drops to one, and even less for the coarsest meshes. eST and $eDIT_{GG}$ solutions are almost indistinguishable, with global errors that favor more the first approach, probably due to the use of flow data explicitly accounting for the jump conditions also in the extrapolation. The $eDIT_{ZZ}$ is still comparable having a trend in the middle between the two. Although the differences are not remarkable, using a more accurate reconstruction seems to have some effect on the magnitude of the error. The improvement is however so small that the eST can be safely replaced by either $eDIT_{GG}$ or $eDIT_{ZZ}$. This also shows that our basic idea can be coupled to different extrapolation methods, and others, possibly improved ones, could be suggested in the future. As expected, $eDIT_{FO}$ performs as the others in the supersonic side of the domain showing a second-order trend. However in the subsonic side, due to the first order extrapolations carried out at the shock, its trend follows the first order slope curve, because it does not include the nodal gradients in the extrapolation.

Finally, Fig. 10, which plots the local discretization error against the radial distance r, clearly reveals the huge reduction of the numerical errors that the various $eDIT$ approaches provide w.r.t. SC. The green line in Fig. 10 represents the actual position where the shock occurs ($r = 1.5$). As already verified in Fig. 8, the error computed with either SC or any of the $eDIT$ versions within the shock-upstream region is almost the same.

Iso-contour lines of the fourth component of $\mathbf{Z}$ computed on *grid level 2* are displayed in Fig. 11: the SC solution is shown in Fig. 11a, whereas Fig. 11b shows the results of two different $eDIT$
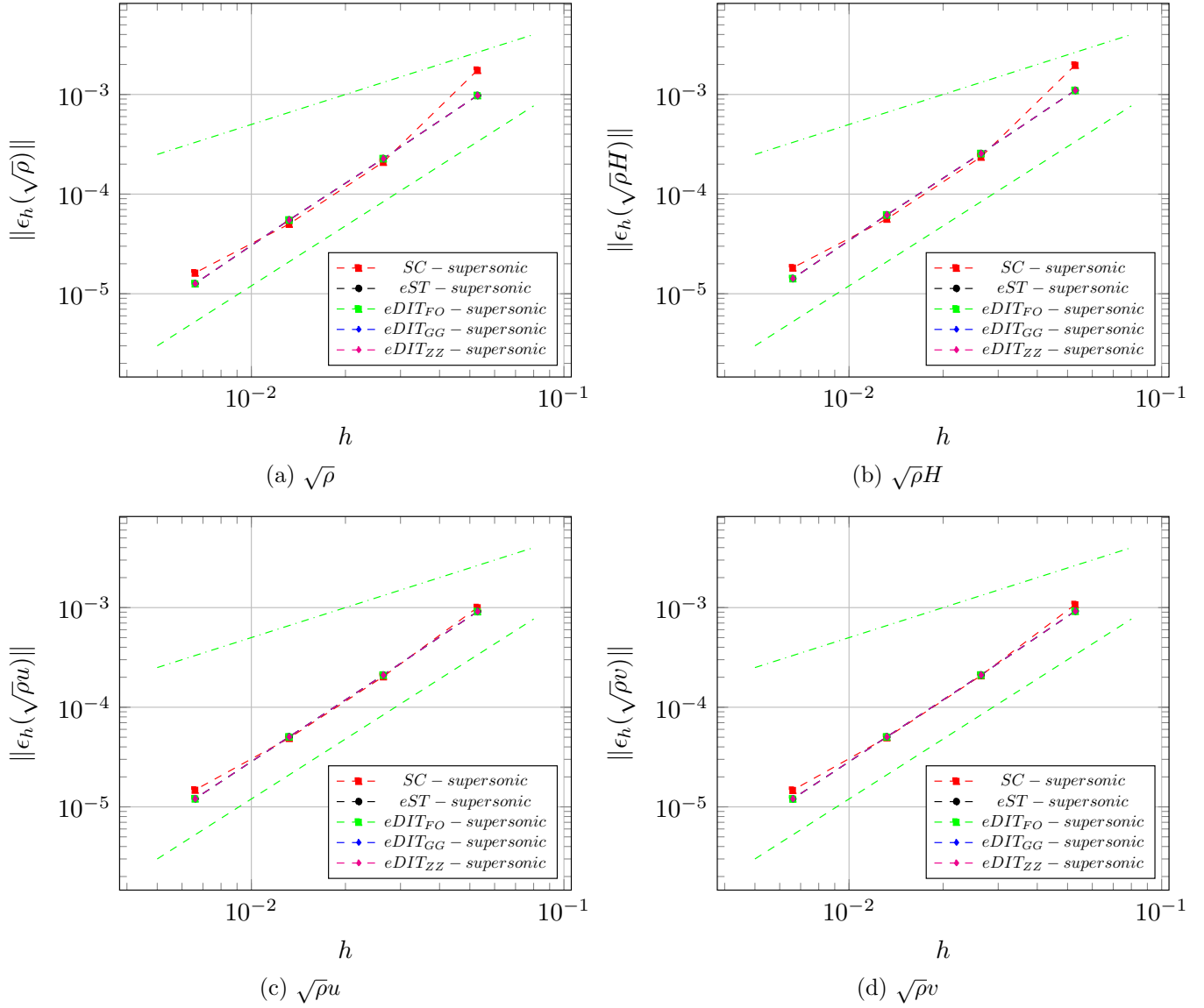
Figure 8: Transonic source flow: grid-convergence analysis within the shock-upstream (supersonic) region.
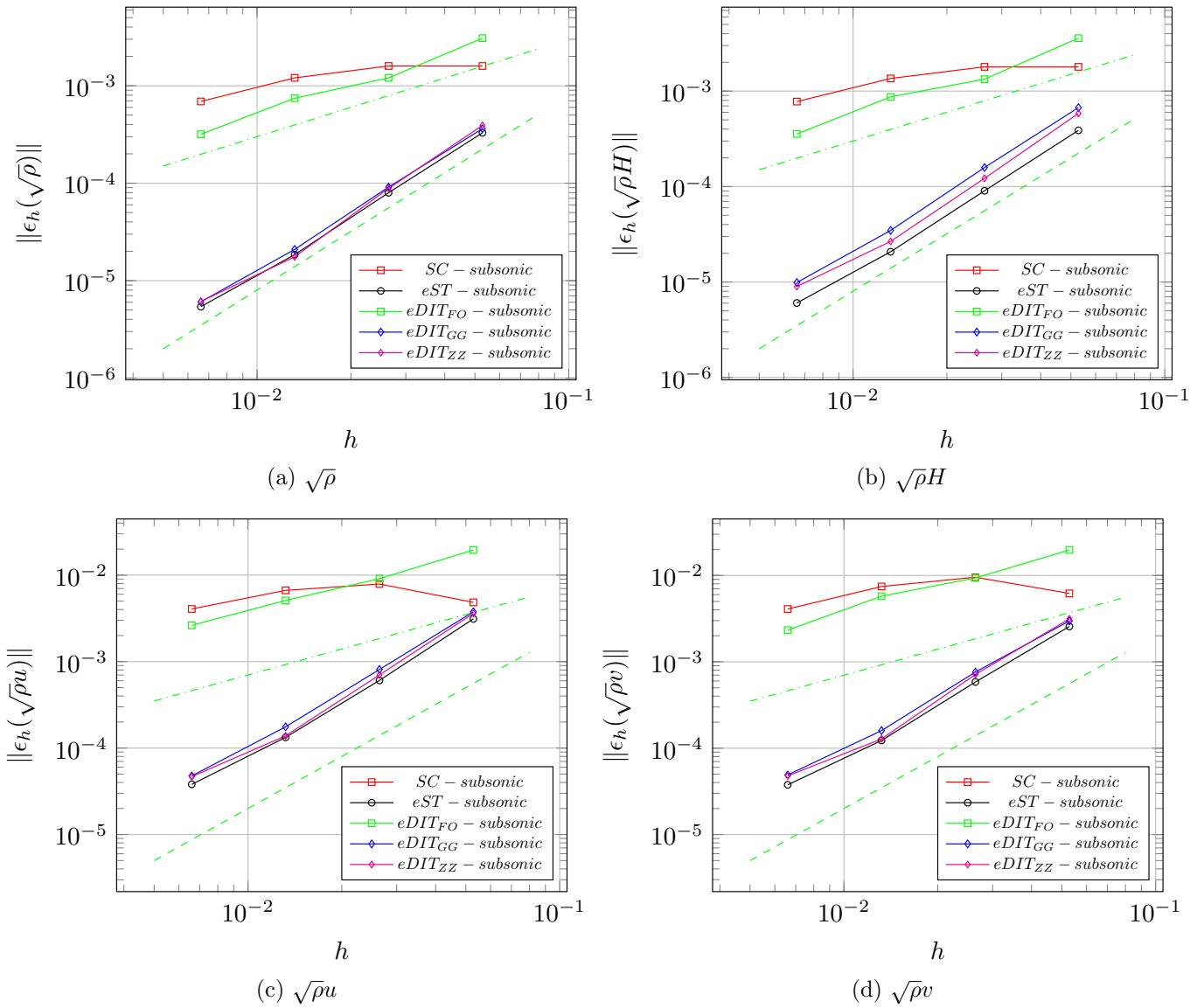
Figure 9: Transonic source flow: grid-convergence analysis within the shock-downstream (subsonic) region.

