Progress in Propulsion Physics **4** (2013) 417-444
DOI: 10.1051/eucass/201304417

elsA-HYBRID: AN ALL-IN-ONE STRUCTURED/UNSTRUCTURED SOLVER FOR THE SIMULATION OF INTERNAL AND EXTERNAL FLOWS. APPLICATION TO TURBOMACHINERY

M. de la Llave Plata, V. Couaillier, M.-C. Le Pape, C. Marmignon, and M. Gazaix

ONERA — The French Aerospace Lab.
Châtillon 92322, France

This paper reports recent work on the extension of the multiblock structured solver *elsA* to deal with hybrid grids. The new hybrid-grid solver is called *elsA-H* (*elsA-Hybrid*), is based on the investigation of a new unstructured-grid module has been built within the original *elsA* CFD (computational fluid dynamics) system. The implementation benefits from the flexibility of the object-oriented design. The aim of *elsA-H* is to take advantage of the full potential of structured solvers and unstructured mesh generation by allowing any type of grid to be used within the same simulation process. The main challenge lies in the numerical treatment of the hybrid-grid interfaces where blocks of different type meet. In particular, one must pay attention to the transfer of information across these boundaries, so that the accuracy of the numerical scheme is preserved and flux conservation is guaranteed. In this paper, the numerical approach allowing to achieve this is presented. A comparison between the hybrid and the structured-grid methods is also carried out by considering a fully hexahedral multiblock mesh for which a few blocks have been transformed into unstructured. The performance of *elsA-H* for the simulation of internal flows will be demonstrated on a number of turbomachinery configurations.

1 INTRODUCTION

Numerical modeling and simulation of turbomachinery flows is an area of keen interest and active research in both industry and academia. The development of new and increasingly powerful simulation tools has helped jet engine manufacturers to gain a greater understanding of the operating performance of their products. It has also allowed them to progress through the design life cycle in a

timely and cost-effective manner, improving reliability and enhancing the quality of the end product.

Coping with the severe thermal and mechanical stresses encountered in gas turbine cycles requires active cooling and leads to highly complex geometries with many small features relative to the characteristic size of the domain. One must account for real geometry effects such as shroud leakage, sealing and tip clearance secondary path flows, and flow around the intricate maze of cooling channels underneath the hub, to mention but a few. This represents a significant challenge and pushes the limits of CFD and grid generation capabilities.

Currently available CFD solvers can be classified according to the type of grid and associated data structure they support. In this sense, a structured-grid solver is defined by the directional nature of the algorithms and methods it implements, and the well-organised Cartesian-type data structure on which these rest. The regular arrangement of the cells allows us to locate any grid entity as well as its neighbors by their three indices (i, j, k). This set of indices thereby defines a new coordinate system local to the grid, with each index representing one of the three mesh-line directions. An unstructured-grid solver, on the other hand, is characterised by having a disordered (close to random) data structure, and by the consequent lack of directionality of the corresponding numerical schemes. This means that in order to identify an element in the grid, it is necessary to define and store additional information with regards to cell-to-cell and face-to-cell connectivity.

Thanks to their mesh-aligned numerical schemes and the generally higher quality of the input grids, structured solvers are known to provide a greater degree of accuracy compared to their unstructured counterparts. They also enjoy higher computational efficiency and lower memory requirements, for a given number of degrees of freedom, as no connectivity information needs to be stored. However, when dealing with complex configurations, as is often the case in turbomachinery applications, the generation of structured grids might become an arduous task. This is the reason why design engineers have turned their attention towards unstructured solvers. Despite the seemingly loss of computational efficiency and accuracy with respect to structured solvers, the major advantage of unstructured-grid methods lies in the ease of mesh generation and the adaptation to the local physics of the flow. Indeed, while generating a good quality structured mesh may take several weeks for a complex geometry, it is possible to obtain a good quality unstructured grid within a few hours. Therefore, finding a trade-off between computational cost, mesh-generation times and accuracy of the numerical solution is desirable.

It would then be possible to devise a hybrid CFD solver able to combine the most valuable assets of both methods. This implies generating hybrid grids. The term hybrid grid means the coexistence of structured and unstructured blocks within the same physical domain. Typically, there will be unstructured subdomains in areas where high geometrical complexity is present (e. g., small design

features in turbine blades), or areas in which the local physics of the flow requires the implementation of a mesh refinement approach, which is most conveniently done in an unstructured framework (e. g., unsteady wake-blade interaction problems). On the other hand, one may want to use a structured grid in near-wall regions in turbulent flows, which with standard solvers are more accurately resolved using a structured approach. In [1, 2], Lefebvre *et al.* demonstrated the versatility of hybrid methods for the simulation of turbulent flows for several turbomachinery and external flow configurations using adaptive hybrid grids. More recently, Yang *et al.* [3] have proposed a hybrid CFD solver based on a conservative hybrid-grid algorithm for mismatched abutting interfaces. In the field of helicopter aerodynamics, Wissink *et al.* [4] have applied the hybrid approach by resorting to the external coupling of two existing CFD solvers: a body-fitted unstructured solver and a high-order block-structured Cartesian solver.

The objective of this work is to exploit the flexibility of hybrid-grid generation and associated algorithms based on an extension of the capabilities of the CFD solver *elsA* [5] to deal with unstructured grids. The new hybrid solver is called *elsA-Hybrid* (*elsA-H*).

The material presented in this paper is organized in the following manner. Firstly, a brief overview of *elsA*'s general architecture and kernel design is provided. Then the features specific to the implementation of *elsA-H*, including the general organization of the code (object-oriented design), data structure, and input/output format, are discussed. The rest of the paper is devoted to the fundamental numerical methods underpinning the new hybrid solver. The numerical schemes available so far are presented in detail, namely, the standard Jameson–Smith–Turkel (JST) scheme and Roe's upwind scheme with MUSCL (monotone upstream-centered schemes for conservation laws) reconstruction. For the implicit stage, an Euler backward time integration scheme associated with the LU-SSOR (lower-upper symmetric successive overrelaxation) technique has been implemented. Special attention is put to the development of an appropriate approach to deal with hybrid-grid matching interfaces, and issues related to the parallelization strategy are discussed. Finally, the flexibility and performance of *elsA-H* is demonstrated on a set of reference test cases, as well as on a number of turbomachinery configurations.

2 GOVERNING EQUATIONS

The equations governing the evolution of a compressible Newtonian fluid are the Navier–Stokes equations. The physical model adopted here is based on the Reynolds-Averaged Navier–Stokes (RANS) approach. This approach is based on the time-averaging of the instantaneous Navier–Stokes equations and constitutes a popular simplification of the original system for complex industrial applications.

In a rotating frame of reference, rotating at angular rate $\mathbf{\Omega}$ about an axis, the three-dimensional (3D) RANS equations are written in terms of relative variables as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0\,; \tag{1}$$

$$\frac{\partial \rho \mathbf{V}}{\partial t} + \nabla \cdot \left(\rho \mathbf{V} \otimes \mathbf{V} + p\bar{\bar{I}} - \bar{\bar{\tau}}\right) = \rho\left(\Omega^2 \mathbf{r} - 2\mathbf{\Omega} \times \mathbf{V}\right)\,; \tag{2}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot \left(\rho E \mathbf{V} + \left(p\bar{\bar{I}} - \bar{\bar{\tau}}\right) \cdot \mathbf{V} - \mathbf{q}\right) = \rho \Omega^2 \mathbf{r} \cdot \mathbf{V}\,. \tag{3}$$

For the sake of compactness, Eqs. (1) to (3) will be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{f_c} - \mathbf{f_v}) = \mathbf{S}(\mathbf{\Omega}) \tag{4}$$

where $\mathbf{U}$, $\mathbf{f_c}$, and $\mathbf{f_v}$ represent the conservative variables, the convective flux, and the viscous flux vectors, respectively,

$$\mathbf{U} = (\rho, \rho \mathbf{V}, \rho E)^{\mathrm{T}}\,; \tag{5}$$

$$\mathbf{f_c} = \left(\rho \mathbf{V}, \rho \mathbf{V} \otimes \mathbf{V} + p\bar{\bar{I}}, (\rho E + p)\,\mathbf{V}\right)^{\mathrm{T}}\,; \tag{6}$$

$$\mathbf{f_v} = (0, \bar{\bar{\tau}}, \bar{\bar{\tau}} \cdot \mathbf{V} - \mathbf{q})^{\mathrm{T}} \tag{7}$$

and $\mathbf{S}(\mathbf{\Omega})$ is the source term that takes into account the rotation of the system and becomes zero for an inertial frame, i. e., when $\mathbf{\Omega} = 0$. In Eqs. (5) to (7), ρ represents the density, E the relative total energy per unit volume, $\mathbf{V}$ is the relative velocity vector $(u, v, w)^{\mathrm{T}}$ and $\otimes$ denotes the tensor product. The static pressure p is related to the conservative variables through the ideal gas law:

$$p = (\gamma - 1)\,\rho \left(E - \frac{\|\mathbf{V}\|^2}{2}\right)$$

where the adiabatic index $\gamma = C_p/C_v$ is the ratio of specific heat capacities at constant pressure and constant volume conditions. The stress tensor $\bar{\bar{\tau}}$ is related to the velocity gradients by Stokes relations. Making use of Boussinesq approximation, the viscous stresses and the heat flux vector $\mathbf{q}$ can be written as

$$\bar{\bar{\tau}} = (\mu + \mu_t)\left[\frac{1}{2}\left(\nabla \mathbf{V} + \nabla \mathbf{V}^{\mathrm{T}}\right) - \frac{2}{3}\nabla \cdot \mathbf{V}\bar{\bar{I}}\right]\,;$$

$$\mathbf{q} = -\left(\frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t}\right) C_p \nabla T \tag{8}$$

In Eq. (8), the Prandtl number Pr and the turbulent Prandtl number Pr_t are assumed to be constant and equal to 0.72 and 0.9, respectively. The Suther-

land law provides an expression for the molecular viscosity as a function of the temperature,

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{3/2} \frac{T_0 + s}{T + s}$$

in which T_0 and μ_0 are the fixed reference values and $s = 110.4$ K.

