ACADÉMIE
DES SCIENCES
INSTITUT DE FRANCE

Comptes Rendus
Mécanique

Mathieu Lugrin, Baptiste Isnard, Clément Benazet, Bruno Maugars and Cédric Content

Towards an automated toolset for mesh adapted Navier–Stokes simulation of hypersonic vehicles

Volume 353 (2025), p. 1451-1475

Online since: 15 December 2025

https://doi.org/10.5802/crmeca.344

The Comptes Rendus. Mécanique *are a member of the Mersenne Center for open scientific publishing*
www.centre-mersenne.org — e-ISSN : 1873-7234

ACADÉMIE DES SCIENCES
INSTITUT DE FRANCE

Comptes Rendus. Mécanique
2025, Vol. 353, p. 1451-1475
https://doi.org/10.5802/crmeca.344

Research article

Towards an automated toolset for mesh adapted Navier–Stokes simulation of hypersonic vehicles

Mathieu Lugrin [*,a], **Baptiste Isnard** [*,a], **Clément Benazet** [b], **Bruno Maugars** [b] **and Cédric Content** [b]

[a] DAAA, ONERA, Institut Polytechnique de Paris, 92190 Meudon, France

[b] DAAA, ONERA, Institut Polytechnique de Paris, 92320 Châtillon, France

E-mail: mathieu.lugrin@onera.fr

Abstract. A fully automated CAD-to-post toolset for Navier–Stokes and Reynolds Averaged Navier–Stokes (RANS) simulations of high speed vehicles flow is presented. The toolset combines the vertex-centered SoNICS solver with anisotropic mesh adaptation based on the REFINE toolbox, eliminating manual meshing and enabling efficient, parallelized simulations from CAD input to converged solutions. The tool is validated on open cases representative of the complexity of the flow encountered around high-speed vehicles including reentry and airbreathing cruise vehicles. This includes complex three-dimensional forebody laminar simulations and axisymmetric triconic cases. For the internal aerodynamics part, given the lack of proper validation references, a direct numerical simulation of the internal conduit of a generic dual mode ramjet air intake is conducted and compared to legacy RANS simulation and mesh-adapted RANS simulation. A demonstration on a full scramjet-powered cruise vehicle then highlights the toolset's scalability and adaptability to industrial applications. The automated workflow reduces setup time and human error, making it a valuable asset for hypersonic vehicle design and optimization.

Keywords. Hypersonic, RANS, mesh adaptation.

Funding. This project was provided with computer and storage resources by GENCI thanks to the grant 2023-A0142A14106 on the SKL partition of the supercomputer Irene at TGCC. The development of the SoNICS CFD software used for this study was partly funded by the DGAC SONICE project.

Note. Article submitted by invitation.

Manuscript received 18 November 2025, accepted 4 December 2025.

1. Introduction

The design of high-speed vehicles heavily relies on Computational Fluid Dynamics (CFD), primarily through steady Navier–Stokes (NS) or Reynolds-Averaged Navier–Stokes (RANS) simulations. These simulations provide access to key performance indicators such as integrated data (e.g., lift to drag ratio, thrust, aerodynamic stability, etc.) or local quantities (heat fluxes, pressure load, etc.) that are then used to evaluate and refine the designs. Even with recent advances in

* Corresponding authors

terms of tools and solvers, setting up a CFD simulation from a CAD drawing of a system is still a time-intensive task that must be performed by a skilled engineer. While the use of unstructured meshes has significantly reduced the human time required to build medium to high quality meshes, meshing remains one of the main blocking points towards the use of CFD in automated toolchains. As the aerospace industry continues to push for more efficient design, the necessity for automated CAD to post-process workflows becomes increasingly evident. The manual CAD to mesh process, which is prone to human error and time-consuming iterations, obstructs the overall productivity of engineering teams. An integrated, automated meshing solution would significantly streamline the CFD pipeline, enabling faster design evaluation and optimization.

A possible solution for automatic CAD to post from the literature is the use of Immersed Boundary Method (IBM) [1]. The main advantage of this technique is that it does not require a body fitted mesh, and thus can use automatically generated Cartesian or octree meshes coupled with automatic masking of the area inside the solid. This method has been widely used for lattice-Boltzmann and steady or unsteady Navier–Stokes simulations, such as wall-modelled LES [2]. However, it usually suffers from multiple drawbacks. First, due to the absence of body fitted meshes, IBM simulations often rely on wall models, which have the common issue of lacking grid convergence [3] and exhibiting spurious behavior in separated regions [4]. This last drawback may be problematic for high-speed vehicle evaluation, particularly for cruise vehicles, where the air intake uses multiple shock-boundary layer interactions to slow down the flow and often reaches a fully separated state. Getting precise laminar baseflow using wall-model can also be challenging. Then, the use of body forces inherently leads to a non-conservative solution, which can again be a major problem for internal aerodynamics, as the mass imbalance may cause a wrong evaluation of the engine performance.

Another very promising solution to alleviate the manual meshing problems without the drawbacks of IBM is anisotropic mesh adaptation. It provides good convergence of the solution even with no physical intuition nor knowledge in the starting mesh, and do not suffer from conservativity or spurious behavior in separated regions, given that it remains resolved and body fitted. Anisotropic mesh adaptation is known for its high efficacy in the automatic generation of optimal computational meshes, first within the context of inviscid flow simulations [5–8] and more recently for RANS simulations [9,10]. At its heart, mesh adaptation [11–15] involves an iterative process: utilising data from a preceding simulation to automatically dictate the necessary mesh adjustments required to enhance either the global solution accuracy or a particular quantity of interest. This is achieved via solution-based sensors that analyse the current results and subsequently define modifications to the mesh. The goal of these prescribed modifications is to balance and optimise the element distribution — and, consequently, the computational resources — throughout the domain [16–18]. Regions characterised by significant gradients in the solution, or those deemed important for the computation of a specific output, are selectively and automatically refined to increase the local accuracy of the discretization. The validity and optimality of this methodology are underpinned by a formal mathematical analysis of the associated numerical error [19].

However, most toolchains relying on mesh adaptation are still not automated CAD-to-post workflows, as the tool often requires manually created surface and initial mesh of the model as well as the tagging of the boundary conditions. They are also often relatively inefficient, even if the mesh adaptation by itself allows for more efficient simulation due to the lower amount of points needed to converge a solution, as they chain tools through Input/Output (I/O, here used to refer to the writing and reading of the solution on a physical disk space) and different executables. Most of the tools are also not built to work in a parallel or distributed manner, which strongly limits performance and, worse, the maximal number of points that can be computed, as one may reach the memory limit of the computing nodes.

In this context, this paper presents a fully automated, parallel distributed CAD-to-post toolset for NS and RANS simulation based on anisotropic mesh adaptation, with a particular focus on the demonstration of its ability to capture the flow around high-speed vehicles.

While the tool has already been extensively validated during development using non-regression tests from the elsA solver, as well as open literature databases such as the NASA Turbulence Modeling Resource, conducting simulations of high-speed vehicles introduces new complexities that are specific to hypersonics and may not be covered by the usual CFD code validation. The tool must be able to handle highly compressible flow, including strong shocks, shock-shock, and shock-boundary layer interactions. This may also be challenging in terms of mesh adaptation, given that metric based adaptation will tend to refine regions with the strongest gradients (i.e. shocks) while overlooking other physically important regions of the flow, such as boundary and mixing layers.

To assess the ability of the tool to capture the flow around high-speed vehicles, it is tested on open cases representative of the complexity of the flow encountered in real-life applications, namely: a canonical reentry vehicle, a complex 3D forebody, and a supersonic internal air intake. Finally, a demonstration on a mock-up full cruise vehicle is conducted.

The article is organised as follows. First, the Navier–Stokes solver, the re-mesher, and the way they are linked together are presented. The manner in which the initial geometry, first mesh generation, and boundary condition tagging are handled is also highlighted, given that they are of great importance for the automation process. A new treatment of the metric used for adaptation and CAD handling is also presented. The tool is then validated on different open cases from the literature, representative of the complexity of real life applications.

The first case of interest is a canonical reentry vehicle that has been widely studied both numerically and experimentally [20,21]. It consists of an axisymmetric cone-cylinder-flare geometry, which offers numerous advantages. First, it allows for the comparison of the results and performances of the toolset on a bi-dimensional case against well established solvers from the literature. Even if geometrically simple, this case includes most of the complexity of the flows encountered around reentry vehicles.

The second and third cases of interest relate to airbreathing cruise vehicles. First, an external aerodynamic case is needed to validate that the tool can handle hypersonic forebodies, potentially with complex flow features such as curved shocks, the linked vortices, and high-Reynolds number laminar boundary layer. One main point of interest here will be the ability of the tool to capture accurate wall heat fluxes, which is both a challenging quantity to obtain numerically and one that is of paramount importance for the design of vehicles, as it will constrain the thermal protection system. Here, the BOLT case is chosen as a reference given that it provides geometrical and physical complexity close to that of real vehicles while being open and widely studied in the literature, with experimental (both ground [22,23] and flight [24,25]) and numerical results [26]. Specifically, a very challenging numerical reproduction of the highest laminar flight Reynolds number of the BOLT II campaign is conducted and compared to numerical results from [26].

The last validation case of interest relates to internal aerodynamics linked with ramjet propulsion. To get a properly functioning ramjet, one needs to slow down and compress the incoming supersonic flow to reach a subsonic state inside the engine. This can be done using both external and internal compression apparatus. Using a series of shock-waves, the flow is slowed down to feed the combustion chamber. To properly design the engine, one needs to have precise data about the flow that will enter the combustion chamber. Being able to easily and accurately compute such flows is thus mandatory to correctly predict vehicle trajectories and validate designs. For this case, the objective is to demonstrate that the tool can reliably reproduce the key internal flow features, such as corner vortices, shock boundary layer interaction (SBLI) and shock-train evolution. Even if some canonical cases of shock trains are available in the literature [27,28], they

are too far from applicative needs to be used in the scope of this paper. As such, a new validation case was built from scratch to reproduce all the flow features of interest. To establish a reference, a DNS of the benchmark configuration is performed, providing high-fidelity data against which results from the tool is compared. Given that the tool relies on turbulence modelling to capture such flow, a "legacy" RANS simulation (conducted with the elsA [29] well established solver) was conducted with the same models in order to discriminate discrepancies between the DNS and the RANS simulations caused by the turbulence modelling from those that may be caused by an ill behaviour of the toolset. Finally, once the ability of the tool to capture both complex internal and external flow has been proven, a demonstration on a full mock-up vehicle (with a modelled scramjet engine) is presented as a demonstration of the possible use of the tool.

2. Toolset description

The toolchain (presented in Figure 1) is built based on two main tools, namely the vertex-centered Navier–Stokes solver SoNICS from ONERA–Safran [30] briefly described hereafter and an anisotropic mesh adaptation toolbox adapted from the open-source toolbox REFINE developed by NASA [31]. Every step that will be described in the following is linked through memory APIs. This means that once the CAD file have be read by process, no I/O has to be conducted (except for post-treatment needs). Also, all following steps are executed in a parallel context, but parallel management is completely transparent to the user.

2.1. *CFD Solver*

SoNICS is a CFD software that solves Euler, Navier Stokes or RANS equations using either cell-centered or vertex-centered second-order finite volume schemes. It can perform multi-species reactive flows simulations. It has been designed with four major requirements: it must be easy to configure and must be able to take advantage of powerful hardware (CPU and GPU), which are all complex, distributed and heterogeneous [32]; it should handle a large variability in terms of Riemann solvers [33–35], flux limiters [36–38], fluid models (equations of state, viscosity laws, mixing laws...), turbulence modelling (RANS, LES, DES); it should be fully differentiable in order to allow modern optimization algorithms [39], input-output analysis [40–45], multi-disciplinary optimization [46,47], data assimilation [48–51], end-to-end machine learning [52]; and finally, it should provide APIs to help couple the tool with other solvers. A full description of the architecture in use for the SoNICS solver can be found in the companion paper by Lienhardt and Michel [53].

