

# AN UNSTRUCTURED SHOCK-FITTING TECHNIQUE FOR THREE-DIMENSIONAL FLOWS WITH SHOCK INTERACTIONS

C. OLLIVIER-GOOCH<sup>1</sup>, R. PACIORRI<sup>2</sup>, A. ASSONITIS<sup>3</sup> AND  
A. BONFIGLIOLI<sup>4</sup>

<sup>1</sup> The University of British Columbia  
6250 Applied Science Lane, Vancouver, BC V6T 1Z4 Canada  
cfog@mech.ubc.ca

<sup>2</sup> Università degli Studi di Roma "La Sapienza"  
Via Eudossiana 18, 00184, Rome, Italy  
renato.paciorri@uniroma1.it

<sup>3</sup> Università degli Studi di Roma "La Sapienza"  
Via Eudossiana 18, 00184, Rome, Italy  
alessia.assonitis@uniroma1.it

<sup>4</sup> Università degli Studi della Basilicata  
Viale dell'Ateneo Lucano 10, 85100 Potenza, Italy  
aldo.bonfiglioli@unibas.it

**Key words:** Compressible flows, shock-waves, Shock-Fitting

**Abstract.** The numerical simulation of hypersonic flows past blunt bodies by means of shock-capturing (S-C) solvers is characterized by some critical challenges, including: stagnation point anomalies, spurious numerical oscillations, the carbuncle phenomenon and the reduction of the order of accuracy of the solution in the entire region downstream of a captured shock worsen the solution quality. This paper describes an updated version of the unstructured shock-fitting (S-F) algorithm for three-dimensional flows. In particular, we present a comparison between the results obtained computing hypersonic flows on blunt bodies using both the S-C and S-F techniques on nearly identical tetrahedral meshes, with a special interest on the grid-convergence properties of the two different shock-modeling options.

## 1 INTRODUCTION

More than 10 years ago a new shock-fitting technique for unstructured meshes was proposed by Paciorri and Bonfiglioli [1]. The technique has been further developed and successfully applied to the simulation of three-dimensional hypersonic flows a few years later [2]. The present paper describes an updated version of the unstructured shock-fitting algorithm for three-dimensional flows: in the technique described in [2], volumetric and surface mesh generation were handled using general-purpose codes, which were not specifically tailored on the requirements of the unstructured shock-fitting algorithm and, therefore, posed strong limitations to the range of

applications that could be handled.

Some years ago, Zaide and Ollivier-Gooch [3] developed software tools based on the GRUMMP library for inserting lines/surfaces into existing unstructured 2D/3D meshes. Their approach provides the core of the mesh generation and handling also required for the unstructured shock-fitting technique, including stitching together multiple shock surfaces at reflection / interaction lines, improving shock surface length scale, and moving shock surfaces through the mesh to reach equilibrium. In particular, the role of the GRUMMP library and its features will be described in detail in the following sections, where different test-cases, computed using both the capturing and fitting shock-modeling options, will be also presented.

## 2 SHOCK-FITTING ALGORITHM

In this section, we provide a general overview of the proposed S-F technique: in particular, it represents the fitted shocks as true discontinuities, since the shock front is modeled using a zero-thickness, double-sided, triangulated surface. An initial guess of the shock front can be determined by evaluating a preliminary S-C calculation, which also provide the initial condition for the S-F computation: indeed, a cloud of points can be identified in order to approximate the location of the discontinuity, and it is meshed in order to obtain a triangulated surface, which is inserted in the background grid, as shown in fig. 1.

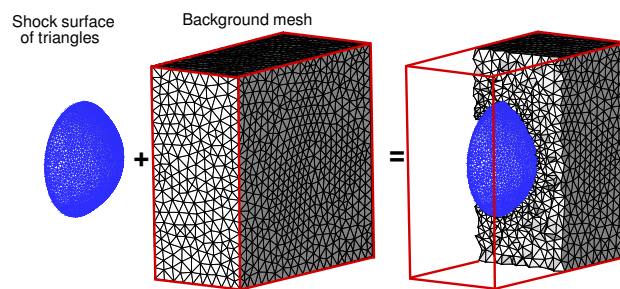


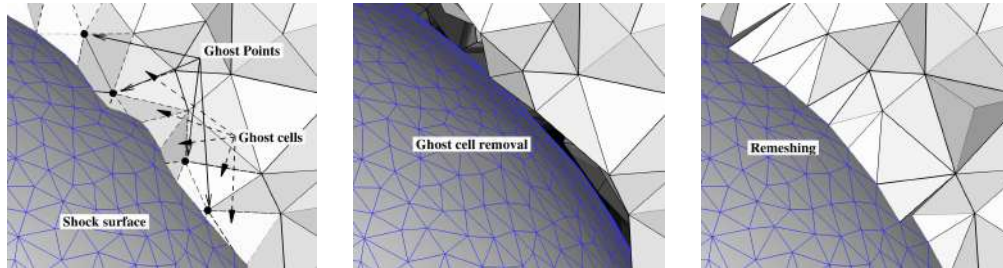
Figure 1: Example of the merging between the shock surface and the background mesh