The system of RANS equations is closed by means of a turbulence model based on transport equations for the turbulent variables. Two turbulence models are considered in this paper, the one-equation RANS model proposed by Spalart–Allmaras (SA) [6] and the κ–ω two-equation model by Wilcox [7].

3 HYBRID FLOW SOLVER

This section gives a general overview of the multiblock structured CFD solver *elsA* and briefly describes the integration of the new unstructured module into the original code architecture. A detailed discussion of the numerical methods implemented in *elsA-H* is provided with emphasis on the special handling of the hybrid-grid interfaces.

3.1 The Multiblock Structured Solver *elsA*

Developed at ONERA since 1997 and initially based on its predecessors *Canari* [8] and *Flu3M* [9], *elsA* is a structured multiblock CFD solver based on a cell-centered finite-volume approach [5, 10].

A large number of engineering problems can be solved using *elsA*, including internal and external flows in the low subsonic to the supersonic regimes. The compressible 3D RANS equations can be solved in a moving frame of reference using a second-order upwind or a centered spatial discretization, stabilized using the scalar or matrix-based viscosity. A high degree of flexibility is achieved thanks to the capability to deal with a large number of grid topologies, including mismatched abutting or nonabutting blocks and overset grids (Chimera approach). The hierarchical mesh refinement technique allows for an increased efficiency and accuracy of the simulation. Available time-marching strategies include the multistage Runge–Kutta scheme with implicit smoothing of residuals, and the implicit backward Euler scheme associated with an LU factorization technique. Time-accurate simulations are performed using the implicit dual time-stepping method.

A number of turbulence models are available in *elsA*, including models based on the turbulent viscosity hypothesis and Reynolds-stress models. More advanced models such as the detached-eddy simulation (DES) and the large-eddy

simulation (LES) techniques can also be used for the simulation of unsteady turbulent flows. Laminar–turbulent transition prediction models, well-adapted to complex configurations, have also been implemented in *elsA*.

Other significant features offered by *elsA* are its aeroelasticity module [11, 12], and an optimization module based on adjoint techniques [13].

3.2 The Hybrid-Grid Solver *elsA-H*

The extension of *elsA* to mixed structured-unstructured configurations is based on the introduction of new generic parent classes from which purely structured and unstructured objects and methods inherit. This approach permitted the development of a new unstructured module within *elsA*, avoiding the need for external code coupling. Following *elsA*'s own numerical choices, *elsA-H* is based on a multiblock cell-centered finite-volume method. The unstructured solver relies on a face-based approach, which involves a preliminary preprocessing stage to convert the element-based connectivity provided by the grid generator into the face-based connectivity required by the solver. A large number of Fortran routines have actually been reused (with minor or no changes) in the development of the new unstructured solver. As an example, most of the original (structured) classes composing the turbulence module have been readily used in *elsA-H*, since the unstructured numerical method is formally identical to the structured one. However, dedicated unstructured classes have been introduced for the computation of the gradient terms and the application of the boundary conditions.

elsA-H supports multielement grids which may contain hexahedra, tetrahedra, pyramids and prisms. The exchange of data between neighboring blocks is performed through one single layer of ghost cells, requiring adapted numerical schemes at the hybrid-grid interfaces. This point will be addressed in detail in subsection 3.5.

The main input data format supported by *elsA-H* is the CFD General Notation System (CGNS) [14], which provides an easy representation of structured, unstructured and hybrid grids in one single file.

3.3 Space–Time Integration of the Reynolds-Averaged Navier–Stokes Equations

The numerical integration of the RANS system (4) is based on a standard space-time decoupling approach. The semidiscrete form of the equations is written as

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{1}{V}\mathbf{R}$$

where $\mathbf{R}$ denotes a space discretization of the volume integral of the convective, diffusive, and source terms, namely,

$$\mathbf{R} \cong \int_V \left(\nabla \cdot (\mathbf{f_c} - \mathbf{f_v}) - \mathbf{S}(\mathbf{\Omega})\right) dV \,. \tag{9}$$

A multistep integration technique is employed in the iterative process. Usually, the four-step Runge–Kutta scheme is used when performing explicit computations. For implicit simulations, the one-step backward Euler scheme with LU-SSOR relaxation is used as described in subsection 3.7. In the case of an explicit simulation, the solution at time $t^{(n+1)}$ is then obtained as follows :

$$\begin{aligned} \mathbf{U}^{(0)} &= \mathbf{U}^{(n)} \,; \\ \mathbf{U}^{(p)} &= \mathbf{U}^{(0)} - \frac{\alpha_p \Delta t}{V} \mathbf{R}^{(p-1)} \quad \text{for} \quad p = 1, \ldots, k \,; \\ \mathbf{U}^{(n+1)} &= \mathbf{U}^{(k)} \end{aligned}$$

For the four-step Runge–Kutta method, $k = 4$ and α_p takes on the values 1/4, 1/3, 1/2, and 1.

3.4 Spatial Discretization

The second-order accurate discretization of the convective fluxes may be performed using either the centered JST scheme or the Roe's upwind scheme with MUSCL reconstruction. Both schemes are described below.

3.4.1 Jameson–Smith–Turkel scheme on unstructured grids

The centered scheme proposed by Jameson *et al.* in [15, 16] is one of the most popular schemes used for structured grids. This is due to its simplicity in terms of implementation, low computational cost, as well as the good spatial accuracy obtained when regular meshes are considered. Starting with Eq. (4), the finite-volume approximation of the convective fluxes is derived by taking the integral of this equation over a cell volume V_i

$$\int_{V_i} \nabla \cdot \mathbf{f_c} \, dV \rightarrow \sum_{j(i)} \mathbf{F_c}_j \cdot \mathbf{n}_j S_j \tag{10}$$

where $\mathbf{F_c}$ represents the numerical flux and $\mathbf{n}_j$ and S_j are the unit normal vector and the surface area, respectively, of the interface $j(i)$. The summation is performed over the faces $j(i)$ that compose the cell i. Jameson *et al.* propose

to express Eq. (10) as the sum of a centered contribution $\mathbf{C}_i$ and an artificial dissipation term $\mathbf{D}_i$:

$$\sum_{j(i)} \mathbf{F}_{\mathbf{c}\,j} \cdot \mathbf{n}_j S_j = \mathbf{C}_i + \mathbf{D}_i \,. \tag{11}$$

The evaluation of the first term in Eq. (11) yields

$$\mathbf{C}_i = \sum_{j(i)} \frac{1}{2} \left(\mathbf{F}_{\mathbf{c}}(\mathbf{U}_{i_j}) + \mathbf{F}_{\mathbf{c}}(\mathbf{U}_i) \right) \cdot \mathbf{n}_j S_j \tag{12}$$

in which the index i refers to the cell where the quantities are being computed and i_j being its neighboring cells. The second term in Eq. (11) acts as an artificial viscosity. It is composed of two contributions:

$$\mathbf{D}_i = \mathbf{D}_{2i} + \mathbf{D}_{4i} \,. \tag{13}$$

The second-order term $\mathbf{D}_{2i}$ is written as the balance of the first differences of the conservative variables across the cell faces,

$$\mathbf{D}_{2i} = \sum_{j(i)} \varepsilon_{2j} \left(\mathbf{U}_{i_j} - \mathbf{U}_i \right) \lambda_j S_j \,. \tag{14}$$

whereas the fourth-order term is given by

$$\mathbf{D}_{4i} = -\sum_{j(i)} \varepsilon_{4j} \left(\delta^2 \mathbf{U}_{i_j} - \delta^2 \mathbf{U}_i \right) \lambda_j S_j \,. \tag{15}$$

Here, the second differences $\delta^2 \mathbf{U}_i$ are defined as

$$\delta^2 \mathbf{U}_i = \sum_{j(i)} \left(\mathbf{U}_{i_j} - \mathbf{U}_i \right) \,. \tag{16}$$

The nonlinear second-order difference term $\mathbf{D}_{2i}$ ensures an adequate representation of the solution in the proximity of a discontinuity, whereas the linear fourth-order difference term $\mathbf{D}_{4i}$ acts as a stabilizing term.