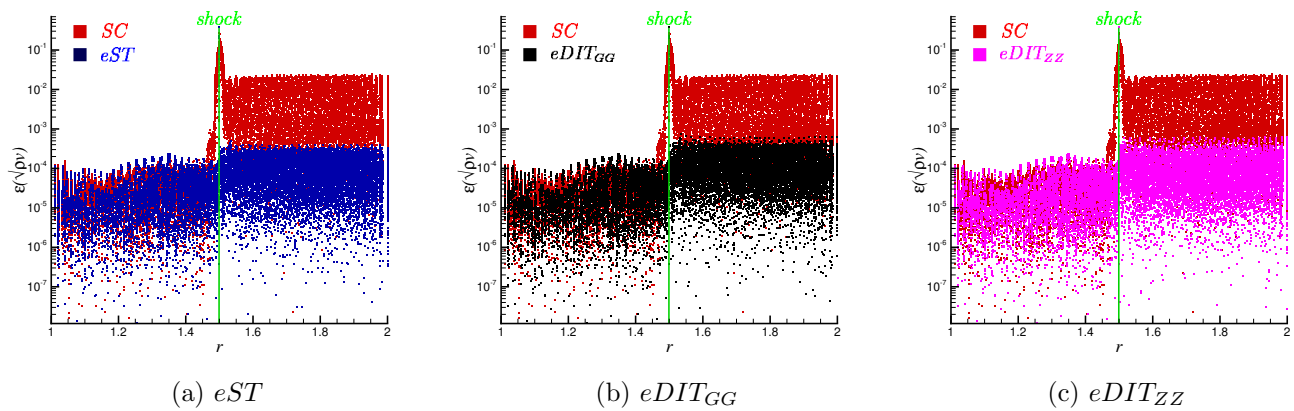(a) eST (b) $eDIT_{GG}$ (c) $eDIT_{ZZ}$

Figure 10: Transonic source flow: local discretization error analysis carried out on *grid level 2* to compare the *SC* computations to the *eDIT* ones w.r.t. $Z_4 = \sqrt{\rho}v$.
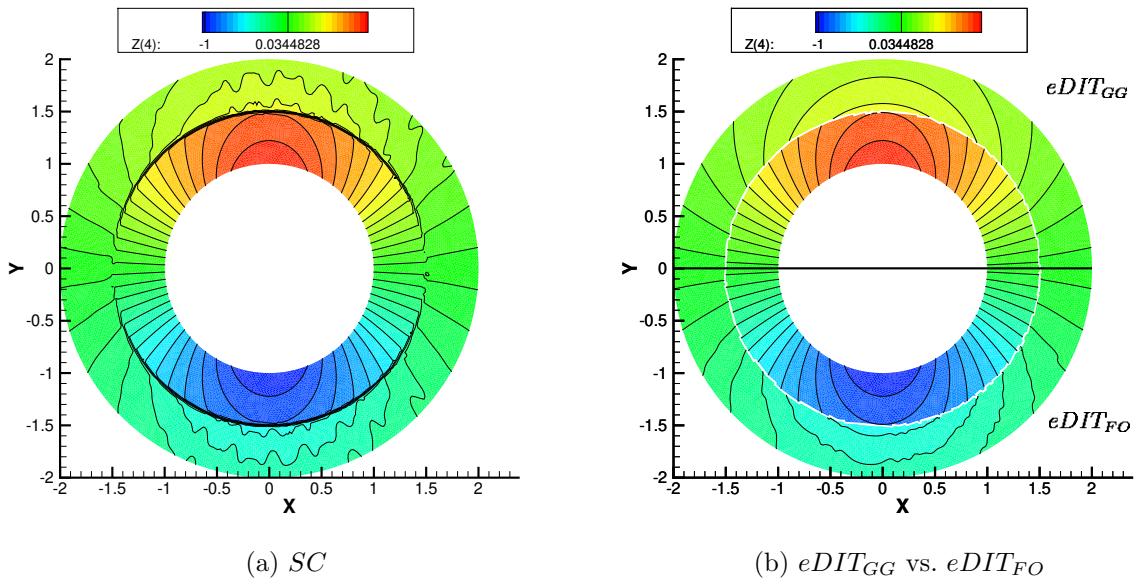
(a) SC (b) $eDIT_{GG}$ vs. $eDIT_{FO}$

Figure 11: Transonic source flow: iso-contours comparison on *grid level 2* w.r.t. $Z_4 = \sqrt{\rho}v$.

calculations. More precisely, the second-order-accurate $eDIT_{GG}$ result is displayed in the upper half of Fig. 11b and $eDIT_{FO}$ is shown in the lower half of the same frame. When mutually comparing the three different calculations, it can be seen that both the SC and $eDIT_{FO}$ calculations are affected by severe oscillations downstream of the shock, which completely disappear when using $eDIT_{GG}$. Even though the $eDIT_{FO}$ solution looks slightly better than the SC one, Fig. 9 confirms that $eDIT_{FO}$ cannot be better than first-order accurate behind the shock.

4.2. Hypersonic flow past a blunt body

We consider now a hypersonic ($M_\infty = 20$) flow past the forebody of a circular cylinder. Compared to the previous test-case, this flow is a comprehensive test-bed for the algorithm, because the entire shock-polar is swept whilst moving along the bow shock which forms ahead of the blunt body. Moreover, as shown in Fig. 12a, the flow-field is much more complex than that examined in § 4.1, due to the presence of a stagnation region within a subsonic pocket, as well as a smooth re-acceleration of the flow to supersonic conditions along the body. Even though no analytical solution is available for the entire flow-field, we know that total enthalpy, H, is preserved along streamlines at steady state. For a constant profile of $H = H_\infty$ ahead of the shock, this leads to an exact solution featuring a homogeneous total enthalpy field, thus the condition $H = H_\infty$ can be used to study the convergence behaviour of the various combinations of numerical schemes and shock-modeling options. Starting from a coarse mesh (shown in Fig. 12b) obtained with the `delaundo` frontal/Delaunay mesh generator [54, 55], we created three levels of nested refined meshes with characteristics summarized in Table 2 for both the background and computational grids.

Table 2: Hypersonic flow past a blunt body: characteristics of the background and computational meshes used to perform the grid-convergence tests.

Grid level	Background grid		Both grids	Computational grid	
	Grid-points	Triangles	h	Grid-points	Triangles
0	351	610	0.16	351	535
1	1,311	2,440	0.08	1,311	2,292
2	5,061	9,760	0.04	5,061	9,460
3	19,881	39,040	0.02	19,881	38,439

16

(a) Sketch of the computational domain.

(b) Detail of the level 0 unstructured mesh.

Figure 12: Hypersonic flow past a blunt body.

The different extrapolated tracking methods discussed in the previous sections are expected to provide orders of convergence similar to those observed for the source flow of § 4.1, as well as smooth and clean results with very low numerical perturbations, even on the coarsest grids; this latter aspect is illustrated in Fig. 13, where the pressure iso-contour lines computed using $eDIT_{GG}$ on the level 0 (coarsest) and level 3 (finest) grids are mutually compared. Figure 14 shows the grid-convergence analysis pointing out again that the different extrapolation techniques provide error levels relatively close, and a second order slope, while the captured result converges with a less than first order rate, and errors of one or two orders of magnitude larger. The patch super-convergent gradient recovery (ZZ) provides a better result for this case.

Figure 15 shows the total-enthalpy discretization error over the entire computational domain for the two solutions obtained with SC and $eDIT_{GG}$ and two different grid-levels: the coarsest and the finest. The maximum value of the error is also explicitly given in each of the four frames. While confirming the lower error levels on the coarse mesh, and the rapid error convergence obtained with second-order shock-tracking, the four frames of Fig. 15 allow to visually highlight the substantial difference in error generation. Indeed, in both captured solutions (Fig. 15a and 15c), the largest error is generated along the shock. Conversely, for the two $eDIT_{GG}$ calculations the numerical error is essentially connected with the wall boundary condition, and undoubtedly related to the entropy generated at the stagnation point and advected downstream. The $eDIT_{ZZ}$ solution is virtually identical and not discussed for brevity.