Then, the fitted shock front acts as an interior boundary for a shock-capturing solver, which is used to solve the governing equations within the computational domain: in particular, the gas-dynamic solver used in the present implementation is EulFS [8], an in-house code based on the Fluctuation Splitting approach which implements both first and second-order-accurate schemes. During time-integration, the shock surface moves according to the Rankine–Hugoniot (R-H) jump relations, causing modifications of the tetrahedral grid, until it reaches its steady state position, corresponding to vanishing shock speed. This technique was applied for the first time on 3D unstructured grids by the authors [2]: in this preliminary version of the algorithm, the mesh generation and handling relied on general propose codes (Tetgen [4] and Yams [7]). Nevertheless, since these mesh generation codes were not specifically designed for unstructured shock-fitting and they posed strong limitations, in the present implementation these programs are replaced by codes based on the GRUMMP library. The main steps of the S-F algorithm

proposed in this paper are briefly described in the following subsections:

## 2.1 Insert the shock surface into the background mesh

The insertion of the shock surface into the background grid[3] first re-discretizes the surface, then removes vertices from the volume mesh that are too near the surface, and finally inserts the new vertices into the volume mesh and recovers the shock surface as a union of triangles in the mesh. The input triangulated shock surfaces are converted to interpolated spline surfaces using CGM [5]. The surface is re-discretized by sampling each smooth surface [6], starting with the surface boundary, then sampling the interior of the surface to ensure topological correctness, geometric fidelity, and mesh quality. The length scale for the sampled surface is determined from the existing volume mesh. Before inserting the surfaces into the volume mesh, we first create a cavity in the volume mesh around the surface. To prevent the creation of short edges, we tag for removal all vertices in the volume mesh that are closer than half of the local length scale to any points on the re-discretized surface. To ensure that we can recover all triangles in the discretized surface after insertion, we also tag for removal any volume mesh points that fall inside a sphere whose equator is the circumcircle of a surface triangle. These are shown in Fig. 2a. We remove these points from the volume mesh, tagging them as phantom points for possible re-addition as the shock moves, as shown in Fig. 2b. Finally, we insert the surface points, recover the surface triangles, and duplicate the shock surface to produce upstream and downstream shock points (Fig. 2c).



(a) Identification of the phantom nodes and ghost cells of the background mesh  
 (b) Crossed elements are removed  
 (c) Generation of the modified mesh

Figure 2: Inserting a shock surface in the mesh

## 2.2 Computation of the tangent and normal unit vectors

The normal unit vectors ( $\mathbf{n}$ ) on the shock surface have to be calculated at each shock point since this is required to compute the jump relations, as described in Sec. 2.3. The computation of these vectors in a generic shock point can be carried out by averaging the face normals of all shock faces that share the given shock point. Similarly to the two-dimensional case [1], however, the averaging process has to take into account only the shock faces that belong to the range of influence of a shock point, see Ref. [2] for further details. By convention, the normal unit vector is oriented from the downstream region to the upstream one.

### 2.3 Solution update on the shock nodes using the capturing code and enforcement of Rankine-Hugoniot jump relations

Using the computational grid as input, a single time-step calculation is performed using the unstructured, vertex-centered, shock-capturing solver EulFS [8], which returns updated nodal values at time  $t + \Delta t$  within all grid-points of the computational mesh generated in Sec. 2.1. As explained in details elsewhere [2], the shock-downstream values of the dependent variables within the shock-points need to be corrected by enforcing the jump relations across each pair of shock-points. This amounts to solving the system of six (in the 3D space) non-linear algebraic equations reported below at each pair of shock-points:

$$\rho_d^{t+\Delta t}(u_{dn} - w) = \rho_u^{t+\Delta t}(u_{un} - w) \quad (1)$$

$$\rho_d^{t+\Delta t}(u_{dn} - w)^2 + p_d^{t+\Delta t} = \rho_u^{t+\Delta t}(u_{un} - w)^2 + p_u^{t+\Delta t} \quad (2)$$

$$\frac{\gamma}{\gamma - 1} \frac{p_d^{t+\Delta t}}{\rho_d^{t+\Delta t}} + \frac{(u_{dn} - w)^2}{2} = \frac{\gamma}{\gamma - 1} \frac{p_u^{t+\Delta t}}{\rho_u^{t+\Delta t}} + \frac{(u_{un} - w)^2}{2} \quad (3)$$

$$\mathbf{u}_d^{t+\Delta t} - u_{dn}\mathbf{n} = \mathbf{u}_u^{t+\Delta t} - u_{un}\mathbf{n} \quad (4)$$

$$\sqrt{\gamma \cdot \frac{p_d^{t+\Delta t}}{\rho_d^{t+\Delta t}} + \frac{\gamma - 1}{2} u_{dn}^2} = R_d^{t+\Delta t} \quad (5)$$