In Eqs. (14) and (15), $\lambda_j = \left(|\mathbf{V} \cdot \mathbf{n}| + c \right)_j$ is the spectral radius of the Jacobian matrix of the convective flux vector. The coefficients ε_{2j} and ε_{4j} are computed by means of a pressure sensor given by

$$\nu_i = \frac{\left| \sum_{j(i)} \left(p_{i_j} - p_i \right) \right|}{\sum_{j(i)} \left(p_{i_j} + p_i \right)}$$

where p_i are the values of the static pressure and

$$\varepsilon_{2j} = k_2 \max\left(\nu_i, \nu_{i_j} \right) \;; \quad \varepsilon_{4j} = k_4 \max\left(0, 1 - \varepsilon_{2j} \right) \,.$$

Typical values for the constants k_2 and k_4 are 0.5 and 0.016, respectively.

3.4.2 Roe's upwind scheme with monotone upstream-centered schemes for conservation laws reconstruction

The distinguishing feature of Roe's approximate Riemann solver [17] is its robustness and performance for a wide variety of flow conditions, ranging from the subsonic to the transonic regime, as is the case in turbomachinery applications. An entropy correction, such as that proposed by Yee *et al.* [18], is, however, necessary in order to guarantee that the approximate solution converges to the correct physical solution, and nonphysical solutions such as expansion shocks are not admitted.

Roe's scheme is based on the Godunov approach [19] which relies on the solution of an approximate Riemann problem at the interface between two distinct states. The discrete flux is computed at the interface j between two constant states as

$$\mathbf{F}_{\mathbf{c}j} = \frac{1}{2}\left(\mathbf{F}_{\mathbf{c}}(\mathbf{W}_L) + \mathbf{F}_{\mathbf{c}}(\mathbf{W}_R) - |\tilde{A}|\,(\mathbf{W}_L - \mathbf{W}_R)\right)$$

in which $\mathbf{W}_L$ and $\mathbf{W}_R$ refer to the primitive variables evaluated at the left- and right-hand sides of the interface j, and $\tilde{A}$ is the Jacobian matrix of the convective fluxes $A = \partial\mathbf{F}_{\mathbf{c}}/\partial\mathbf{U}$ evaluated at some average state which must be uniquely defined as a function of the left and right states.

The first-order Roe scheme consists in taking $\mathbf{W}_L$ and $\mathbf{W}_R$ equal to the values of $\mathbf{W}$ in the cells sharing the interface j, which leads to a very dissipative scheme. In order to increase the accuracy of this scheme it is necessary to use the MUSCL technique [20] which is based on a linear reconstruction of the left and right states at the interface using the information contained in the neighboring cells.

For unstructured grids and using the notation introduced in the previous section, one has at interface $j(i)$:

$$\mathbf{W}^L_{j(i)} = \mathbf{W}_i + \frac{1}{2}\,\Phi\left(\nabla\mathbf{W}_i \cdot \mathbf{d}_{i\to i_j}, \mathbf{W}_{i_j} - \mathbf{W}_i\right)\,; \tag{17}$$

$$\mathbf{W}^R_{j(i)} = \mathbf{W}_{i_j} - \frac{1}{2}\,\Phi\left(\nabla\mathbf{W}_{i_j} \cdot \mathbf{d}_{i_j\to i}, \mathbf{W}_{i_j} - \mathbf{W}_i\right) \tag{18}$$

where Φ is a limiter function that limits the slope of the piecewise approximation. Available limiter functions in *elsA-H* include *Minmod*, *Van Albada*, *Van Leer*, and *Superbee*. Vector $\mathbf{d}_{i_j\to i}$ is the vector connecting the centroids of the adjacent cells i_j and i. It is, therefore, assumed that the interface $j(i)$ is located halfway between i and i_j. This choice is consistent with the structured version of Roe's scheme implemented in *elsA*. The evaluation of the gradient $\nabla\mathbf{W}_i$ is done using the Green–Gauss formula:

$$\nabla\mathbf{W}_i = \frac{1}{V_i}\sum_{j(i)}\frac{1}{2}\,\left(\mathbf{W}_i + \mathbf{W}_{i_j}\right)\cdot\mathbf{n}_j S_j\,. \tag{19}$$

3.4.3 Viscous discretization

A simple scheme is used for the discretization of the viscous terms. It is based on the evaluation of the gradient at the interface $j(i)$ as the average of the cell-centered gradients in cells i and i_j computed as in Eq. (19):

$$\overline{(\nabla \mathbf{W})}_j = \frac{1}{2}\left(\nabla \mathbf{W}_i + \nabla \mathbf{W}_{i_j}\right) .$$

The extension of this method to unstructured grids is straightforward.

3.5 Conservative Hybrid-Grid Algorithm

Unlike the unstructured-grid formulation, the discrete convective fluxes are computed for a structured block as the sum of three contributions along each of the three mesh directions i, j, and k. It is therefore necessary to modify the numerical scheme at the hybrid-grid interfaces so that flux conservation is ensured. This section describes the numerical approach which makes it possible to accomplish this for the second-order schemes discussed above.

3.5.1 Hybrid Jameson–Smith–Turkel scheme

The methodology used to enforce flux conservation across a hybrid interface consists in computing its contribution to the numerical flux in exactly the same way in both domains.

From Eqs. (11) to (15), it is deduced that the contribution of a hybrid interface, which is denoted by h, to the integral (10) reads:

$$\mathbf{F}_{\mathbf{c}h} \cdot \mathbf{n}_h S_h = \mathbf{c}_h S_h + \left(\varepsilon_{2h}\mathbf{d}_{2h} - \varepsilon_{4h}\mathbf{d}_{4h}\right)\lambda_h S_h$$

where the index i refers to the interior cell adjacent to the block boundary, and g is the corresponding ghost cell bearing the information from the neighboring subdomain. Thus, the terms $\mathbf{d}_{2h}$ and $\mathbf{d}_{4h}$ are defined as

$$\mathbf{c}_h = \frac{1}{2}\left(\mathbf{F}_{\mathbf{c}}(\mathbf{U}_i) + \mathbf{F}_{\mathbf{c}}(\mathbf{U}_g)\right) \cdot \mathbf{n}_h \,; \tag{20}$$

$$\mathbf{d}_{2h} = \mathbf{U}_g - \mathbf{U}_i \,; \tag{21}$$

$$\mathbf{d}_{4h} = \delta^2 \mathbf{U}_g - \delta^2 \mathbf{U}_i \tag{22}$$

in which the second difference term $\delta^2 \mathbf{U}_i$ is given by relation (16). Figure 1 provides a schematic representation of the stencil used in the proximity of a hybrid-grid interface. It is clear from Eqs. (20) to (22) that only the artificial dissipation term $\mathbf{d}_{4h}$ needs a special treatment. The second-order dissipation

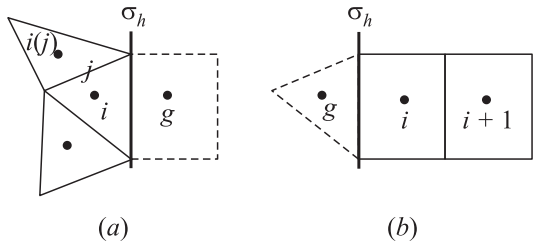


Figure 1 Hybrid-grid stencil near the boundary: (*a*) unstructured and (*b*) structured domains

term $\mathbf{d}_{2h}$ and the centered component $\mathbf{c}_h$ only require knowledge of the conservative variables in the adjacent block. They will thereby take the same values in both subdomains. The computation of $\mathbf{d}_{4h}$ involves, however, using information from interior cells beyond the immediate neighbors. Moreover, the second differences $\delta^2\mathbf{U}$ are evaluated in different ways depending on the type of domain. There are several alternatives to solving this problem. The approach adopted in *elsA-H* aims at remaining as consistent as possible with the structured method. Note that in the purely structured multiblock algorithm implemented in *elsA* two ghost layers are employed for a second-order scheme, whereas the purely unstructured and hybrid methods developed in *elsA-H* consider using one single ghost layer. Using the notation of Fig. 1, for an interface normal to the i-direction in the structured mesh, one can write:

$$\mathbf{d}_{4h}^S = \delta^2\mathbf{U}_g^U - \delta^2\mathbf{U}_i^S\,;$$
$$\mathbf{d}_{4h}^U = \delta^2\mathbf{U}_g^S - \delta^2\mathbf{U}_i^U$$

in which the superscripts S and U refer to quantities computed following the structured and unstructured methods, respectively. The second difference terms are then computed as follows:

$$\delta^2\mathbf{U}_g^U = \sum_{j(g)} \left(\mathbf{U}_{g_j} - \mathbf{U}_g\right)\,; \tag{23}$$
$$\delta^2\mathbf{U}_i^S = \mathbf{U}_{i+1} - 2\mathbf{U}_i + \mathbf{U}_g$$

which defines $\mathbf{d}_{4h}^S$ in the structured domain. Similarly, we have

$$\delta^2\mathbf{U}_g^S = \mathbf{U}_{i+1} - 2\mathbf{U}_i + \mathbf{U}_g\,;$$
$$\delta^2\mathbf{U}_i^U = \sum_{j(i)} \left(\mathbf{U}_{i_j} - \mathbf{U}_i\right)$$

which define $\mathbf{d}_{4h}^U$ in the unstructured domain. It is obvious from these relations that $\mathbf{d}_{4h}^S = -\mathbf{d}_{4h}^U$ and, therefore, flux conservation is guaranteed across the hybrid interface. Note that the sum in Eq. (23) contains a contribution from the

ghost (structured) cell equal to $(\mathbf{U}_g - \mathbf{U}_i)$. Note as well that in order to evaluate these terms, the values of $\delta^2\mathbf{U}_g^U$ and $\delta^2\mathbf{U}_g^S$ must be precomputed and stored in the corresponding ghost cell. This is done in two communication steps. First, the updated values of the conservative variables are exchanged and used to compute the values of $\delta^2\mathbf{U}_i$ in each domain (in those cells adjacent to a hybrid-grid interface). Second, the newly calculated differences are exchanged between adjacent blocks and stored in the ghost cells.