As mentioned above, the vertex-centered solver of SoNICS is used. The spatial discretization is based on the one proposed by A. Dervieux and C. Debiez [54] on simplex mesh (triangle/tetrahedron). The convective terms are solved by the finite volume method on the dual mesh composed of median cells. Fluxes are determined through the use of the AUPM approximate Riemann solver proposed by Chen et al. [34]. Second order space accuracy is achieved through MUSCL type reconstruction method [55]. The MUSCL procedure is applied as proposed in A. Dervieux and C. Debiez [54] or Alauzet and Frazza [56] to recover the so-called V4-scheme. A shock capturing similar to the one of [57] is also implemented. Viscous fluxes are computed with the conventional expression described as Cell Based Viscous scheme in Liu et al. [58].

Boundary conditions are weakly imposed by evaluating the flux at each facet at the boundaries of the domain. As in Alauzet and Frazza [56], each Dirichlet boundary (such as the velocities on a no-slip wall) are strongly enforced at each iteration, i.e. the DOF concerned by the boundary are removed from the whole CFD problem.

In order to reach the steady state, we consider an implicit backward Euler scheme with local time stepping, approximately solved by an efficient LU-SGS solver similar to the one proposed in [59]. For RANS simulation, the Spalart–Allmaras (SA) [60] model with the compressibility correction of [61] and the Quadratic Constitutive Relation (QCR) [62] are used.

2.2. *Anisotropic mesh adaptation on full tetrahedron meshes workflow*

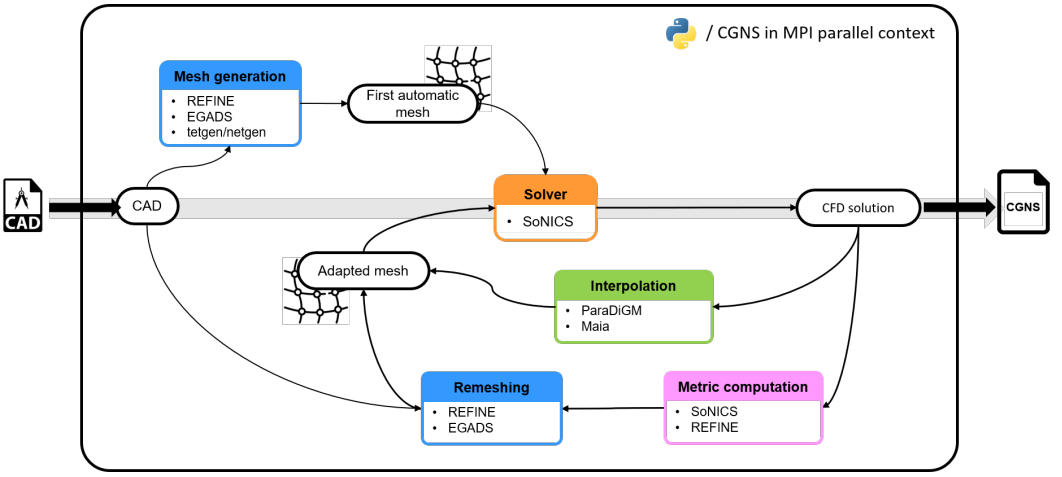


Figure 1. Diagram of the toolchain used for the automated computations.

The automated toolchain for mesh adaptation is presented in Figure 1. The input of the process is a CAD description of a closed shell representing the fluid volume to simulate. The first step is then to prove the existence of the mesh. For this purpose, REFINE provides an EGADS [63] wrapper to generate a surface mesh of the CAD shell. One can note that for the applications presented here, the CAD model is provided as a STEP file, which allows interaction with any CAD modeling software; however, a direct link with ESP [64] could also be utilized to take advantage of its differentiation. From the STEP input, EGADS tessellates the CAD edges and surfaces to build a closed surface mesh. Tesselation provides meshes whose elements have poor quality for a volume mesher, especially given that each CAD patch is meshed independently. As such, the tesselation process results in an overall low-quality surface mesh. In order to improve its quality before providing it to the tetrahedral mesher, REFINE performs surface mesh adaptation driven by a curvature-based metric that is evaluated thanks to the CAD parametrisation at each mesh node. Several passes can be performed with various parameters (minimal and maximal cell size, mesh complexity, size gradation) in order to obtain a decent surface mesh before volume mesh generation. Once done, a tetrahedral mesh generator (here tetgen [65] or netgen [66]) is used to mesh the interior of the fluid volume. This first mesh generation is illustrated in Figure 2.

The mesh structure — including coordinates, connectivity, and CAD information — is converted into a Python/CGNS structure [67] (CGNS tree), which serves as the entry point for the SoNICS solver. The CGNS tree data is natively distributed over MPI processes following Maia's parallel rules for CGNS trees (see [68,69]). It will remain distributed for the rest of the workflow.

In industrial configurations, the number of ridge and surface patches can be exceedingly large, which implies some extra work for the user to retrieve link between CAD patches and the CFD boundary conditions (BCs). In order to relieve the user of this step, if CAD patches are

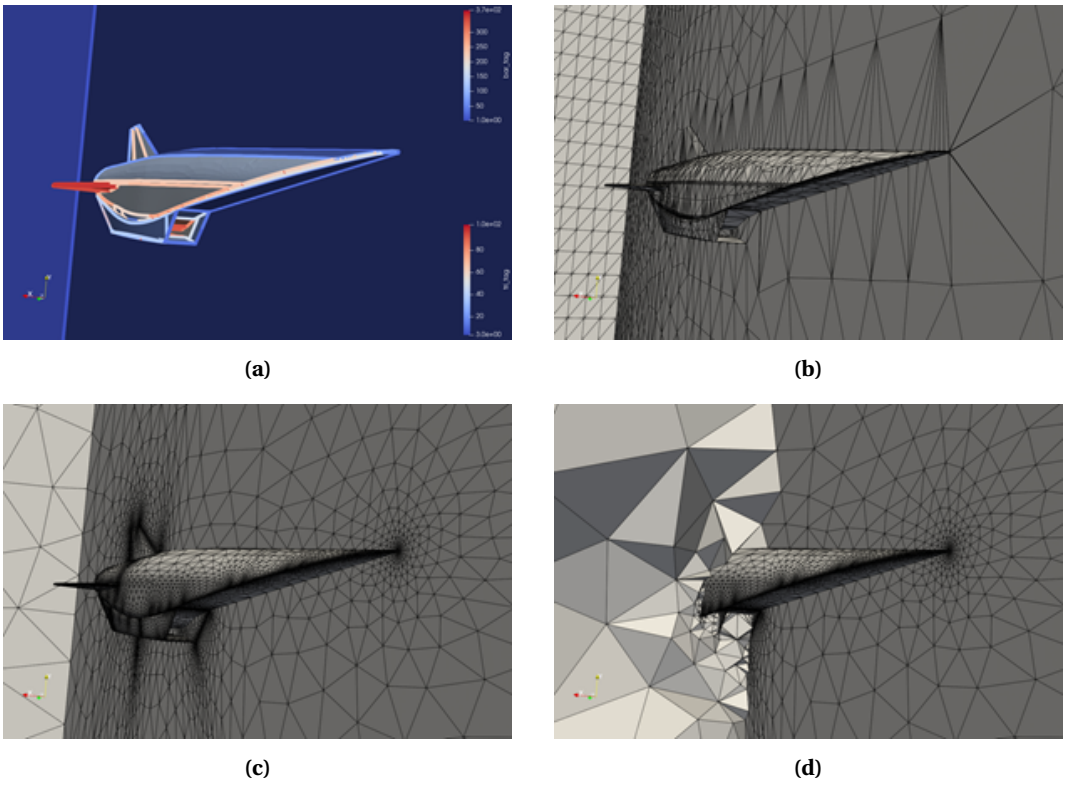


Figure 2. Automated first mesh generation process illustrated by: (a) CAD model; (b) the EGADS tesselation of the CAD model; (c) the surface mesh after remeshing by REFINE; (d) volume mesh created by tetgen.

tagged with names using the advanced face option of STEP format, the toolchain preserves these names during the mesh generation process, so boundaries can easily be retrieved for the solver boundary condition tagging. Patches from the same families can also be concatenated together in a single CGNS node to ease I/O and tree handling.

The Python/CGNS tree is then prepared by adding a family of boundary conditions (BCs) (automatically, thanks to the preservation of patch names), the reference state, and the solver options. Steady compressible Navier–Stokes or RANS simulations are then conducted using the vertex-centered solver of SoNICS described here-before. Once a solution is found, the computation of the mesh adaptation metric is performed directly in the CFD solver and is fed to the remesher. In this work, the metric is computed as the absolute value of the Hessian of the Mach number (albeit the user can decide to use any field). A pre-processing is applied by REFINE to compute the error estimate for the desired p-norm and complexity. Then an anisotropic size gradation algorithm is used to smooth the metric. REFINE then performs a parallel remeshing of the surface and volume meshes. The geometric description of the fluid domain is preserved with high fidelity using EGADS's CAD projection when inserting or moving boundary mesh vertices. The current flow solution is finally interpolated on the new adapted mesh using the open-source ParaDiGM library [70] (see also [71]), which is available through the Python/CGNS API of Maia. The interpolated solution is used as an initial state and the simulation is run again. The process is looped, potentially with an increase in complexity depending on the steps, until a proper solution is converged. Convergence indicators are case-dependent and are to be included in the Python

script directly by the person skilled in the art. The same goes for complexity jumps, gradation, and most of the CFD solver parameters which are also application-dependent.

The whole procedure that is described above is performed by a single Python script that takes as inputs the STEP file and the run conditions. All pre/co/post-processing, partitioning and distribution are performed using the open-source Maia library.

2.3. *Convergence*

Given that the first generated mesh is very rough and does not include any physical intuition, the first NS or RANS computation at the beginning of the loop can be numerically challenging. The usual way to start the mesh convergence process is to run the solver computation over a few pseudo-time iterations to avoid solver crashes. The number of iterations can then be slowly increased over remeshing passes to gradually converge the proper solution. A trick allows to accelerate this solution convergence process: the Mach Hessian used as a metric for feature-based remeshing is a good criterion for mesh convergence of Navier–Stokes equations resolution [72]. However, if used as is, it can lead to slow mesh convergence because it is not very efficient at quickly capturing the boundary layer. Especially for flows including shocks, in the first steps where meshes have poor discretization, the chosen metric highlights the very strong gradients in compression regions compared to viscous regions. Frazza [9] proposed to boost the Hessian near viscous walls in order to add more points into the boundary layer, which will help the process converge quickly to a steady flow solution. For this purpose, metric diagonal is boosted by a factor in the wall normal direction.

3. Reentry vehicles aerodynamics

As stated in the introduction, the first type of high-speed vehicles of interest are the reentry vehicles. To assess the ability of the solver to treat those cases, the axisymmetric cone-cylinder-flare CCF12 geometry, which was designed as a canonical and open case to study the flow features found in atmospheric reentry vehicles is chosen as a benchmark. It includes a simple developing hypersonic laminar boundary layer followed by more complex features such as boundary layer interaction with expansion fans, and most importantly, separation induced by the shock-boundary layer interaction. The numerical results of Caillaud et al. [20] (see Table 1) are used as a reference given that they provide high-order solutions cross-validated between two well established solvers from ONERA [73] and NASA [74]. For this case, the axisymmetric solver of SoNICS is used alongside the 2D remesher of REFINE, in this case the mesh is only composed of triangles.

