



Metric-based mesh adaptation for hypersonic flows: capturing wall heat flux with anisotropic triangles and tetrahedrons

Guillaume Puigt¹, Valentin Golliet², Frédéric Alauzet³

Abstract

Mesh adaptation consists of optimizing the mesh to decrease the discretization error under the constraints that the number of degrees of freedom of the computations is kept constant. To do so, an error estimate is first computed, associated with the notion of local metrics. It is then transferred to the metric-based remesher, and once the new mesh is obtained, the initial solution is the interpolated one of the previous mesh. In addition, the final solution is the best one for a given number of degrees of freedom (or computational cost). Then, multiplying by 2 the required degrees of freedom and starting again the computational loop enables the verification of mesh convergence. In this context, the present paper is dedicated to using and modifying the computational chain of mesh adaptation for hypersonic flows. Our goal is to focus attention on the computation of the wall heat flux, a quantity that depends strongly on mesh quality for unstructured grids. One key aspect is the solution of the adjoint problem to get the metric estimation, which is an input of the remesher. The performance of the proposed computational chain is assessed with test cases of increasing complexity, both in 2D and 3D.

Keywords: *mesh adaptation, triangle, tetrahedron, wall heat flux, perfect gas*

Nomenclature

<i>Latin</i>	<i>node j</i>
C_p – Wall pressure coefficient	Ma – Mach number
C_h – Heat flux coefficient at the wall	Re – Reynolds number
V_j – volume of the triangle/tetrahedron indexed j	<i>Subscripts</i>
$\mathcal{V}_j$ – volume of the median cell associated with	∞ – any inflow quantity

1. Introduction

Reentry is the capability of an object launched from Earth to leave Earth's atmosphere and then come again. During reentry at the hypersonic regime, the object encounters essentially two energetic phenomena: a strong bow shock and the transformation of kinetic velocity into heat in the boundary layer. The latter is so intense that the temperature is high enough to activate chemistry and transform oxygen and nitrogen into other products. Of course, a huge part of the energy is transferred to the wall and the object by conduction. The most important quantity to compute is the wall heat flux, since dimensioning the object requires good confidence in its computation.

Heat flux in the boundary layer varies mainly in the direction normal to the wall. So, a standard prerequisite is the definition of a mesh with mesh lines perpendicular to the wall. Standard schemes that mimic the Finite Difference method can easily estimate the wall heat flux. However, the straightforward way to define such a mesh is to consider structured grids. The mesh generation requires a long time for the blocking definition and to mesh the blocks. The procedure is not automatic and depends strongly

¹*GammaO, Equipe Inria-ONERA, DMPE, ONERA, Université de Toulouse, 31000 Toulouse, France, guillaume.puigt@onera.fr*

²*GammaO, Equipe Inria-ONERA, DMPE, ONERA, Université de Toulouse, 31000 Toulouse, France, valentin.golliet@onera.fr*

³*GammaO Team, Inria Saclay Ile-de-France, Palaiseau, France, frederic.alauzet@inria.fr*

on the user's expertise/experience. The technique cannot be applied successfully to complex geometry because the required human resources are larger.

On the other hand, unstructured grids are generated automatically, even on complex geometries. However, controlling mesh elements is more challenging: the alignment of flow physics with mesh lines is not guaranteed. The quality of the solution is ultimately inferior to that of using structured grids. For hypersonic flows during reentry, this statement is transferred into a bad alignment in the boundary layer, and wall heat flux can be false (its maximum value, at the stagnation point, is lost).

In this work, we introduce a metric-based mesh adaptation procedure. Contrary to standard computation, the mesh adaptation procedure converges the couple (mesh, solution) for a given set of numerical scheme. The automaticity of the process relies on specific ingredients:

- a CFD solver robust on anisotropic meshes composed of simplexes (triangles in 2D and tetrahedron in 3D)
- an error estimate procedure to define the metrics for generating the new mesh
- a mesh generator method to build the mesh from metrics
- an interpolation algorithm to interpolate the previous solution on the new mesh

The full procedure is applied using a node-centered solver, and the remesher is the library `feflo.a` from Inria.

The present paper is organized as follows. After this short introduction, the discretization is summarized, and the adaptation loop is introduced. Mesh adaptation has proven its efficiency for aircraft and turbomachinery, so the focus is on additional ingredients in Sec. 4. Several numerical experiments for 2D and 3D flows are presented in Sec. 5 before concluding.

2. Navier-Stokes equations and discretization using a mixed Finite Volume / Finite Element approximation

The Navier-Stokes equations in $\mathbb{R}^d$ are written in compact form as:

$$\frac{\partial W}{\partial t} + \nabla \cdot F(W) + \nabla \cdot G(W, \nabla W) = 0, \quad (1)$$

where the vector W is the vector of the conservative variables. It is composed of the density ρ , the d components of the momentum $\rho \vec{U}$, and the total energy ρE . F is the convection flux density and G is the diffusion flux density. This standard system is closed with the perfect gas assumption, with constant heat capacity at constant pressure C_p or constant volume C_v , constant polytropic coefficient γ (for air $\gamma = 1.4$), and perfect gas equation $p = (\gamma - 1)\rho e = \rho C_v T$. e is the internal energy, T the temperature, and p the pressure. The symbols $\nabla \cdot$ (respectively ∇) represent divergence (gradient resp.) operator concerning spatial directions. Eq. 1 is solved using the Finite Volume approximation. To do so, the domain of interest is first covered by 2D or 3D elements (defined hereafter). In 3D, two elements are connected if they share an edge, a node, or a face. In 2D, they are connected if they share an edge or a node.

Following the Finite Volume approximation, Eq. 1 is integrated over control volumes; in our approach, they are defined using the median-dual approach as:

- Mesh elements are triangles ($d = 2$) or tetrahedrons ($d = 3$) only,
- Control volumes for the Finite Volume approximation follow the median cell formulation. The dual cell in 2D is built around mesh nodes by introducing fictitious edges linking the cell center with edge midpoints. In 3D, the dual cell boundary associated with the edge inside a tetrahedron is closed by segments based on the centers of the adjacent faces, the edge midpoint, and the cell center. The dual cell boundary is always a four-node planar face, and the definition of the face normal is ubiquitous using nodes only.

- As a consequence, there are as many degrees of freedom as mesh nodes.

The main interest of such a formulation is the possibility of performing hyperbolic flux computation by a loop over mesh edges. Diffusion is computed using the Finite-Element approach on the simplices. In addition, the number of neighboring cells is generally much larger than that for a cell-centered formulation because a node has many more connected edges than an element has connected faces.

Let $\mathcal{T}_i$ be a control volume built around node i and $\mathcal{V}_i$ be the measure of its volume. Let $\overline{W}_i$ be the averaged quantity of W over the control volume $\mathcal{T}_i$:

$$\mathcal{V}_i \overline{W}_i = \int_{\mathcal{T}_i} W d\mathbf{x}.$$

Eq. 1 is integrated over $\mathcal{T}_i$, and using Green's relation for the divergence term and Fubini's theorem to invert integration and derivation, the following Finite Volume equation Eq. 2 is obtained:

$$\frac{d}{dt}(\mathcal{V}_i \overline{W}_i) + \sum_{f \in \mathcal{L}_i} \int_f F(W) \cdot \vec{n} d\gamma + \sum_{f \in \mathcal{L}_i} \int_f G(W, \nabla W) \cdot \vec{n} d\gamma = 0, \quad (2)$$

where the surface integrals over the boundary $\partial\mathcal{T}_i$ of $\mathcal{T}_i$ are cast into several boundary integrals, according to the definition of the boundary faces. The boundary of $\mathcal{T}_i$ is composed of faces shared with adjacent volumes $\mathcal{T}_j$ and boundary faces associated with the computational domain limits. In Eq. 2, $\mathcal{L}_i$ refers to $\mathcal{T}_i$'s full list of faces.