3.5.2 Hybrid Roe–MUSCL scheme

In the case of Roe's scheme, the procedure is a little simpler. Since the flux at the interface is computed from the values of its left and right states, it is needed just to make sure that these states are the same in both domains, in which case flux conservation will be automatically satisfied. As known from subsection 3.4, the left and right states at an interface j are given by Eqs. (17) and (18), respectively. At a hybrid interface, one has:

$$\mathbf{W}_h^L = \mathbf{W}_i + \frac{1}{2}\,\Phi\left(\nabla\mathbf{W}_i \cdot \mathbf{d}_{i\to g}, \mathbf{W}_g - \mathbf{W}_i\right) ;$$

$$\mathbf{W}_h^R = \mathbf{W}_g - \frac{1}{2}\,\Phi\left(\nabla\mathbf{W}_g \cdot \mathbf{d}_{g\to i}, \mathbf{W}_g - \mathbf{W}_i\right)$$

in which the gradient terms are constructed for the unstructured, as well as for the structured grid, using the Green–Gauss formula:

$$\nabla\mathbf{W}_i = \frac{1}{V_i}\sum_{j(i)}\frac{1}{2}\left(\mathbf{W}_{j(i)} + \mathbf{W}_i\right)\cdot\mathbf{n}_j S_j\,.$$

Effective computation of $\mathbf{W}_h^L$ and $\mathbf{W}_h^R$ requires knowledge of the quantities $\mathbf{W}_g$ and $\mathbf{d}_{g\to i}$ in the adjacent domain. As explained in the previous section, this is achieved via two communication steps. In the first step, the values of the recently computed conservative variables are transferred to the adjacent block. Using these values, the gradients of the primitive variables can be calculated and exchanged together with the position of the cell centroids.

3.6 Boundary Conditions

At inflow/outflow boundaries, the boundary conditions imposed are based on characteristic wave relations derived from the Euler equations. The local algebraic relations are written at the boundary face centers, regardless of the type of grid, as

$$\vec{\ell}_i^{\,n}\left(\mathbf{W}^n - \mathbf{W}^*\right) = 0$$

where $\mathbf{W}^n$ is the primitive variables vector at time t_n and $\mathbf{W}^*$ represents an internal state which is obtained via local reconstruction of the computed flow variables. This reconstruction is based here on a low-order directional extrapolation. The quantities $\vec{\ell}_i$ are the left eigenvectors of the Jacobian matrix of the flux vector expressed in terms of the primitive variables. The characteristic relations associated with an incoming wave are replaced by suitable Dirichlet or Neumann boundary conditions in order to close the system.

At inviscid or viscous walls, slip or no-slip boundary conditions, respectively, are imposed.

Rotational periodicity, however, is not treated as a physical boundary condition in *elsA-H*, but in the way explained in subsection 3.5. The flux terms are therefore evaluated at the periodic boundary using the information contained in the ghost cells. In order to account for the angular shift between this boundary and its partner boundary, an appropriate rotation must be applied to all vector quantities coming from the partner domain.

3.7 Implicit Time-Marching Scheme

This section presents the implicit time discretization scheme implemented in *elsA-H*. It is based upon the implicit backward Euler scheme with LU-SSOR approximate factorization. The method can be split into two steps, namely, the derivation of the linear system to be solved, and its actual solution.

The semi-discrete form of Eq. (4) reads

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{1}{V}\,\mathbf{R} \tag{24}$$

in which the R is given by (9). Using the implicit backward Euler scheme, the discrete form of Eq. (24) reads

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = -\frac{1}{V}\,\mathbf{R}^{n+1}\,. \tag{25}$$

The right-hand side of Eq. (25) can be linearized using a second-order Taylor expansion as

$$\mathbf{R}^{n+1} = \mathbf{R}^n + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\,\Delta U^n \tag{26}$$

where $\Delta \mathbf{U}^n = \mathbf{U}^{n+1} - \mathbf{U}^n$. Injecting Eq. (26) into Eq. (25) gives

$$A \Delta U^n = -\mathbf{R}^n\,; \qquad A = \frac{V}{\Delta t}\,\bar{\bar{I}} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}}\,. \tag{27}$$

The Jacobian matrix in Eqs. (27) is evaluated using the first-order van Leer flux vector splitting (FVS) scheme, so that A is sparse. The matrix A can be written

as the sum of a strictly lower triangular matrix $\mathcal{L}$, a diagonal matrix $\mathcal{D}$, and a strictly upper triangular matrix $\mathcal{U}$. The solution of the linear system is based on an approximation to the exact matrix:

$$A = (\mathcal{L} + \mathcal{D} + \mathcal{U}) \approx (\mathcal{L} + \mathcal{D})\mathcal{D}^{-1}(\mathcal{U} + \mathcal{D}) \tag{28}$$

and is obtained through the computation of a sequence of approximate solutions. Considering a relaxation method composed of forward and backward sweeps across the computational domain, one has:

$$(\mathcal{L} + \mathcal{D})\Delta\mathbf{U}^{p+1/2} = -\mathbf{R} - \mathcal{U}\Delta\mathbf{U}^{p}\,;$$
$$(\mathcal{U} + \mathcal{D})\Delta\mathbf{U}^{p+1} = -\mathbf{R} - \mathcal{L}\Delta\mathbf{U}^{p+1/2}$$

where p is the subiteration number. The maximum number of subiterations is chosen so that a satisfactory rate of convergence is obtained. In general, two subiterations are enough to provide reasonable results.

In the structured version of the LU-SSOR method, the relaxation sweeps are usually performed across the hyperplanes $i + j + k = ctt$ so that the matrix A can be written as in Eq. (28). In the forward sweep, the newly updated values at $(i-1, j, k)$, $(i, j-1, k)$, and $(i, j, k-1)$ are used, while the backward sweep involves using the values in cells $(i+1, j, k)$, $(i, j+1, k)$, and $(i, j, k+1)$. Unfortunately, this method does not extend easily to unstructured grids, and a special reordering procedure is thereby required. The reordering procedure used in *elsA-H* is the one proposed by Soetrisno *et al.* [21].

3.8 Parallel Implementation Strategy

All the features discussed in the previous sections are also available in parallel mode using *elsA-H*. Based on the same parallelisation strategy adopted in the original *elsA* system, *elsA-H* can run in parallel using the MPI communication library. Thus, taking advantage of the multiblock capability, each processor is responsible for handling a set of subdomains. The mapping between the blocks and the processors must be done using a suitable partitioning technique.

3.8.1 Hybrid-grid partitioning issues

To the best of our knowledge, there exists no hybrid-grid partitioner able to work on configurations composed of structured and unstructured blocks within the same computational domain. Some adaptation is, therefore, needed in order to tailor the existent tools to meet our needs.

The first solution would consist in performing the grid-partitioning operation on the structured and the unstructured blocks separately. The structured blocks can be partitioned using *elsA*'s own load balancing tool, which relies on geometric

considerations. On the other hand, the unstructured blocks can be processed using available unstructured-grid partitioners such as Scotch [22] or Metis [23], based on graph theory. To achieve a suitable load balance in the final hybrid partition, the number of subdomains in the partition must be the same for both the structured and the unstructured domains. This would guarantee that an equal number of blocks of each type can be mapped onto each processor, thereby ensuring that the same amount of work is allocated to each of them. This option, though seemingly straightforward, is somewhat restrictive, if not invalid in some particular cases. For instance, think of a case in which there is a predominant block-structured grid with only a few unstructured blocks in isolated patches distributed across the grid. This could well be the case of a shrouded turbomachinery blade with cooling holes, in which the area between the blade tip and the shroud as well as the holes would be more conveniently meshed using unstructured methods.

A more sophisticated approach would imply handling the structured and unstructured blocks simultaneously within the same partition process. A suitable set of weights must be assigned to each element type in order to account for the uneven computational effort involved in the simulation of the different regions. Note that in the case of an unstructured grid it appears more natural to assign those weights to the grid interfaces, as most of the operations are carried out in loops over the interfaces rather than the cells. The first guess for the weights can be obtained by performing a number of structured and unstructured reference computations on the same hexahedral grid, and studying the relative amount of work involved in each of the simulations. It is also important to account for the extra cost at hybrid-grid interfaces, which require additional operations. ONERA is currently working on this new tool based on the graph partitioning software Scotch.

