



Hybrid grid generation for two-dimensional high-Reynolds flows

D. Dussin^a, M. Fossati^b, A. Guardone^{b,*}, L. Vigevano^b

^aCD-adapco, Computational Dynamics, Via Siviglia 25, 00143 Roma, Italy

^bPolitecnico di Milano, Department of Aerospace Engineering, Via La Masa 34, 20156 Milano, Italy

ARTICLE INFO

Article history:

Received 15 July 2008

Received in revised form 6 April 2009

Accepted 8 April 2009

Available online 3 May 2009

ABSTRACT

An hybrid mesh generation algorithm for two-dimensional viscous flows at high-Reynolds number is presented. An advancing-front method is used close to solid surfaces and in the wake region(s). The boundary layer and wake grids possibly contain both highly stretched quadrilateral and triangular elements. The latter are inserted locally to improve the quality of the grid, thus circumventing some drawbacks of standard structured grid advancing-front methods. An advancing-front/Delaunay algorithm triangulates the remaining portion of the computational domain. An overview of both algorithms is given and results for viscous laminar and turbulent compressible flows around single and multi-element airfoils are shown to support the present approach.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Viscous flows at high-Reynolds numbers around aerodynamic bodies are characterized by the presence of boundary layers and wakes which strongly influences the overall aerodynamic forces acting on the body itself. In fact, although these regions usually cover only a small portion of the flowfield, an incorrect evaluation of the flow quantities in boundary layers and wakes may affect the solution in the whole flowfield, due to the large gradients of the flow variables in these regions. Moreover, the accurate evaluation of the velocity and temperature gradients at solid boundaries is mandatory for a correct prediction of both the drag due to friction and the heat flux.

In flow solvers relying on computational meshes to obtain a geometrical description of the flowfield, the local accuracy of the solution depends on both the local size of the grid elements and on the magnitude of the solution gradients [21]. It is therefore appropriate, to minimize the discretization errors, to cluster grid points in regions where large gradients occur: for compressible viscous flows, these include also shock and contact surfaces. If the local gradient of the solution is aligned to specific directions, the computational effort can be further reduced by the adoption of anisotropic elements, namely, by elements whose characteristic dimension along the solution gradient is much lower than that normal to it. This is indeed the case of the boundary layer region, in which the solution is slowly varying along the streamwise direction and solution gradients are usually aligned along the normal to the body surface. Therefore, the most suitable elements for the

boundary layer regions are characterized by high stretching and are aligned with the local flow direction.

Both structured and unstructured grid generation techniques are capable of providing the anisotropic discretization required in high-Reynolds flows computations. For example, anisotropic unstructured grids of triangles or tetrahedra can be obtained via the advancing-front/Delaunay approach [22,10] by specifying a suitable Riemannian metrics which locally depends on the body geometry or on the flow features [17]. However, highly stretched grids result in very small internal angles of the triangles/tetrahedra, which is often a source of additional discretization errors [1,21]. Moreover, although the elements are mostly aligned with the flow, some grid segments are not, which usually results in a relevant amount of artificial dissipation being introduced in the streamwise direction and hence to numerical errors. On the contrary, structured grids made of quadrilaterals or hexaedra are indeed very well suited in the boundary layer region, since internal angles are not affected by the elements' stretch. As it is well known, however, structured grid methods are by their very nature usually unsuitable for complex geometries, such as those usually considered in aerospace or turbomachinery applications [31].

An alternative approach, which combines the advantages of both structured and unstructured grid techniques, is the use of the so-called hybrid grids, in which boundary layer regions are described by structured grids, which allow for good-quality highly stretched elements, whereas the outer domain is discretized by means of standard isotropic unstructured grids for geometrical flexibility [20,29]. The grid describing the computational domain is therefore unstructured and hybrid, namely, it is made of elements of different types. In two-dimensional applications, these are quadrilateral elements in the boundary layer and wake regions and triangles in the remaining portion of the domain, see Fig. 1.

* Corresponding author. Tel.: +39 0223998393; fax: +39 0223998334.
E-mail address: alberto.guardone@polimi.it (A. Guardone).

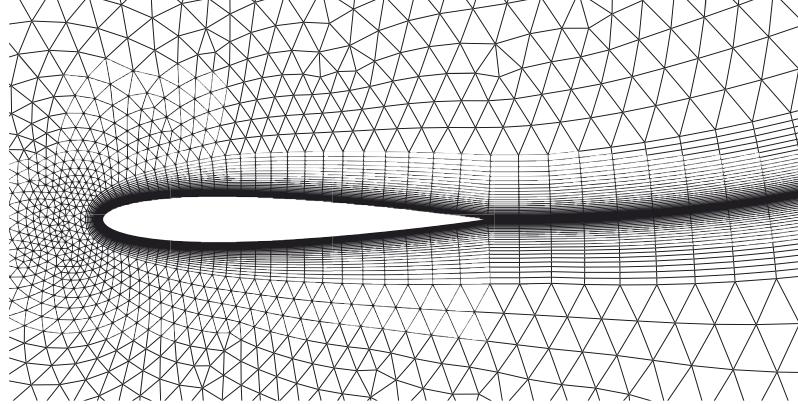


Fig. 1. Two-dimensional hybrid grid around a NACA 0012 airfoil, including the structured boundary layer grid of quadrilateral elements and the outer unstructured grid of triangles.

Three-dimensional applications can be found in Ref. [14]. Flow solvers have been modified accordingly to deal with grids made of elements of different types. In particular, edge-based solvers [25,26,2,6] have been developed in which geometrical data are described in terms of edges of the triangulations, namely, by the segments connecting two grid nodes belonging to the same element. Edges are indeed the simplest structure which is common to all elements and also allows for an easy extension of existing two-dimensional solvers to three-spatial dimensions.

The hybrid approach is now widely used for generating grids suitable for high-Reynolds flow simulations and a variety of approaches to boundary layer grid generation can be found in the literature. These include methods relying on an existing preliminary triangulation whose elements are then combined to form quadrilateral elements close to solid boundaries [19,18]. Structured grid techniques have also been used in which an existing structured mesh is locally refined in the boundary layer region. Transition to coarser structured grid regions is achieved by introducing triangles in between structured grid regions [13]. However, most commonly used structured grid generation techniques for boundary layer do not rely upon an existing triangulation. These are, for example, elliptic schemes in which an elliptic set of equations is solved in the domain to obtain the node's coordinates [24,33]. Advancing-front methods are instead based on the idea of creating a element layer or front along the surface of the body and to propagate it outward up to the required thickness, see Fig. 2. Hyperbolic and algebraic schemes are examples [31]. In the former, the outward direction for front propagation is obtained by solving an hyperbolic system of equations, whereas in algebraic schemes the propagation direction is simply obtained as an average of the local direction

normal to the front itself. In advancing-front schemes, front collisions can possibly occur and front overlap is to be controlled explicitly. Exemplary situations are the overlap of two boundary layer grids around different bodies close to each other (collision of two fronts) or the overlap of a front growing from a convex geometry (single-front collision), in which the grid is literally “turned into itself” as the front advances. Concave geometries represent indeed the most challenging problem, in terms of both front overlap and grid quality. In this case, points accumulates near the center of the concave internal domain, thus leading to low-quality elements, often up to the point of preventing a complete mesh generation after a number of element layers has been inserted. Possible solutions to these problem include modifications to the front advancing behavior, namely, corrections to the insertion directions for new grid points. The above is easily accomplished in hyperbolic grid generation by adding dissipative terms to the set of equations to be solved [31]. Alternatively, the height of new quadrilaterals can be reduced to fit all layers into the available space [9,12]. As an alternative, front overlap can be dealt with after the grid generation, by simply eliminating the elements which overlap and by filling the gaps with triangles [30,16,9].