Our node-centered Finite Volume formulation was introduced by Dervieux [1] (from INRIA) in 1985 and by Rostand (from Inria) and Stoufflet (from Dassault Aviation) [2] in 1988. The approach was the subject of intense development and analysis, and the full review is out of the scope of the paper. Recent results on the node-centered formulation are summarized in [3, 4]. It is shown that the formulation can handle strongly anisotropic adapted meshes with dedicated numerical schemes. In this article, only the most important ingredients of the approach are recalled.

2.1. Convective flux computation

The faces of the control volumes are grouped and associated with the edges. In 2D, any internal edge has two bi-segments based on the edge midpoint and cell centers. In 3D, it is still possible to do the same association using all the faces based on the same edge midpoint. In practice, the normal vector is the sum of the normal vectors of the faces scaled by the area of the face. The normal vector is therefore not a unit vector. Its norm represents the equivalent face area. This approximation enables the computation of the convection flux using a single loop over the mesh edges.

The convection flux depends on the left and right states evaluated at the edge midpoint, coupled with the MUSCL technique, and on the numerical flux. In the following, the flux is always computed using the HLLC approximate Riemann solver proposed by Batten *et al.* [5]. The second-order MUSCL [6] approximation is less standard.

2.2. Second-order MUSCL approximation

For an edge $P_i P_j$ between nodes P_i and P_j , the flux is integrated over the sum of the facets based on the edge midpoint, using the definition proposed before (median cell). The computation of the flux depends on the linearly extrapolated states:

$$\begin{cases} W_{ij} = W_i + \frac{1}{2}(\nabla W)_i^\beta \cdot \overrightarrow{P_i P_j} \\ W_{ji} = W_j + \frac{1}{2}(\nabla W)_j^\beta \cdot \overrightarrow{P_j P_i}. \end{cases} \quad (3)$$

The location of the edge midpoint explains the coefficient of $1/2$ associated with a linear displacement of half the edge vector. The β -gradient is a particular sum of centered and upwind contributions:

$$\begin{cases} (\nabla W)_i^\beta \cdot \overrightarrow{P_i P_j} = (1 - \beta)(\nabla W)_{ij}^C \cdot \overrightarrow{P_i P_j} + \beta(\nabla W)_{ij}^U \cdot \overrightarrow{P_i P_j} \\ (\nabla W)_j^\beta \cdot \overrightarrow{P_j P_i} = (1 - \beta)(\nabla W)_{ij}^C \cdot \overrightarrow{P_j P_i} + \beta(\nabla W)_{ij}^D \cdot \overrightarrow{P_j P_i}. \end{cases} \quad (4)$$

The centered gradients are computed along the edge and:

$$\begin{cases} (\nabla W)_{ij}^C \cdot \overrightarrow{P_i P_j} = W_j - W_i \\ (\nabla W)_{ij}^C \cdot \overrightarrow{P_j P_i} = W_i - W_j, \end{cases} \quad (5)$$

while upwind (U) and downwind (D) gradients are computed using the finite element approach on simplex, using the gradient of the shape function for upwind and downwind elements. Those elements are defined using a geometric algorithm, as introduced in Fig. 1. In practice, we consider the V4-scheme for which $\beta = 1/3$ [7].

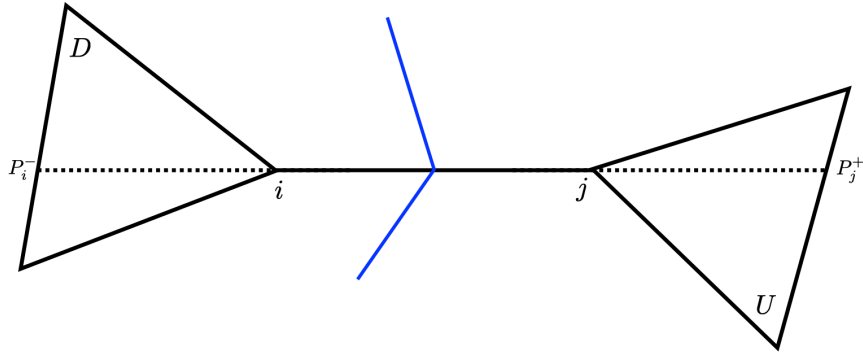


Fig 1. Representation of the mesh edge ji , downwind (D), and Upwind (U) triangles. The blue segments represent the face of the convective flux computation. Upwind and downwind triangles are defined by the intersection between the line ij and the opposite face of the triangle, leading to the additional points P_i^- and P_j^+ .

The present scheme is not bound-preserving, especially near a discontinuity. The MUSCL formulation requires a slope limiter so that the extrapolated values W_{ij} and W_{ji} are not invalid. The complete reconstruction starts from Eq. 3 and includes now the slope limiter Ψ :

$$\begin{cases} W_{ij} = W_i + \Psi \left(\frac{\nabla W^C \cdot \overrightarrow{P_i P_j}}{\nabla W^U \cdot \overrightarrow{P_i P_j}} \right) \frac{1}{2} (\nabla W)_i^\beta \cdot \overrightarrow{P_i P_j} \\ W_{ji} = W_j + \Psi \left(\frac{\nabla W^C \cdot \overrightarrow{P_j P_i}}{\nabla W^D \cdot \overrightarrow{P_j P_i}} \right) \frac{1}{2} (\nabla W)_j^\beta \cdot \overrightarrow{P_j P_i}. \end{cases} \quad (6)$$

In the present work, we always consider Piperno's limiter [8, 9], which is an extension of the Van Albada limiter to the β -scheme.

2.3. Viscous flux

The Finite-Volume discretization requires the definition of the gradient on the surface. Surfaces are defined inside mesh elements. Assuming that the unknown is located at the mesh nodes, the Finite Element approximation provides a unique gradient by control volume. As introduced in Eq. 14 in [3], the closure of the viscous terms is performed element by element, using the trace of the dual cell faces per element.

2.4. Boundary conditions

Boundary conditions are imposed on the mesh boundary faces, following the same principles as for a centered Finite Volume formulation. However, some edges have one or two vertices located on boundary conditions. In this case, the upwind or downwind gradients cannot be established due to the lack of upwind/downwind elements. In that case, the lacking gradient is replaced by the nodal gradient using Clément's L^2 -projection. Mathematically, Clément's projection leads to the definition of a nodal gradient defined as the average of element-wise Finite Element gradients [3], using mesh simplex volume as a weighting coefficient.

2.5. Implicit time integration

The present work focuses on the steady solution of the laminar Navier-Stokes equations. Efficiency for solution convergence is a prerequisite to diminish or control the total CPU time for the computation. In that context, we use a pseudo-transient continuation method using the pseudo-time method with an implicit time integration. For a steady solution, the backward Euler approach is coupled with a Symmetric Gauss-Seidel (SGS) implicit solver for the linear system, as in [3]. Moreover, an automatic CFL law is implemented to minimize the restitution time. Finally, the convergence of the implicit system can be discarded when limit cycles are created during the process. A standard way to circumvent it is by limiter freezing, consisting here of freezing gradients, except if the new gradient is smaller than the previous value.

2.6. Partial conclusion

The discretisation and the numerical ingredients are standard, as published in the literature. They serve as a basis for the extension to hypersonic flows. During the computations, computational difficulties were highlighted, and additional ingredients were required. They are introduced in the next section.