In that specific 2D axisymmetric case, the CAD file is generated from the analytical description of the body using GMSH and exported as a STEP file. An example of an adapted mesh on that case, alongside a Mach number contour, is presented in Figure 3.

Taking the low Reynolds case as an example, the toolset is run for 121 seconds on 4 cores of an Intel Cascade Lake CPU, allowing us to conduct 6 mesh adaptations and steady computations

Table 1. Flow conditions from [20] used in this study.

$\mathrm{Re}_m \times 10^6$ (m^{-1})	R_n (mm)	u_∞ (m/s)	ρ_∞ ($\mathrm{kg/m}^3$)	T_∞ (K)	T_0 (K)	T_{wall} (K)	P_0 (kPa)
2.229	0.1	907.92	0.00920	56.977	467.21	300	237.50
4.614	0.1	990.41	0.02128	67.800	555.96	300	653.90

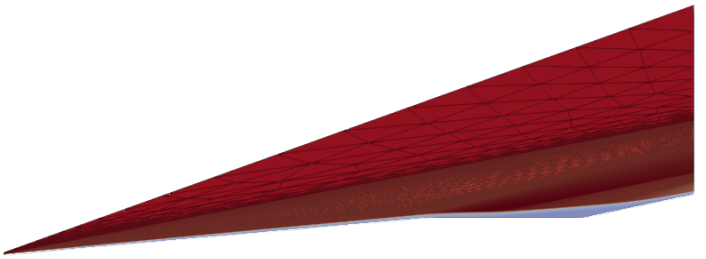


Figure 3. Adapted mesh and Mach number contour for the low Reynolds number case of [20].

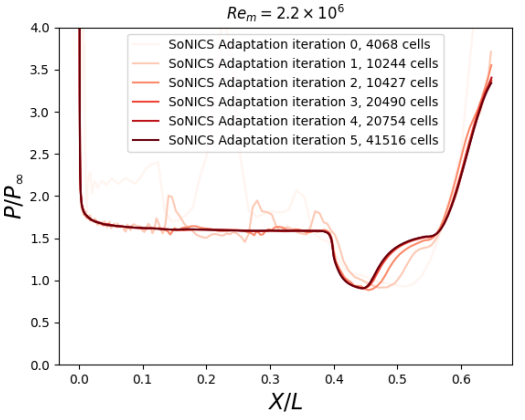


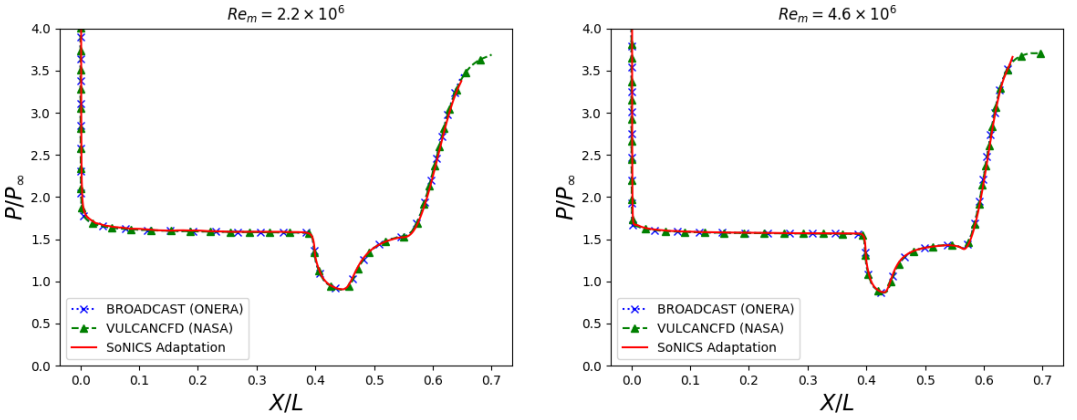
Figure 4. Evolution of the wall pressure distribution on the CCF12 at low Reynolds number at each iteration showing the convergence path.

until complete convergence is reached, the evolution of the wall pressure distribution for each step is presented in Figure 4.

The wall pressure distribution for two different Reynolds numbers is presented in Figure 5 alongside results from two high-order solvers. Given the sensitivity of the strongly separated shock boundary layer interaction to both the resolution of the boundary layer gradients and the associated separation and reattachment shock, obtaining the correct pressure signature guarantees that these features are accurately captured in the solution. This result validates the good behaviour of the toolset for hypersonic axisymmetric configurations.

4. Cruise vehicles aerodynamics

The second type of vehicle of interest is air-breathing cruise vehicles, which, in addition to external aerodynamics, involve complex internal flow. This validation is split into two parts: first, a 3D forebody case is studied; then, a canonical internal aerodynamic reference case is built and used as a reference to validate the toolset.



(a) Low Reynolds case

(b) Intermediate Reynolds number case

Figure 5. Wall pressure distribution for the two different Reynolds number alongside reference data from [20].

4.1. *Forebody*

The first subcase of interest for cruise vehicles concerns forebodies, which are often complex 3D geometries on which the main points of interest will be drag, lift and heat-flux prediction. The BOLT case is chosen as it provides plenty of open data (including flight) and its 3D geometry is representative of the complexity of real designs. It also displays an intricate baseflow structure with multiple vortices, which is challenging to capture and is thus a good benchmark. The chosen condition for the computation is a challenging high Reynolds case with both CFD and sparse flight data coming from [26]. This case is at a flight altitude where laminar-turbulent transition starts to appear, which means that it is the highest Reynolds number for which the flow remains laminar; the full flow conditions are presented in Table 2. Given the high Mach number and the relatively low altitude, this case is fully representative of a scale one hypersonic cruise vehicle forebody in flight. This case would also serve future validation of transition prediction tools integrated to the solver.

The case is run on a quarter of the full scale flight BOLT II model using symmetry planes. The computation is conducted on 2400 Intel Cascade Lake cores for a single job of 15 hours, allowing for 8 adaptation/computation steps and achieving a final mesh of 61 million nodes in order to reach proper convergence of the wall heat fluxes. This can be compared to the 110M cells used in the study of [26], keeping in mind that the solution on the structured grid is computed using a sixth order central finite difference scheme, while the mesh adapted solution uses a second order method. This gives insight on the ability of mesh adaptation to strongly reduce the required number of DOF to achieve the same convergence. This could be even more drastic when using goal-orientated adaptation (on the parietal heat-flux) [75], given that the convergence of

Table 2. Flow conditions from [26] used for the BOLT forebody reference, data coming from the BOLT-II flight.

Time (s)	$\mathrm{Re}_\infty \times 10^6$ (m^{-1})	Altitude (m)	u_∞ (m/s)	ρ_∞ (kg/m^3)	T_∞ (K)	Mach	T_{wall} (K)
401.9	6.0914	23908	1768.5	0.048302	213.15	6.0424	390

wall heat-fluxes requires a lot of degrees of freedom, especially with meshes composed only of tetrahedron and using metric based adaptation.

An illustration of the geometry, alongside the wall heat-flux distribution and a Mach number field at the outlet of the domain, is presented in Figure 6. It highlights the complex nature of the flowfield with multiple vortical structures appearing, especially in the center region. Quantitative comparisons of heat-fluxes at two different longitudinal positions with the CFD solution of [26] are presented in Figure 7. Overall, excellent agreement is found between the two simulations. Results from the mesh adaptation display a relatively good smoothness given that they come from anisotropic tetrahedron only meshes. Some discrepancies are still visible in the peaks linked with local vortices, mostly in terms of amplitude of the fluctuation. Figure 7(a) presents the most upstream slice, from left to right, the relatively low heat-flux region is linked with the upwash of the centerline vortices (caused by the 3D nature of the geometry and thus of the bow shock which leads to the creation of vorticity), those vortices are clearly visible in Figure 6. This region is then followed by a first peak linked with the downwash of those vortices and then a long plateau of relatively high heat flux linked with the thin boundary layer in the outboard region. In all those regions, the two computations are in almost perfect agreement.

The origin of these discrepancies is still unknown; they may be linked to differences in the mesh resolution; some vortices may be under-resolved in either simulation. Another possible origin could be the growth of vortices linked to numerical errors from the time integrators, as the baseflow is extremely unstable; small numerical errors could lead to the growth of structures that are not part of the baseflow. Future work on that case will require additional simulations from other meshes and/or solvers to assess the origin of these discrepancies. Again, a Newton method could be used once the Jacobian operator is available to compute a fixed point of the equations.

Overall, the tool appears to be fully capable of computing a very complex baseflow such as the one encountered on the BOLT forebody. The key point to highlight here is that the user time needed to set up such a computation is minimal (of the order of minutes) with no effort whatsoever put into meshing, which strongly contrasts with other studies on that case.

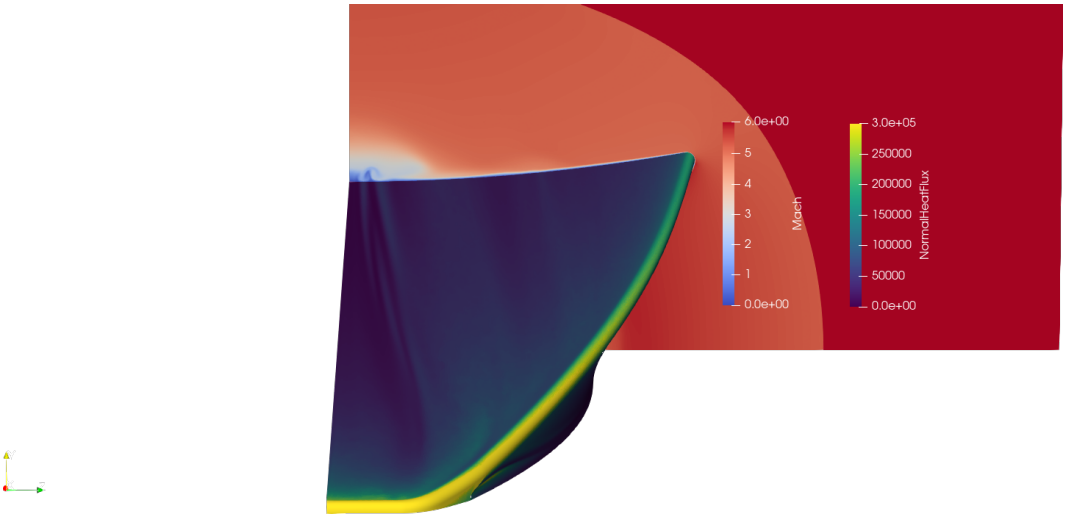


Figure 6. BOLT geometry and simulation results (wall heat flux and Mach number field at the outlet of the simulation domain) for the descent case of [26].

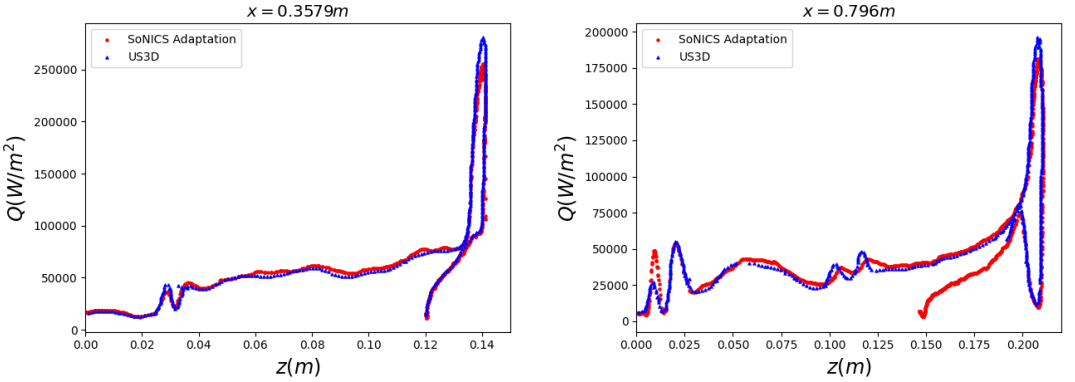


Figure 7. Spanwise heat-fluxes distribution at two longitudinal sections of the BOLT forebody alongside data from the US3D [76] computation of [26].