In the present paper, a novel approach to hybrid grid generation in two-spatial dimensions is presented and applied to complex geometries. The proposed approach uses an advancing-front scheme for boundary layer grids; the advancing-front/Delaunay technique of Rebay [22] is then used to generate the unstructured grid in the remaining portion of the computational domain. The main feature of the present scheme lies in the generation of the boundary layer grids, which are mainly made of highly stretched quadrilateral elements but contains also triangles that are locally

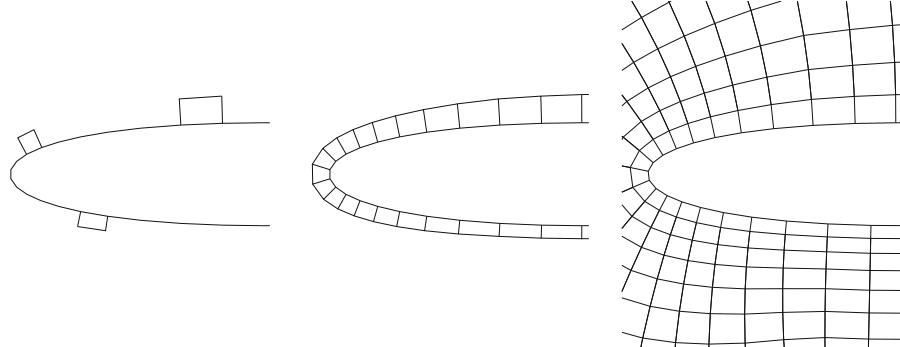


Fig. 2. Structured grid advancing-front scheme. Left: the first layer element size along the streamwise and normal direction is specified at a number of control points. Center: the first layer is generated respecting the imposed metric. Right: subsequent layers or fronts are then included.

inserted to cure the aforementioned drawbacks of standard front-advancing techniques for convex and concave geometries. Such hybrid grids *within* the boundary layer region is meant to ensure a smooth transition between the boundary layer grid itself and the outer unstructured grid, which is usually not assured by standard front-advancing methods. In fact the latter produces at the outer layer very stretched elements for highly convex/concave geometries. The present method, instead, assures at the boundary layer grid limit the *same* metric characteristics as specified by the user over the body geometry. We refer to this feature as an increased grid quality, which can be quantified by a suitable smoothness indicator, as shown later on. Finally the proposed approach is highly automatic, since a very limited user interaction is required to produce a high-quality mesh with specified anisotropic element dimensions.

The algorithm for the generation hybrid grid is presented as follows. In Section 2, the advancing-front technique for boundary layer and wake grids is described. In particular, the strategies for triangles insertion in convex and concave geometries are detailed in Sections 2.1 and 2.2, respectively; front collision is dealt with in Section 2.3. In Section 2.4, a short description of the advancing-front/Delaunay schemes is given and the procedure to match boundary layers grids and the triangulation of the outer domain is described. A brief comment on grid quality assessment is given in Section 3. Finally, in Sections 4.1, 4.2 and 4.3, laminar compressible flows around the NACA 0012 airfoil and turbulent compressible flows around the RAE 2822 and around a two-element airfoil configuration are studied to assess the performance of the proposed technique. Numerical results are compared with available numerical and experimental data.

2. The hybrid front-advancing method

In the present section, the front-advancing method for boundary layer and wake grid generation is detailed. The grid front is initially made of the boundaries of the domain around which a boundary layer grid is to be generated. In addition, one or more curves representing wakes inside the domain can be specified by the user as polylines, Fig. 3. Element sizes in the direction tangent to the boundary—the boundary nodes spacing—is imposed at user-defined control points along the boundary, together with the local height of the first layer elements (length in the normal direction) and the metric propagation function, see Fig. 2.

The point insertion inside the computational domain then proceeds as follows. For each node P_i belonging to the i th layer of elements, the local outward normal to the front \mathbf{n}_i is computed as the average value of the outward normal to neighboring elements. A new grid point P_{i+1} is then inserted in the next layer with coordinates $P_{i+1} = P_i + d_i \mathbf{n}_i$ where d_i is the local height of the new elements, namely, the thickness of the new element layer. The height d_i at node i is computed from its corresponding value at

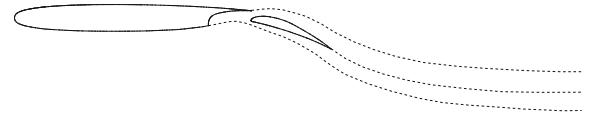


Fig. 3. Arrangement of wake polylines for a two-element airfoil configuration, see also Figs. 19–22.

the boundary and modified in accordance with the user-specified metric propagation law. Points are stored into circular bidirectional lists, one for each independent advancing front. The element connectivity is then recovered from the point data structure.

The point insertion procedure outlined above allows for the creation of subsequent layers of quadrilateral elements which form the boundary layer or wake grids. However, in the presence of highly convex or concave geometries, the resulting grid no longer satisfies the prescribed element size and stretching. Moreover, front overlap can possibly occur in highly concave geometries. These cases require a special treatment as detailed in the next sections, where the grid generation procedure is described.

2.1. Convex geometries

Standard front-advancing schemes for boundary layer grids usually fail to preserve the user-specified element size and stretch around highly convex geometries, as shown, e.g., in Fig. 4 (left). In fact, to keep the number of points/elements constant on each new layer, the distance between two given adjacent grid points increases as the distance from the body surface increases.

The limitation of structured grid front-advancing methods is here circumvented by selectively adding triangular elements as the front advances to keep the size constraints set by the user. This amounts to split some of the quadrilateral elements into three triangles as shown in Fig. 4 (right). The element splitting is controlled by enforcing a maximum value for the element stretching, namely, for the value of the ratio of the element size along the normal and tangential directions of the front. To allow for a variable maximum stretch value, the latter is specified by the user at control points along the body and propagated to subsequent layer as the front advances. After completion of a new layer, the actual stretch value of each quadrilater is compared to the user-assigned function and the quadrilaterals which do not satisfy the constraint are split into three triangles.