The Eqs. 1-5 are the Rankine–Hugoniot (R-H) jump relations. The last equation is referred to the Riemann variable which is associated with the acoustic wave which moves upstream towards the shock

$$R_d^{t+\Delta t} = \tilde{a}_d^{t+\Delta t} + \tilde{\mathbf{u}}_d^{t+\Delta t} \cdot \mathbf{n} \quad (6)$$

where  $\mathbf{n}$  is the unit vector normal to the shock surface already computed in sec. 2.2,  $\tilde{a}_d^{t+\Delta t}$  and  $\tilde{\mathbf{u}}_d^{t+\Delta t}$  are respectively the values of sound speed and flow velocity computed by the unstructured solver in the downstream state of the shock nodes. Due to the upwind nature of the discretization used in the unstructured capturing code, the acoustic signal related to  $R_d^{t+\Delta t}$  is correctly evolved, even though the individual quantities  $\tilde{a}_d^{t+\Delta t}$  and  $\tilde{\mathbf{u}}_d^{t+\Delta t}$  may be wrong. The left hand side of Eq. 6 is therefore assumed to be correct, whereas the quantities on its right hand side are only provisional values (hence the tilde) that will be correctly updated enforcing the R-H relations. The aforementioned system is solved using a Newton–Raphson algorithm and supplies also the local shock speed  $w$ .

### 2.4 Shock displacement and interpolation of the phantom nodes

The position at time-level  $t+\Delta t$  is computed moving all shock points according to the shock speed ( $w_{sh}^{t+\Delta t}$ ) obtained in Sect. 2.3. If  $\Delta t$  is kept sufficiently small, the shock surface will overtake only phantom nodes but none of the active point of the computational mesh. All phantom vertices are checked to determine if they are far enough from the shock that they should be re-inserted into the active mesh. Those that will be re-activated must have their solution state updated. This is done by identifying the volume cell they fall within. The solution at all active vertices in the mesh has already been updated to  $t + \Delta t$ , and this solution is linearly interpolated to the phantom vertices. These phantoms are then re-inserted into the mesh, taking care to preserve triangles on

the shock. Non-phantom vertices near the shock are checked to see if they are too close to the shock. Those that are too close are removed from the mesh topology and tagged as phantoms.

### 3 NUMERICAL SIMULATION OF HIGH-SPEED FLOWS PAST BLUNT BODIES

In the following sections we will analyze and discuss the results obtained in the simulation of an inviscid high-speed flow past two blunt bodies using both S-C and S-F in order to highlight the differences between these two shock-modeling options. In particular, we performed the numerical solutions by means of the in-house second-order-accurate S-C solver described in Ref. [8], which is the same solver used in the smooth regions by the S-F technique. Computed results for the first test-case (Sec. 3.1) will be analyzed using the analytically computed values at stagnation point and a quantitative assessment of the order-of-convergence of the two techniques will also be made.

#### 3.1 Hemisphere

The hypersonic flow ( $M_\infty = 10$ ) past an hemisphere has been numerically computed using the computational domain shown in Fig. 3: the hemisphere has been carved within an orthogonal parallelepiped, half of which is shown in Fig. 3;  $R$  is radius of the hemisphere and the triangulated surface of the bow-shock is shown in blue.

S-C and S-F calculations have been performed on nearly identical tetrahedral grids: the S-C mesh coincides with the background mesh used in the S-F calculation; the modified S-F mesh differs from the background tessellation only in the neighborhood of the fitted shock-surface (compare Figs. 4a and 4b) where it has been locally modified by the GRUMMP library to give room to the triangulated shock surface. In order to evaluate the grid-convergence properties of the two different shock-modeling options, calculations have been performed using two nested grid levels, the finest one obtained by splitting each cell of the coarser mesh into eight tetrahedra.