4.2. *Internal aerodynamics case*

As explained in the introduction, given the lack of open cases representative of applied vehicle technologies, a reference case is built to validate the toolset on hypersonic internal aerodynamics. The motivation for the geometry and flow conditions choices as well as the setup for the reference simulation is presented in this section, followed by comparison between reference, legacy RANS and automatic toolset results.

4.2.1. *Geometry and flow conditions*

The geometry under study is designed to be representative of the complexity of the internal flow of a ramjet powered vehicle. As such, it should include an internal compression region, followed by a throat, and then a slowly diverging section in which a shock train will appear. A back pressure at the outlet of the diverging section is necessary to simulate the engine. To facilitate the sharing and accurate reproduction by other researchers, the geometry must be canonical and analytically describable. This led to the geometry described in Appendix A. The dimensions of the geometry, while arbitrary, are chosen such that reproductions in conventional supersonic wind tunnels are facilitated. The flow conditions are chosen to match a reasonable value for most supersonic wind tunnels to allow for further cross comparison.

The Reynolds number is chosen as high as possible, while still allowing for a highly resolved computation within the given computing allocation. The inlet Mach number is set to 3, the stagnation temperature is low (300 K) to facilitate experimental reproduction, and the static pressure of the inlet is chosen to adjust the Reynolds number and thus the computational cost. The complete conditions, as well as the reference values used for the computations, are presented in Table 3.

Table 3. Reference conditions for the computations.

M_∞	P_∞	T_i	α	β	T_{wall}	Re_L	L
3	4000 Pa	300 K	0°	0°	300 K	4.029×10^6	0.4 m

The boundary conditions are supersonic inlet and outlet, and isothermal viscous wall at 300 K. The turbulence generation technique is described in Appendix C. The chosen conditions lead

to the approximate Reynolds number presented in Table 4, θ and δ_1 being the boundary layer momentum and displacement thickness, respectively.

Table 4. Approximate boundary layer thickness and boundary layer based Reynolds number in the central plane $z = 0$.

	Re_θ	δ_1
Before first ramp ($x = 0$)	1500	0.00077 m
At bottom wall separation point (see Figure 9)	6875	0.00169 m

4.2.2. *Reference numerical simulation setup*

In order to conduct the DNS and the legacy RANS simulations, a structured mesh is created. The characteristic sizes of the meshes are presented in Table 5, the wall units sizing are computed from an initial SA QCR RANS computation, and the maxima are given compared to the local wall unit of the closest wall, ensuring good resolution in the entire domain. Stream-wise cell sizes are adapted using linear laws, whereas wall-normal ones are using a collection of geometrical laws. The final DNS mesh contains more than 3.5 billion points without ghost cells. The RANS mesh uses the same construction, but with much less resolution.

Table 5. Mesh characteristics.

	n_x	n_y	n_z	n_{tot}	$\Delta x^+_{\mathrm{max}}$	$\Delta y, z^+_{\mathrm{max}}$	$\Delta y, z^+_{\mathrm{wall}}$
DNS	9807	605	605	3.59×10^9	< 12	< 10	< 1.4
RANS	442	195	195	16.8×10^6	< 300	< 100	< 1.4

The DNS (see Figure 8) is performed using the open-source HPC oriented FAST compressible Navier–Stokes solver from ONERA [77]. The solver uses a finite volume approach on multi-block structured mesh using hybrid MPI/OpenMP parallel programming, cache blocking, and vectorization to improve performance. Inviscid fluxes are solved using an extended version of the scheme described in [78], details about the spatial scheme are given in Appendix B. The viscous fluxes are computed using a second-order accurate centered scheme. Time integration is performed via an explicit low-storage third-order 3-step Runge–Kutta scheme; the time step is set to $\Delta t = 3 \times 10^{-9}$ s to ensure a CFL number lower than 0.5 in the whole domain. The molecular viscosity is computed using Sutherland's law. The computation was run using a hybrid MPI/OpenMP parallelism on 64 computing nodes (3072 cores), using 1024 MPI processes and 3 OpenMP threads per process. The mean flow is averaged over 2×10^6 iterations, which corresponds to a physical time of 6 ms.

The legacy RANS simulations are performed using the elsA software (Safran–ONERA property [29]) on the structured mesh presented before (see Table 5). The inviscid fluxes are computed using the AUSM+up of Liou [79] and an unlimited third-order MUSCL reconstruction. The viscous fluxes are solved using a centered scheme on five points, and the time integration to find the steady solution is done implicitly with a local time-stepping approach. In order to obtain the correct solution, the simulation is initialized with a subsonic region in a subpart of the diverging section and free-stream values elsewhere, the same initialisation strategy is used for the mesh adapted simulation.

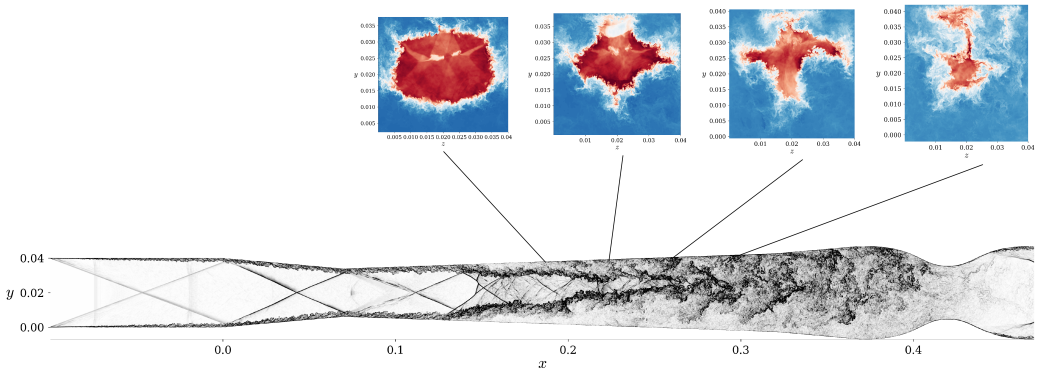


Figure 8. Illustration of the simulation thanks to instantaneous numerical pseudo-schlieren in a constant z-plane ($z = 0.01$) of the configuration and instantaneous stagnation pressure contour in various x-plane.

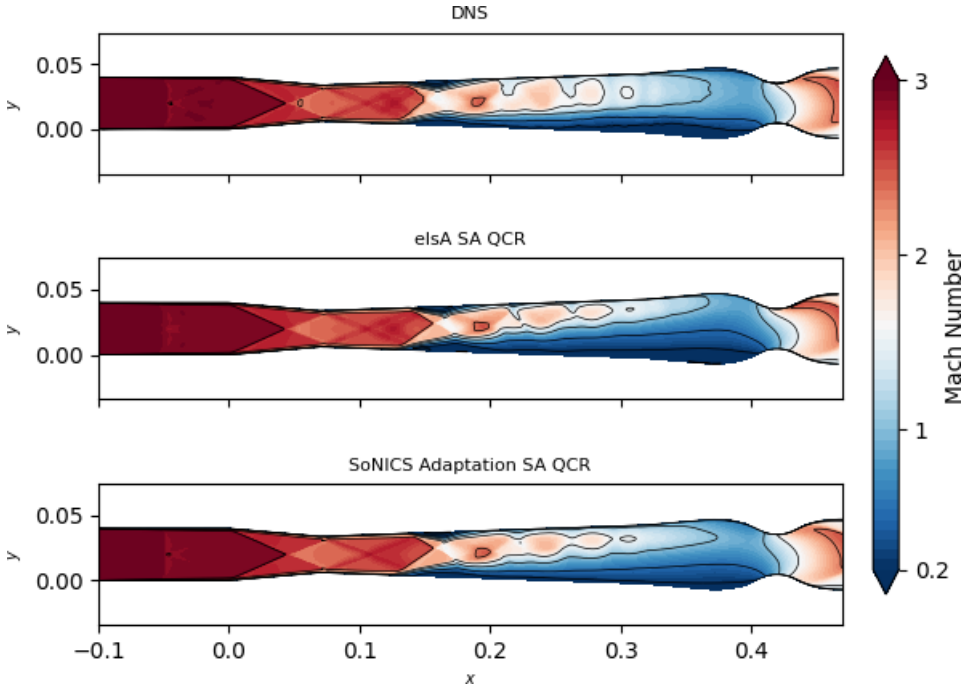


Figure 9. Mach number distribution in the central z-plane. The DNS results correspond to time-averaged flow.

4.2.3. *Results*

For this case, given the slow time convergence of internal flow, the tool is run for 33 adaptation steps with a varying number of time iterations, to reach a final mesh of 4 million nodes (to be compared to the 17 million cells of the legacy setup). The time-averaged Mach number fields (in the central plane) for the DNS and both RANS simulations are presented in Figure 9. The

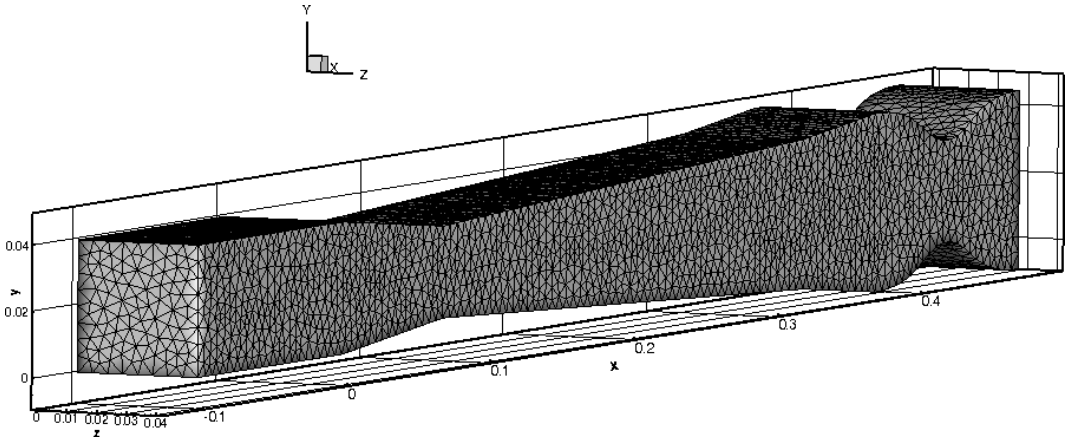


Figure 10. Starting mesh for the mesh adapted RANS simulation.