As a final note, we remark that total enthalpy is a conserved variable if and only if the time derivative in Eq. (1) vanishes. This means that the convergence to steady state plays a major role in

Figure 13: Hypersonic flow past a blunt body: comparison between the solutions obtained with $eDIT_{GG}$ on *Grid level 0* and *Grid level 3* in terms of pressure iso-lines.
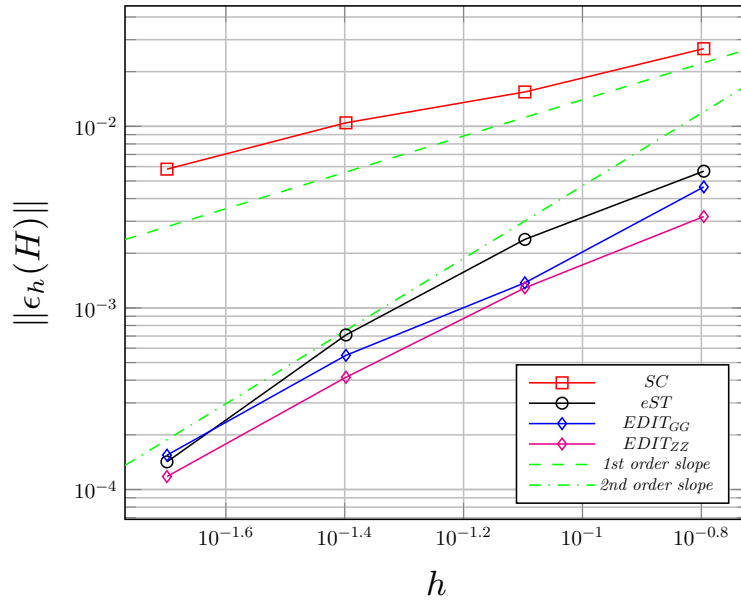

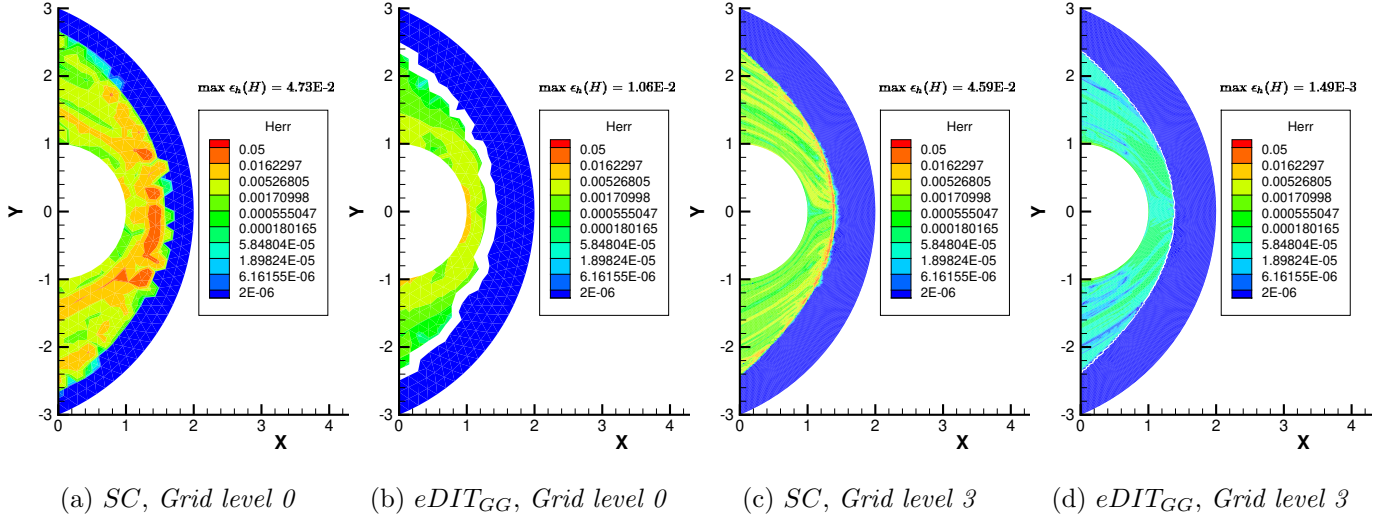Figure 14: Hypersonic flow past a blunt body: grid-convergence analysis.

(a) *SC, Grid level 0*   (b) *eDIT$_{GG}$, Grid level 0*   (c) *SC, Grid level 3*   (d) *eDIT$_{GG}$, Grid level 3*

Figure 15: Hypersonic flow past a blunt body: discretization error of the total enthalpy, $\epsilon_h(H)$, obtained with *SC* and *eDIT$_{GG}$* over the entire computational domain for *Grid level 0* and *Grid level 3*.

(a) *eST*   (b) *eDIT$_{GG}$*   (c) *eDIT$_{ZZ}$*

Figure 16: Hypersonic flow past a blunt body: Pseudo-time shock convergence.

allowing to correctly measure the decay rate of the discretization error. Convergence to steady-state is an important aspect of this type of methods, which is why we also report on Fig. 16 the iterative convergence of the *eDIT* algorithm when starting from a captured solution. In all our calculations we have set as a stopping criterion a threshold on the norm of the shock speed of $\sim 10^{-7}$. As Fig. 16 shows, this level is reached quite monotonically in all the computations. This despite the shock crossing some nodes during the iterations, which produces a change in topology of the surrogate discontinuities and of the computational domain which can be seen in some of the peaks appearing locally in the iterative convergence plots. Similar convergence curves are often hard to obtain with non-linear shock capturing methods.

4.3. Interaction between two shocks of the same family

The present and subsequent test-cases address the interaction among different discontinuities, a kind of flow-topology which had not been addressed in the first journal appearance [1] of the *eST* algorithm and thus represent one of the key novelties of this study.

19

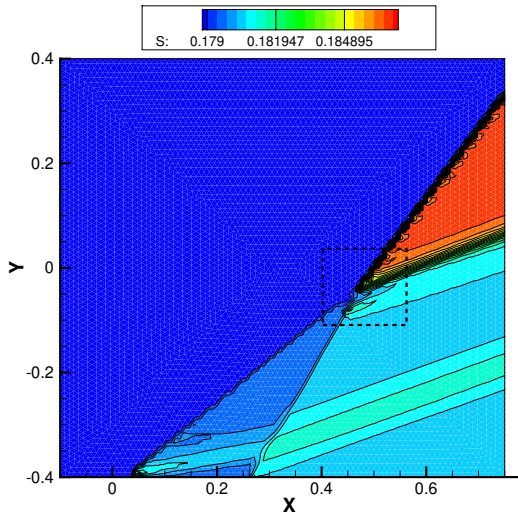Figure 17: Interaction between two shocks of the same family: sketch of the flow.

We consider here the interaction between two oblique shocks colliding and giving rise to a five-waves interaction. As shown in the sketch of Fig. 17, the two incident shocks (IS1 and IS2) of the same family interact in a quadruple point, referred to as QP in the figure, generating three reflected discontinuities: a strong reflected shock (RS1), with jumps of magnitude larger than the two incident ones; a contact discontinuity (CD); a weak reflected shock (RS2), with jumps so small that it could be considered as a Mach wave. More in general, depending on the free-stream Mach number and the flow deflection angles caused by the incident shocks, RS2 can be either a weak expansion or compression wave. With reference to Fig. 17, in this case the free-stream Mach number and the flow deflection angles are: $M_2 = 2$, $\theta_1 = 10°$ and $\theta_2 = 20°$ which can be shown to lead to states 4 and 5 characterized by $p_4 = p_5 = 2.822$, $M_4 = 1.218$, $M_5 = 1.28$ and $\theta_4 = \theta_5 = 19.87°$. For this choice of parameters, RS2 is so weak that we decided not to track this wave since its resolution has negligible overall effect on the flow.

Simulations of the interaction have been performed using a Delaunay triangulation containing 7,921 nodes and 15,514 triangles. As before, the same mesh is used to perform the SC simulation and as a background grid for the $eDIT_{GG}$ computations. Results with the $eDIT_{ZZ}$ are virtually identical to the latter, and not discussed.

As mentioned in § 3.5.2, the interaction points need to be modelled explicitly on a case by case basis. If only some of the discontinuities meeting at the interaction point are tracked, the only possibility is to solve independent algebraic problems arising from the jump conditions for each discontinuity, and use some approximate formula for QP. For the type of interaction considered here, we have used the following relation to compute the velocity of the interaction point QP:

$$\boldsymbol{\omega}_{QP} = \boldsymbol{\omega}_{IS1} + (\boldsymbol{\omega}_{IS2} \cdot \boldsymbol{\tau_1}) \, \boldsymbol{\tau_1} \tag{20}$$
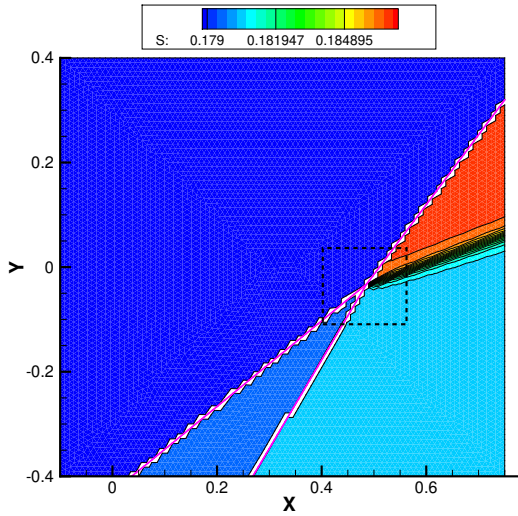
where $\boldsymbol{\tau_1}$ is the vector tangential to IS1 (see Fig.17a). Equation (20) is similar, but not identical, to the formula proposed in [19] for the same purpose.
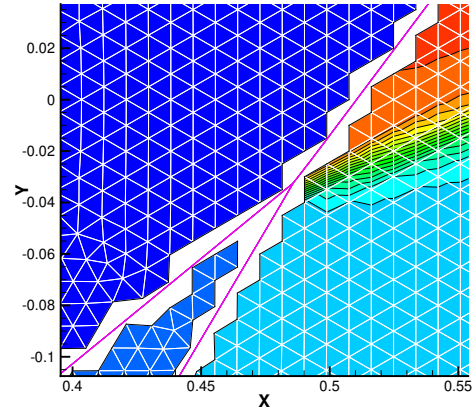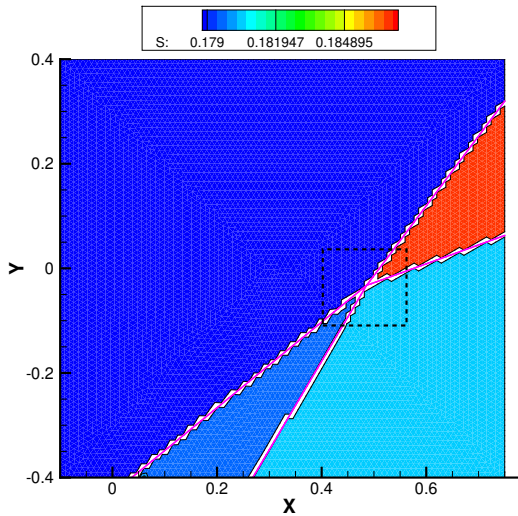
20

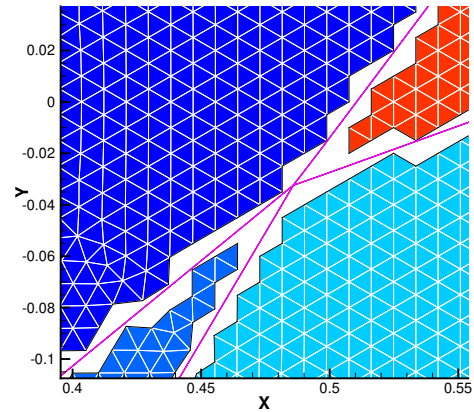(a) *SC* solution

(b) close-up around the QP

(c) Hybrid $eDIT_{GG}$: the CD is captured.