Fig. 5 shows boundary layer grid near a sharp edge, where the element stretch is usually poorly controlled by structured front-advancing schemes (left plot). The introduction of a limited amount of triangular element allows in this case to recover the required element sizes and also to enhance the grid quality. In Fig. 5 (center and right plots), an automatic treatment for sharp corners is also clearly visible in the first layer of elements, where two triangles are introduced. In this way, the grid quality is improved

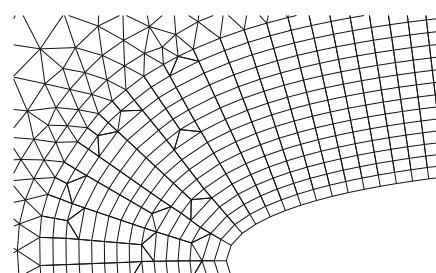
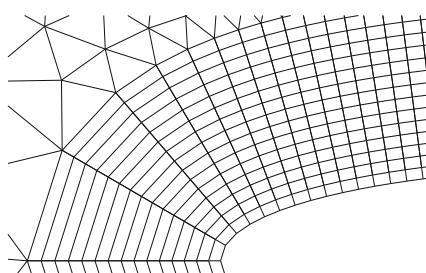


Fig. 4. Boundary layer close to the airfoil leading edge. Left: standard structured boundary layer grid of quadrilateral elements. Right: the hybrid solution.

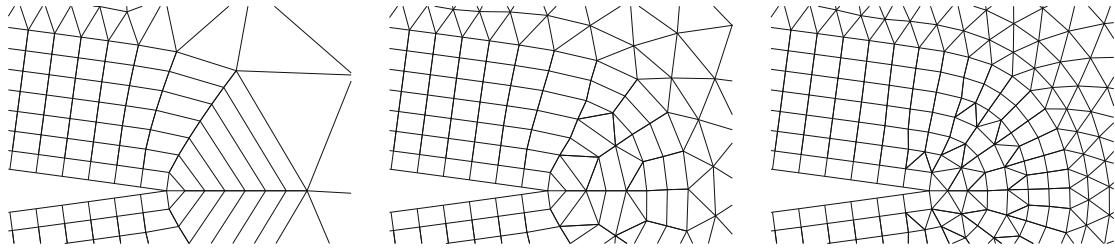


Fig. 5. Boundary layer grid close to the airfoil trailing edge. Different meshes obtained by varying the stretch bounding limit. The grid on the left is obtained by imposing a large value of the limit, which is decreased to a value of about two and one to obtain the grid in the center and on the right, respectively.

and the edge separating the two triangles can be used as the first portion of a fictitious boundary stemming from the trailing edge, namely, of a polyline curve representing the wake in the computational domain.

It should be noted however that the splitting of quadrilaterals into three triangles can result in very low-quality triangular elements in the presence of highly stretched grids. In these cases, the minimum angle of the inserted triangles can be in fact very low.

2.2. Concave geometries

The hybrid advancing-front method is now described in the case of concave geometry. An example is given in Fig. 6 (left), where a structured boundary layer grid around a convex corner is obtained by means of an hyperbolic grid generation technique. A standard solution to avoid front overlap in this case is to locally change the front propagation direction. Such a solution is however not satisfactory in terms of grid quality and of fulfillment of prescribed element sizes. As in the convex case, the introduction of triangles during the front advancing process is proposed here to preserve the specified metrics at the boundary layer grid outer limit, as shown in Fig. 6 (right). In this way, the local propagation direction of the front is not modified.

The triangles insertion procedure is as follows. After completion of the current grid layer, an index N_{ol} is assigned to each quadrilateral element; N_{over} is the number of layer after which the sides of the considered element normal to the front will cross, thus leading to an inadmissible element, and it is easily computed from the local advancing direction. If $N_{over} \leq 0$, the quadrilateral is substituted by a triangle by simply condensing its two upper points on their center of gravity.

This approach produces satisfactory results in the presence of highly concave boundaries, as it is the case for example of the sharp corner shown in Fig. 6. On the contrary, element collapsing due to low-curvature concave boundaries usually requires numerous steps to be performed before occurring and the aforementioned special treatment is usually deferred up to a stage at which new triangles are highly stretched along the advancing direction. To avoid the introduction of such low-quality triangles, an alternative approach to quadrilateral substitution is devised which does not only prevent quadrilateral elements from collapsing but also guarantees an adequate quality of the additional triangles. Adjacent couples of quadrilateral elements located in a concave front region (thick continuous line in Fig. 7) are monitored as the front progresses and taken into consideration for a possible transformation into a triangle, whose shape is subject to a quality constraint. To this purpose, a minimum angle α_{min}

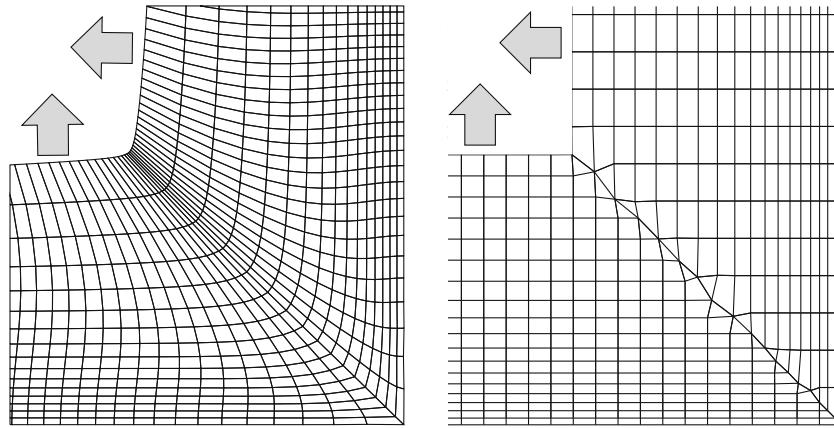


Fig. 6. Front intersection at a concave corner (front is advancing from bottom right to top left). Left: standard hyperbolic front advancing scheme. Right: present method.

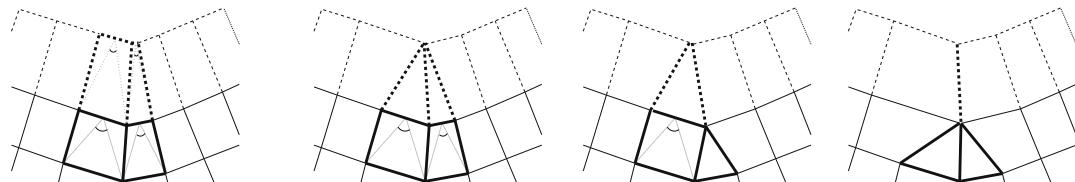


Fig. 7. From left to right: quality of possible triangles to be inserted, hybrid solution for upper vertex angles greater than 10°, greater than 30° and greater than 40°.

is prescribed by the user which controls the minimum allowed angle encompassed by the two segments of the triangle having the triangle's upper vertex in common. Four triangular elements are considered at the same time; these are the triangles obtained by collapsing the two considered quadrilateral elements as well as the two corresponding elements in the next layer. Starting from the two active elements and proceeding to the next layer as in Fig. 7, the quality constraint is enforced and quadrilateral elements with associated triangles violating the minimum angle rule are collapsed. An exemplary application of this procedure is given in Fig. 8.

It is however to be noted that the local character of the present approach can be unsatisfactory in the presence of very highly concave geometry, in which elements collapsing along the front occurs at such a rate that collapsing two or one element per layer cannot be sufficient. In this case, a local modification of the advancing direction is also enforced as it is standard practice in front-advancing schemes.

2.3. Front overlap

The grid generation procedure for concave and convex boundaries outlined in the previous section guarantees a high-quality meshes and prevents front overlapping for grids originating from a given boundary. It is however possible that fronts moving from different boundaries can collide, as it is often the case for multiple connected geometries, e.g., the flapped airfoil configurations in Fig. 9; in this case, front collision is to be treated explicitly.