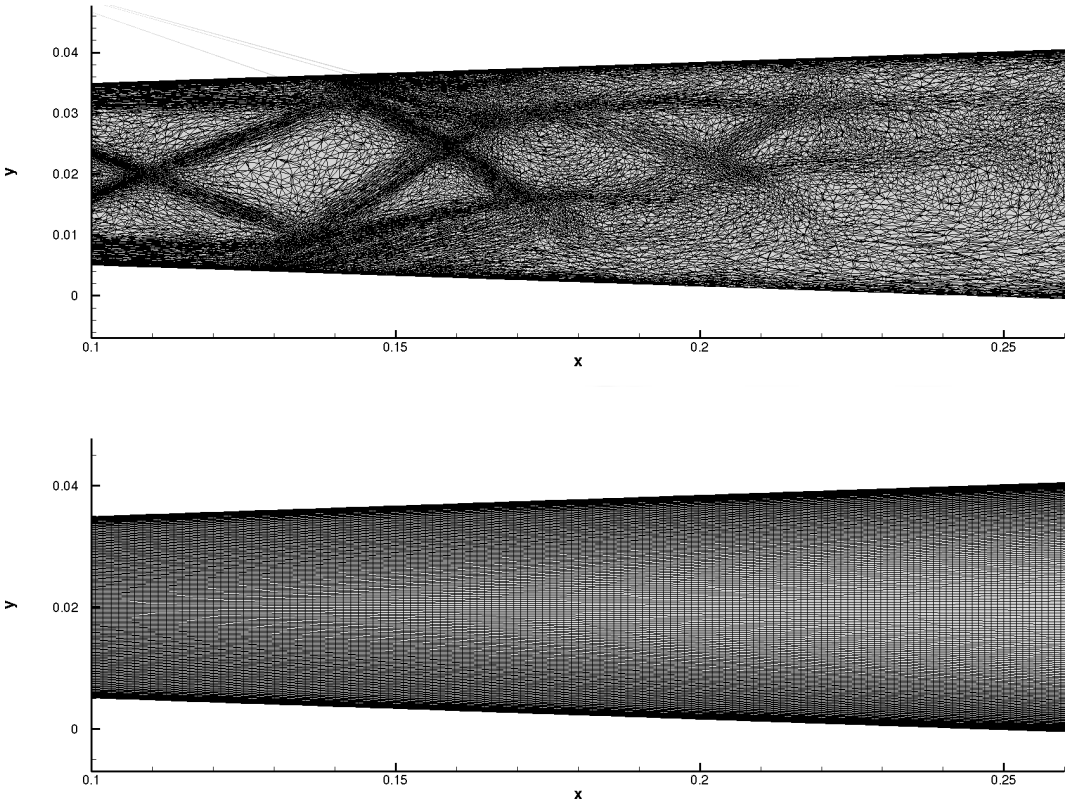


Figure 11. Meshes from the adapted (top) and elsA (bottom) SA QCR simulations in the central plane for the shock-train region.

flow displays all the main expected features: first, a boundary layer develops, then encounters a first shock wave as it reaches the compression ramp, followed by an expansion at the entrance of the diffuser. The flow ends up separating via a SBLI and is slowed down by a shock train to finally reach a purely subsonic state. Both RANS simulations display results that are surprisingly

close to the DNS reference, given the complexity of the flow (one can note here that the use of the QCR strongly improves the quality of the RANS solution, as described in Appendix C). From an engineering standpoint, both methods are acceptable for vehicle design. This validates the good behavior of the toolset for turbulent internal aerodynamics. Upon closer examination of the solutions, it is evident that the elsA results show a shock train located closer to the top wall than in the mesh-adapted simulation. Consequently, the latter is in better agreement with the DNS reference. This may be due to the traditional meshing procedure used for the standard simulation, where the mesh is highly refined near the walls to properly capture the boundary layer, leading to a mesh-induced displacement of the shock train. As the refinement of the mesh-adapted simulation is fully feature-driven, no such best practice is used (see the initial mesh presented in Figure 10), which leads to a better resolution of the mesh further away from the wall. The difference between both final meshes in the central plane for the shock train region is displayed in Figure 11. The adapted mesh clearly displays a refined region further away from the wall in addition to the near-wall refinement for the boundary layer, which may allow a better capture of the physics of the flow.

5. Demonstration on a full vehicle

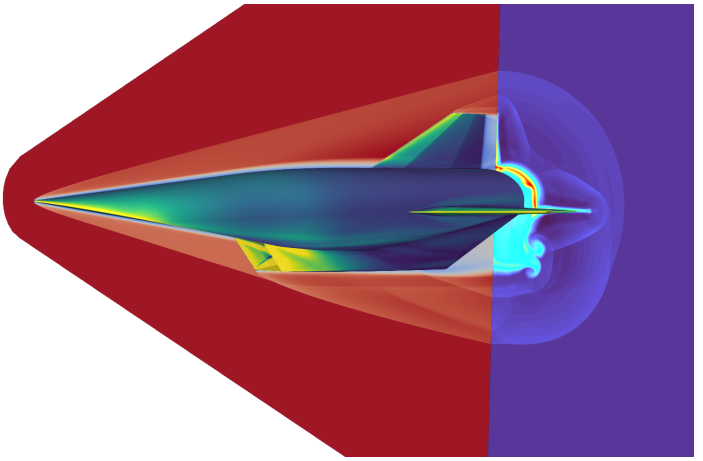


Figure 12. Skin-friction, Mach number slice in the central plane and temperature slice at the outlet illustrating the external aerodynamics of the full vehicle.

Now that the ability of the toolset to capture both complex internal and external high speed aerodynamic flowfield has been documented, a demonstration on a full mock-up vehicle can be conducted. The case consists of a high Mach number cruise of a scramjet powered vehicle. The combustion chamber is not computed but replaced by a 0D model extending from the air intake throat to the nozzle entrance. This simulation only serves as a demonstration of how we can leverage the ability of the toolset to handle complex geometry at no additional cost. Setup time for such a simulation requires only few minutes of human work. The toolset is then run on 600 Intel Cascade Lake cores for a single job of 15 hours, which allowed conducting 12 adaptation/simulation steps to reach convergence of the aero-propulsive balance, the final mesh has 35 million nodes. Samples of results are presented in Figures 12 and 13. Although quantitative data cannot be presented here, the objective is to demonstrate the tool's ability to manage complex geometries (e.g., control surfaces and bleed systems). A key advantage is that

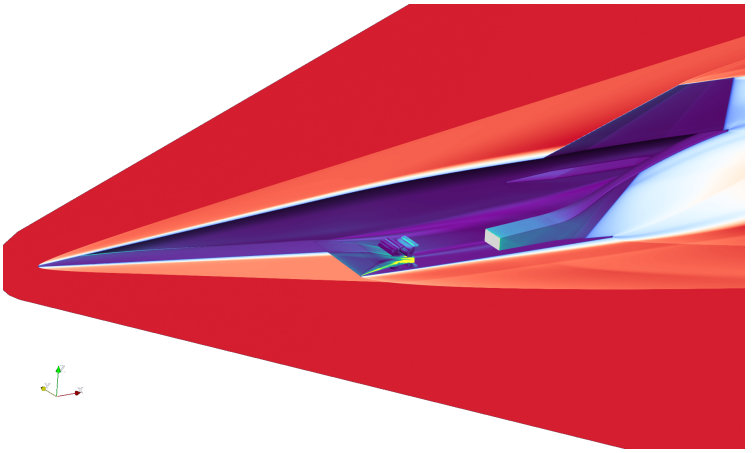


Figure 13. Skin pressure distribution and Mach number slice in the central plane illustrating the internal aerodynamics of the full vehicle.

geometric variability is fully transparent to the user, for example, the same simulation can be performed with different or additional control surfaces simply by modifying the input CAD file.

6. Conclusion

This study presents the development and validation of a fully automated, CAD-to-post toolset designed for Navier–Stokes and Reynolds-Averaged Navier–Stokes (RANS) simulations of complex hypersonic vehicles. The toolset integrates a 2D, 2D-axi and 3D vertex-centered finite volume Navier–Stokes solver, SoNICS, with an anisotropic mesh adaptation framework based on the REFINE toolbox. By leveraging automated CAD handling, surface and volume meshing, and metric-based remeshing, the toolset eliminates the need for manual intervention in the meshing process, significantly reducing setup time and human error. The workflow is entirely parallelized and managed through a Python/CGNS interface, ensuring scalability and efficiency on modern high-performance computing architectures. The toolset was rigorously validated using a series of open benchmark cases representative of the complex flow physics encountered in hypersonic applications. First, the axisymmetric cone-cylinder-flare geometry, a canonical reentry vehicle configuration, demonstrated the toolset's ability to accurately capture shock-boundary layer interactions and flow separation, with results closely matching high-order reference solutions. Second, the BOLT forebody case, characterized by its three-dimensional geometry and intricate vortical structures, highlighted the toolset's robustness in predicting wall heat fluxes and complex laminar boundary layers at high Reynolds numbers. The comparison with experimental and numerical data from the literature confirmed the toolset's reliability in resolving challenging external aerodynamic features. For internal aerodynamics, a novel validation case was designed to replicate the flow conditions within a ramjet-powered vehicle's air intake. This case included a compression ramp, diffuser, and shock-train formation, all of which are critical for engine performance evaluation. The toolset's results were benchmarked against a Direct Numerical Simulation (DNS) and a legacy RANS simulation using the elsA solver. The comparison revealed that the mesh-adapted RANS simulations, particularly when using the Quadratic Constitutive Relation (QCR) turbulence model, provided results remarkably close to the DNS reference. This underscores the toolset's capability to capture key flow features such as shock-boundary layer interactions, corner vortices, and shock-train evolution, even in highly turbulent and separated flow

regimes. Finally, a demonstration on a full mock-up of a scramjet-powered cruise vehicle illustrated the toolset's scalability and adaptability to complex geometries. The simulation, which included both external and internal aerodynamics, was set up with minimal human effort and efficiently converged on high-performance computing resources. This showcases the toolset's potential for industrial applications, where rapid design iterations and high-fidelity simulations are essential. The results presented in this study validate the toolset's ability to handle the full spectrum of hypersonic flow complexities, from external aerodynamics to internal propulsion systems. The automated workflow, combined with anisotropic mesh adaptation, ensures both accuracy and computational efficiency, making it a valuable asset for future hypersonic vehicle design and optimization. Future work will focus on extending the toolset's capabilities to include moving bodies (through CAD manipulation), hybrid meshes, multi-species flows, advanced turbulence modeling, and GPU-accelerated remeshing to further enhance its performance and applicability. Additionally, the integration of adjoint-based optimization and data assimilation techniques will enable more sophisticated design and analysis workflows, paving the way for next-generation hypersonic vehicle development.

Acknowledgments

The manuscript was written through contributions of all authors.

Philippe Duveau and Pierre Grenson are acknowledged for the discussion about internal aerodynamics and help to define the reference geometry, as well as Sebastien Deck, Nicolas Renard, Jaime Vacquero, Julien Husson and Lucas Manueco for the discussions on turbulence modelling and high-fidelity simulations. The authors would like to thank Bastien Andrieu for his constant effort in Paradigm improvement and for all the discussions on geometry, mesh and partitioning. Last, the authors would like to thank Ivan Mary for the help and support regarding the use of the FastS solver and numerical scheme implementation.

Declaration of interests

The authors do not work for, advise, own shares in, or receive funds from any organization that could benefit from this article, and have declared no affiliations other than their research organizations.

Appendix A. Geometry description for the internal aerodynamics case

This appendix describes the geometry that was designed to build the internal aerodynamics reference case.