(d) close-up around the QP

(e) Hybrid $eDIT_{GG}$: the CD is tracked.

(f) close-up around the QP

Figure 18: Interaction between two shocks of the same family: numerical solutions (comparisons in terms of $S = p\rho^{-\gamma}$).
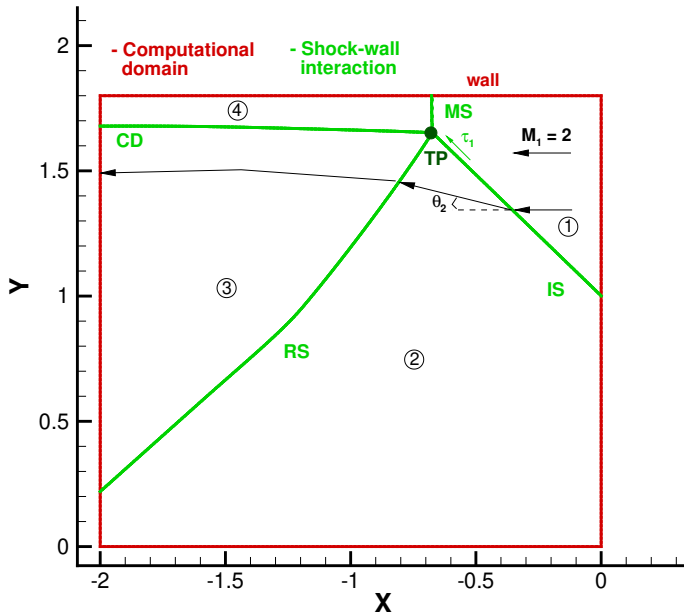
21

Figure 19: Mach reflection: sketch of the flow.

We compare in Fig. 18 three sets of results: *i*) a fully captured solution (top row of Fig. 18); *ii*) a hybrid solution in which CD is captured, and all the shocks (IS1, IS2 and RS1) are tracked (central row of Fig. 18); *iii*) a fully tracked result (bottom row of Fig. 18) where also CD is fitted. First of all, this result shows that it is possible to track several discontinuities and capture others, all involved in the same interaction. Compared to the fully captured solution, this hybrid simulation with captured CD provides a much cleaner entropy field, as seen comparing the top and central rows of frames in Fig. 18. However, we can also see a small anomaly due to the capturing of the CD, which is still observable in part of the flow downstream of the reflected shock RS1: see the close up of Fig 18d. These spurious disturbances are removed by adding the contact discontinuity to the set of discontinuities to be tracked by the algorithm, as visible on the bottom row of results of Fig. 18. Note that despite its apparent simplicity, the results obtained are surprisingly good as this is a difficult simulation. Indeed, on a mesh as coarse as the one used here, the elements crossed by the discontinuity generate a relatively large cavity around the interaction point. This is clearly visible for example in Fig 18f. The extrapolation strategy proposed allows to handle this delicate geometrical situation.

4.4. Shock-wall interaction: Mach reflection

An oblique shock that impinges on a straight wall gives rise to either a regular or a Mach reflection, depending on the combination of free-stream Mach number, M_1, and flow deflection angle, θ_2, that the free-stream flow undergoes while passing through the oblique shock (hereafter also referred to as the incident shock). Whenever θ_2 is larger than the maximum deflection that the supersonic stream behind the incident shock can sustain, a Mach reflection takes place. As sketched in Fig. 19, a Mach reflection consists in a fairly complex three-shocks system (the incident shock, IS, the reflected shock, RS, and the Mach stem, MS) interacting in the triple point, TP, from which a contact discontinuity, CD, also arises. Note that differently from the test-case addressed in § 4.3, here not all the regions surrounding TP involve uniform flow. Moreover, CD presents a slight angle, which

makes it in general not mesh-aligned.

For the chosen setting, $M_1 = 2$ and $\theta_2 = 14°$, we consider four different types of simulations:

1. fully captured;
2. an extrapolated shock-tracking setting in which the Mach stem (MS) and the reflected shock (RS) are tracked as a single discontinuity;
3. a hybrid setting in which only the shocks are tracked, but CD is captured;
4. full-fledged discontinuity-tracking for all shocks plus the CD.

Note that no modelling of the triple point TP is necessary in configuration 2 which involves a unique shock-mesh. In case 3 we need to provide an explicit model for the triple-point. Not having enough equations to compute the TP, because the CD is captured, rather than being tracked, we need to resort to a heuristic approach. As in the case of § 4.3, a nonlinear algebraic problem is solved independently for each discontinuity, and for the TP velocity we use the following simplified relation:

$$\boldsymbol{\omega}_{TP} = \boldsymbol{\omega}_{IS} + (\boldsymbol{\omega}_{MS} \cdot \boldsymbol{\tau_1}) \, \boldsymbol{\tau_1} \tag{21}$$

where $\boldsymbol{\tau_1}$ is the unit vector tangential to the IS (see Fig.19a).
Equation (21) is similar, but not identical, to the formula used in [19] for the same purpose.
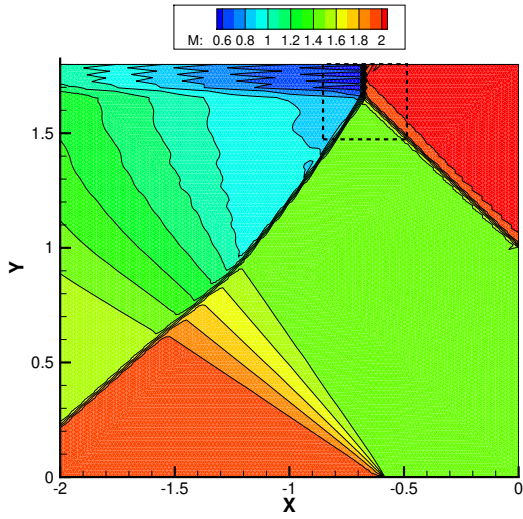
Finally, in configuration 4, all four states surrounding the TP and its unkown velocity, $\boldsymbol{\omega}_{TP}$, can be coupled into a single non-linear system of algebraic equations, whose solution provides updated values downstream of the RS and MS as well as $\boldsymbol{\omega}_{TP}$. Full details are given in [13].
The results are arranged in four rows in Figures 20 and 21, showing on the left an overview of the Mach contours, and on the right a zoom of the TP. Compared to the SC calculation (top row in Fig. 20), tracking MS and RS (second row in Fig. 20) already provides an enormous improvement in the quality of the solution. The CD remains however poorly captured on this coarse mesh. Adding the IS to the tracking set (third row in Fig. 20) leads to a cleaner flow, which is visible in the nicer straight contour lines downstream of RS. However, CD is still poorly resolved. Finally, in full-fledged *eDIT* mode of Fig. 21 we are able to resolve the CD in a single row of elements. A small kink in the contour lines in the non-constant region around CD is still visible. We assume these to be due to perturbations arising in correspondence of the steps in the surrogate CD and propagating downstream. A qualitative view of the error cleaning and pseudo-time convergence of the full-fledged simulation is displayed in Fig. 22.
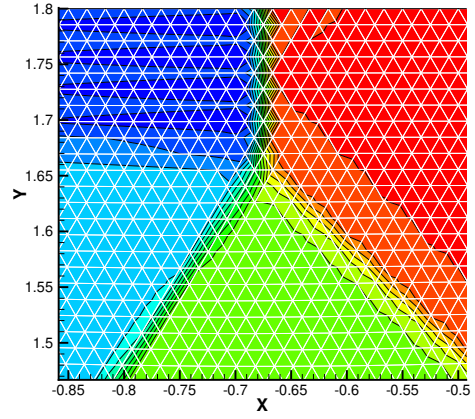
4.5. Supersonic channel flow