3.8.2 Interblock communication across hybrid-grid interfaces

In *elsA-H*, the exchange of information between unstructured blocks (or processors in a parallel context) is performed via one layer of ghost cells (or *halo* cells), whereas the transfer of information between two structured blocks takes place via two ghost layers. It is straightforward to see that in the case of two blocks of different type, the restriction imposed by the unstructured solver implies that only one layer of halo cells be used in the implementation of the hybrid scheme.

As already discussed in subsection 3.5, the specific data exchanged between blocks or processors, depends on the numerical scheme considered. In the case of the JST scheme, it was noted that only the values of the second difference terms need to be transferred in order to compute the fourth-order artificial viscosity (15). The application of Roe's scheme with MUSCL reconstruction requires, on the other hand, that the gradients of the primitive variables and the position of the cell centroids be exchanged.

4 EULER SIMULATION RESULTS

This section gives several examples of the use of *elsA-H* for the solution of the Euler equations for several unstructured and hybrid-grid configurations.

4.1 Two-Dimensional Euler Simulation of the Flow Past a NACA 0012 Airfoil: Structured Versus Hybrid-Grid Solver

It is interesting to compare the performance of the structured and the hybrid solvers to simulate a given flow configuration on the same grid. To this end, a hexahedral structured grid has been transformed into a hybrid grid by destroying a portion of the computational domain to transform it into an unstructured subdomain (red subdomain in Fig. 2).

The flow conditions correspond to a Mach number of $M_\infty = 0.8$ and an angle of attack $\alpha_\infty = 1.25°$. The spatial discretization scheme used is the JST scheme with standard values of the artificial viscosity coefficients. The four-step Runge–Kutta and the implicit time integration schemes were tested for a Courant–Friedrichs–Lewy (CFL) condition of 1 and 1000, respectively.

Figure 3*a* compares the solutions obtained from both solvers. These results reveal a good agreement between both simulations.

The same conclusion is reached by inspecting the evolution of the residuals in Fig. 3*b*, which suggests that the convergence properties of the structured and the hybrid JST discretizations are comparable.

The comparison study presented in subsection 5.3 for a 3D Navier–Stokes configuration further corroborates these findings.

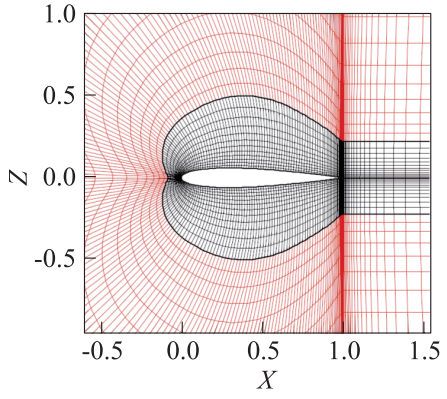


Figure 2 Hybrid grid around the NACA 0012 airfoil.

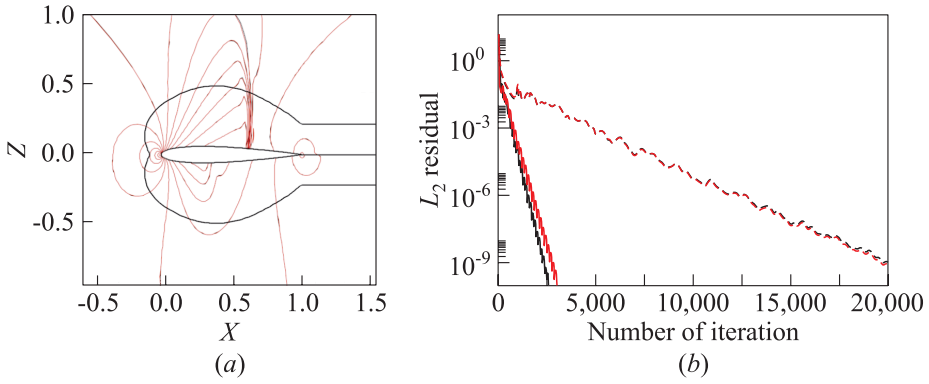


Figure 3 Simulation of the flow past a NACA 0012 airfoil: structured (black lines) vs. hybrid (red lines) solver: (*a*) isocontours of the Mach number; and (*b*) evolution of the L_2 residuals (solid lines — implicit, LU-SSOR; and dashed lines — explicit, four-step Runge–Kutta).

4.2 Three-Dimensional Euler Simulation of the Flow Through an Axisymmetric Channel

Euler simulations of the flow through an axisymmetric channel have been performed using the tetrahedral grids shown in Figs. 4*a* and 4*b*. The first config-

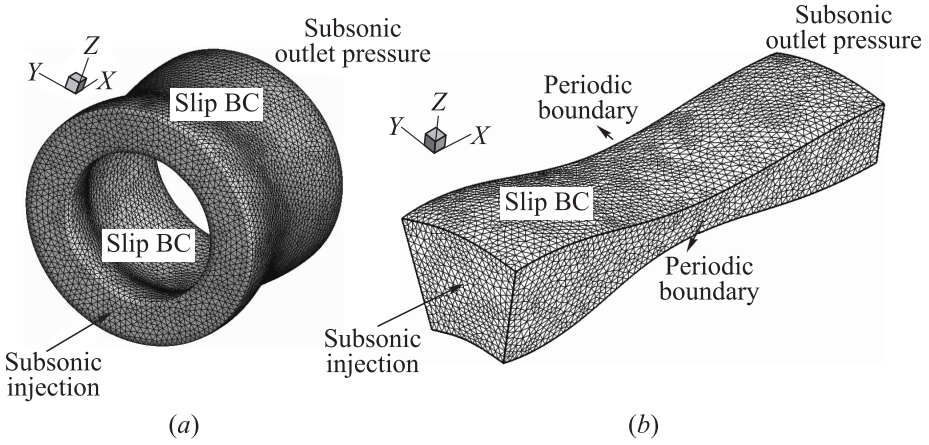


Figure 4 Three-dimensional axisymmetric channel configuration: (*a*) complete axisymmetric configuration; and (*b*) 30 degree segment. BC — boundary condition

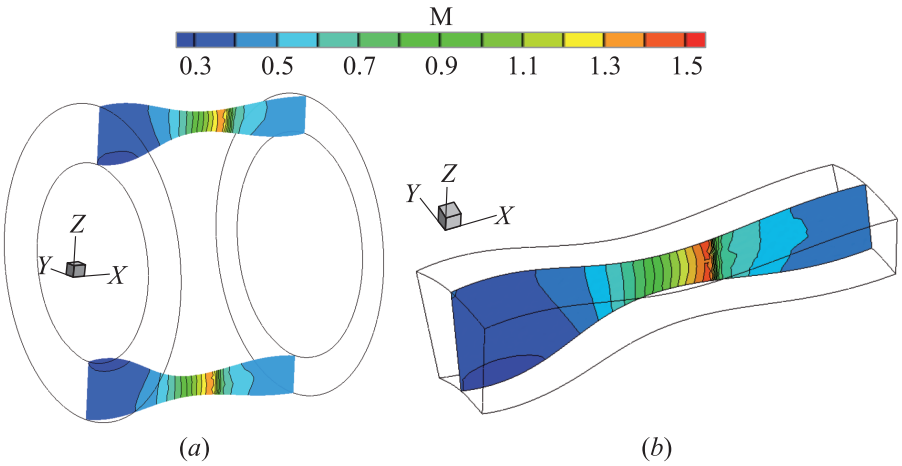


Figure 5 Flow through axisymmetric channel: flow solution in the channel (a) and flow solution in the channel segment (b).

uration corresponds to the complete channel, whereas the second represents an angular segment of 30° of the former, with imposed rotational periodicity in the azimuthal direction.

The equations are solved for the relative variables in a relative frame of reference rotating at the speed $\Omega = 3250$ rpm. A subsonic inflow condition is imposed at the inlet by specifying constant values of the total enthalpy, h_a, the total pressure, p_a, and the direction of the velocity vector, $\vec{d}_0$. At the outlet, the outflow boundary condition is imposed by providing the value of the static pressure p_s. The ratio of the static pressure at the outlet to the inlet total pressure corresponds to an outlet Mach number of 0.5. Slip boundary conditions are specified on the outer and inner walls. The simulations have been carried out using the JST scheme with implicit time integration. Figure 5 shows the isocontours of the Mach number for both configurations. Observe that the solutions are qualitatively similar. The small discrepancies found in the middle region of the channel are due to the fact that the tetrahedral grids are not perfectly periodic with respect to the azimuthal angle.