The cross plane is top/bottom symmetrical (see Figure 14) consisting of first a 10 cm long flat plate to allow for the formation of a proper turbulent boundary layer (the method used to ensure a turbulent inflow is presented in Appendix C). Then, the internal compression is caused by a 7 cm long, 5° compression ramp. This ramp is followed by a long $-2°$ diverging section that spans over 25 cm. This section is representative of the diffuser in a ramjet air intake, and its angle was selected to prevent boundary layer thickening from further restricting the inviscid flow area. Finally, the flow enters a fast-diverging section with a $-5°$ angle, extending over 5 cm. To ensure a well-posed boundary condition, back pressure is generated by incorporating a choked converging-diverging nozzle downstream of the diverging section. This configuration enables the use of a supersonic outflow boundary condition. The nozzle shape is defined using a Hermitian spline. An analytical (or discretized) description is available upon request. The throat closure has

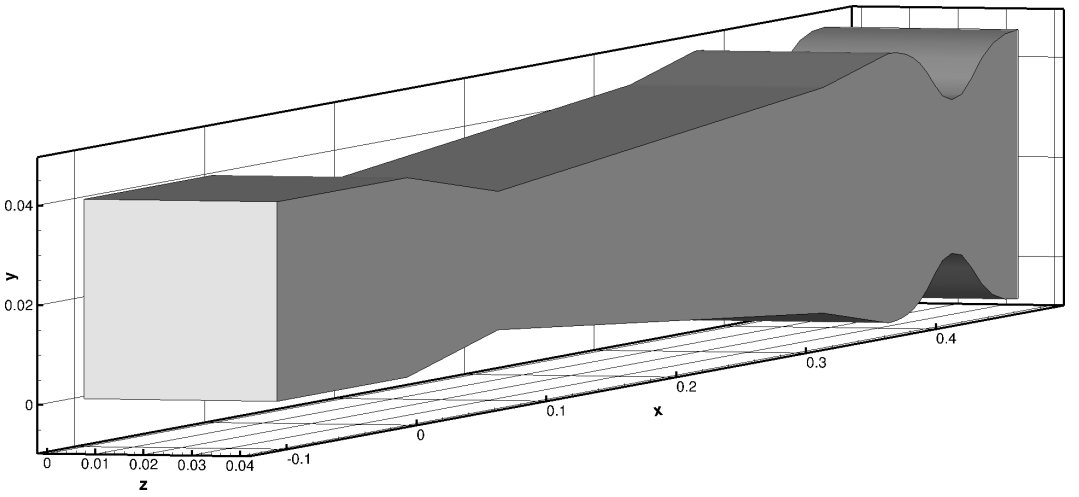


Figure 14. 3D representation of the geometry under study.

been chosen so that the isolator behaviour is close to critical for the flow conditions under study (terminal compression near the start of the diffuser). However, being too close to the critical point could have led to the unstart of the air intake either during DNS transient or when using some turbulence models. To alleviate this problem, the final condition under study is what could be considered an off-design condition, where the air intake does not provide the maximal efficiency. The total height of the inlet plane in 4 cm. The 3D geometry is then created by extruding that plane over 4 cm, creating a square shape at the inlet of the domain.

Appendix B. Numerical method for the DNS

The approximate Riemann solver is the simplified AUSM(P):

$$F_c^{i+\frac{1}{2}} = U_1 \frac{\mathbf{q}_R + \mathbf{q}_L}{2} - \Phi U_{\text{dis}} \frac{\mathbf{q}_R - \mathbf{q}_L}{2} + P, \tag{1}$$

U_1 and U_{dis} being the interface fluid velocity and the dissipation velocity similar to the original formulation of the scheme [78], $\mathbf{q}$ is the conservative state variable vector and the subscript $\cdot_L$ (resp. $\cdot_R$) corresponds to left (resp. right) state computed thanks to a MUSCL reconstruction, Φ is the dissipation sensor which is discussed hereafter.

In order to limit the dissipation, Mary [78] has proposed to identify regions where cell-to-cell oscillations vanish in order to fine tune a 2nd order low dissipation scheme. The proposed oscillations sensor reads:

$$\Delta_i^k = \begin{cases} -1 & \text{if } (\phi_{i+1}^k - \phi_i^k)(\phi_i^k - \phi_{i-1}^k) < 0, \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

with ϕ_i^k the k-th component of the state vector of primitive variables at point i. Then the dissipation sensor is defined as:

$$\Phi^i = \max_k \begin{cases} 1 & \text{if } \sum_{l=i}^{i+1} \Delta_l^k \leq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

If the solution is fully monotonous in the stencil, the scheme behaves as a fully centered scheme. If the solution is nonmonotonous, the dissipation is fully activated.

In this work, we proposed to extend the methodology to fifth-order MUSCL reconstruction. Due to the extent of the stencil the upwinding function of the scheme must be reworked. The smoothness of the solution is still evaluated by the number of slope changes in the stencil. If there are one, two, or more, the effective dissipation is set to an increased amount of the standard upwind scheme:

$$\Phi^i = \max_k \begin{cases} 1 & \text{if } \sum_{l=i-1}^{i+2} \Delta_l^k < 0, \\ \alpha_2 & \text{if } \sum_{l=i-1}^{i+2} \Delta_l^k = 0, \\ \alpha_1 & \text{if } \sum_{l=i-1}^{i+2} \Delta_l^k = 2, \\ \alpha_0 & \text{if } \sum_{l=i-1}^{i+2} \Delta_l^k = 4, \end{cases} \tag{4}$$

with $0 < \alpha_i \leq 1$ and $\alpha_0 < \alpha_1 < \alpha_2$. This approach strongly limits the dissipation of the scheme in regions where the solution is smooth, while preserving the stability of the scheme in non-smooth regions.

For supersonic and hypersonic flows, in order to ensure the robustness of the simulation while preserving the high resolvability properties of the scheme in the smooth flow regions, the shock capturing based on the work of [57] is used. The discontinuity sensor reads:

$$\kappa_i = \frac{10}{2}\left[1 - \tanh\left(2.5 + 10\frac{\delta x_i}{c}\nabla\cdot\mathbf{u}\right)\right] \times \frac{(\nabla\cdot\mathbf{u})^2}{(\nabla\cdot\mathbf{u})^2 + |\nabla\times\mathbf{u}|^2 + \epsilon} \times \left|\frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}}\right|, \tag{5}$$

with δx_i the size of the cell in the direction of interest, c the speed of sound, p_i the pressure at the point i and $\mathbf{u}$ the velocity vector. The sensor is used to switch from fifth to first order MUSCL state reconstruction, as following:

$$\mathbf{q}_{R,L} = (1-\kappa)\mathbf{q}_{R,L}^{o5} + \kappa\mathbf{q}_{R,L}^{o1}. \tag{6}$$

In strong shocks, the reconstruction tends towards first order, while elsewhere it stays at fifth order, ensuring a proper stability of the numerical method in discontinuities. As the sensor is directional, a different coefficient is applied to the reconstruction in each direction. An instantaneous map of κ for the i-direction in a constant z-plane ($z = 0.01$) of the DNS is presented in Figure 15, showing that the first order reconstruction use is confined to shock-wave and does not reduce the order of the scheme in turbulence nor boundary layers.

For instance, the numerical scheme depicted above with tuning parameters $\alpha_0 = 0.05$, $\alpha_1 = 0.4$, $\alpha_2 = 0.6$ has been successfully used in [80].

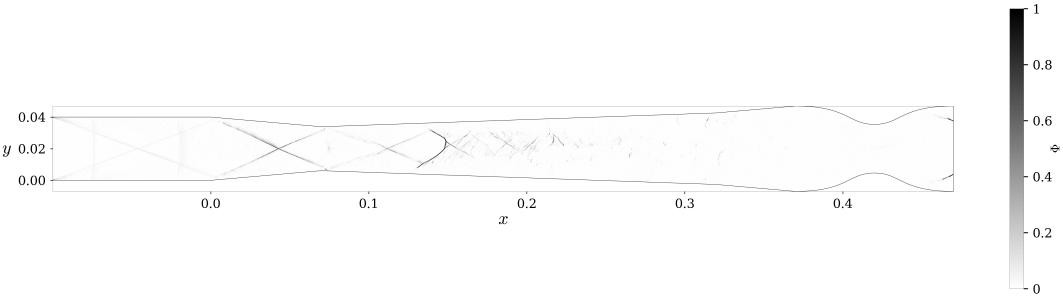


Figure 15. Instantaneous map of κ in the i-direction in a constant z-plane ($z = 0.01$) of the configuration.

Appendix C. Turbulent inflow generation for the DNS

One important aspect for such DNS is the need to inject turbulence at the inlet of the domain. In the present case it was decided not to use a standard turbulence generation method (such as Synthetic Eddy Method of Digital Inflow Filtering, which may both introduce nonphysical acoustic fluctuations which might be detrimental in a confined environment), instead a method inspired from recent advances in natural transition simulation [81,82] is used. To create a quick, and low noise transition, streaks are first generated by alternating viscous and non-viscous wall boundary conditions patches at the inlet of the domain. The wavenumber of the created streaks is chosen to be as small as possible, the lower limit being linked to the resolution of the scheme. The patches are 4 cells wide and are repeated every 12 cells. The length of the patches is set to 40 cells. The created streaks represent a very efficient support for transitional instabilities while being small enough to not contaminate the turbulent flow with large coherent structures. Then, to trigger the transition, a random perturbation on the density ($\rho' = \rho\big(1 + 0.002 * A(i,j,k)\big)$, with A a Gaussian filtered white noise, and ρ the density field, see [82]) is injected into the boundary layer, the support of the noise injection is 130 cells in the streamwise direction and 15 cells in the wall normal direction, starting from the inlet and the wall respectively. As such, it is confined to the viscous region near the wall. This noise seeds all the instabilities of the streaks, which then quickly break down to turbulence. Thanks to that method, turbulent state is reached after around $\sim 37\delta_1$, with δ_1 taken at the beginning of the convergent section, of flat plate with minimal noise emission. The flow then has $\sim 90\delta_1$ to regularize before the start of the actual geometry under study. This process is qualitatively illustrated in Figure 16, showing an instantaneous snapshot of the creation of the streaks and their breakdown. Figure 17 presents the boundary layer profile and the streamwise velocity fluctuations statistics in the central $z = 0$ plane just before the first ramp ($x = 0$), both display a behavior typical of low-Reynolds number wall turbulence, with no spurious fluctuations coming from the turbulence generation method. The DNS profile is very close to the SA-QCR profile, except in the buffer region, where the model is known to behave poorly.

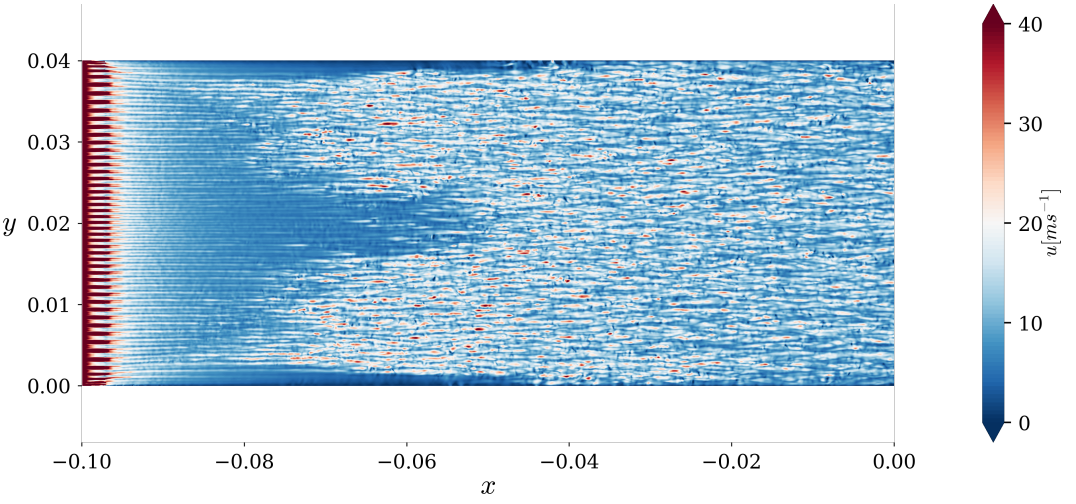


Figure 16. Streamwise velocity in the first cell in the wall normal direction, from an instantaneous snapshot of the DNS, illustrating the turbulence generation method, with first the inviscid patches to create the streaks, and then the quick breakdown to turbulence.