3. Mesh adaptation computational loop

3.1. Description of the loop

The principle of the metric-based mesh adaptation is to converge at the same time the mesh and the solution in a computational loop. The process is therefore nonlinear. Starting from a coarse mesh, its associated solution, and a list of targeted mesh sizes (typically the number of mesh nodes), the user first defines the maximum of metric-based mesh adaptations for any targeted mesh size and a measure of convergence (for instance, evolution of the lift coefficient is lower than a threshold). Then, the computational loop becomes:

■ **External loop:** for a given expected number of mesh nodes, do:

• **Adaptation loop:** for any metric-based mesh adaptation, do:

- ⇒ Error estimate
- ⇒ Remesher
- ⇒ Interpolation
- ⇒ Computation
- ⇒ Solution analysis

• end for **Adaptation loop**

■ end for **External loop**

The error estimate is computed and defines the local modification of edge lengths, surfaces, and volumes, angles. This error estimate provides privileged directions and local metrics for the mesher. Our remesher differs from the standard Adaptive Mesh Refinement method. Indeed, the standard Adaptive Mesh Refinement that starts from a given mesh and can split volumes ; our procedure requires the meshing of the full computational domain. The meshing uses the mesher *feflo.a* developed at Inria. *Feflo.a* builds the new mesh from the previous one and respects the metrics imposed on the new mesh. Instead of starting the solution from a constant state, the solution is interpolated to the new mesh to take advantage of the information available at the previous mesh. Solution in a new mesh node is defined by a standard linear reconstruction using barycentric weights. Then, the computation is started again to provide the solution. If several solutions on different adapted meshes vary under a threshold, mesh convergence is attained and the procedure continues with a refined mesh (the number of degrees

of freedom is typically multiplied by 2).

To resume, the interest of the proposed computational loop lies in the fact that two parameters can be introduced. The first one is associated with the number of mesh nodes (also called complexity in literature). The second one is the possibility to increase the number of mesh nodes in the remesher step of the process from error estimates computed on a given mesh. If the number of mesh nodes is multiplied by two, one can easily demonstrate the mesh convergence of the results.

3.2. Error estimate

Local mesh size and anisotropy are required by the remesher `fefflo.a` to define the new mesh. Two kinds of error estimate are compared hereafter.

The feature-based approach relies on the definition of a sensor from which the Hessian is computed at the mesh nodes. The procedure requires nodal gradients computed using Clément's approach, and the Hessian is estimated using the very same Clément's rule on gradient. The procedure cannot guarantee the equality of cross derivatives of Hessian. Symmetry is therefore enforced by taking average of contributions. Finally, the error estimate is derived in L^p -norm using the Hessian of the quantity, as in [3]. In practice, $p = 4$ in the following as Park *et al.* have shown its efficiency to capture the boundary layer earlier in the adaptation loop [10].

The goal-oriented approach relies on the definition of a sensor estimated from the solution of the CFD problem and from the solution of the adjoint problem. The goal is to obtain the sensitivity of the considered output functional to the flow. Standard output functional relies on the lift and drag coefficients, but another problem will be presented in Sec. 4 and dedicated to the wall heat flux for hypersonic flows. The goal-oriented solution is computed by solving a linear problem using the GMRES method.

4. Additional ingredients for mesh adaptation

4.1. For boundary conditions

The MUSCL formulation is modified using Clément's gradient for boundary nodes. When a shock impacts an outflow boundary condition, such a formulation becomes unstable, leading to unphysical results. To overcome this limitation, a shock location algorithm is implemented, and for nodes associated with boundary conditions, the flux is reverted to first-order accuracy by removing the MUSCL reconstruction.

4.2. Definition of the adjoint problem

The definition of the optimal problem using Lift and Drag coefficients was shown to be efficient for aircraft computation. However, interest in hypersonic flows lies in capturing the wall heat flux. We propose the integral of the wall heat flux over the whole wall as a new criterium for mesh adaptation. The key point is the linearization (Jacobian) of the wall heat flux with respect to the unknowns. Let's illustrate the procedure.

Starting from the wall heat flux $h = \lambda \nabla T \cdot \vec{n}$, it is clear that:

$$\frac{\partial h}{\partial W} = \left(\frac{\partial \lambda}{\partial W} \nabla T + \lambda \frac{\partial \nabla T}{\partial W} \right) \cdot \vec{n}.$$

For a perfect gas, λ is related to T by Sutherland's law, and $\partial \lambda / \partial W$ is easily computed. The second term needs the linearization of the temperature gradient. The nodal temperature gradient in node P_i is estimated by Clément's relation:

$$(\nabla T)_{P_i} = \frac{\sum_{P_j \in K_j} V_j \left(\sum_{P_k \in K_j} T_k \nabla \Phi_k \right)}{\sum_{P_j \in K_j} V_j},$$

where the first sum is performed over any elements having P_j as a node, and the term inside the parentheses is the Finite-Element gradient, using the P^1 shape functions Φ_k associated with node k of the triangle K_j . The Jacobian of this term becomes:

$$\frac{\partial(\nabla T)_{P_i}}{\partial W} = \frac{\sum_{P_i \in K_j} V_j \left(\sum_{P_k \in K_j} \frac{\partial T_k}{\partial W} \nabla \Phi_k \right)}{\sum_{P_i \in K_j} V_j}.$$

Such a term involves any triangle/tetrahedron connected to P_i , which makes the linear system composed of as many terms as the number of edges connected to a node.

5. Numerical results

The goal of the present section is to analyze numerical results, comparing them with experimental data or other numerical solutions, and validating the formulation. In the following, the wall pressure coefficient C_p is defined by:

$$C_p = \frac{p - p_\infty}{0.5 \rho_\infty u_\infty^2},$$

where p is the local pressure, and p_∞ , ρ_∞ , u_∞ represent the inflow pressure, density, and velocity, respectively. The wall heat flux coefficient C_h is defined by:

$$C_h = \frac{\lambda \nabla T \cdot \vec{n}}{0.5 \rho_\infty u_\infty^3}.$$

Computations can start with one without the MUSCL formulation (order 1). Once the flow is converged, a restart from the first-order accurate solution is performed at order 2. In many cases, the simulations start directly at order 2. The second-order solution serves as the initial solution to the adaptive process. Steady solution convergence is monitored. Any steady computation is stopped once relative residuals on Drag, C_p , and C_f are below 10^{-4} , on Lift below 10^{-2} , and on heat flux below 10^{-2} . In the following, another criterion is associated with the mesh convergence for a given expected number of degrees of freedom. The associated convergence residual is therefore linked with the adaptation process, and not a single CFD computation.

5.1. Cylinder at Mach $Ma = 5.73$

We start with the computation of the laminar flow over a cylinder at Mach number $Ma = 5.73$ and Reynolds number $Re = 2050$. The fluid temperature is $39.6K$ and the (isothermal) wall temperature is $210K$. This test case is provided as Case 1, Section 2.2 in [11]. The interest of such a case lies in the experiments and computations available. The reference solution is provided by the authors using their "MP7-AUSMP+" scheme regarding the distribution of the wall pressure coefficient C_p and the wall heat transfer C_h . In the following, feature-based and goal-oriented computations will be compared.

All computations start with a mesh composed of 2048 mesh nodes (Fig. 2). Three conditions are applied: an inflow condition (pink line), the isothermal wall (green and red lines), and an outflow condition (grey and yellow lines). The mesh is refined near the outflow condition because our adjoint solver could not resolve properly triangles sharing two adjacent boundary conditions. Finally, the last segment won't be accounted for during the adjoint computation, and the adjoint is applied only on the red surface. Once the second-order solution is obtained, the adaptation process is started, using parameters defined in Tab. 1. The mesh size represents the targeted number of mesh nodes. The computational process cannot guarantee the exactness of the number of mesh nodes, but our experience shows that the variation is below 10%. The maximum number of mesh adaptations gives the maximum number of remeshing accepted during the computation, for any targeted number of mesh nodes. In practice, the number of iterations is strongly reduced because the three other convergence parameters are true. The first two correspond to one lift count and one drag count. The last one corresponds to the integrated

heat flux at the wall. It is measured as a relative variation between two consecutive computations (here below 1%). The computational loop stops once the three errors are below the targeted threshold.