Differently from other approaches—in which front collision is usually checked only after the completion of the grid generation process—front collision tracking is performed here at each step of the front-advancing scheme, to take advantage of the active data structure describing the current layer of elements. In this way, no grid post-processing, such as element cutting or stretching, is required and the grid quality is locally preserved also if front collision occurs.

To verify the occurrence of front overlap, the front position as a whole is tracked my means of an adaptive tree data structure. A bounding box for each front is first built containing all points belonging to a given layer. If bounding boxes belonging to different fronts share common regions of the domain, then each over-

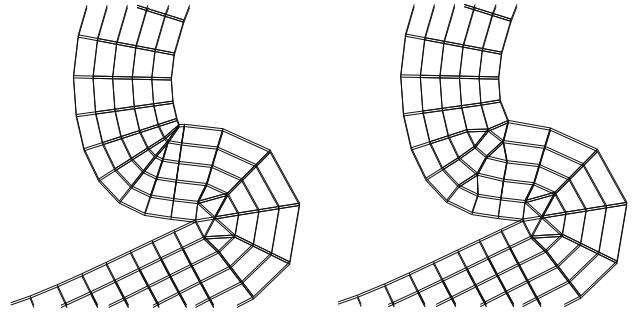


Fig. 8. Left: concave boundary without quality constraint. The front is advancing from left to right. Right: quality constraint enforced.

lapping box is divided into two smaller ones, to provide a more refined description of the front location. This procedure is repeated until no overlap is found or up to a box encompassing only two points of the layer. In this case, the segment made of the two nodes is inspected to verify whether it intersects adjacent fronts. If this is the case, the corresponding nodes and element are removed from the triangulations. Otherwise, the minimum distance between adjacent segments is computed and compared with the local height of the elements, to stop the front from advancing before an actual collision occurs and to provide a sufficiently large gap in between fronts for the Delaunay grid generation procedure.

2.4. Delaunay triangulation of the outer domain

The unstructured mesh generation outside the boundary layer and wake grids is performed here by using the software UM^{2D} [23], which is capable of producing isotropic and anisotropic meshes of triangles in two-dimensional domains. The software is an implementation of the front-advancing/Delaunay triangulation technique of Rebay [22]. The algorithm is boundary-conforming, i.e., the insertion of grid points starts from the boundary, where a metric (element size) is given which specifies the spacing between points located on the boundary itself. This information, together with metric specifications at selected points inside the domain, is sufficient to produce an isotropic meshes of triangles,

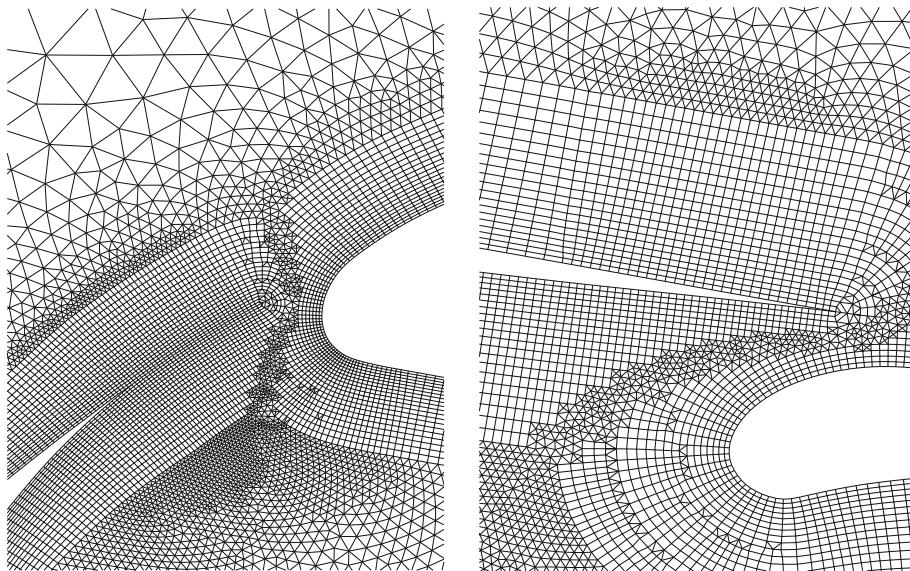


Fig. 9. Boundary layer/wake grid collisions for a flapped configurations. Left: interaction of the wake grid from the slat with the boundary layer grid around the main airfoil. Right: wake/boundary layer grid from the main airfoil interacting with the boundary layer grid around the flap.

as it is required in the present approach. Anisotropic grids have not been considered for simplicity and because the use of highly stretched elements outside the boundary layer and wake region is not deemed to be necessary. A background grid is first generated which includes all boundary points and the additional internal user-specified internal point where the metric is imposed. The background grid is used as the grid generation proceeds to compute the local element size which is interpolated on the background grid. The insertion of additional grid points is then performed by means of the well-known Bowyer–Watson [3,32] algorithm, so that a constrained Delaunay triangulation is obtained at every step, until all grid elements satisfy the prescribed metric [22].

An interface between the Delaunay triangulator and the present boundary layer and wake grid generation process has been coded, in which the last layers of the boundary layer and wake grids are included as the new boundaries of the Delaunay algorithm. The local metric is recovered by computing the local spacing of the elements constituting the last front, with no intervention from the user. These new fictitious boundaries are added to those in which no boundary layer grid has been generated to define a new (closed) region to be triangulated.

2.5. Algorithm structure and user interaction with the grid generation process

A brief outline of the sequence of the meshing operations together with the definition of the parameters required by user are now presented.

Let us start with the list of input parameters that will drive mesh generation. The input data are organized by records that define each line of the boundary of the computational domain and possible wakes. For each record, i.e., for each boundary line or wake line, information on its geometry and on the grid spacing is specified. Geometry can be specified either by means of primitives, i.e., straight line, circle, ellipse, or by points coordinates. Further translation/rotation of the each line can be specified explicitly. The grid spacing to provide for each record differs as the boundary/wake line is associated to a boundary layer mesh or not. In the latter case, only a tangential spacing has to be provided as input. The latter is specified by enforcing the desired tangential element length at a desired curve coordinate along the line. If the boundary/wake line is associated with a boundary layer grid further information has to be provided. Namely, the number of structured layers, the height of the first structured layer in correspondence of each curve coordinate in which a tangential spacing has been provided, together with the metric propagation function and the maximum element stretching have to be specified for each record. Note that the maximum element stretching is the parameter that controls the insertion of triangles in the structured layers of quadrilateral elements in presence of convex boundary/wake lines and in presence of sharp trailing edges. The triangles insertion in correspondence of concave boundaries is controlled by user through the specification of a minimum allowed angle.

Control over the elements spacing in the Delaunay region is allowed by the possibility of specifying the coordinates of a set of backgrid and steiner points inside the domain in which enforce a desired spacing of the triangular elements.