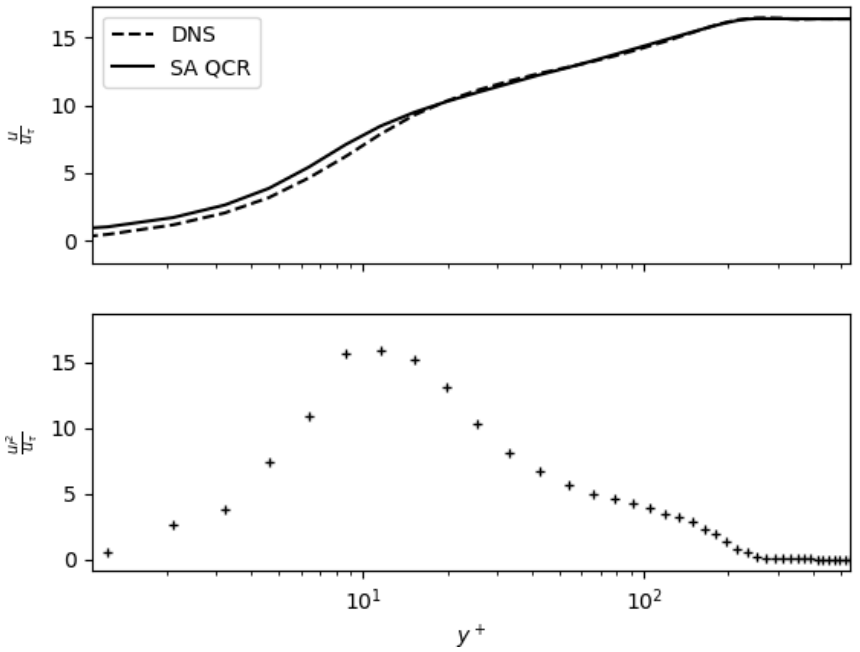


Figure 17. Time-averaged streamwise velocity profile (top) and streamwise velocity fluctuation root-mean-square value (bottom) showing the good behavior of the turbulence generation. The profile is extracted from the time-averaged flow just before the compression ramp ($x = 0$) in the central $z = 0$ plane. DNS data are sampled one point over four.

Appendix D. Impact of the turbulence model on the results

While outside the scope of the study, different turbulence models have been benchmarked before choosing the SA-QCR one. The benchmarked RANS turbulence models are the following: the Spalart–Allmaras [60] model with the compressibility correction of [61] (labeled SA) and the SST model of Menter [83] (labeled as k-ω SST). For both models, the impact of the Quadratic Constitutive Relation (labeled QCR) [62] is also tested. This non-linear addition is known to drastically improve the accuracy of linear eddy viscosity turbulence models for corner flows [84].

Figure 18 presents the Mach number in the central z-plane for each cases (the DNS correspond to time-averaged data). On the impact of the turbulence modeling strategy, one can notice that the non-QCR simulations predict a completely different separated region topology compared to the QCR ones and the DNS. The first separation point is located much more upstream, and it is caused by a weak, poorly defined shock-wave, which may be a sign that the separation comes from strong 3D effects at the corners of the geometry. The k-ω simulation quickly displays a larger separated region than the SA one. The use of QCR strongly improves the behavior of the turbulence models and leads to results in the central plane that are much closer to the DNS reference, with a separation point just upstream of $x = 0.15$ and a separation shock that is a well-defined oblique shock. The main differences between all the QCR simulations and the DNS is that the shock-train is a bit more confined to the top part of the diffuser and the low-speed region in the lower downstream part of the diffuser is larger. However, the results of the QCR RANS simulations are surprisingly close to the reference, demonstrating once again the accuracy improvement provided by the QCR on RANS simulations in the presence of corner flow.

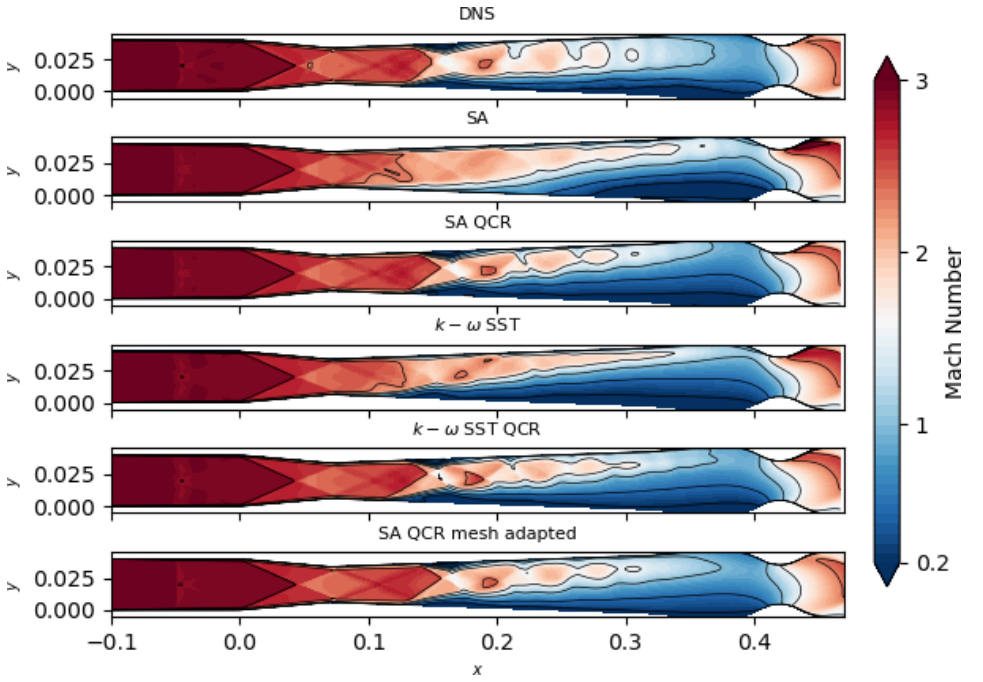


Figure 18. Mach number distribution in the central z-plane. The DNS results correspond to time-averaged flow.

References

[1] R. Mittal and G. Iaccarino, "Immersed boundary methods", *Annu. Rev. Fluid Mech.* **37** (2005), no. 1, pp. 239–261.

[2] R. Verzicco, "Immersed boundary methods: historical perspective and future outlook", *Ann. Rev. Fluid Mech.* **55** (2023), no. 1, pp. 129–155.

[3] M. Terracol and L. Manueco, "A dynamic linearized wall model for turbulent flow simulation: towards grid convergence in wall-modeled simulations", *J. Comput. Phys.* **521** (2025), article no. 113555.

[4] T.-H. Shih, L. A. Povinelli and N.-S. Liu, "Application of generalized wall function forcomplex turbulent flows", *J. Turbul.* **4** (2003), no. 1, article no. 015.

[5] A. Loseille, A. Dervieux and F. Alauzet, "Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations", *J. Comput. Phys.* **229** (2010), no. 8, pp. 2866–2897.

[6] F. Alauzet and A. Loseille, "High-order sonic boom modeling based on adaptive methods", *J. Comput. Phys.* **229** (2010), no. 3, pp. 561–593.

[7] A. Loseille, A. Dervieux, P. J. Frey and F. Alauzet, "Achievement of global second order mesh convergence for discontinuous flows with adapted unstructured meshes", in *18th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2007.

[8] D. A. Venditti and D. L. Darmofal, "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows", *J. Comput. Phys.* **176** (2002), no. 1, pp. 40–69.

[9] F. Alauzet and L. Frazza, "Feature-based and goal-oriented anisotropic mesh adaptation for RANS applications in aeronautic and aerospace", *J. Comput. Phys.* **439** (2021), article no. 110340.

[10] C. Tarsia Morisco, E. Guilbert, M. Maunoury and F. Alauzet, "Chapter Two - Practical use of metric-based anisotropic mesh adaptation for industrial applications", in *Error Control, Adaptive Discretizations, and Applications, Part 3* (F. Chouly, S. P. A. Bordas, R. Becker and P. Omnes, eds.), Advances in Applied Mechanics, vol. 60, Elsevier, 2025, pp. 57–111.

[11] M. J. Castro-Díaz, F. Hecht, B. Mohammadi and O. Pironneau, "Anisotropic unstructured mesh adaption for flow simulations", *Int. J. Numer. Methods Fluids* **25** (1997), no. 4, pp. 475–491.

[12] A. Dervieux, D. Leservoisier, P.-L. George and Y. Coudière, "About theoretical and practical impact of mesh adaptation on approximation of functions and PDE solutions", *Int. J. Numer. Methods Fluids* **43** (2003), no. 5, pp. 507–516.

[13] P. J. Frey and F. Alauzet, "Anisotropic mesh adaptation for CFD computations", *Comput. Methods Appl. Mech. Eng.* **194** (2005), no. 48, pp. 5068–5082.

[14] C. Gruau and T. Coupez, "3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric", *Comput. Methods Appl. Mech. Eng.* **194** (2005), no. 48, pp. 4951–4976.

[15] X. Li, M. S. Shephard and M. W. Beall, "Accounting for curved domains in mesh adaptation", *Int. J. Numer. Methods Eng.* **58** (2003), no. 2, pp. 247–276.

[16] K. J. Fidkowski and D. L. Darmofal, "Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results", *AIAA J.* **49** (2011), no. 4, pp. 673–694.

[17] G. Rogé and L. Martin, "Goal-oriented anisotropic grid adaptation", *Comptes Rendus. Mathématique* **346** (2008), no. 19–20, pp. 1109–1112.

[18] F. Hecht, "Error indicator and mesh adaption, in FreeFem++", 2009. Online at https://www.ljll.fr/hecht/ftp/FH-Mamerm09.pdf. Notes for the 3rd International Conference on Approximation Methods and numerical Modeling in Environment and Natural Resources (MAMERN '09).

[19] A. Loseille, *Génération et Adaptation de Maillages pour le Calcul Scientifique*, Université Paris-Saclay (France), 2020.

[20] C. Caillaud, A. Scholten, J. Kuehl, et al., "Separation and transition on a cone-cylinder-flare: computational investigations", *AIAA J.* **63** (2025), no. 7, pp. 2615–2634.

[21] E. K. Benitez, J. S. Jewell and S. P. Schneider, "Separation bubble variation due to small angles of attack for an axisymmetric model at Mach 6", in *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021.

[22] L. Sombaert, M. Lugrin, R. Bur, et al., "Ground tests on bolt at Mach 7: Cross-facility comparison and stability analysis", *AIAA J.* **63** (2025), no. 6, pp. 2135–2150.

[23] B. C. Chynoweth, A. Lay and J. S. Jewell, "Boundary-Layer Transition on BOLT in High Reynolds Number Mach-6 Quiet Flow", *J. Spacecr. Rock.* (2025).

[24] B. M. Wheaton, D. C. Berridge, T. D. Wolf, R. T. Stevens and B. E. McGrath, "Boundary layer transition (BOLT) flight experiment overview", in *2018 Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2018.

[25] J. M. Wirth, R. D. Bowersox, A. T. Dufrene and T. P. Wadhams, "Measurements of natural turbulence during the BOLT II flight experiment", *J. Spacecr. Rock.* **61** (2024), no. 5, pp. 1281–1292.

[26] C. Vu, A. Knutson, P. K. Subbareddy and G. V. Candler, "Postflight Receptivity and Transition Analysis of BOLT-II Side A Descent Trajectory", in *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024.