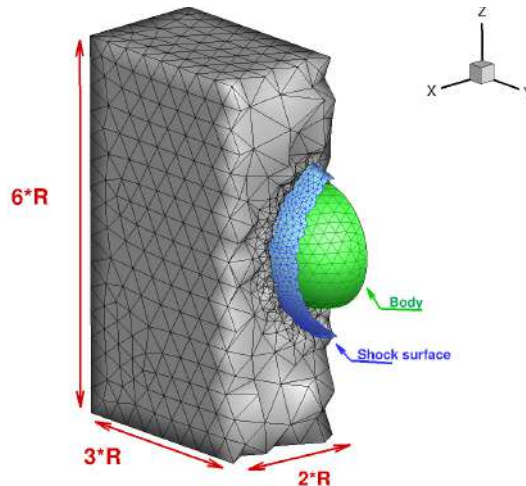


Figure 3: Computational domain

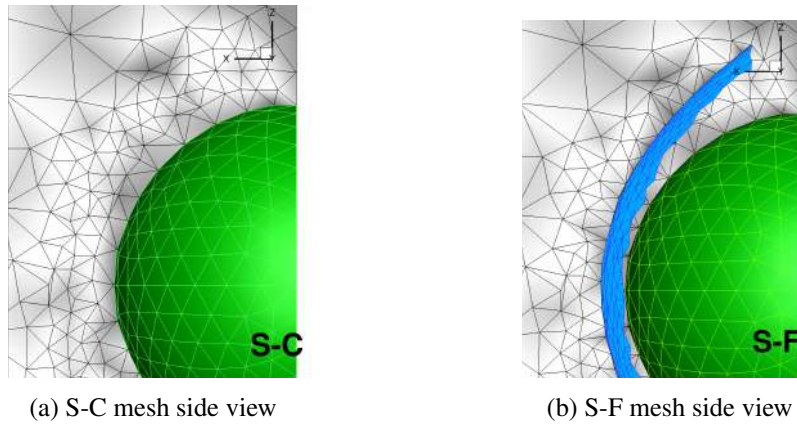


Figure 4: Hypersonic flow past a hemisphere: grid detail (plane XZ view)

The superior performance of the S-F technique is revealed by Fig. 5, which compares the dimensionless pressure field within the XZ plane computed using both S-C and S-F on the coarsest grid. The S-C calculation, Fig. 5a, features a shock-thickness which is comparable to the shock stand-off distance and is characterized by the presence of spurious disturbances within the entire shock-downstream region, due to the mis-alignment between the shock-surface and the triangular faces of the grid. On the contrary, the S-F calculation, Fig. 5b, features a discontinuity of zero thickness and a smooth pressure-field downstream of the bow shock.

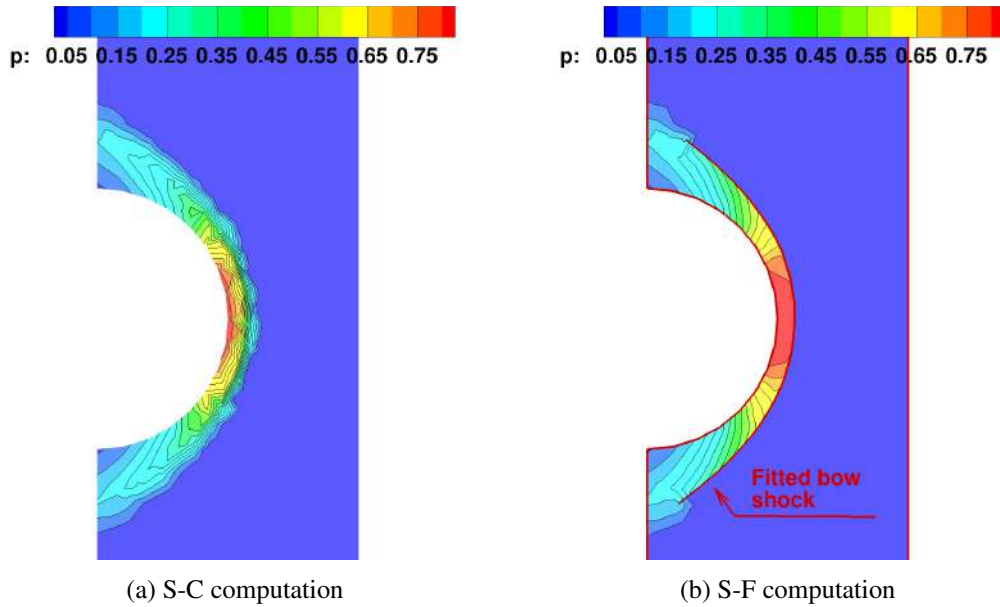


Figure 5: Hypersonic flow past a hemisphere: dimensionless pressure field in the XZ plane.)