Two sets of computations are performed. For the Feature-Based mesh adaptation, the criterion is the Hessian of the temperature, and the error estimate is based upon the L^4 -norm. The goal-oriented error estimate is associated with the integral of the wall heat flux.

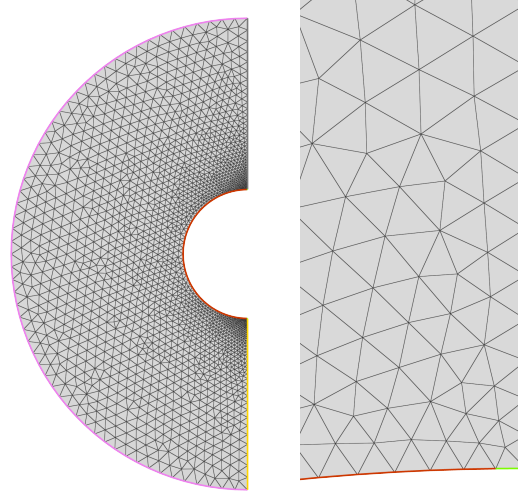


Fig 2. Cylinder at $Ma = 5.73$: initial mesh (left) and zoom (right) near the outflow condition. Boundary conditions are represented by colors.

Table 1. Targeted parameters for mesh adaptation

Mesh size	4K	8K	16K	32K	64K	128K	256K	512K
Max. number of adaptations	20	15	10	10	10	10	10	10
Error on Lift	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.01
Error on Drag	0.0004	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
Error on heat flux	0.05	0.02	0.02	0.01	0.01	0.01	0.01	0.01
Feature-Based	8	6	3	3	3	3	3	3
Goal-oriented	8	5	3	3	3	3	3	3

5.1.1. Analysis of the Feature-Based mesh adaptation solutions

Fig. 3 and 4 show the distribution of the wall pressure coefficient obtained on the last computation for the expected number of degrees of freedom. The mesh convergence of the coefficient is easily seen. During the mesh adaptation process, the distribution of nodes at the wall is not forced symmetric around the stagnation line. But the solution becomes symmetric. The very same kind of results is obtained with the goal-oriented mesh adaptation based on the wall heat flux distribution. The zoom highlights that the convergence of the wall pressure coefficient requires fewer degrees of freedom for the goal-oriented approach (the solutions using the goal-oriented approach are almost identical since 8K mesh nodes).

Fig. 5 represents the wall heat flux distribution, and our results fit well with the reference from [11].

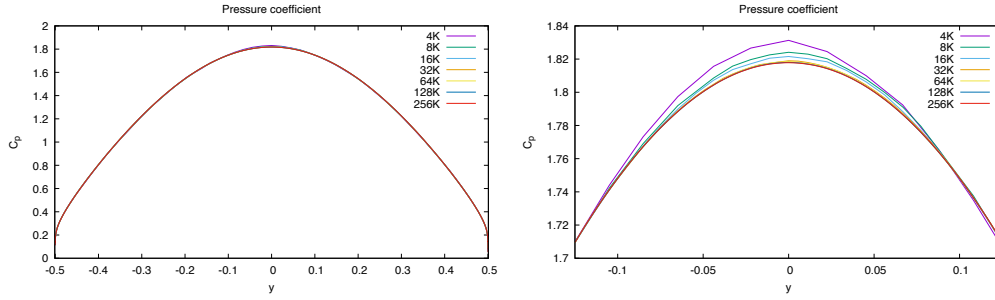


Fig 3. Cylinder at $Ma = 5.73$: C_p distribution (left) and zoom (right) near the stagnation line for the feature-based mesh adaptation.

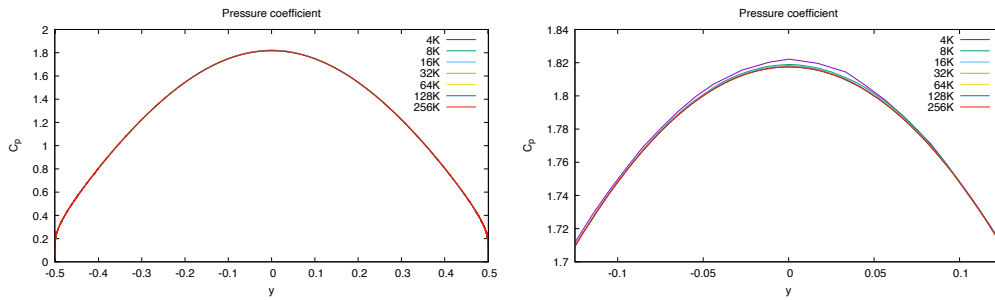


Fig 4. Cylinder at $Ma = 5.73$: C_p distribution (left) and zoom (right) near the stagnation line for the goal-oriented mesh adaptation.

The convergence of the wall heat flux is easily seen. Here again, the solution is shown to converge more slowly to the reference with the feature-based method than with the goal-oriented technique. Any solution still presents a few oscillations of the wall heat flux. This is indeed a consequence of the computation of the gradient with a non-uniform grid. These oscillations should be removed by a post-processing tool (polynomial interpolation, for instance), but we prefer to present results directly issued from the CFD solver.

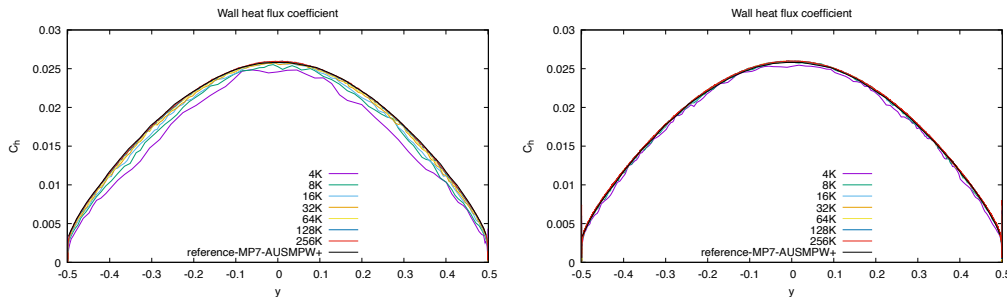


Fig 5. Cylinder at $Ma = 5.73$: C_h distribution with feature-based (left) and goal-oriented (right) methods.

Feature-based and goal-oriented techniques are now compared for their efficiency. First, the converged solutions with the mesh at 216K nodes are compared with the reference computation (Fig. 6). All solutions and the reference computation are very close. An interesting question concerns the efficiency of the method to attain a level of accuracy for the wall heat flux distribution. Two sets of data are compared. Despite the similarities of the very coarse solutions (Fig. 7) and the solutions on intermediate meshes (Fig. 8), the goal-oriented method requires between 4 and 6 times fewer mesh nodes than the

feature-based technique. This is not surprising because the goal-oriented method focuses explicitly on the wall heat flux.

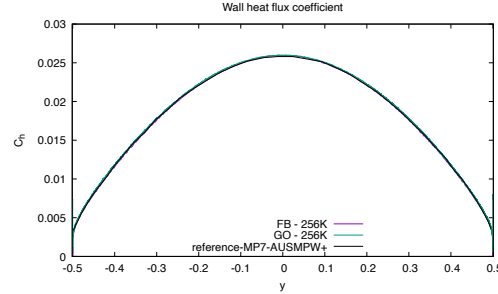


Fig 6. Cylinder at $Ma = 5.73$: C_h distribution with feature-based and goal-oriented methods on the most refined mesh. Both methods are equivalent.

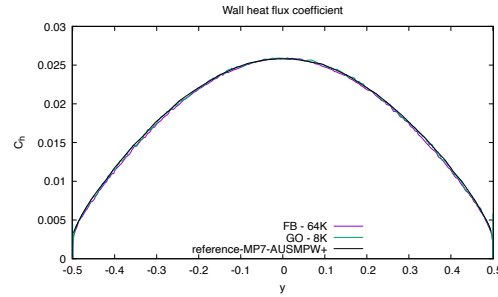


Fig 7. Cylinder at $Ma = 5.73$: C_h distribution with feature-based and goal-oriented methods on the very coarse meshes.