[27] R. Fiévet, H. Koo, V. Raman and A. Auslender, "Numerical simulation of shock trains in a 3D channel", in *54th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2016.

[28] J. C. Dutton and B. F. Carroll, *A numerical and experimental investigation of multiple shock wave/turbulent boundary layer interactions in a rectangular duct*, technical report, University of Illinois at Urbana-Champaign, no. UILU-ENG-88-4001, 1988.

[29] L. Cambier, S. Heib and S. Plot, "The Onera elsA CFD software: Input from research and feedback from industry", *Mechanics and Industry* **14** (2013), no. 3, pp. 159–174.

[30] D. Gueyffier, S. Plot and M. Soismier, "SoNICS: a new generation CFD software for satisfying industrial users needs", 2022. Online at https://hal.science/hal-04045165/document. Conference paper: OTAN/STO/Workshop AVT-366.

[31] A. Balan, M. A. Park, S. Wood and W. K. Anderson, "Verification of anisotropic mesh adaptation for complex aerospace applications", in *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, 2020.

[32] U.S. Department of Energy (DOE), *DOE Exascale Requirements Review — Basic Energy Sciences (BES)*, technical report, U.S. Department of Energy, Office of Science, no. DOE/SC-0187, 2017. Online at https://blogs.anl.gov/exascaleage/wp-content/uploads/sites/67/2017/05/DOE-ExascaleReport_BES_R48.pdf.

[33] P. L. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", *J. Comput. Phys.* **43** (1981), pp. 357–372.

[34] S.-s. Chen, C. Yan, K. Zhong, H.-c. Xue and E.-l. Li, "A novel flux splitting scheme with robustness and low dissipation for hypersonic heating prediction", *Int. J. Heat Mass Transfer* **127** (2018), pp. 126–137.

[35] F. Renac, "Entropy stable, robust and high-order DGSEM for the compressible multicomponent Euler equations", *J. Comput. Phys.* **445** (2021), article no. 110584.

[36] V. Venkatakrishnan, "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters", *J. Comput. Phys.* **118** (1995), no. 1, pp. 120–130.

[37] G. D. van Albada, B. van Leer and J. W. W. Roberts, "A comparative study of computational methods in cosmic gas dynamics", *Astron. Astrophys.* **108** (1982), no. 1, pp. 76–84.

[38] S. Piperno and S. Depeyre, "Criteria for the design of limiters yielding efficient high resolution TVD schemes", *Comput. Fluids* **27** (1998), no. 2, pp. 183–197.

[39] G. K. W. Kenway, C. A. Mader, P. He and J. R. R. A. Martins, "Effective adjoint approaches for computational fluid dynamics", *Prog. Aerosp. Sci.* **110** (2019), article no. 100542.
[40] L. N. Trefethen, A. E. Trefethen, S. C. Reddy and T. A. Driscoll, "Hydrodynamic Stability Without Eigenvalues", *Science* **261** (1993), no. 5121, pp. 578–584.
[41] F. Alizard, S. Cherubini and J.-C. Robinet, "Sensitivity and optimal forcing response in separated boundary layer flows", *Phys. Fluids* **21** (2009), no. 6, article no. 064108.
[42] B. J. McKeon and A. S. Sharma, "A critical-layer framework for turbulent pipe flow", *J. Fluid Mech.* **658** (2010), pp. 336–382.
[43] D. Sipp and O. Marquet, "Characterization of noise amplifiers with global singular modes: the case of the leading-edge flat-plate boundary layer", *Theor. Comput. Fluid Dyn.* **27** (2013), no. 5, pp. 617–635.
[44] G. Rigas, D. Sipp and T. Colonius, "Nonlinear input/output analysis: application to boundary layer transition", *J. Fluid Mech.* **911** (2021), article no. A15.
[45] B. Bugeat, J. C. Chassaing, J.-C. Robinet and P. Sagaut, "3D global optimal forcing and response of the supersonic boundary layer", *J. Comput. Phys.* **398** (2019), article no. 108888.
[46] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary Design Optimization: A Survey of Architectures", *AIAA J.* **51** (2013), no. 9, pp. 2049–2075.
[47] G. K. W. Kenway and J. R. R. A. Martins, "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration", *J. Aircraft* **51** (2014), no. 1, pp. 144–160.
[48] E. J. Parish and K. Duraisamy, "A paradigm for data-driven predictive modeling using field inversion and machine learning", *J. Comput. Phys.* **305** (2016), pp. 758–774.
[49] K. Duraisamy, G. Iaccarino and H. Xiao, "Turbulence Modeling in the Age of Data", *Ann. Rev. Fluid Mech.* **51** (2019), pp. 357–377.
[50] Z. Li, H. Zhang, S. C. C. Bailey, J. B. Hoagg and A. Martin, "A data-driven adaptive Reynolds-averaged Navier–Stokes k-ω model for turbulent flow", *J. Comput. Phys.* **345** (2017), pp. 111–131.
[51] L. Franceschini, D. Sipp and O. Marquet, "Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high Reynolds number backward-facing step", *Phys. Rev. Fluids* **5** (2020), article no. 094603 (24 pages).
[52] S. Bai, J. Z. Kolter and V. Koltun, "Deep Equilibrium Models", in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds.), Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., 2019.
[53] M. Lienhardt and B. Michel, "SoNICS: design and workflow of a new CFD software", *Comptes Rendus. Mécanique* **353** (2025), pp. 1261–1287.
[54] C. Debiez and A. Dervieux, "Mixed-element-volume MUSCL methods with weak viscosity for steady and unsteady flow calculations", *Comput. Fluids* **29** (2000), no. 1, pp. 89–118.
[55] B. van Leer, "Towards the Ultimate Conservative Difference Scheme", *J. Comput. Phys.* **135** (1997), no. 2, pp. 229–248.
[56] F. Alauzet and L. Frazza, "Feature-based and goal-oriented anisotropic mesh adaptation for RANS applications in aeronautics and aerospace", *J. Comput. Phys.* **439** (2021), article no. 110340.
[57] L. Sciacovelli, D. Passiatore, P. Cinnella and G. Pascazio, "Assessment of a high-order shock-capturing central-difference scheme for hypersonic turbulent flow simulations", *Comput. Fluids* **230** (2021), article no. 105134.
[58] Y. Liu, B. Diskin, H. Nishikawa, W. K. Anderson, G. C. Nastac, E. J. Nielsen, A. Walden and L. Wang, "Edge-Based Viscous Method for Mixed-Element Node-Centered Finite-Volume Solvers", *AIAA J.* **62** (2024), no. 1, pp. 209–230.
[59] H. Song, K. V. Matsuno, J. R. West, A. Subramaniam, A. S. Ghate and S. K. Lele, "Scalable parallel linear solver for compact banded systems on heterogeneous architectures", *J. Comput. Phys.* **468** (2022), article no. 111443.
[60] P. R. Spalart and S. R. Allmaras, "One-equation turbulence model for aerodynamic flows", *Rech. Aerosp.* **439** (1994), no. 1, pp. 5–21.
[61] S. Deck, P. Duveau, P. d'Espiney and P. Guillen, "Development and application of Spalart–Allmaras one equation turbulence model to three-dimensional supersonic complex configurations", *Aerosp. Sci. Technol.* **6** (2002), no. 3, pp. 171–183.
[62] P. R. Spalart, "Strategies for turbulence modelling and simulations", *Int. J. Heat Fluid Flow* **21** (2000), no. 3, pp. 252–263.
[63] R. Haimes and M. Drela, "On the construction of aircraft conceptual geometry for high-fidelity analysis and design", in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2012.
[64] J. Dannenhoffer, "An overview of the engineering sketch pad", in *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024.
[65] H. Si, "TetGen, a Delaunay-based quality tetrahedral mesh generator", *ACM Trans. Math. Softw.* **41** (2015), no. 2, article no. 11 (36 pages).

[66] J. Schöberl, "NETGEN An advancing front 2D/3D-mesh generator based on abstract rules", *Comput. Vis. Sci.* **1** (1997), no. 1, pp. 41–52.

[67] C. Rumsey, B. Wedan, T. Hauser and M. Poinot, "Recent updates to the CFD general notation system (CGNS)", in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2012.

[68] ONERA, *onera/Maia: Distributed algorithms and manipulations over CGNS meshes*, 1.8. Online at https://github.com/onera/Maia (accessed on December 10, 2025).

[69] ONERA, "Maia CGNS trees", in *Maia documentation*, 2021. Online at https://onera.github.io/Maia/1.8/introduction/introduction.html#maia-cgns-trees (accessed on December 10, 2025).

[70] ONERA, *onera/paradigm: Library for managing meshes in a massively parallel distributed environment*, 2024. Online at https://github.com/onera/paradigm/ (accessed on December 11, 2025).

[71] B. Andrieu, B. Maugars and E. Quémerais, "Dynamic load-balanced point location algorithm for data mapping", 2025. To appear in *Comptes Rendus. Mécanique*.

[72] A. Loseille and F. Alauzet, "Continuous mesh framework part II: validations and applications", *SIAM J. Numer. Anal.* **49** (2011), no. 1, pp. 61–86.

[73] A. Poulain, C. Content, D. Sipp, G. Rigas and E. Garnier, "BROADCAST: A high-order compressible CFD toolbox for stability and sensitivity using Algorithmic Differentiation", *Comput. Phys. Commun.* **283** (2023), article no. 108557.

[74] R. A. Baurle, J. A. White, M. D O'Connell, T. G. Drozda and A. T. Norris, *VULCAN-CFD User Manual: Ver. 7.2. 0*, technical memorandum, NASA, no. NASA/TM-20220008781, 2022.

[75] G. Puigt, V. Golliet and F. Alauzet, "Metric-based mesh adaptation for hypersonic flows: capturing wall heat flux with anisotropic triangles and tetrahedrons", 2025. Conference paper: The 4th International Conference on High-Speed Vehicle Science & Technology.

[76] G. V. Candler, H. B. Johnson, I. Nompelis, V. M. Gidzak, P. K. Subbareddy and M. Barnhardt, "Development of the US3D code for advanced compressible and reacting flow simulations", in *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 2015.

[77] ONERA, *Fast: a compressible flow solver python package*, 4.0, 2024. Online at https://onera.github.io/Fast/ (accessed on December 11, 2025).

[78] I. Mary and P. Sagaut, "Large Eddy Simulation of Flow Around an Airfoil Near Stall", *AIAA J.* **40** (2002), no. 6, pp. 1139–1145.

[79] M.-S. Liou, "A sequel to AUSM, Part II: AUSM+-up for all speeds", *J. Comput. Phys.* **214** (2006), no. 1, pp. 137–170.

[80] A. Poulain, C. Content, A. Schioppa, P. Nibourel, G. Rigas and D. Sipp, "Adjoint-based optimisation of time- and span-periodic flow fields with Space-Time Spectral Method: Application to non-linear instabilities in compressible boundary layer flows", *Comput. Fluids* **282** (2024), article no. 106386 (17 pages).

[81] C. Hader and H. F. Fasel, "Towards simulating natural transition in hypersonic boundary layers via random inflow disturbances", *J. Fluid Mech.* **847** (2018), article no. R3.

[82] M. Lugrin, S. Beneddine, C. Leclercq, E. Garnier and R. Bur, "Transition scenario in hypersonic axisymmetrical compression ramp flow", *J. Fluid Mech.* **907** (2021), article no. A6.

[83] F. R. Menter, *Improved two-equation k-omega turbulence models for aerodynamic flows*, technical memorandum, NASA, no. NASA-TM-103975, 1992.

[84] J. Dandois, "Improvement of corner flow prediction using the quadratic constitutive relation", *AIAA J.* **52** (2014), no. 12, pp. 2795–2806.