The improved resolution offered by S-F is further confirmed by the dimensionless density distribution over the body surface, which is shown in Figs. 6a and 6b for the coarse, resp. fine grid levels. Indeed, in the S-C calculations shown in the two frames on the left of Fig. 6, the density iso-contour lines are seen to be plagued by severe oscillations that pollute the solution, in particular close to the stagnation point. In contrast, the S-F calculation reveals a reasonably good circumferential symmetry (particularly on the finer mesh) despite the use of a fully unstructured tetrahedralization.

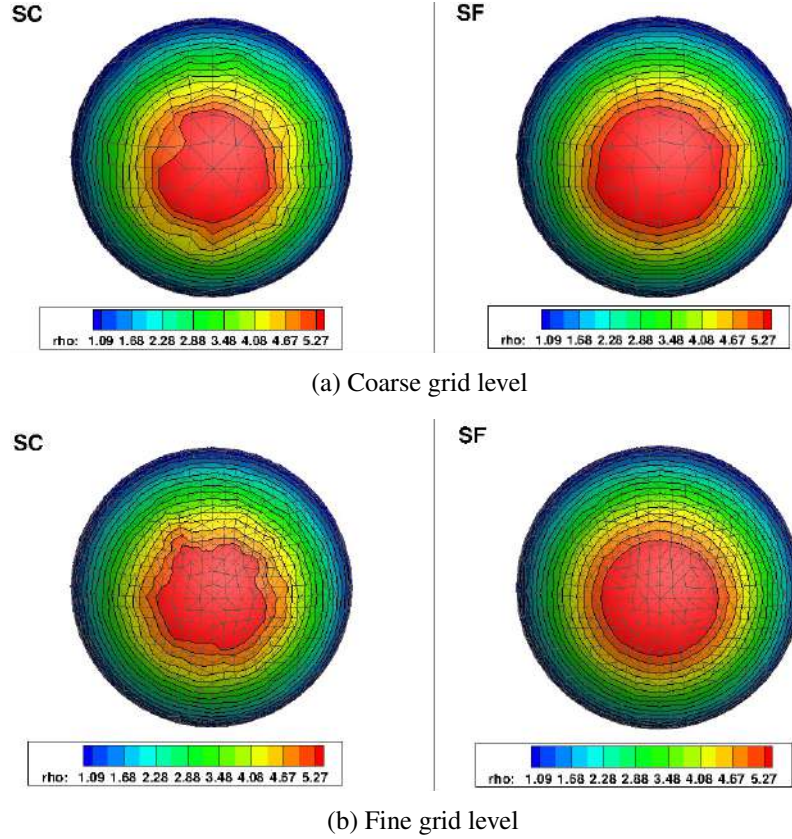


Figure 6: Hypersonic flow past a hemisphere: dimensionless density field over the body surface.

Nested meshes also allow to assess the observed order of accuracy,  $\tilde{n}$ , and discretization error,  $\epsilon_h$ , i.e. the difference between the numerical solution computed on a mesh of spacing  $h$  and the exact solution,  $u_0$ . Following [9], the observed order of accuracy has been calculated using:

$$\tilde{n} = \frac{\log R^{-1}}{\log r} \quad (7)$$

where the grid refinement ratio is equal to  $r = 2$  for nested meshes and the grid convergence ratio  $R$  is:

$$R = \frac{u_2 - u_0}{u_1 - u_0} = \frac{\epsilon_h}{\epsilon_{2h}} \quad (8)$$

In Eq. (8) the subscripts 1 and 2 refer to the coarse, resp. fine grid levels, whereas  $u_0$  is the exact stagnation point value of either pressure or density. Values at stagnation point can be analytically determined because the bow shock is a normal shock at the point where it is crossed by the stagnation streamline. Table 1, which shows the numerically and analytically computed values of pressure and density at stagnation point clearly reveals that S-F is much closer than S-C to the exact value.

Table 1: Inviscid flow past a hemisphere: stagnation point data.

	Shock capturing	Shock fitting	Analytical
Quantity	fine grid	fine grid	
$P/P_\infty$	125.64	128.93	129.21
$\rho/\rho_\infty$	5.80	6.13	6.15

The results of the grid-convergence analysis are reported in Tab. 2: on both grid levels the discretization error incurred by S-F is one order of magnitude smaller than that of S-C. This observation holds true for both pressure and density. As far as the observed order of convergence is concerned, however, the behavior is different between pressure and density. When looking at the pressure error, both S-C and S-F exhibit an observed order which is close to second, i.e. design order. When moving from the coarse to the fine mesh, the discretization error is reduced by a factor  $R^{-1}$  nearly equal to four, as expected. This is not any longer true when looking at the density error, in which case both shock-modeling options appear to be far from the asymptotic range of convergence, with S-F doing a little bit better (1.20 vs. 0.97) than S-C. Additional calculations using finer grid levels will be necessary to obtain more reliable grid-convergence trends.