4.3 Three-Dimensional Euler Simulation of the Flow over a Turbine Blade

This subsection presents the results of an Euler computation on the unstructured grid shown in Fig. 6a. The simulation was performed using the JST scheme with

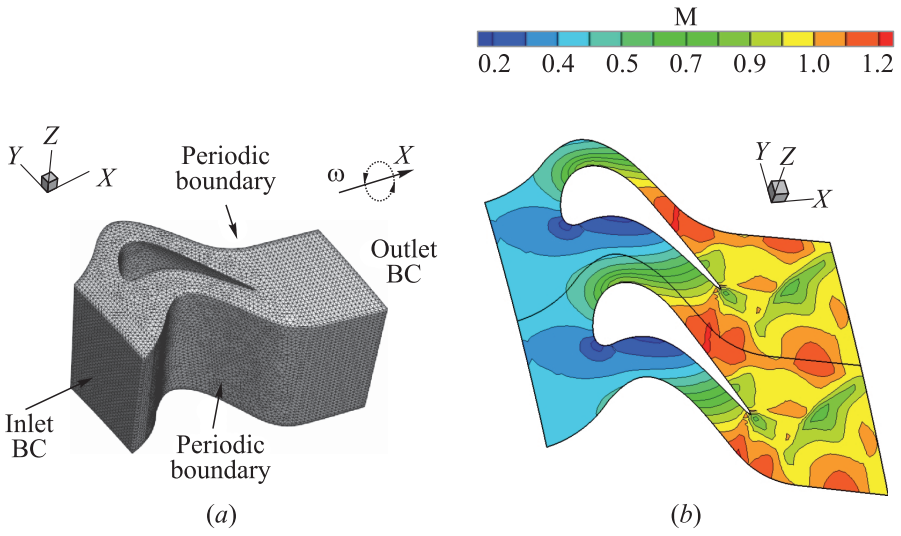


Figure 6 Euler simulation of the flow over a turbine blade: (*a*) unstructured grid; (*b*) iso-Mach contours.

implicit time-stepping at CFL = 100, and in a relative frame of reference rotating at the speed $\Omega = 18{,}600$ rpm.

The imposed boundary conditions are also shown in Fig. 6*a*. The inlet condition is specified via a data file in which the values of $(h_a, p_a, \vec{d_0})$ are provided in the relative frame of reference. The values of the static pressure at the outlet are also read from an external data file. These values come from a previous structured-grid RANS simulation of a geometrically similar configuration. Therefore, it has been necessary to map the data provided in the files onto the new unstructured grid boundaries by using a simple interpolation technique. Finally, rotational periodicity is imposed in the azimuthal direction, with period $\theta = 3.75°$.

Figure 6*b* shows the isocontours of the Mach number on a section at a constant radial position. The computational domain has been duplicated and rotated in order to emphasise the periodic character of the solution.

5 REYNOLDS-AVERAGED NAVIER–STOKES SIMULATION RESULTS

This section reports RANS simulation results obtained using *elsA-H* for a number of unstructured and hybrid-grid configurations.

5.1 Two-Dimensional Reynolds-Averaged Navier–Stokes Simulation of the VKI LS 89 Turbine Blade

Figure 7*b* shows simulation results obtained with *elsA-H* on a hybrid grid* composed of one unstructured (black) and four structured (coloured) domains, as illustrated in Fig. 7*a*. It corresponds to the viscous flow over the VKI LS 89 turbine blade cascade using the SA RANS model. The simulation has been performed using the standard second-order JST scheme and implicit time stepping. No-slip boundary conditions have been imposed on the viscous walls, whereas a translational periodicity condition is prescribed between the lower and upper bounds of the unstructured domain.

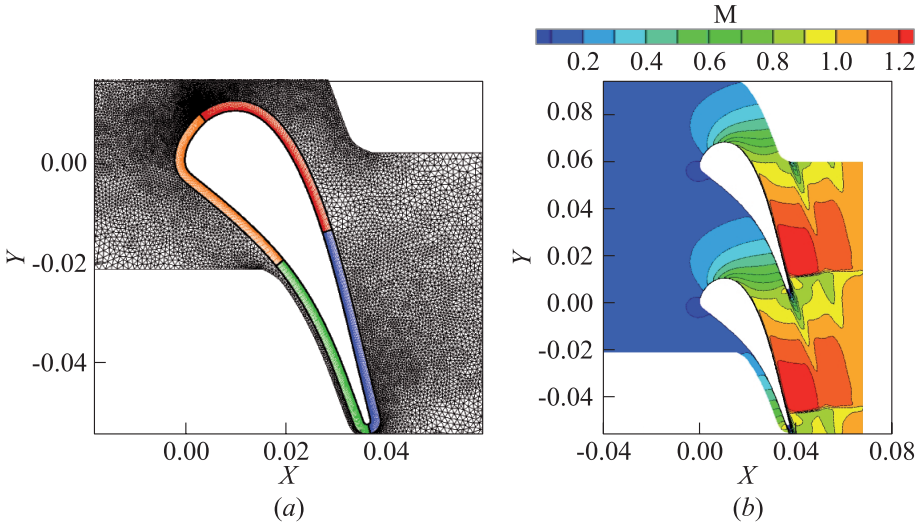


Figure 7 The RANS (SA) simulation of the flow over the VKI LS 89 turbine-blade cascade: (*a*) hybrid grid; and (*b*) iso-Mach contours.

5.2 Three-Dimensional Reynolds-Averaged Navier–Stokes Simulation of the ONERA M6 Wing

An unstructured-grid simulation of the ONERA M6 Wing (Fig. 8) has been carried out using the SA RANS model. The reference flow conditions are $M_\infty = 0.84$ and $\alpha_\infty = 3.06°$. In this case, the spatial discretization scheme used is Roe's scheme with MUSCL reconstruction and a Harten correction parameter of 0.01. The CFL number of the implicit simulation was set to 1000. The results of the simulation are shown in Fig. 9*a*, in which we can appreciate the good

*This grid has been generated using the freeware mesh generator Gmsh [24].

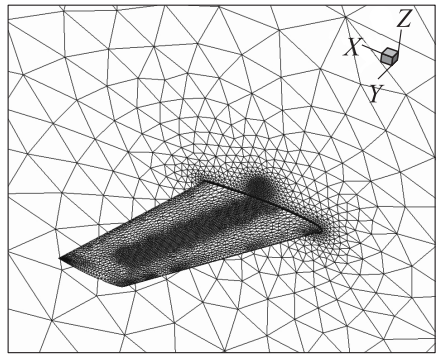
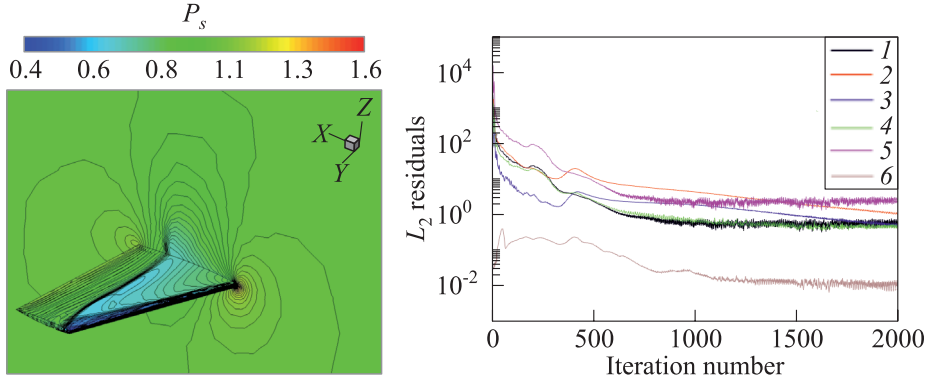


Figure 8 Unstructured grid for the ONERA M6 Wing configuration



(*a*) (*b*)

Figure 9 The RANS (SA) simulation of the ONERA M6 Wing configuration: (*a*) isocontours of the static pressure; and (*b*) evolution of the L_2 residuals: *1* — ρ; *2* — ρu; *3* — ρv; *4* — ρw; *5* — ρE; and *6* — $\rho\overline{v} \cdot 10^4$.

representation of the solution in the proximity of the shock. The evolution of the L_2 residuals of the solution is presented in Fig. 9*b*.

5.3 Three-Dimensional Reynolds-Averaged Navier–Stokes Simulation of a Rotating High-Pressure Turbine Blade: Structured vs. Hybrid-Grid Solver on a Hexahedral Configuration

A question often asked when discussing hybrid-grid methods is the following: if a given structured (therefore, hexahedral) mesh is considered, and several blocks

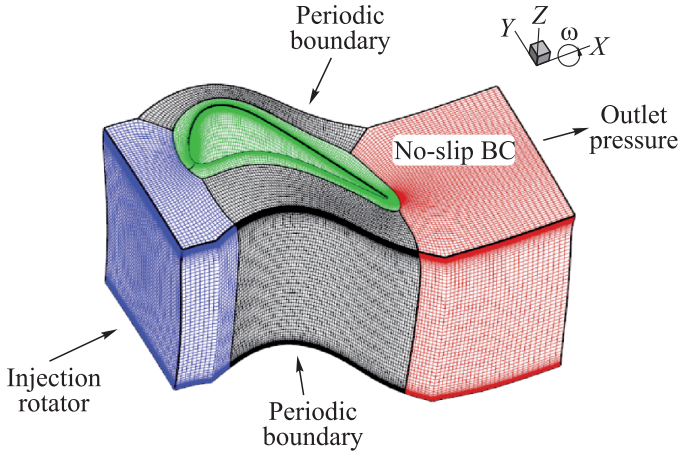


Figure 10 Five-block hexahedral hybrid mesh for the high-pressure turbine blade configuration. Unstructured blocks are black and structured blocks are coloured.