The last test-case considered consists in the supersonic, $M_\infty = 3.5$, flow in a planar channel, whose variable-area geometry is shown in Fig. 23a and reported in [18]: the two constant-area portions of the duct are joined through a double ramp. The flow pattern is as follows: two oblique, straight shocks of the same family, labeled IS1 and IS2, originate at the two convex corners of the ramp and their interaction gives rise to a wave configuration already explored in § 4.3 where IS1, IS2, a new shock RS1, a slip-stream CD and an expansion fan EF1 meet at the interaction point QP. A second, stronger expansion fan EF2 takes place at the concave corner of the lower wall and interacts with RS1, which bends before being reflected from the upper wall. The reflected shock RS2 is again reflected by the lower wall and leaves the duct as shock RS3.
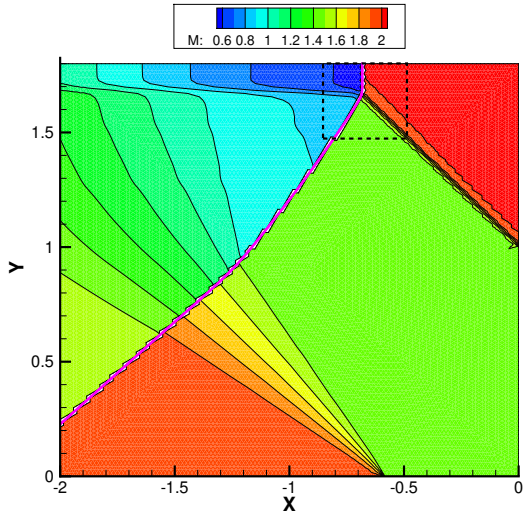
The goal of the present test-case consists in assessing that the different features that have been implemented within the *eDIT* algorithm, already described in the previous sections, work well also when combinedly used to simulate a fairly complex shock-pattern, such as the one illustrated in Fig. 23a. To improve simulation fidelity, we track as many discontinuities as we can, i.e. the three
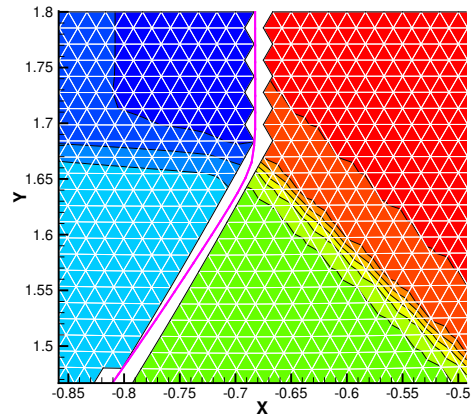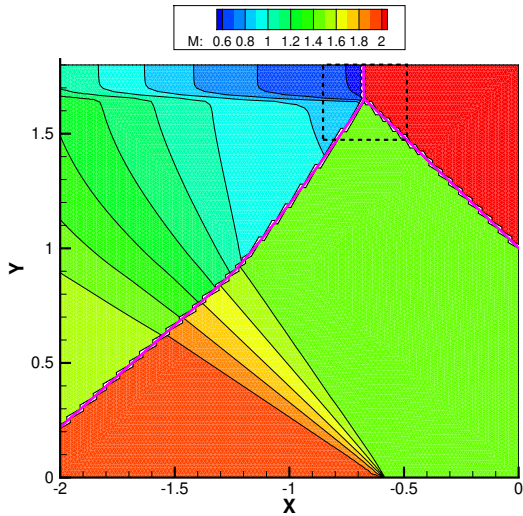
(a) *SC* solution
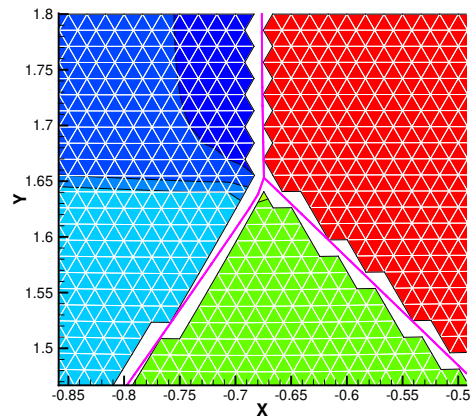
(b) close-up around the TP

(c) Hybrid $eDIT_{GG}$: configuration 2

(d) close-up around the TP

(e) Hybrid $eDIT_{GG}$: configuration 3

(f) close-up around the TP

Figure 20: Mach reflection: the Mach number iso-lines of the numerical solutions.
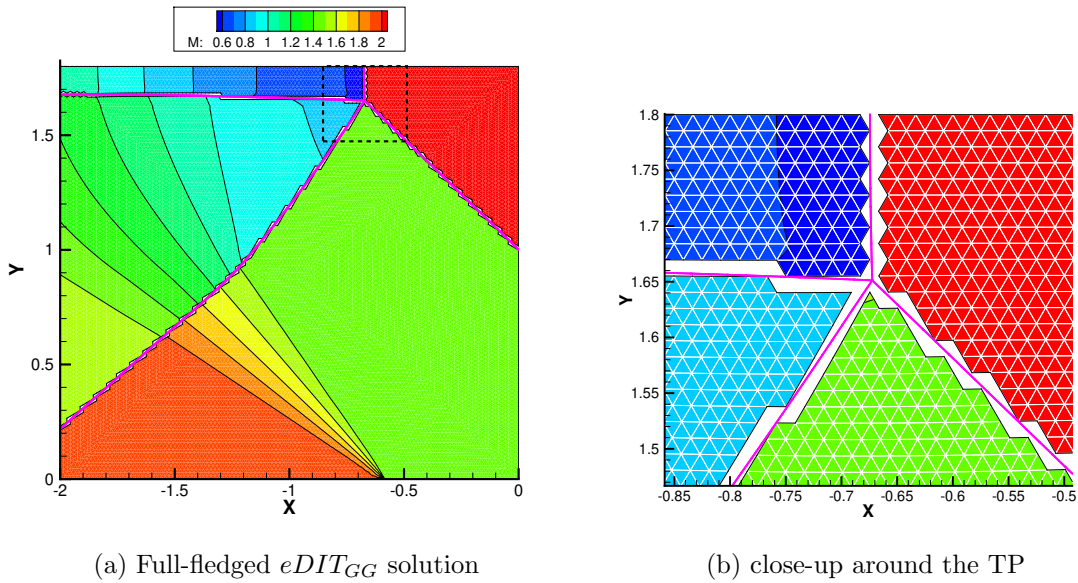
24

(a) Full-fledged $eDIT_{GG}$ solution　　　　　(b) close-up around the TP

Figure 21: Mach reflection: the Mach number iso-lines of the full-fledged (configuration 4) numerical solutions.

shock waves IS1, IS2 and RS1, which meet at QP, and the two regular reflections made up by RS1, RS2 and RS3 taking place on the upper and lower walls. Only the CD has been captured.

When simulating this testcase with the $eDIT$ algorithm, two different models have been employed: the one already described in § 4.3 for QP, where the interaction between shocks of the same family takes place, and a different one for points IP1 and IP2, where a regular reflection takes place on the upper, resp. lower wall. As shown in Fig. 24, the velocity of points IP1 and IP2 is set equal to the component tangential to the wall of the corresponding incident shock velocity.

Simulations have been performed using both SC and $eDIT_{GG}$. The unstructured grid used in the SC calculation and as the background triangulation in the $eDIT_{GG}$ calculation, which is made up of 12,417 grid-points and 24,310 nearly equilateral triangles, has been generated using the frontal mesh generator of the `gmsh` software [53]. A detail of the mesh is shown in Fig. 23b.

The comparison between the two sets of calculations is reported in Figs. 25 and 26, were density and Mach number iso-contours are respectively displayed. It can be seen that the capture of the discontinuities, see Figs. 25a and 26a, gives rise to spurious disturbances (in particular downstream of RS2) which are not present in the $eDIT_{GG}$ calculation (Figs. 25b and 26b) pointing out a globally cleaner flow-field featuring smoother iso-contours. The oscillations that occur downstream of RS2 are more clearly visible in Fig. 27, which shows the dimensionless pressure profiles along the upper and lower walls. It is evident that by resorting to a shock-tracking approach, not only we are able to get rid of the major oscillation introduced by SC and exactly represent the discontinuities as having zero-thickness, but we also get a better prediction of wall pressure peaks, which are smoothed out in the SC calculation.