Table 2: Inviscid flow past an hemisphere: convergence analysis.

	Pressure analysis				Density analysis			
	$\epsilon_{2h}$	$\epsilon_h$	$R^{-1}$	$\tilde{n}$	$\epsilon_{2h}$	$\epsilon_h$	$R^{-1}$	$\tilde{n}$
S-C	11.76	3.27	3.59	1.85	0.698	0.357	1.96	0.97
S-F	1.07	0.29	3.73	1.90	0.062	0.027	2.28	1.20



### 3.2 Cylinder flare body

In this section we consider the supersonic flow past a blunt-nosed body. The geometry consists of a cylinder with an hemispherical nose, a conical flare making an angle of  $30^\circ$  w.r.t. the body axis and a cylindrical tail. The free-stream Mach number is  $M_\infty = 4.04$  and the angle of attack equal to  $20^\circ$ , as shown in Fig. 7c. The flow-field is characterized not only by a bow shock, but also by an embedded shock which originates at the cylinder-cone junction; these two shocks interact giving rise to a type VI shock-shock interaction, as classified by Edney [10]. As far as the computational

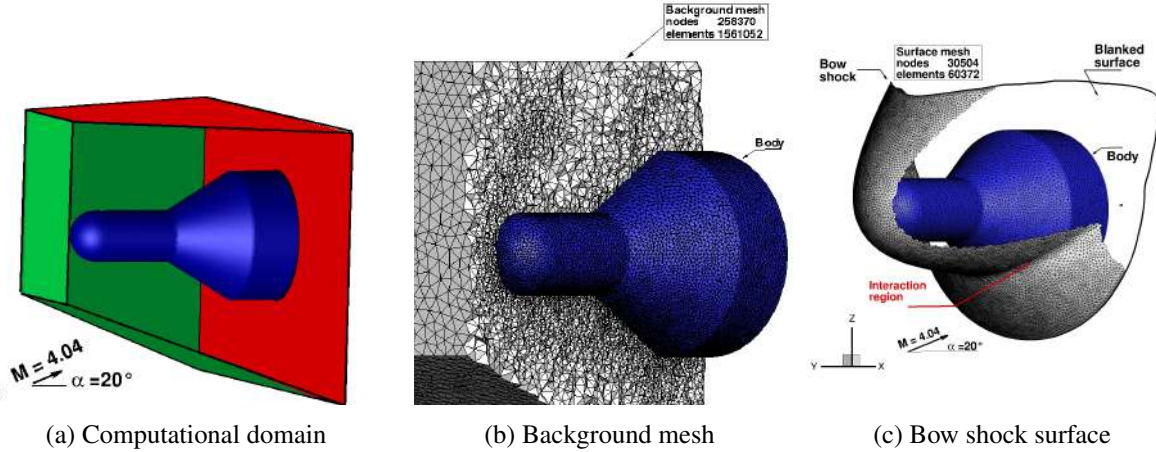


Figure 7: Hypersonic flow past a flare: computational domain and mesh details

setting is concerned, Fig. 7a shows the computational domain and the boundary conditions used to perform both the S-C and S-F computations: supersonic inflow boundary conditions have been imposed on the four upstream faces, highlighted in green in Fig. 7a, whereas extrapolation boundary conditions have been prescribed on the remaining boundary faces, marked in red in Fig. 7a. The usual slip condition has been applied along the body walls. Fig. 7 also shows the background grid used in both the S-C and S-F calculations and, in Fig. 7c, the triangulated surface of the bow shock obtained at steady-state in the S-F calculation. The embedded shock is not visible in Fig. 7c, because the S-F technique has been used in a hybrid manner, whereby only the bow shock has been fitted, whereas the embedded one has been captured.

A qualitative comparison between the S-C and the S-F solutions is available in Fig. 8, which shows the dimensionless pressure field computed using the two shock-modeling options. In Fig. 8b, which refers to the S-F calculation, the intersection between the triangulated shock-surface and the  $XZ$  plane has been marked using a red solid line. Figure 8b clearly reveals the hybrid nature of the S-F calculation, with the bow shock being fitted, whereas the embedded shock and the shock-shock interaction have been captured. Figures 9 and 10 show details of the pressure field close to the hemispherical nose and in the region where the shock-shock interaction takes place. The comparison between the S-C (Figs. 9a and 10a) and S-F (Figs. 9b and 10b) calculations leads to conclusions that are very similar to those already made in Sect. 3.1. First of all, the un-physical thickness of the captured shock (which spans a few cells) is comparable