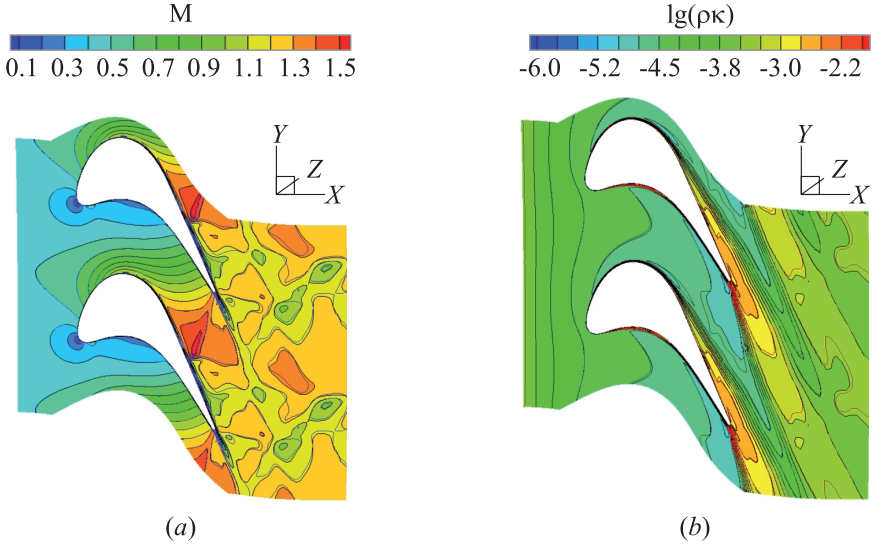


Figure 11 The RANS (κ–ω) simulation of the flow over a high-pressure turbine blade. Isocontour lines for a slice at a constant radial position. Structured (coloured lines) vs. hybrid (flood and black lines) simulations: (a) isocontours of the Mach number; and (b) isocontours of lg($\rho\kappa$).

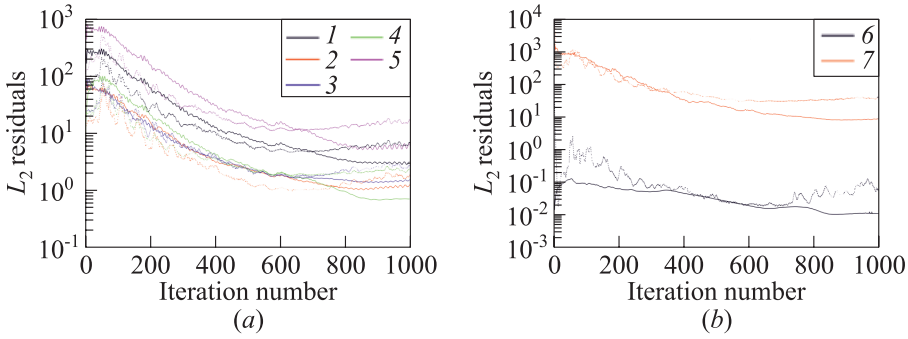


Figure 12 The RANS (κ–ω) simulation of a high-pressure turbine blade. Evolution of the L_2 residuals for conservative (a) and turbulent (b) variables. Structured (dotted curves) vs. hybrid (solid curves) simulation: *1* — ρ; *2* — ρu; *3* — ρv; *4* — ρw; *5* — ρE; *6* — $\rho\kappa$; and *7* — $\rho\omega \cdot 10^{-6}$.

are destroyed, how would the solution from the purely structured solver on the original mesh compare to the solution which results from the hybrid simulation on the new (hybrid) grid? This subsection is aimed at answering this question. With that goal in mind, the five-block hexahedral structured mesh shown in Fig. 10 has been modified so that the two blocks located at the extrados and the intrados of the blade have been transformed into unstructured blocks.

The RANS equations are solved in a relative frame of reference rotating at the speed $\Omega = 18\,600$ rpm, using the two-equation κ–ω model by Wilcox with SST (sea-surface temperature) correction. The inflow and outflow conditions are identical to those specified for the test case described in subsection 4.3. In this case, however, in addition to p_a, h_a, and $\vec{d}_0$, it is also necessary to specify the values of the turbulent variables κ and ω. Adiabatic no-slip conditions are prescribed on the blade skin and on the walls near the hub and the casing.

The results from the structured and the hybrid-grid simulations are compared in Fig. 11. A good agreement is observed between both solutions for the isocontours of the Mach number and the turbulent kinetic energy. It is seen from Fig. 12 that the evolution of the residuals and the number of time steps required to reach the steady state are comparable in both cases. Measurement of the CPU time consumed in both simulations reveals an extra cost of about 25% for the hybrid simulation.

5.4 Shrouded Stator of the Three-Stage CREATE Compressor

In this section, the results from the 3D RANS hybrid simulation of the CREATE shrouded stator are presented. Two different views of the hybrid grid are shown in Fig. 13. The grid contains two unstructured blocks, located upstream and

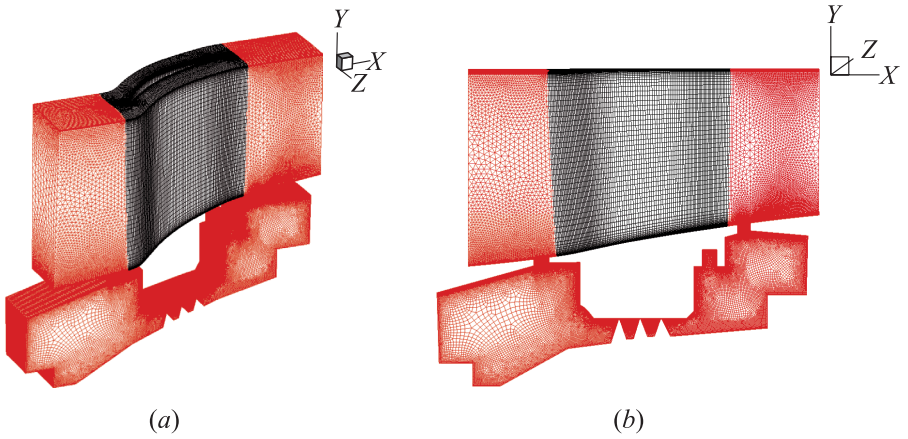


Figure 13 Hybrid grid of the CREATE configuration. Unstructured blocks are red. View in the XY plane is presented on (b).

downstream of the blade, respectively, which are mainly composed of tetrahedra, but also contain hexahedra (e. g., within the boundary layer) and pyramids next to the hybrid interfaces.

This is an interesting test case in which the use of an unstructured grid generator presents a significant advantage over its structured counterpart. In fact, meshing the cooling cavity underneath the hub (see Fig. 13) using a structured

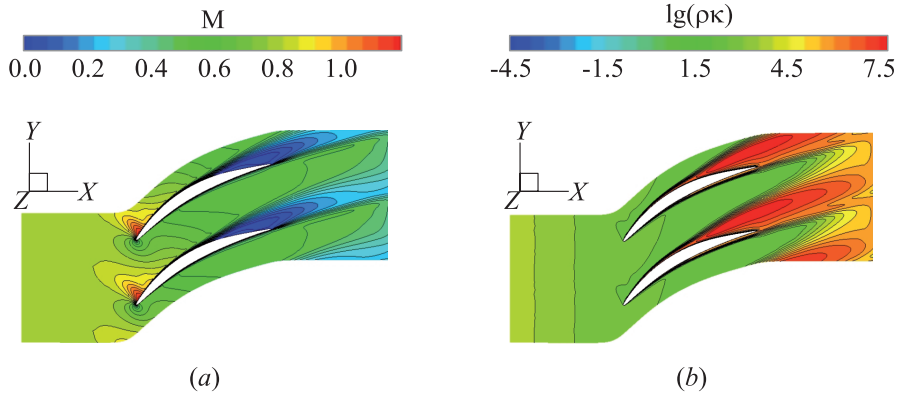


Figure 14 The RANS (κ–ω) simulation of the CREATE configuration. Isocontours for a slice at constant radial position: (a) Mach number; and (b) lg($\rho\kappa$).

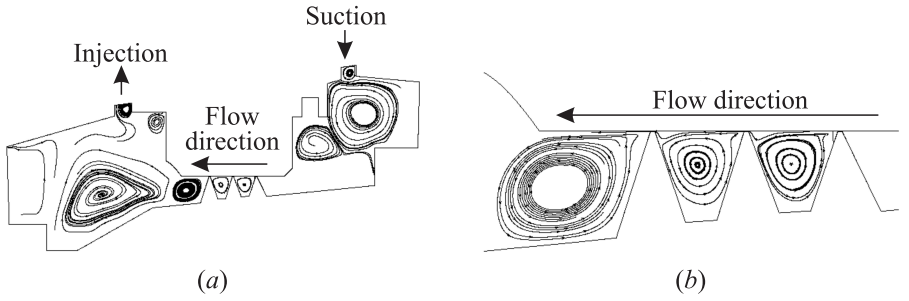


Figure 15 The RANS (κ–ω) hybrid simulation of the CREATE stator configuration. Streamlines around blade and in the cavity: (a) general view; and (b) detail of the flow through the middle region of the cavity

method requires the generation of over 30 blocks. In contrast, an unstructured grid generator can do the job by building one single block.