5. Conclusion

A novel technique recently proposed in [1] to simulate flows with shock waves has been further improved to make it capable of dealing with different kinds of discontinuities (both shock waves and contact discontinuities) as well as shock-shock and shock-wall interactions, thus opening the possibility to compute complex flows. Moreover, since the algorithm can be run in hybrid mode, it is also possible to study complicated flows by tracking some of the discontinuities and leaving others to be captured. The proposed technique provides genuinely second-order-accurate results even for flows
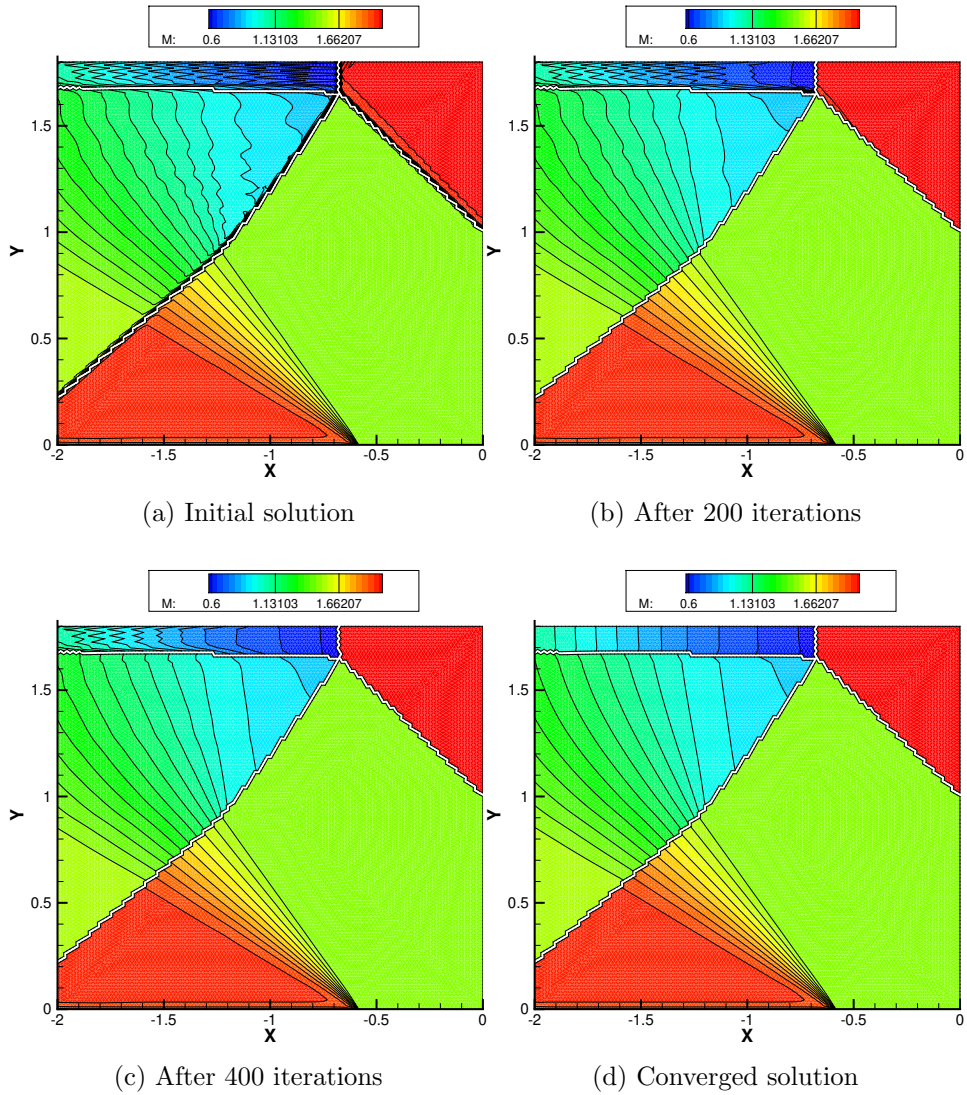
(a) Initial solution

(b) After 200 iterations

(c) After 400 iterations

(d) Converged solution

Figure 22: Mach reflection: pseudo-time convergence of the full-fledged $eDIT_{GG}$ simulation (Mach number iso-contour lines)

(a) Sketch of the computational domain and flow-pattern.

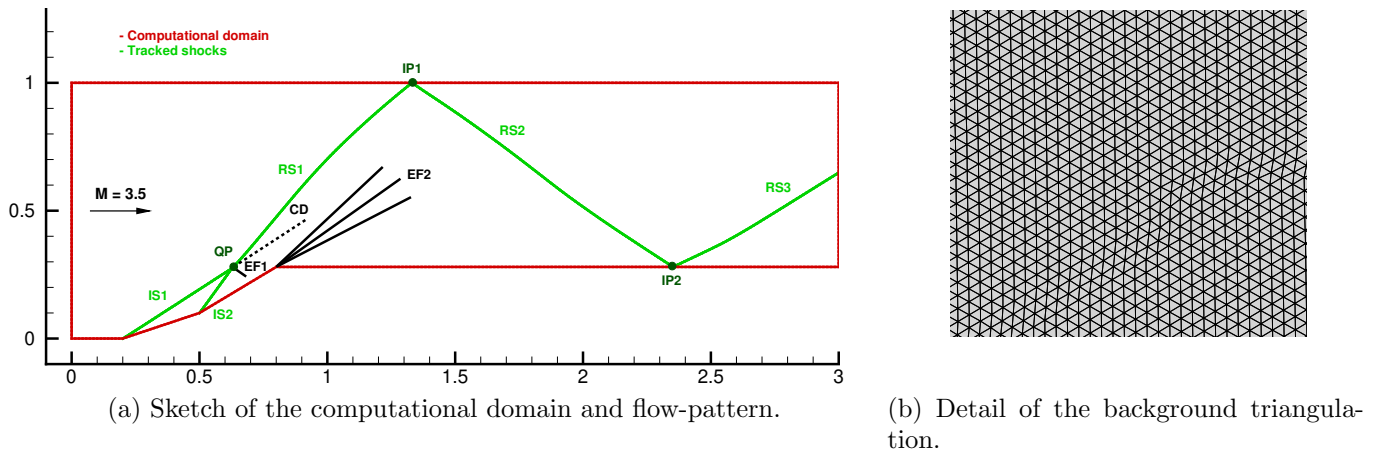(b) Detail of the background triangulation.
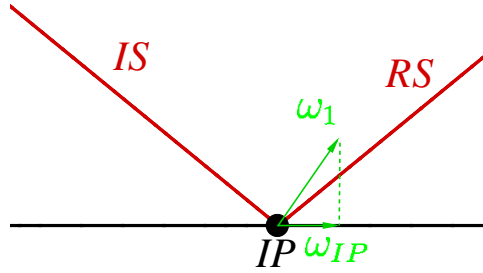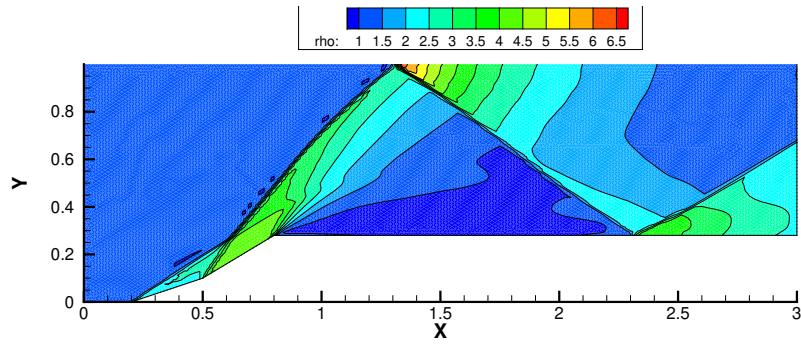
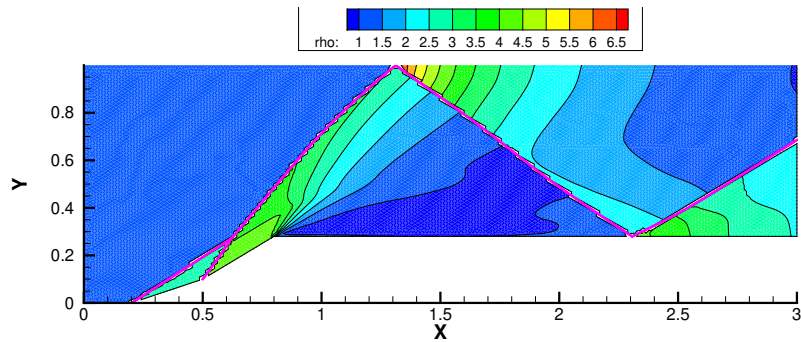Figure 23: Supersonic channel flow.

26

Figure 24: Supersonic channel flow: regular reflection at a solid wall and calculation of of the displacement velocity of the point, IP, where the IS and RS meet.
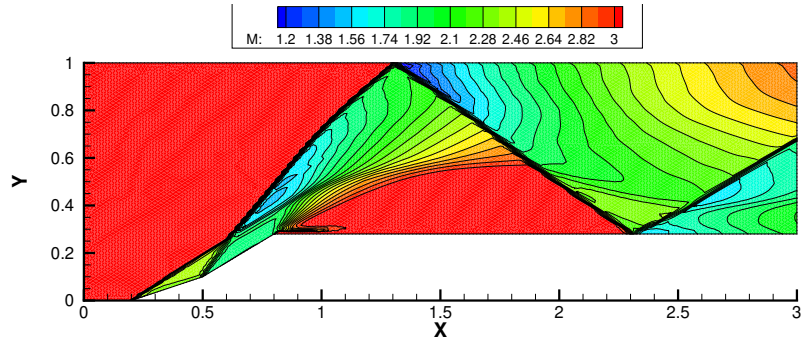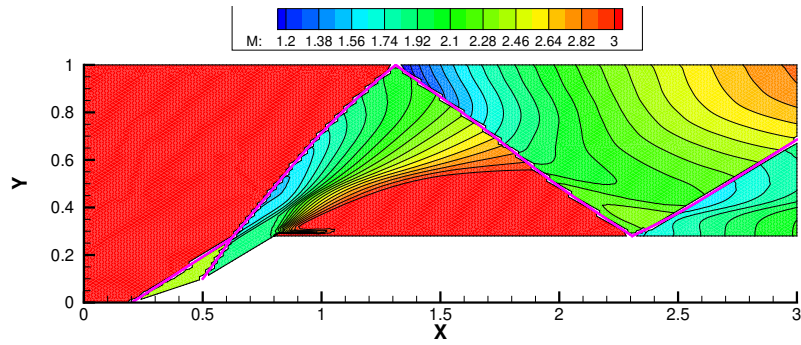