The algorithm is structured as follows: after reading the input data and generating the geometry of the boundary and wake lines, the first operation is to generate the layers of quadrilateral elements around the boundary and wake lines for which a structured grid is requested by user. The generation of the structured layers starts from the boundary wake line. Elements are then inserted according to the specified metric and propagation function. During this phase, a prediction is made on the quality of the next layer

quadrilateral element. If an element with bad quality is foreseen in correspondence of a concave or convex boundary line, or an overlapping of the structured advancing fronts is detected, then the insertion of suitable triangles is prescribed for the next layer, according to the criteria explained in Sections 2.1, 2.2 and 2.3. As indicated above, the outer layer of the quadrilateral grid defines a new closed region that is then processed by the well-known Bowyer–Watson algorithm.

3. Grid quality estimate

In order to provide a quantitative measure of the performance of the present grid generation technique a simple numerical example have been considered and the resulting grid quality have been evaluated on the basis of a suitable indicator, a nodal functional commonly employed in an optimization approach to the grid generation [15]. This indicator is computed for each internal node i of the grid and is based on the quality of the elements that have in common the node i . The functional is obtained by suitable vector quantities associated to the M nodes directly connected to node i by an edge of the triangulation. Namely, let us indicate the coordinates of node i by \mathbf{x}_i and a generic neighbor node by \mathbf{x}_m with $m \in (1, M)$ and $\mathbf{x}_1 = \mathbf{x}_M$. The following quantities can be introduced [15]:

$$\begin{aligned} \mathbf{e}_m &= \mathbf{x}_m - \mathbf{x}_i \\ l_m^2 &= \mathbf{e}_m \cdot \mathbf{e}_m, \quad l_{m+1}^2 = \mathbf{e}_{m+1} \cdot \mathbf{e}_{m+1}, \quad \beta_m = \mathbf{e}_m \cdot \mathbf{e}_{m+1} \\ g_m &= l_m^2 l_{m+1}^2 - \beta_m^2 \end{aligned} \quad (1)$$

Then the following functional associated to the smoothness of the grid can be obtained:

$$F_s = \frac{1}{2M} \sum_m \frac{(l_m^2 + l_{m+1}^2)}{\sqrt{g_m}} \quad (2)$$

In this work, the functional F_s is used only to evaluate the quality of a grid. To this purpose, two different grids are produced. One grid is obtained without applying the quality constraints of the present approach, then a second grid is generated by adopting the expedients described above. The quality functional is then evaluated for the two different grids and a comparison between the distribution of the functional in the two cases is presented.

The test case considered is the problem of grid generation around an lenticular airfoil. Fig. 10 shows the two considered grids. Fig. 11 shows instead the comparison of the distribution of the functional F_s in the two cases. Note that the distribution has been normalized with the total number of internal nodes. Grids with good quality with respect to F_s are characterized by uniform distribution of such an indicator. Moreover for the case considered here higher quality grids are associated to small values of F_s . From the comparison of the two distributions it can be observed that the adoption of triangles insertion technique results in a grid in which the distribution of the functional is more uniform and the percentage of nodes with smaller value of F_s is increased, showing the improvement of the grid quality. Moreover in order to give a more quantitative feeling of the quality improvement a comparison of the L_2 norm of the functional over the grid nodes for the two grids is reported here:

$$\begin{aligned} \text{No triangles insertion : } \|F_s\|_{L_2} &= 1.415 \\ \text{With triangles insertion : } \|F_s\|_{L_2} &= 1.187 \end{aligned} \quad (3)$$

A final consideration on grid quality concerns the ratio of the tangential spacing specified along the boundary line and the tangential spacing along the last layer of the structured region. From the point of view of the quality of the numerical solution it would be desir-

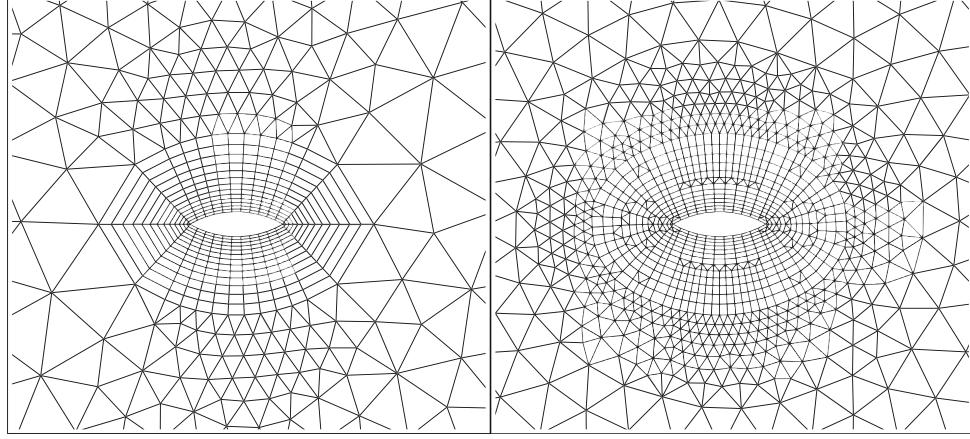


Fig. 10. Grid obtained without triangles insertion in the structured layers (left). Grid with triangles insertion technique (right).

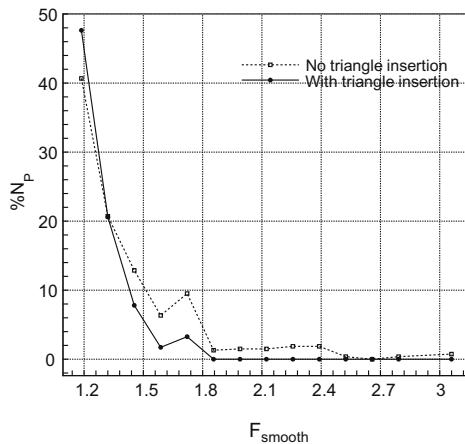


Fig. 11. Comparison of F_s distribution for the two computed grids. The y-axis reports the percentage of nodes, i.e., $N_p/N_{\text{tot}} \cdot 100$.

able to have the same tangential spacing through all the layers of the structured region. In this sense, it can be observed that by inserting suitable triangles in the structured region the present algorithm successfully keeps the desired spacing during the front-advancing procedure, Fig. 10 (right). On the other hand if no triangle is allowed the resulting grid may present significative differences in the spacing obtained at the boundary line and the one in correspondence to the outer layer of the structured region, Fig. 10 (left).

4. Numerical examples

The grid generation procedure presented in the previous sections is now applied to the computation of exemplary cases of viscous compressible flows in which the influence of boundary layers and wake regions is relevant. These are: a supersonic and a trans-

onic laminar flow around the NACA 0012 airfoil, two turbulent transonic flows around the RAE2822 and a turbulent flow past a two-element airfoil configuration. The latter high-Reynolds flows have been included to investigate the capabilities of the method to produce good-quality highly stretched boundary layer grids.

All computations have been performed using a mixed finite element/volume code that implements the node-pair hybrid finite volumes/finite elements method of Selmin [26], Selmin and Formaggia [27].