The RANS equations are solve based on the Wilcox κ–ω turbulence model in an absolute frame of reference, and are advanced in time using the implicit time integration. The two spatial discretization schemes discussed in subsection 3.4 were tested, which provided similar results. Here, the solution from the JST simulation is presented.

The imposed boundary conditions are a subsonic injection condition at the inlet with specified inlet data from a previous structured simulation (values of p_a, h_a, $\vec{d}_0$, κ, and ω), a static pressure at the outlet verifying the radial equilibrium condition (specified values provided in an input data file), and a rotational periodicity condition in the azimuthal direction. Adiabatic wall boundary conditions are imposed on all walls, including the cavity wall.

The simulation results are as shown in Fig. 14. It is interesting to have a look at the flow within the cavity as shown in Fig. 15. It is characterized by an aspiration of the main flow through the small passage on the right-hand-side of the cavity, and an injection into the main flow through the passage on the left-hand side. Figure 15b shows a close-up of the middle region of the cavity. These results are in agreement with the observations of Marty *et al.* [25] who performed similar simulations using a fully structured grid. The evolution of the residuals is shown in Figs. 16a and 16b for the conservative and the turbulent variables, respectively.

6 CONCLUDING REMARKS

The new 3D Navier–Stokes hybrid solver *elsA-H* is presented and validated on a variety of reference test cases and industrial configurations. The general architecture of the solver relies on the modular approach provided by the object-oriented

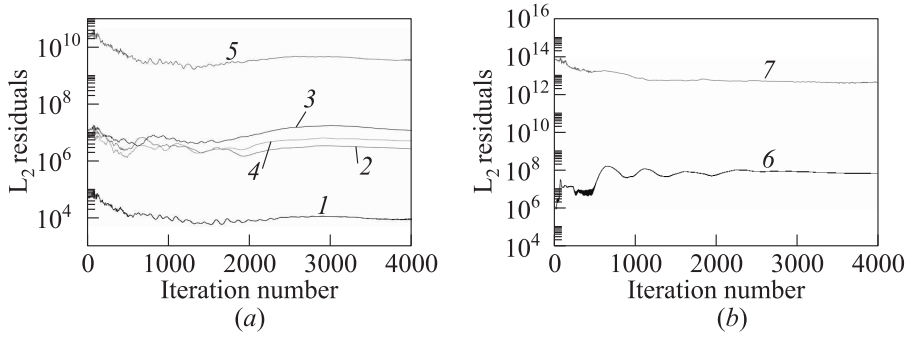


Figure 16 The RANS (κ–ω) simulation of the complete CREATE stator configuration. Evolution of the L_2 residuals for conservative (a) and turbulent (b) variables (κ–ω): *1* — ρ; *2* — ρu; *3* — ρv; *4* — ρw; *5* — ρE; *6* — $\rho\kappa$; and *7* — $\rho\omega$

framework upon which the structured CFD solver *elsA* is built. This has permitted for the implementation of a new unstructured module within the originally structured code.

A detailed description of the implemented numerical methods is provided, with special emphasis on the hybrid conservative methodology used at hybrid-grid interfaces. As is seen, standard second-order schemes can be easily adapted at the matching block boundaries so that flux conservation is guaranteed. This requires a communication step during which structured and unstructured subdomains must provide each other with the information necessary to the computation of the second-order fluxes at the interfaces (second differences in the case of the JST scheme, and gradients of the primitive variables and the position vector of the cell-centers in the case of the Roe-MUSCL scheme).

A work is currently carried out on the development of unstructured schemes based on reconstruction through gradients in order to improve the spatial accuracy of the hybrid schemes (second-order and higher). A comparison between the structured and the hybrid solvers is carried out on a multiblock hexahedral mesh for which two blocks were transformed into unstructured. These qualitative results suggest that the solutions obtained by using both solvers separately are comparable, in terms of quality and convergence properties.

The flexibility and performance of *elsA-H* is demonstrated on a complex turbomachinery configuration corresponding to the shrouded stator of the three-stage CREATE compressor. This configuration is an interesting example in which the use of hybrid methods represents a true advantage, since the generation of the mesh within the cooling cavity is significantly easier using an unstructured grid generator.

Today, the major challenge concerning hybrid solvers lies in the need for a hybrid-grid generator able to produce structured (ijk-type) and unstructured

meshes within the same computational domain. Efforts in this direction have to be made by the CFD community to develop such a tool, which when mature, will bring out the full potential of hybrid-grid CFD solvers.

ACKNOWLEDGMENTS

elsA-H is supported by the French Ministry of Transport (DGAC) and by the Safran Group. The authors would like to thank Snecma and Turbomeca for the grids provided and for permission to publish these results. The authors also wish to thank Dr. Bertrand Michel for technical support and very useful discussions.

REFERENCES

1. Lefebvre, M., V. Couaillier, and J. M. Duboué. 1998. Numerical methods on adaptive hybrid grids for the solution of Euler and Navier–Stokes equations. *4th ECCOMAS Computational Fluid Dynamics Conference Proceedings*. Athens, Greece. John Wiley & Sons, Ltd. 184–96.
2. Lefebvre, M. 1998. Développement de nouvelles techniques numériques pour la résolution des équations de Navier–Stokes tridimensionnelles stationnaires sur des maillages hybrides "structurés/non-structurés." Ecole Centrale de Lyon, France.
3. Yang, H., D. Nuernberger, and H. P. Kersken. 2006. Toward excellence in turbomachinery Computational Fluid Dynamics: A hybrid structured-unstructured Reynolds-averaged Navier–Stokes solver. *J. Turbomachinery* 128:390–402.
4. Wissink, A., M. Potsdam, V. Sankaran, J. Sitaraman, Z. Yang, and D. Mavriplis. 2010. A coupled unstructured-adaptive Cartesian CFD approach for hover prediction. *American Helicopter Society 66th Annual Forum and Technology Display Proceedings*. Phoenix, Arizona.
5. Cambier, l., and M. Gazaix. 2002. elsA: An efficient object-oriented solution to CFD complexity. AIAA Paper No. 2002-0108.
6. Spalart, P. R., and S. R. Allmaras. 1994. A one-equation turbulence model for aerodynamic flows. *Recherche Aerospatiale* 1:5–21.
7. Wilcox, D. C. 1994. Simulation of transition with a two-equation turbulence model. *AIAA J.* 32(2):247–55.
8. Vuillot, A. M., V. Couaillier, and N. Liamis. 1993. 3D turbomachinery Euler and Navier–Stokes calculations with a multidomain cell-centered approach. AIAA Paper No. 1993–2576.
9. Boniface, J., P. Guillen, M. C. Le Pape, D. Darracq, and P. Beaumier. 1998. Development of a Chimera unsteady method for the numerical simulation of rotorcraft flow fields. AIAA Paper No. 1998-0421.
10. Cambier, L., and J. P. Veuillot. 2008. Status of the elsA CFD software for flow simulation and multidisciplinary applications. AIAA Paper No. 2008-664.

11. Girodroux-Lavigne, P. 2007. Recent Navier–Stokes aeroelastic simulations using the elsA code for aircraft applications. *International Forum on Aeroelasticity and Structural Dynamics Proceedings*. Stockholm, Sweden.
12. Girodroux-Lavigne, P. 2009. Fluid–structure coupling using Chimera grids. *International Forum on Aeroelasticity and Structural Dynamics Proceedings*. Seattle, USA.
13. Peter, J. E. V., and R. P. Dwight. 2010. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers Fluids* 39(3):373–91.
14. The CGNS steering sub-committee of the AIAA CFD committee on standards. 2005. The CFD General Notation System. Standard Interface Data Structures. AIAA R-101A-2005.
15. Jameson, A., W. Schmidt, and E. Turkel. 1981. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. AIAA Paper No. 1981-1259.
16. Jameson, A., and T. J. Baker. 1987. Improvements to the aircraft Euler method. AIAA Paper No. 1987-0452.
17. Roe, P. L. 1981. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* 43(2):357–72.
18. Yee, H. C., R. F. Warming, and A. Harten. 1985. Implicit total variation diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.* 57(3):327—60.
19. Godunov, S. K. 1959. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Math. Sbornik* 47:271–306.
20. Van Leer, B. 1979. Towards the ultimate conservative difference scheme. V. A second order sequel to Godunov's method. *J. Comput. Phys.* 32(1):101–36.
21. Soetrisno, M., S. T. Imlay, and D. W. Roberts. 1994. A zonal implicit procedure for hybrid structured-unstructured grid. AIAA Paper No. 94-0645.
22. Pellegrini, F. 2010. *Scotch and LibScotch 5.1 User's Guide*. INRIA Bordeaux Sud-Ouest, IPB & LaBRI.
23. Abou-Rjeili, A., and G. Karypis. 2006. Multilevel algorithms for partitioning power-law graphs. *IEEE International Parallel & Distributed Processing Symposium (IPDPS) Proceedings*.
24. Geuzaine, C., and J. F. Remacle. 2009. Gmsh: A 3D finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.* 79:1309–31.
25. Marty, J., B. Aupoix, and J. Bert. 2008. Interaction of shrouded stator flow and main flow and its influence on performances of a three-stage high pressure compressor. ONERA Internal Report.