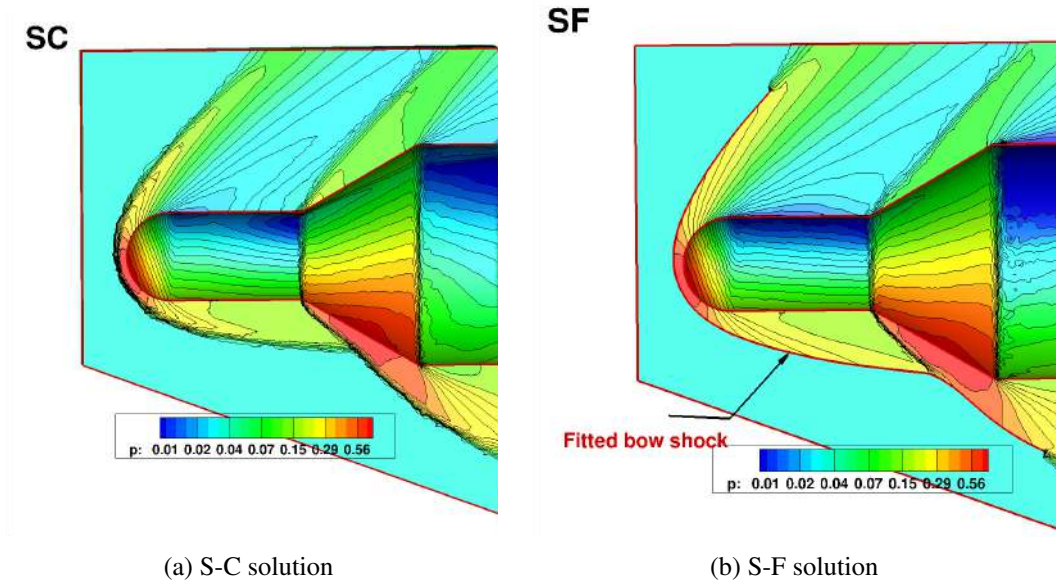


Figure 8: Hypersonic flow past a flare: computed pressure field in the XZ plane

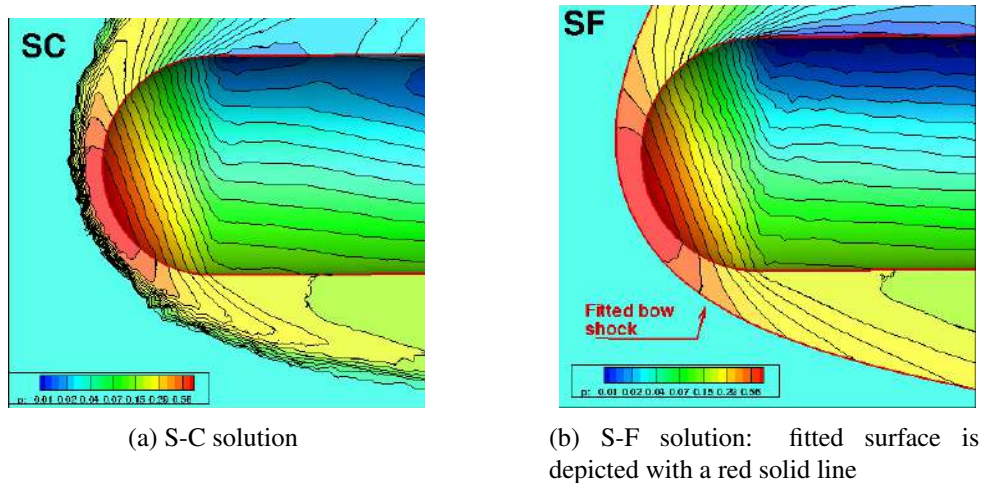


Figure 9: Hypersonic flow past a flare: hemispherical-nose enlargement in the XZ plane.

to the shock stand-off distance. In the present test-case, however, possibly because of the lower shock-strength, the solution within the shock-downstream region is not so badly affected by the capture of the shock as it is in the hypersonic case of Sect. 3.1: compare Figs. 9a and 5a. Anyway, by comparing the left and right frames of Figs. 8, 9 and 10, it is clear that significant differences exist between the pressure field computed by S-C and S-F downstream of the bow shock, both in the symmetry plane and over the body surface. Finally, Fig. 10, which shows a zoom of the region where the shock-shock interaction takes place, suggests that the S-F solution may be