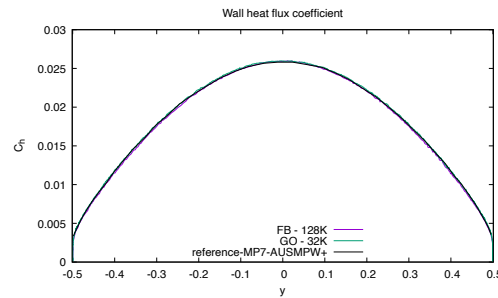


Fig 8. Cylinder at $Ma = 5.73$: C_h distribution with feature-based and goal-oriented methods on intermediate meshes.

5.2. Edney-type IV shock/shock interaction, with focus on the wall heat flux

The second test case is the shock-shock interaction of type IV (as defined by Edney [12]), at $Ma = 9.95$. Experiments were conducted in the ONERA R5Ch wind tunnel.

The geometry is composed of a wedge installed to generate a detached shock that interacts with the detached shock over a cylinder. The computation is purely 2D. The goal is to capture the flow introduced in Fig. 9 near the cylinder.

The simulation starts with a uniform mesh composed of 735 nodes, and the solution is a rough approximation because the mesh is not built to capture physics (Fig. 10).

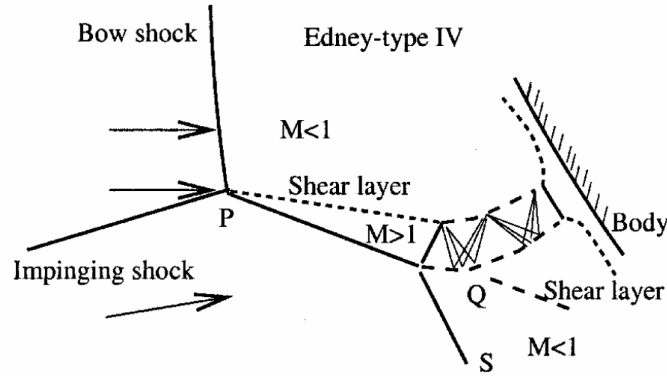


Fig 9. Shock/shock interaction defined by Edney as type IV

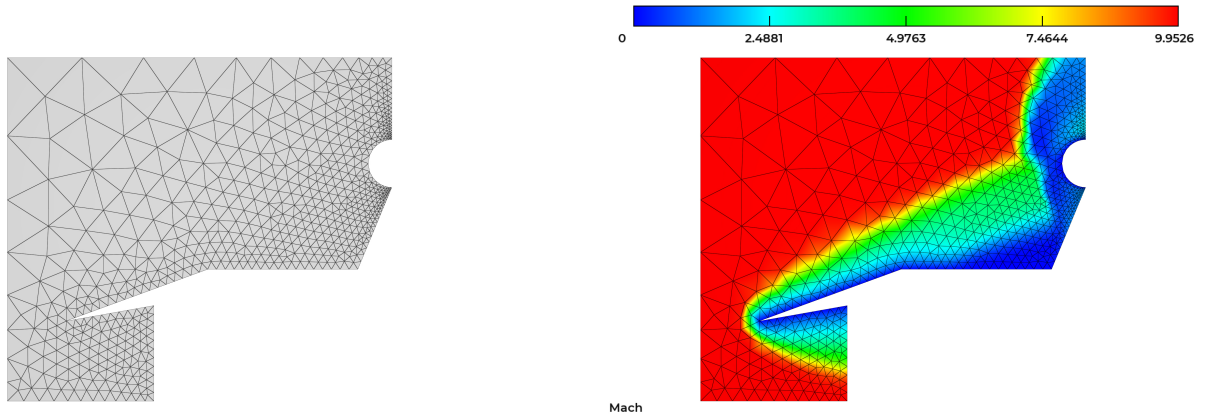


Fig 10. Uniform mesh to initialize the computation over the Edney type 4 test case, Mach number on the same mesh on the right.

The metric-based mesh adaptation then starts. Starting from the objective of 8,000 nodes, several meshes are built to demonstrate the mesh convergence. The whole set of simulations is summarized in Tab. 2. Two sets of computations are performed. The feature-based mesh adaptation is based on the Mach number, associated with its L^2 norm. The goal-oriented simulation is based on the integration of the wall heat flux on the cylinder. In the computational loop, the simulation on mesh index i starts from an interpolation of the solution over mesh $i - 1$, which strongly reduces the overall CPU cost. A key ingredient concerns the convergence of the computation: the metric-based mesh adaptation requires a strong convergence to cope with the mathematical prerequisite. This is to avoid the procedure to capture a non-physical solution associated with a lack of convergence. The residuals during the whole process are presented in Fig. 11.

The solution on the latest mesh with 149,791 nodes (Feature-Based) or 146,225 (Goal-Oriented) and the mesh itself are presented in Fig. 12. One immediately notes that the refinement under the wedge has disappeared for the Goal-Oriented computation since it does not affect the distribution of the wall heat flux on the cylinder.

Our solutions can be compared to those of D'Ambrosio, published in the Journal of Spacecraft and Rockets in 2003 [14]. We focus our attention on the distribution of the wall heat flux over the cylinder. Mesh convergence of the wall heat flux is provided in Fig. 13 for feature-based and goal-oriented simulations. The maximum flux is larger with goal-oriented mesh adaptation. In addition, the effect of the goal-oriented mesh adaptation is clearly visible: fewer degrees of freedom are required for the goal-

Table 2. Parameters for shock/shok interaction case

Mesh size	8K	16K	32K	64K	128K
Max. number of adaptations	10	10	10	10	5
Error on Lift	0.02	0.02	0.01	0.01	0.01
Error on Drag	0.0002	0.002	0.0001	0.0001	0.0001
Error on heat flux	0.02	0.02	0.01	0.01	0.01
Feature-Based	6	3	3	3	3
Goal-oriented	7	3	3	3	3

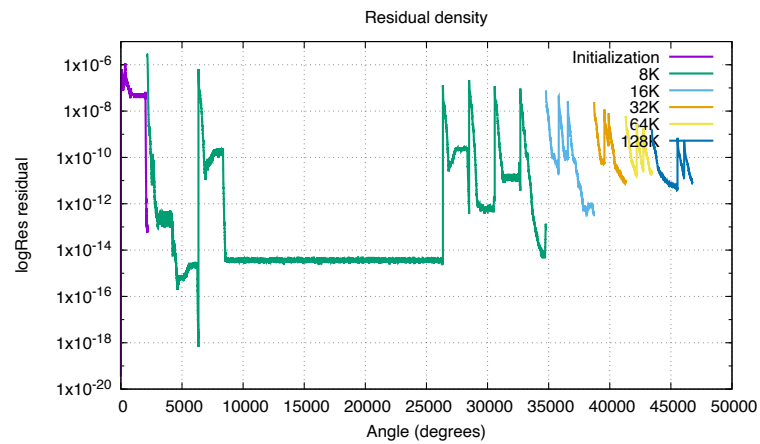


Fig 11. Convergence history for the density using the log residual as defined in [13]. One color corresponds to a number of nodes. Any discontinuity represents a restarting in the process. The residual are shown for the goal oriented mesh adaptation.