4.1. Laminar compressible flows around the NACA 0012 airfoil

Two test cases proposed in the GAMM workshop [4] are now considered for the evaluation of the grid generation algorithm. These are the laminar steady flows around the NACA 0012 labeled A4 and A6 in Ref. [4], for which more recent numerical results [8] are also available for comparison. The flow conditions and the grid details are given in Table 1. Both grids extend up to 20 chords lengths from the airfoil. The semi-structured region of the hybrid grid has been prolonged to a short distance from the trailing edge to allow for wake capturing. The resulting grid for the A4 case is made of 21,270 nodes and 26,210 elements, while the grid for the A6 case is made of 20,627 nodes and 25,852 elements. Fig. 12 shows the normalized smoothness functional distribution for the two grids, the L_2 norms of F_s in the two cases are, respectively, $\|F_{s,A4}\|_{L_2} = 4.01$ and $\|F_{s,A6}\|_{L_2} = 4.49$. The functional distribution for both cases is characterized by a peak in correspondence of the lower values of F_s in accordance with the quality criteria indicated above.

The grid arrangement, the Mach number contours, and the distribution of the pressure coefficient and of the shear stress for the A4 test case are shown in Fig. 13. The presence of a negative shear stress coefficient along the lower part of the airfoil is due to the choice of the reference frame for the computation of the shear stress, that is consistent with the one chosen by Forsyth and Jiang [8]. The present results are compared with those of Forsyth and Jiang [8] who employed a structured grid made of 46,848 (244

Table 1

Flow conditions for the laminar test cases A4 and A6: Mach number M, Reynolds number Re and angle of attack α . Relevant characteristics of the grids used in the computations are also listed.

Case	M	Re	α	Grid name	Grid type	Airfoil nodes	Total nodes
A4	2.00	1000	10°	A4-H	Hybrid	490	21,270
A6	0.85	2000	0°	A6-H	Hybrid	474	20,627

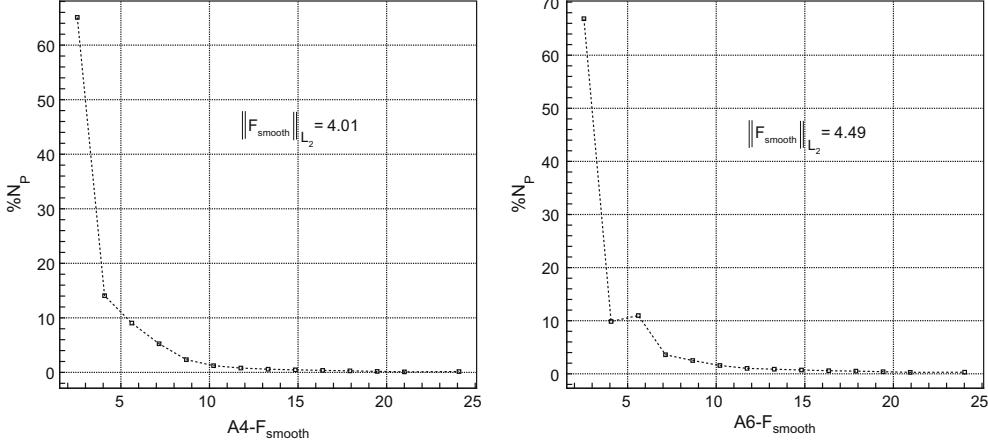


Fig. 12. Smoothness functional distribution. A4 grid (left), A6 grid (right).

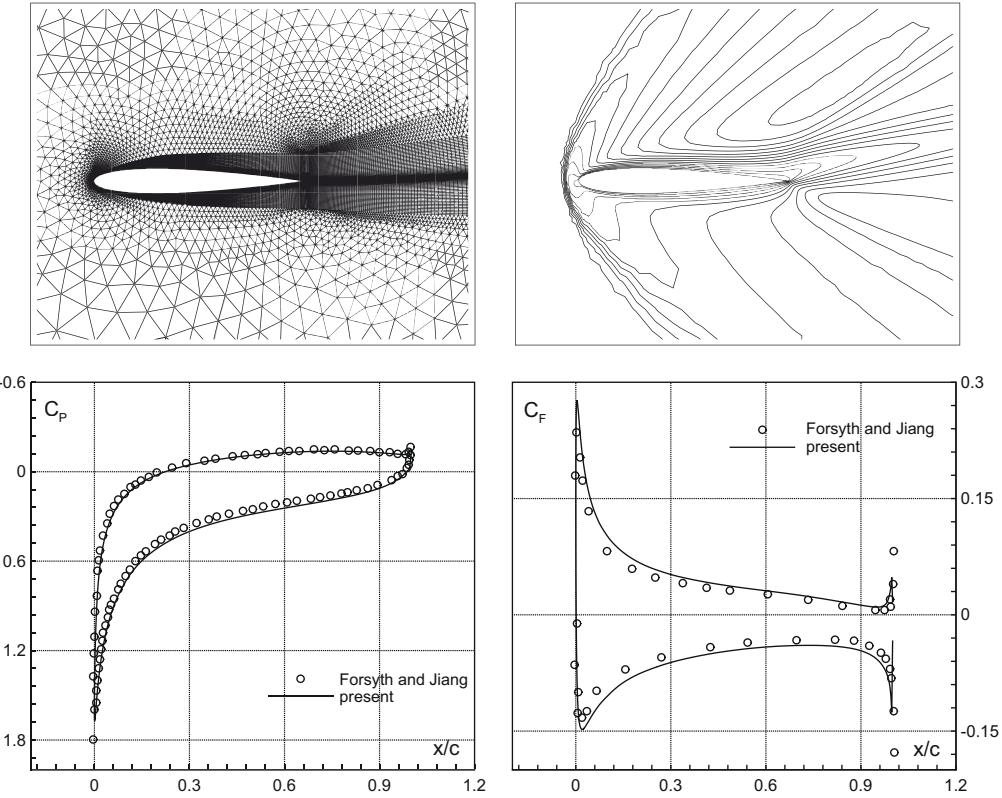


Fig. 13. Grid arrangement, Mach number contours together with pressure and shear stress coefficient distributions on the airfoil for the A4 test case.

Table 2
Lift and drag coefficient for the A4 test case.

	Present	Forsyth and Jiang
C _L	0.331	0.339
C _D	0.235	0.254

$\times 192$) quadrilateral elements. Table 2 summarizes the results in terms of aerodynamic coefficients.

The second test case considered features a limited transonic flow. Fig. 14 shows the results in terms of Mach number contours, pressure and shear stress coefficients. Only the upper distribution

is shown, due to the symmetry of the problem. Table 3 summarizes the results in terms of lift and drag coefficients. It can be shown that the agreement with the reference data of Forsyth and Jiang is quite good.

4.2. Turbulent compressible flows around the RAE2822 airfoil

In the present section, the turbulent compressible flow around the RAE2822 profile is computed and compared to experimental data of Ref. [5] in two different flight conditions, see Table 4. In both cases a shock wave is present on the upper surface of the airfoil at about 50% of the chord. Note that the T2 case is more critical in that a separation region occurs past the shock wave.