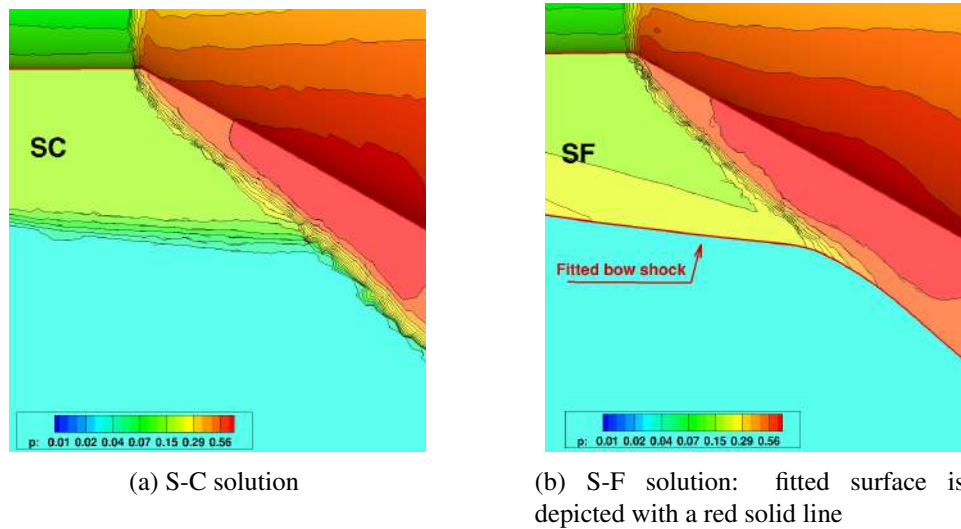


Figure 10: Hypersonic flow past a flare: interaction region enlargement in the  $XZ$  plane.

further improved if the embedded shock and the shock-shock interaction were also fitted, rather than being captured as in Fig. 10b. This is work underway, which requires improvements in the GRUMMP mesh-generation package to make it capable of dealing with interacting surfaces and the computational modeling of three-dimensional shock-shock interactions.

#### 4 CONCLUSIONS

In this paper the S-F algorithm described in [2] has been further developed and enhanced: in particular, we highlight the role of the GRUMMP library in handling the fitted shock fronts within the background grid, which provided a significant improvement in our code capability. The lack of a tailored software capable of performing tetrahedral mesh generation around the moving triangular shock surface, has been the main obstacle that prevented further developments of three-dimensional S-F algorithm described in [2], in order to make it capable of dealing with shock-shock interactions as already done in the 2-D case [1]. These limitations can be overcome with the use of the GRUMMP library: although the simulations analyzed in this article do not consider the interaction between fitted shock surfaces (indeed, in the second test-case only a hybrid interaction between a fitted and a captured shock has been described), these results show the current progress due to the GRUMMP library implementation in the S-F technique and the challenges ahead.

#### REFERENCES

- [1] Paciorri R, Bonfiglioli A. A shock-fitting technique for 2D unstructured grids. *Computers & Fluids* (2009) **38(3)**:715–26
- [2] Bonfiglioli, A., Grottaurea, M., Paciorri, R., Sabetta, F. . An unstructured, three-dimensional, shock-fitting solver for hypersonic flows. *Computers & Fluids* (2013) **73**.,

162-17.

- [3] Zaide D.W., Ollivier-Gooch C.F., Inserting a surface into an existing unstructured mesh *International Journal for Numerical Methods in Engineering* (2016) **106 (6)**: 484–500
- [4] Si Hang. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)* (2015) **41.2**: 1-36.
- [5] Tautges, T. J., CGM: A Geometry Interface for mesh generation, analysis and other applications. *Engineering with Computers* (2001) **17(3)**: 299–314.
- [6] Gosselin, S., and Ollivier-Gooch, C., Constructing Constrained Delaunay Tetrahedralizations of Volumes Bounded by Piecewise Smooth Surfaces. *International Journal of Computational Geometry & Applications* (2011) **21**: 571-594.
- [7] Frey, Pascal. Yams a fully automatic adaptive isotropic surface remeshing procedure. 2001.
- [8] Bonfiglioli A., Fluctuation splitting schemes for the compressible and incompressible Euler and Navier-Stokes equations. *International Journal of Computational Fluid Dynamics* (2000) **14(1)**: 21-39.
- [9] A. Bonfiglioli, R. Paciorri, Convergence analysis of shock-capturing and shock-fitting solutions on unstructured grids, *AIAA Journal* (2014) **52 (7)**:1404–1416.
- [10] Edney B., Anomalous heat transfer and pressure distributions on blunt bodies at hypersonic speeds in the presence of an impinging shock, TR 115, *Flygtekniska Forsoksanstalten (The Aeronautical Research Institute of Sweden)* (1968).