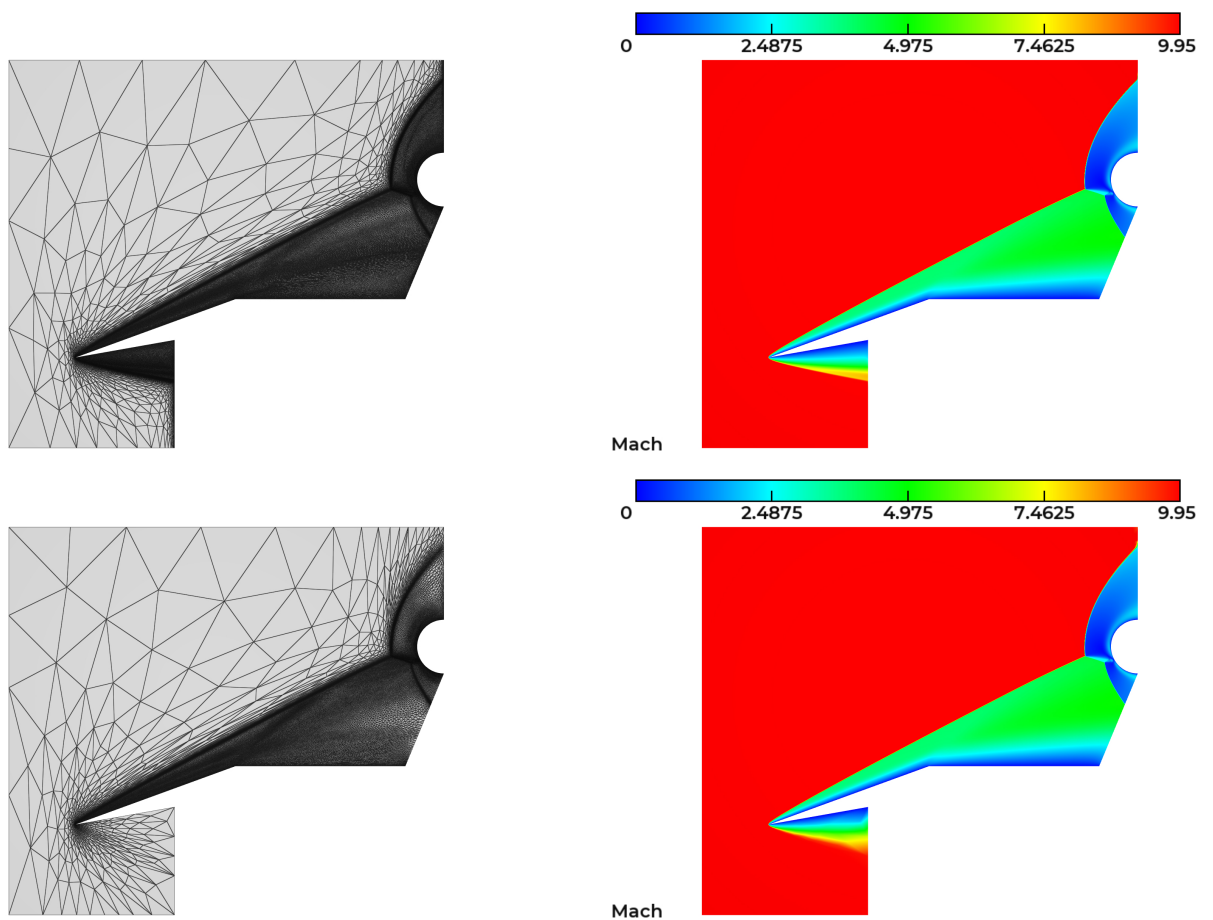


Fig 12. Final meshes for the Edney type 4 test case on the left, Mach number distribution on the same mesh on the right. Feature-Based (Up) and Goal-Oriented (Down) mesh adaptation.

oriented simulation to capture the wall heat flux. Indeed, the 32K goal-oriented solution agrees well with the 128K feature-based solution. Results are in fair agreement with D'Ambrosio and experimental results.

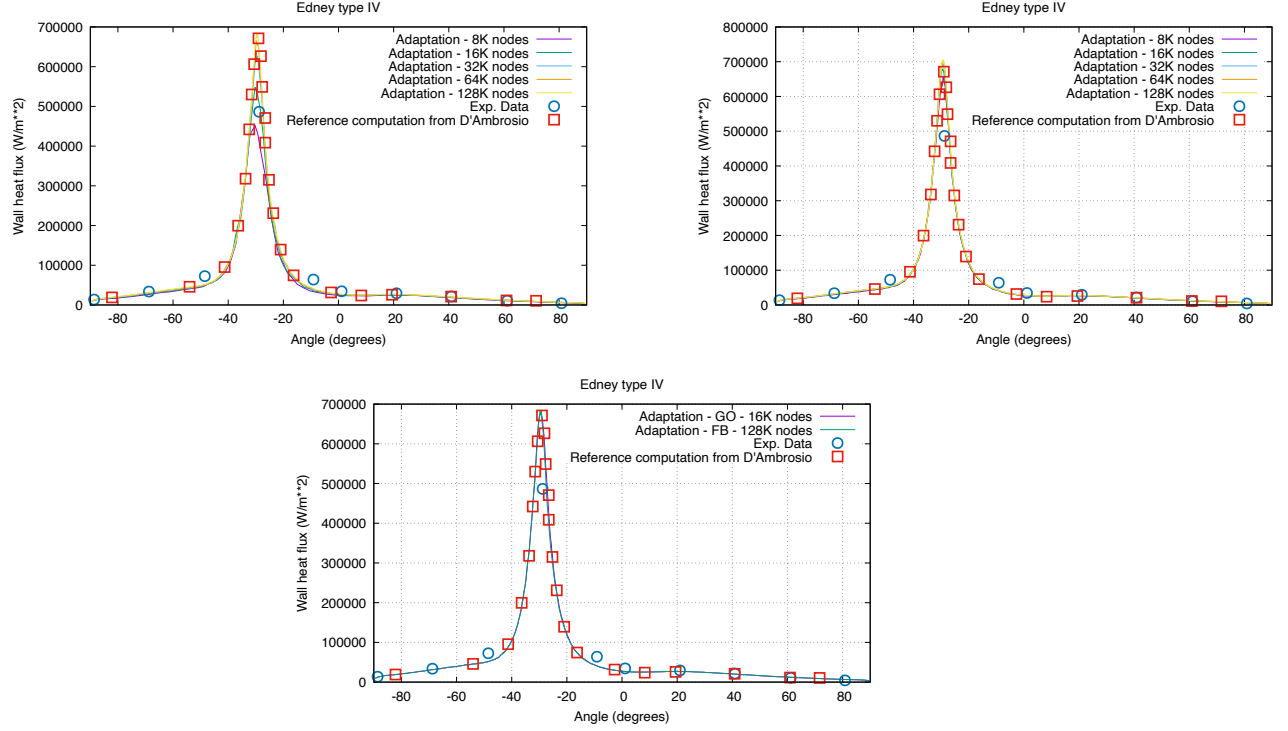


Fig 13. Convergence history for the wall heat flux using Feature-Based (left) and Goal-Oriented (right) approaches. Comparison of goal-oriented and feature-based solutions.

The goal-oriented approach enables a reduction by 8 of the number of grid points required for the same accuracy.

Because of the best mesh convergence of the goal-oriented computational loop, it will be considered as a basic ingredient of the adaptation procedure in the following. The feature-based error estimated will be discarded in the following.

5.3. 2D and 3D laminar compression ramp

The last test case is a three dimensional compression ramp at $M_a = 14.1$ with Reynolds number per unit length $Re/m = 23.6 \cdot 10^4 \text{ m}^{-1}$. This steady state flow configuration was proposed in the work of [15]. A laminar boundary layer develops over the flat plate and then separates ahead of the inclined ramp. As a result, a leading edge shock develops on top of the boundary layer edge. As the flow travels over the flat plate, a separated flow region develops which produces a second oblique shockwave. This shockwave interacts with the leading-edge shockwave. The freestream conditions are:

$$M_\infty = 14.1, \quad Re_\infty/L = 23.6 \cdot 10^4 \text{ m}^{-1}, \quad T_\infty = 88.89 \text{ K}, \\ T_w = 297.22 \text{ K}, \quad \beta = 24^\circ, \quad L = 0.439 \text{ m}$$

where L is the length of the flat plate and β the inclination angle of the ramp.