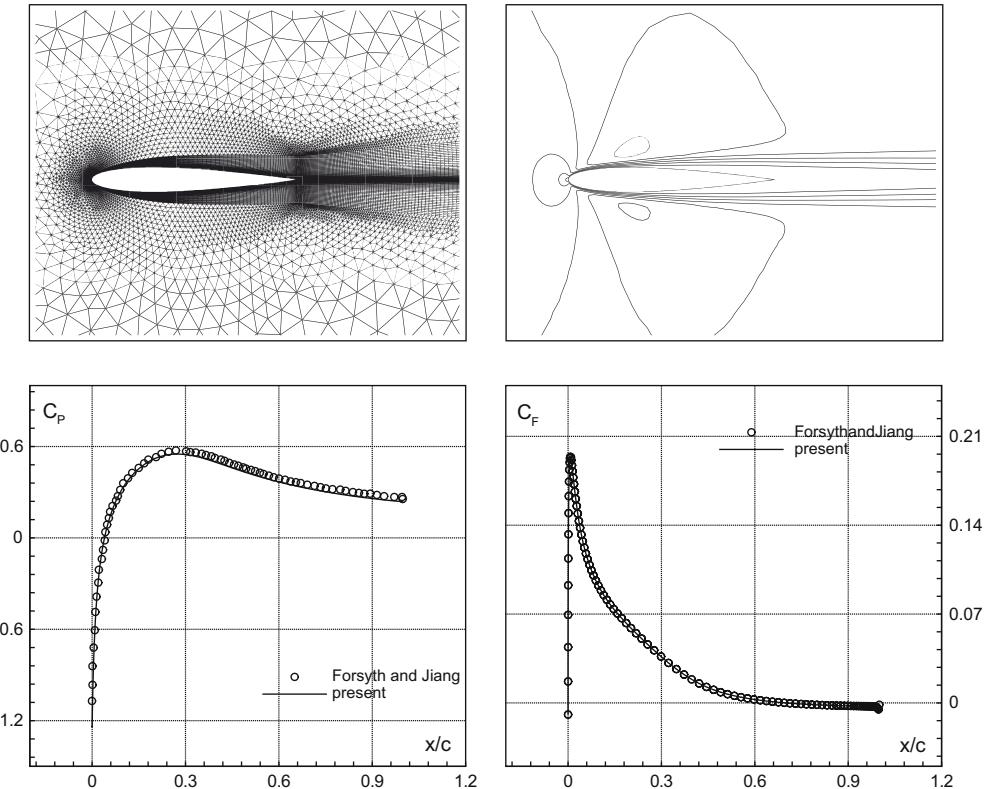


Fig. 14. Grid arrangement, Mach number contours together with pressure and shear stress coefficient distributions on the airfoil for the A6 test case.

Table 3
Lift and drag coefficient for the A6 test case.

	Present	Forsyth and Jiang
C_L	4.321×10^{-3}	1.5×10^{-4}
C_D	0.122	0.126

The Spalart–Allmaras [28] turbulence model is used to compute the turbulent viscosity coefficient. A constant turbulent Prandtl number of 0.72 is assumed to compute the thermal conductivity coefficient. Transition point from laminar to turbulent flow (trip point) is set at 3% of the chord length on both the upper and lower portion of the airfoil in all computations. The angle of attack has been corrected to account for wind tunnel effects according to the suggestions in Ref. [28,27].

Table 4 provides also relevant data for the grid used in the computations. The distance of the first internal node from the airfoil surface is about $10^{-5}c$, with c chord length, which corresponds to $y^+ \approx 2$ for both cases. As in the previous section, the computational domain extends to a distance of $20c$ from the airfoil. Fig. 15 shows the smoothness distribution for the grids used in the two different cases. In Fig. 16, the computational grid and the Mach number distribution in the flowfield are shown for case T1. The pressure and shear stress coefficient along the airfoil for this test case are shown

in the lower row of Fig. 16. The numerical results are found to be in good agreement with experimental data from Ref. [5]. The computed lift and drag coefficients C_L and C_D are 0.787 and 0.0177, respectively, as summarized in Table 5. The value of C_L is within 2% of the measured value of 0.803; the difference between the numerical and measured (0.0168) value of the drag coefficient is less than 5%.

Numerical results for test case T2 are shown in Fig. 17, where the same convention of Fig. 16 has been adopted to show the shear stress coefficient. Numerical and experimental results differ in determining the location of the shock wave, which is however very sensitive to the angle of attack and hence to wind tunnel corrections. The pressure difference across the shock wave is also predicted to be larger than the measured value. However, the same discrepancy is found by other authors [5,28,11] and it is not clear whether this difference is related to measurement errors or to deficiencies in the turbulence models. From Fig. 17, the shear stress coefficient is seen to be computed fairly well and flow separation past the shock wave is correctly captured by the numerical simulations. This is indeed a very complex behavior to simulate: it is to be recalled that some two-equations turbulence model are not capable of capturing the shock-boundary layer interaction correctly in this case [11]. Aerodynamic coefficients are $C_L = 0.759$ and $C_D = 0.0242$, to be compared with their experimental counterparts of 0.743 and 0.0242, respectively, Table 6.

Table 4

Turbulent compressible flow around the RAE2822 airfoil. The Reynolds number is computed using the chord as the reference length. The angle of attack $\alpha_{\text{exp.}}$ is the geometrical angle of the experiment as in Ref. [5]; $\alpha_{\text{corr.}}$ is the corrected angle including wind tunnel effects [28].

Case	M	Re	$\alpha_{\text{exp.}}$	$\alpha_{\text{corr.}}$	Grid type	Airfoil nodes	Total nodes
T1	0.73	6.5×10^6	3.15°	2.79°	Hybrid	280	15,745
T2	0.75	6.2×10^6	3.19°	2.81°	Hybrid	290	17,205

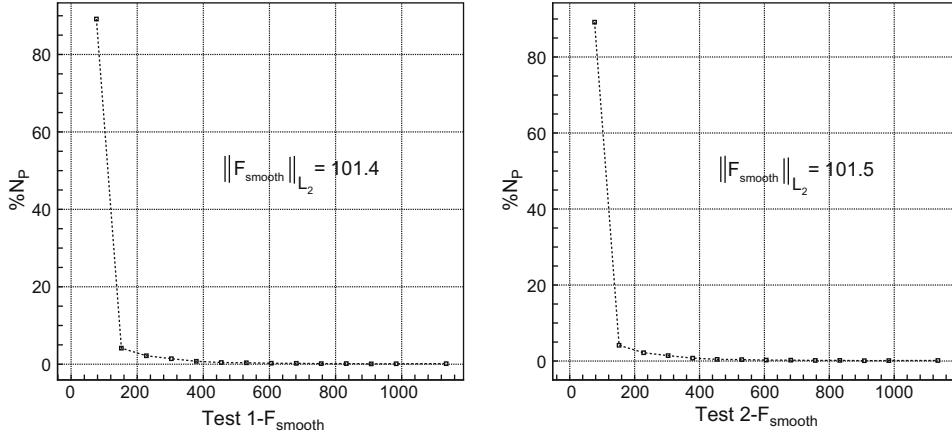


Fig. 15. Smoothness distribution for the two grids of the Rae test cases.