(a) SC solution

(b) $eDIT_{GG}$ solution

Figure 25: Supersonic channel flow: dimensionless density ρ/ρ_∞ iso-contours.
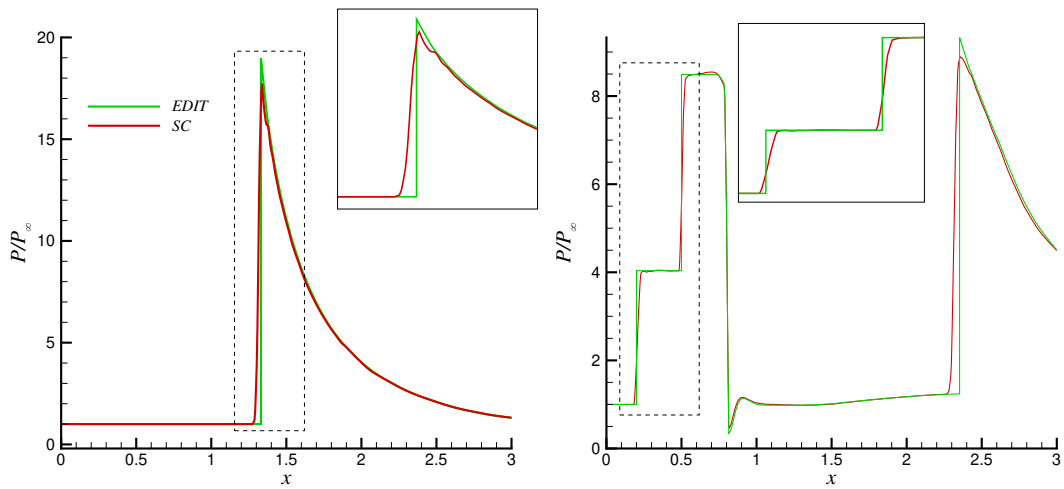
(a) SC solution

(b) $eDIT_{GG}$ solution

Figure 26: Supersonic channel flow: Mach iso-contours.

(a) Upper wall.

(b) Lower wall.

Figure 27: Supersonic channel flow: dimensionless pressure distribution along the walls of the channel computed by means of SC and $eDIT$.

featuring very strong shocks, without the complexity of the re-meshing/adaptation phase of previous fitting approaches. For this reason, the present algorithm is substantially independent from the data structure of the gasdynamic solver and, therefore, it can be applied with small modifications to both cell-centered and vertex-centered solvers on both unstructured and structured grids. Although many improvements of the fitting/tracking technique have been made in the present paper, further developments are in progress, including its coupling with a structured-grid, finite volume solver [38].

Appendix A. Nodal gradient reconstruction

As explained in § 3.1, in the present version of the $eDIT$ algorithm, the computational domain is tesselated into triangles and bi-linear shape functions with unknowns stored at the vertices of the triangulation that provide the functional representation of the dependent variable, which is Roe's parameter vector $\mathbf{Z} = \sqrt{\rho}\,(1, H, u, v)^t$. Using the aforementioned setting, the gradient of $\mathbf{Z}$ is constant within each triangular cell T and can be readily computed as follows:

$$\nabla \mathbf{Z}_T = \frac{\sum_{k=1}^{3}(\mathbf{Z}_k\,\mathbf{n}_k)}{2\,A_T} \tag{A.1}$$

where $\mathbf{n}_k$ denotes the normal to the face opposite vertex k, scaled by its length and pointing inside triangle T and A_T denotes the area of triangle T.

As explained in § 3.5.1, however, in order to perform the data trasfer between the surrogate boundaries and the discontinuity-front, it is also necessary to compute the gradient of $\mathbf{Z}$ within every grid-point of the surrogate boundaries; this is accomplished using either the Green-Gauss (GG) or Zienkiewicz-Zhu (ZZ) approaches to be detailed hereafter.

Green-Gauss reconstruction. The GG reconstruction consists in camputing the nodal gradient in grid-point i as the area-weighted average of the cell-wise constant gradients of all triangles that surround grid-point i:

$$\nabla \mathbf{Z}_i = \sum_T \theta_T \nabla \mathbf{Z}_T \tag{A.2}$$

where:

$$\theta_T = \frac{A_T}{\sum_T A_T} \quad \text{such that} \quad \sum_T \theta_T = 1 \tag{A.3}$$

The summations in Eqs. (A.2) and (A.3) range over all triangles surrounding the grid-point i.

Zienkiewicz-Zhu reconstruction. Similarly to the GG reconstruction, all steps involved in the ZZ gradient-reconstruction have to be repeated within all grid-points of the surrogate boundaries. In order to alleviate the notation, however, we shall hereafter drop the index that refers to the grid-point and use subscript k to refer to one of the four components of the parameter vector and subscript j to refer the cartesian coordinates, i.e. $(x_1, x_2) = (x, y)$. The starting point in the ZZ reconstruction consists in using a linear polynomial expansion of each component of the gradient, i.e.

$$\left(\frac{\partial Z_k}{\partial x_j}\right) = \mathbf{P} \cdot \mathbf{a}_{k,j} \tag{A.4}$$

where:

$$\mathbf{P} = [\,1\,,\,x_1\,,\,x_2\,] \quad \text{and} \quad \mathbf{a}_{k,j} = [\,a_0\,,\,a_1\,,\,a_2\,]_{k,j} \tag{A.5}$$

Following the Zienkiewicz-Zhu (ZZ) patch recovery procedure, the unknown parameters $\mathbf{a}_{k,j}$ are computed via a least-squares fit which amounts to minimizing the following test function:

$$F(\mathbf{a}_{k,j}) = \sum_T \theta_T \left[\left(\frac{\partial Z_k}{\partial x_j} \right)_T - \mathbf{P}(x_1^T, x_2^T) \cdot \mathbf{a}_{k,j} \right]^2 \tag{A.6}$$

The summation in Eq. (A.6) ranges over all triangles surrounding the given grid-point, the gradient in Eq. (A.6) is computed element-wise according to Eq. (A.1) and vector $\mathbf{P}$ is computed in the barycentric coordinates of triangle $T : \left(x_1^T, x_2^T \right)$. Given that three unknowns, $\mathbf{a}_{k,j}$, must be computed for each component of $\mathbf{Z}$ and each cartesian coordinate, a stencil of at least three triangles is required. The minimization problem defined by Eq. (A.6) can be solved in matrix form:

$$\mathbf{a}_{k,j} = \underline{\underline{A}}_{k,j}^{-1} \cdot \mathbf{b}_{k,j} \tag{A.7}$$

where:

$$\underline{\underline{A}}_{k,j} = \sum_T \theta_T \, \mathbf{P}^t(x_1^T, x_2^T) \, \mathbf{P}(x_1^T, x_2^T) \quad \text{and} \quad \mathbf{b}_{k,j} = \sum_T \theta_T \, \mathbf{P}^t(x_1^T, x_2^T) \left(\frac{\partial Z_k}{\partial x_j} \right)_T \tag{A.8}$$

The GG reconstruction, Eq. (A.2), can be recovered from the ZZ reconstruction (A.7) by choosing a constant polynomial expansion: $\mathbf{P} = [\, 1\,,\, 0\,,\, 0\,]$.

References

[1] M. Ciallella, M. Ricchiuto, R. Paciorri, A. Bonfiglioli, Extrapolated shock tracking: bridging shock-fitting and embedded boundary methods, Journal of Computational Physics (2020) 109440doi:https://doi.org/10.1016/j.jcp.2020.109440.

[2] J. VonNeumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, Journal of Applied Physics 21 (3) (1950) 232–237. doi:10.1063/1.1699639.

[3] P. D. Lax, Hyperbolic systems of conservation laws and the mathematical theory of shock waves, SIAM, 1973.

[4] H. W. Emmons, The numerical solution of compressible fluid flow problems, NACA-TN 932, NASA (1944).
URL https://apps.dtic.mil/dtic/tr/fulltext/u2/b814416.pdf

[5] H. W. Emmons, Flow of a compressible fluid past a symmetrical airfoil in a wind tunnel and in free air, NACA-TN 1746 (1948).
URL https://ntrs.nasa.gov/citations/19930082372

[6] R. Marsilio, G. Moretti, Shock-fitting method for two-dimensional inviscid, steady supersonic flows in ducts, Meccanica 24 (4) (1989) 216–222. doi:https://doi.org/10.1007/BF01556453.

[7] G. Moretti, Thirty-six years of shock fitting, Computers & Fluids 31 (4-7) (2002) 719–723. doi:10.1016/s0045-7930(01)00072-x.

[8] R. Paciorri, A. Bonfiglioli, A shock-fitting technique for 2D unstructured grids, Computers & Fluids 38 (3) (2009) 715–726. doi:https://doi.org/10.1016/j.compfluid.2008.07.007.

[9] 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:https://doi.org/10.1016/j.jcp.2011.01.018.

[10] A. Bonfiglioli, R. Paciorri, L. Campoli, Unsteady shock-fitting for unstructured grids, International Journal for Numerical Methods in Fluids 81 (4) (2016) 245–261. doi:https://doi.org/10.1002/fld.4183.

[11] L. Campoli, P. Quemar, A. Bonfiglioli, M. Ricchiuto, Shock-fitting and predictor-corrector explicit ale residual distribution, in: Shock Fitting, Springer, 2017, pp. 113–129.

[12] A. Bonfiglioli, M. Grottadaurea, R. Paciorri, F. Sabetta, An unstructured, three-dimensional, shock-fitting solver for hypersonic flows, Computers & Fluids 73 (2013) 162–174. doi:https://doi.org/10.1016/j.compfluid.2012.12.022.