Several works have been published in the literature [16, 17] based on two-dimensional computations of the flow field, by considering the central longitudinal plane ($y = 0$) of the geometry. Inclinaison angles β of 15° , 18° and 24° were investigated and results from [16] shows good agreements of the 2D model with respect to the experimental data. However, the 2D model fails to predict accurately the separated-flow region when the inclination angle is 24° . Indeed, three dimensional effects are more important as the inclination angle β increases, which cause the airflow to spill on the ramp's sides.

All the mentioned numerical computations from the literature were performed using structured-type meshes for space discretization. In the present study, fully unstructured meshes are employed in line with Goal-Oriented (GO) anisotropic mesh adaptation. The functional of interest is a function of the wall heat flux.

Regarding the computational domain, we take advantage of the symmetry of the geometry to simulate only one half-domain. Employed mesh sizes appear in Tab. 3 with final mesh size of 512K nodes. Mesh adaptation parameters are listed in Tab. 3. The following computation highlights the ability of the method to cope with such compression case flows by considering an 24° inclined ramp. The initial mesh is shown on Fig. 14. Obtained results for wall pressure, friction and heat flux coefficients along the symmetry plane are presented respectively in Fig. 15a, Fig. 15b and Fig. 15c for various meshes to demonstrate that mesh convergence is reached with 256K nodes in the mesh.

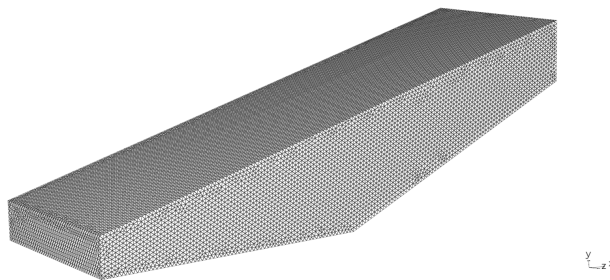


Fig 14. Initial mesh of the 24° compression ramp case (half-domain).

On Fig. 16a, pressure coefficient C_p results match that of the experimental work of Holden [15] for total mesh size of 128K cells and higher. More points are needed for the wall friction C_f and wall heat flux C_h coefficients (see respectively Fig. 16b and Fig. 16c). Some discrepancies are visible for the wall friction coefficient compared to Rudy's results [16] over the inclined ramp portion. Our results show a good match for 256K and 512K mesh sizes. A higher peak in wall friction coefficient C_f is reached

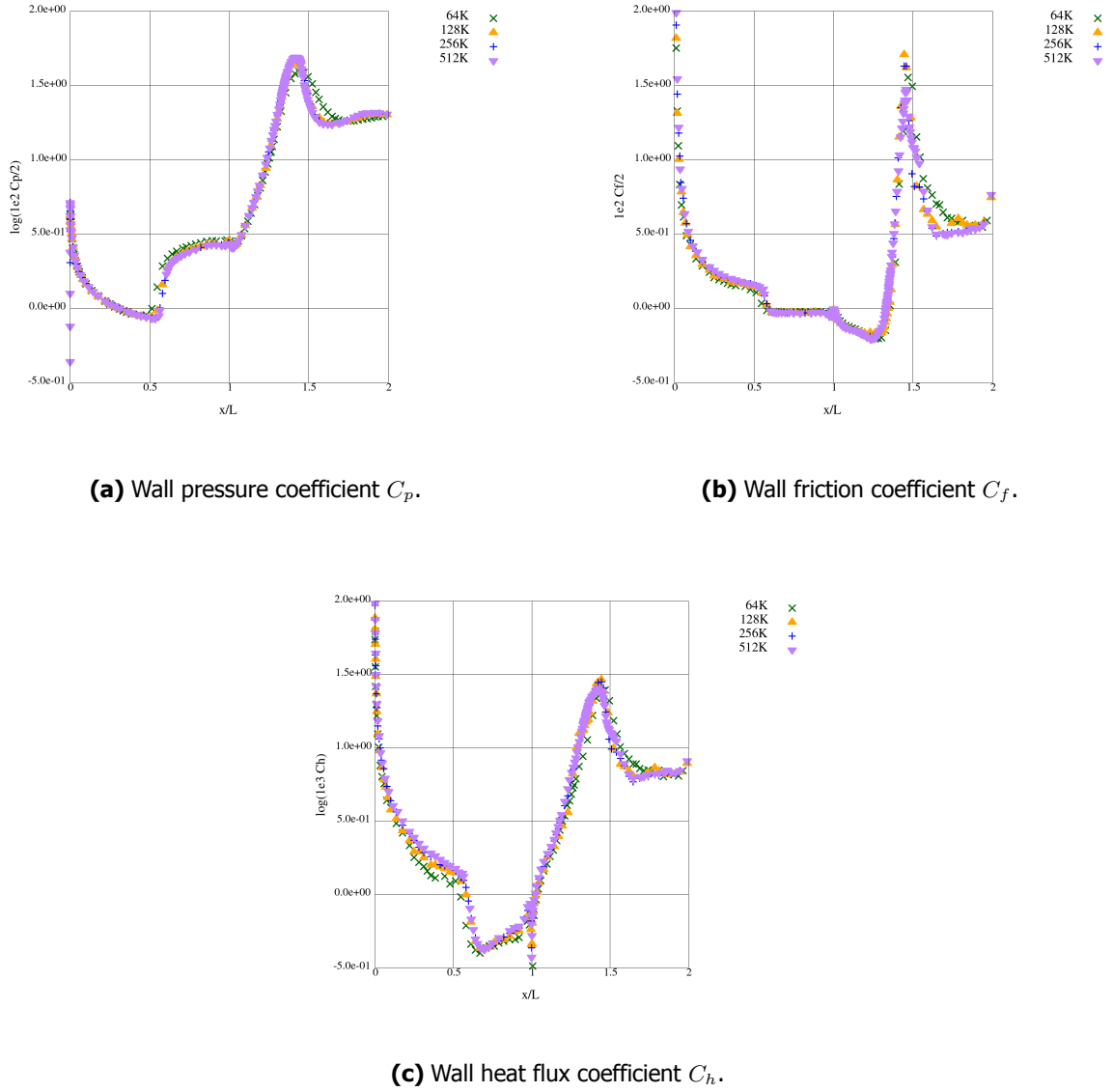
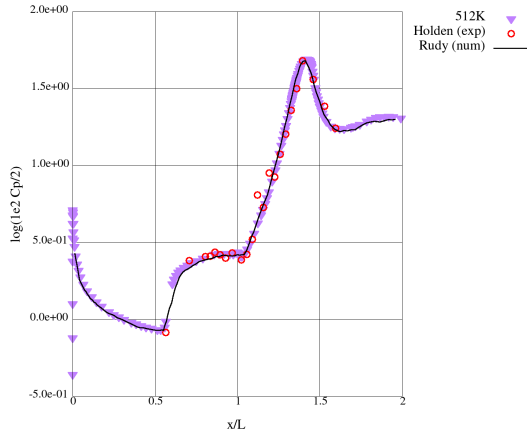
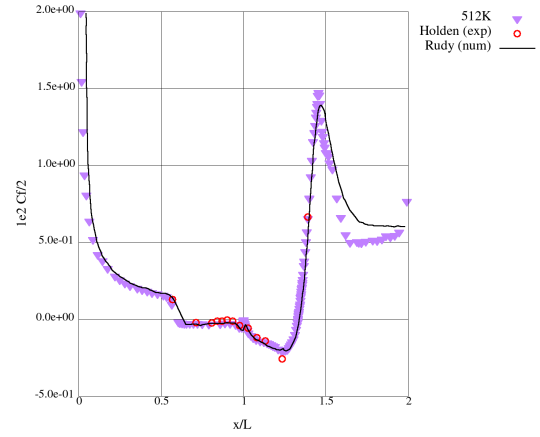


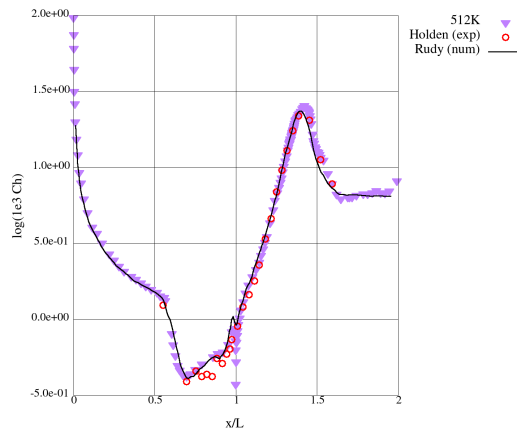
Fig 15. Wall pressure, friction and heat flux coefficients of the 24° compression ramp case. Comparison between different total number of nodes in the mesh.