[13] 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:https://doi.org/10.1007/s00193-010-0266-y.

[14] L. Campoli, A. Assonitis, M. Ciallella, R. Paciorri, A. Bonfiglioli, M. Ricchiuto, Undifi-2d: an unstructured discontinuity fitting code for 2d grids, arXiv preprint arXiv:2105.14269.

[15] 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.

[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:https://doi.org/10.1002/fld.4697.

[17] M. J. Zahr, A. Shi, P.-O. Persson, Implicit shock tracking using an optimization-based high-order discontinuous galerkin method, Journal of Computational Physics 410 (2020) 109385. doi:https://doi.org/10.1016/j.jcp.2020.109385.

[18] D. Zou, A. Bonfiglioli, R. Paciorri, J. Liu, An embedded shock-fitting technique on unstructured dynamic grids, Computers & Fluids 218 (2021) 104847. doi:https://doi.org/10.1016/j.compfluid.2021.104847.

[19] S. Chang, X. Bai, D. Zou, Z. Chen, J. Liu, An adaptive discontinuity fitting technique on unstructured dynamic grids, Shock Waves (2019) 1–13doi:https://doi.org/10.1007/s00193-019-00913-3.

[20] X. Chen, S. Fu, Convergence acceleration for high-order shock-fitting methods in hypersonic flow applications with efficient implicit time-stepping schemes, Computers & Fluids 210 (2020) 104668. doi:https://doi.org/10.1016/j.compfluid.2020.104668.

[21] T. Huang, M. J. Zahr, A robust, high-order implicit shock tracking method for simulation of complex, high-speed flows, arXiv preprint arXiv:2105.00139.

[22] M. Luke, B. T. Helenbrook, A. Mazaheri, A novel stabilization method for high-order shock fitting with finite element methods, Journal of Computational Physics 430 (2021) 110096. doi:https://doi.org/10.1016/j.jcp.2020.110096.

[23] M. J. Zahr, J. M. Powers, High-order resolution of multidimensional compressible reactive flow using implicit shock tracking, AIAA Journal 59 (1) (2021) 150–164. `doi:https://doi.org/10.2514/1.J059655`.

[24] C. S. Peskin, Flow patterns around heart valves: a numerical method, Journal of computational physics 10 (2) (1972) 252–271. `doi:https://doi.org/10.1016/0021-9991(72)90065-4`.

[25] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, Computers & structures 81 (8-11) (2003) 491–501. `doi:https://doi.org/10.1016/S0045-7949(02)00404-2`.

[26] R. Abgrall, H. Beaugendre, C. Dobrzynski, An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques, Journal of Computational Physics 257 (2014) 83–101. `doi:https://doi.org/10.1016/j.jcp.2013.08.052`.

[27] L. Nouveau, H. Beaugendre, C. Dobrzynski, R. Abgrall, M. Ricchiuto, An adaptive, residual based, splitting approach for the penalized Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 303 (2016) 208–230. `doi:https://doi.org/10.1016/j.cma.2016.01.009`.

[28] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Computer methods in applied mechanics and engineering 191 (47-48) (2002) 5537–5552. `doi:https://doi.org/10.1016/S0045-7825(02)00524-8`.

[29] E. Burman, Ghost penalty, Comptes Rendus Mathematique 348 (21-22) (2010) 1217–1220. `doi:https://doi.org/10.1016/j.crma.2010.10.006`.

[30] E. Burman, P. Hansbo, Fictitious domain methods using cut elements: iii. a stabilized Nitsche method for Stokes' problem, ESAIM: Mathematical Modelling and Numerical Analysis 48 (3) (2014) 859–874. `doi:10.1051/m2an/2013123`.

[31] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of computational physics 152 (2) (1999) 457–492. `doi:https://doi.org/10.1006/jcph.1999.6236`.

[32] C. Farhat, A. Rallu, S. Shankaran, A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions, Journal of Computational Physics 227 (16) (2008) 7674–7700. `doi:https://doi.org/10.1016/j.jcp.2008.04.032`.

[33] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. part I: Poisson and Stokes problems, Journal of Computational Physics 372 (2018) 972–995. `doi:https://doi.org/10.1016/j.jcp.2017.10.026`.

[34] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. part II: Linear advection–diffusion and incompressible Navier–Stokes equations, Journal of Computational Physics 372 (2018) 996–1026. `doi:https://doi.org/10.1016/j.jcp.2018.01.023`.

[35] D. She, R. Kaufman, H. Lim, J. Melvin, A. Hsu, J. Glimm, Front-tracking methods, in: Handbook of Numerical Analysis, Vol. 17, Elsevier, 2016, pp. 383–402. `doi:https://doi.org/10.1016/bs.hna.2016.07.004`.

[36] R. Pepe, A. Bonfiglioli, A. D'Angola, G. Colonna, R. Paciorri, An unstructured shock-fitting solver for hypersonic plasma flows in chemical non-equilibrium, Computer Physics Communications 196 (2015) 179–193. doi:https://doi.org/10.1016/j.cpc.2015.06.0.

[37] T. Song, A. Main, G. Scovazzi, M. Ricchiuto, The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows, Journal of Computational Physics 369 (2018) 45–79. doi:https://doi.org/10.1016/j.jcp.2018.04.052.

[38] A. Assonitis, R. Paciorri, M. Ciallella, M. Ricchiuto, A. Bonfiglioli, A new shock-fitting approach for 2-d structured grids, in: Submitted to the 2022 AIAA SciTech Forum.

[39] M. D. Salas, A shock-fitting primer, CRC Press, 2009.

[40] A. Assonitis, R. Paciorri, A. Bonfiglioli, Numerical simulation of shock/boundary-layer interaction using an unstructured shock-fitting technique, Computers & Fluids (2021) 105058doi:https://doi.org/10.1016/j.compfluid.2021.105058.

[41] R. Paciorri, A. Bonfiglioli, Accurate detection of shock waves and shock interactions in two-dimensional shock-capturing solutionis, Journal of Computational Physics 406 (2020) 109196. doi:https://doi.org/10.1016/j.jcp.2019.109196.

[42] A. Beck, J. Zeifang, A. Schwarz, D. Flad, A neural network based shock detection and localization approach for discontinuous galerkin method, Journal of Computational Physics 423 (2020) 109824. doi:https://doi.org/10.1016/j.jcp.2020.109824.

[43] A. Bonfiglioli, Fluctuation splitting schemes for the compressible and incompressible Euler and Navier-Stokes equations, International Journal of Computational Fluid Dynamics 14 (1) (2000) 21–39. doi:https://doi.org/10.1080/10618560008940713.

[44] H. Deconinck, M. Ricchiuto, Residual Distribution Schemes: Foundations and Analysis, John Wiley & Sons, Ltd, 2017, pp. 1–53.

[45] R. Abgrall, M. Ricchiuto, High-Order Methods for CFD, John Wiley & Sons, Ltd, 2017, pp. 1–54.

[46] A. Bonfiglioli, R. Paciorri, A mass-matrix formulation of unsteady fluctuation splitting schemes consistent with Roe's parameter vector, International Journal of Computational Fluid Dynamics 27 (4-5) (2013) 210–227. doi:https://doi.org/10.1080/10618562.2013.813491.

[47] P. L. Roe, Approximate riemann solvers, parameter vectors, and difference schemes, Journal of computational physics 43 (2) (1981) 357–372. doi:https://doi.org/10.1016/0021-9991(81)90128-5.

[48] H. Deconinck, P. Roe, R. Struijs, A multidimensional generalization of Roe's difference splitter for the Euler equations, Computers and Fluids 22 (2/3) (1993) 215–222. doi:https://doi.org/10.1016/0045-7930(93)90053-C.

[49] G. Giorgiani, H. Guillard, B. Nkonga, E. Serre, A stabilized powell–sabin finite-element method for the 2d euler equations in supersonic regime, Computer Methods in Applied Mechanics and Engineering (2018) 216–235doi:https://doi.org/10.1016/j.cma.2018.05.032.

[50] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique, International Journal for Numerical Methods in Engineering 33 (7) (1992) 1331–1364. `doi:https://doi.org/10.1002/nme.1620330702`.

[51] A. Bonfiglioli, R. Paciorri, Convergence analysis of shock-capturing and shock-fitting solutions on unstructured grids, AIAA journal 52 (7) (2014) 1404–1416. `doi:https://doi.org/10.2514/1.J052567`.

[52] M. S. Campobasso, M. H. Baba-Ahmadi, Ad-Hoc Boundary Conditions for CFD Analyses of Turbomachinery Problems With Strong Flow Gradients at Farfield Boundaries, Journal of Turbomachinery 133 (4) (2011) 041010–041010–9. `doi:https://doi.org/10.1115/1.4002985`.

[53] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities, International journal for numerical methods in engineering 79 (11) (2009) 1309–1331. `doi:https://doi.org/10.1002/nme.2579`.

[54] J.-D. Müller, P. L. Roe, H. Deconinck, A frontal approach for internal node generation in delaunay triangulations, International Journal for Numerical Methods in Fluids 17 (3) (1993) 241–255. `doi:https://doi.org/10.1002/fld.1650170305`.

[55] J. Müller, Delaundo mesh generator, Available at
`http://www.ae.metu.edu.tr/tuncer/ae546/prj/delaundo/`.