(a) Wall pressure coefficient C_{p+} .



(b) Wall friction coefficient C_f .



(c) Wall heat flux coefficient C_h .

Fig 16. Wall pressure, friction and heat flux coefficients of the 24° compression ramp case. Comparison with experimental data.

Table 3. Parameters for 3D compression ramp case

Mesh size	64K	128K	256K	512K
Max. number of adaptations	8	8	8	8
Error on Lift	0.02	0.02	0.02	0.02
Error on Drag	0.004	0.004	0.002	0.002
Error on heat flux	0.05	0.05	0.02	0.02

for $x/L = 1.5$. Finally, wall heat flux coefficient C_h is well predicted as it fits the experimental data correctly.

6. Conclusion and perspectives

Computing the wall heat flux distribution is of paramount importance for reentry. Standard computational approaches require a regular, structured grid. It can be managed easily for any simple geometry, but cannot be considered for complex geometry, since the time for mesh generation becomes prohibitive.

In this context, we propose to introduce the mesh adaptation procedure for reentry. Meshes are now composed of triangles and tetrahedra, elements generally not retained to capture gradients. However, converging the solution and the mesh at the same time enables an efficient alignment of grid points and physics. We demonstrated that this approach can capture the wall heat flux, even if the mesh is strongly irregular. In addition, the goal-oriented approach based on the wall distribution of the heat flux enables a strong reduction in the required number of mesh nodes. In this case, mesh nodes are located by solving the direct and the associated adjoint problem (here based on the heat flux). To our knowledge, the paper presents the very first results of this technique for such flows.

We are continuing to work on the technique, and we plan to validate the approach by considering new 3D cases in the near future.

Acknowledgement

The present work was initiated during the master thesis of Anton Schotte and Ludovic Chan.

References

- [1] A. Dervieux, Steady Euler simulations using unstructured meshes, in: 16th VKI Lecture Series on Computational Fluid Dynamics 1985-04, Von Kármán Institute, Rhode-St.-Genèse, Belgium, 1985.
- [2] P. Rostand, B. Stoufflet, Finite volume Galerkin methods for viscous gas dynamics, Tech. Rep. RR-863 (1988).
- [3] F. Alauzet, L. Frazza, Feature-based and goal-oriented anisotropic mesh adaptation for RANS applications in aeronautics and aerospace, Journal of Computational Physics 439 (2021) 110340.
- [4] F. Alauzet, L. Frazza, D. Papadogiannis, Periodic adjoints and anisotropic mesh adaptation in rotating frame for high-fidelity RANS turbomachinery applications, Journal of Computational Physics 450 (2022) 110814.
- [5] P. Batten, B. Clarke, C. Lambert, D. Causson, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. 18 (6) (1997) 1553–1570.

- [6] B. V. Leer, Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method, *Journal of Computational Physics* 32 (1979) 109,136. doi:[https://doi.org/10.1016/0021-9991\(79\)90145-1](https://doi.org/10.1016/0021-9991(79)90145-1).
- [7] C. Debiez, A. Dervieux, Mixed-element-volume muscl methods with weak viscosity for steady and unsteady flow calculations, *Computers and Fluids* 29 (2000) 89,118. doi:[https://doi.org/10.1016/S0045-7930\(98\)00059-0](https://doi.org/10.1016/S0045-7930(98)00059-0).
- [8] S. Piperno, Schémas TVD d'ordre élevé pour la résolution de l'équation de burgers, Rapport de recherche du Cermics (1996). doi:<https://cermics.enpc.fr/cermics-rapports-recherche/1996/CERMICS-1996/CERMICS-1996-49.pdf>.
- [9] S. Piperno, S. Depeyre, Criteria for the design of limiters yielding efficient high resolution TVD schemes, *Computers and Fluids* 27 (1998) 183,197. doi:[https://doi.org/10.1016/S0045-7930\(97\)00045-5](https://doi.org/10.1016/S0045-7930(97)00045-5).
- [10] M. Park, A. Balan, W. Anderson, M. Galbraith, P. Caplan, H. Carson, M. T., J. Krakos, D. Kamenetski, A. Loseille, F. Alauzet, N. Barral, Verification of unstructured grid adaptation components, in: *AIAA Scitech 2019 Forum*, January 7-11, San Diego (Ca), AIAA Paper 2019-1723, 2019.
- [11] X. Chen, J. Fang, C. Moulinec, D. R. Emerson, A high-order hybrid numerical scheme for hypersonic flow over a blunt body, *Flow, Turbulence and Combustion* 110 (4) (2023) 799–833. doi:[10.1007/s10494-022-00389-9](https://doi.org/10.1007/s10494-022-00389-9).
URL <https://doi.org/10.1007/s10494-022-00389-9>
- [12] B. Edney, Anomalous heat transfer and pressure distributions on blunt bodies at hypersonic speeds in the presence of an impinging shock, Tech. rep., THE AERONAUTICAL RESEARCH INSTITUTE OF SWEDEN (1968).
- [13] L. F. F. Alauzet, Feature-based and goal-oriented anisotropic mesh adaptation for rans applications in aeronautics and aerospace, *Journal of Computational Physics* (2021). doi:<https://doi.org/10.1016/j.jcp.2021.110340>.
- [14] D. D'Ambrosio, Numerical prediction of laminar shock/shock interactions in hypersonic flow, *Journal of Spacecraft and Rockets* 40 (2) (2003) 153–161. arXiv:<https://doi.org/10.2514/2.3947>, doi:[10.2514/2.3947](https://doi.org/10.2514/2.3947).
URL <https://doi.org/10.2514/2.3947>
- [15] M. S. Holden, J. R. Moselle, Theoretical and experimental studies of the shock wave–boundary layer interaction on compression surfaces in hypersonic flow, Report, Aerospace Research Laboratories, Wright-Patterson AFB, Ohio (1970). doi:[10.21236/AD0706135](https://doi.org/10.21236/AD0706135).
- [16] D. H. Rudy, J. L. Thomas, A. Kumar, P. A. Gnoffo, S. R. Chakravarthy, Computation of laminar hypersonic compression-corner flows, *AIAA Journal* 29 (7) (1991) 1108–1113. arXiv:<https://doi.org/10.2514/3.10710>, doi:[10.2514/3.10710](https://doi.org/10.2514/3.10710).
URL <https://doi.org/10.2514/3.10710>
- [17] E. Chaput, F. Dubois, J. Vaudescal, D. Lemaire, G. Moules, Flu3pns - a three-dimensional thin layer and parabolized Navier-Stokes solver using the MUSCL upwind scheme, in: *29th AIAA Aerospace Sciences Meeting*, January 7-10, Reno (NV), AIAA Paper 1991-0728, 1991.