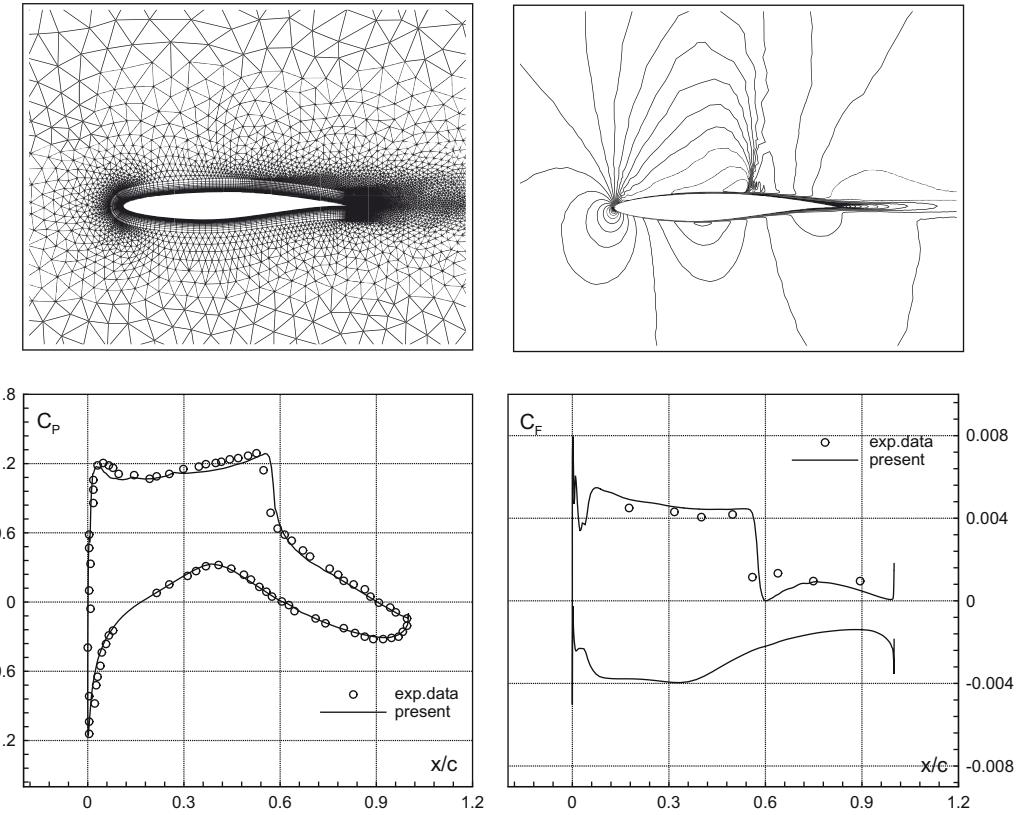


Fig. 16. Grid, Mach number contours together with the pressure and shear stress coefficient around the RAE2882 airfoil for the T1 case of Table 4.

Table 5

Lift and drag coefficient for the T1 case.

	Present	Experimental
C_L	0.787	0.803
C_D	0.0177	0.0168

For case T2, velocity profiles measured at two different stations located at 75% and 90% of the chord length are also available from [5] and are compared with numerical results in Fig. 18, together with the numerical results from Ref. [11], in which the Spalart–Allmaras model was also used. Major differences between experimental and numerical results are observed at both measurement

stations, although the present results agree fairly well with those computed in Ref. [11]. The reader is referred to Ref. [11] for a discussion on the numerical results.

4.3. Turbulent compressible flow around a two-element airfoil

The present grid generation algorithm has been tested also for the two elements airfoil flap configuration proposed by Omar and collaborators [7]. The turbulent steady flow at Mach 0.201, Reynolds 2.83×10^6 with an incidence of 0.01° has been simulated, and the results have been compared with wind tunnel experimental data [7].

Fig. 19 displays an overview of the grid adopted for the computation and the Mach number contours. Three wakes have been con-

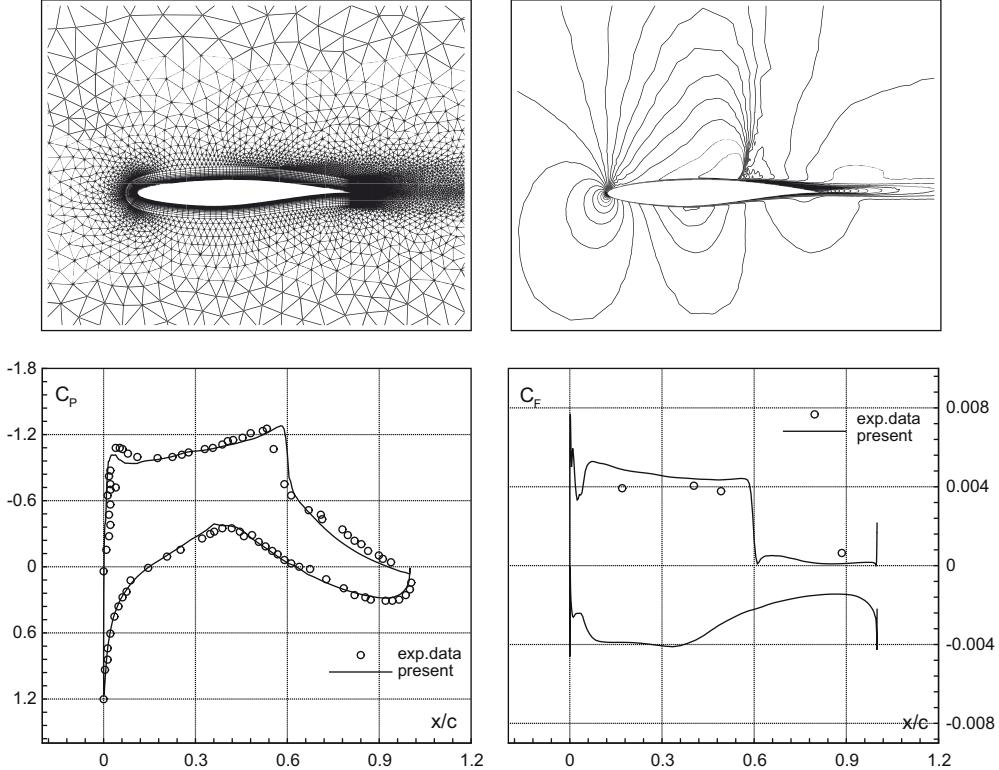


Fig. 17. Grid, Mach number contours together with the pressure and shear stress coefficient around the RAE2882 airfoil for the T2 case of Table 4.

Table 6
Lift and drag coefficient for the T2 test case.

	Present	Experimental
C_L	0.759	0.743
C_D	0.0242	0.0242

sidered: one for the flap and two for the main element. All the algorithm's features concerning the wake/boundary layer interaction as well as the treatment of concave geometries are exploited for the present configuration. Figs. 20–22 show grid details and flow streamlines in proximity of the flap well and the trailing edge of the flap. The generation process of this fairly complex grid made of 124,266 elements and 108,596 nodes required only 82.5 s of CPU time on a AMD k7 2 GHz processor. Fig. 23 presents the

smoothness functional distribution together with the comparison of the pressure coefficient obtained from the simulation with the experimental data obtained by Omar [7]. The agreement between numerical and experimental data is fairly good.

5. Conclusions

An hybrid grid generation algorithm for two-dimensional high-Reynolds compressible flows is presented and applied to the computation of both laminar and turbulent flows around airfoils. High-quality grids are obtained around convex and concave geometry by locally modifying the structured boundary layer grid via the introduction of triangular elements. In this way, the user-specified element size and stretch is easily obtained throughout all the boundary layer region. The procedure for triangle insertion re-

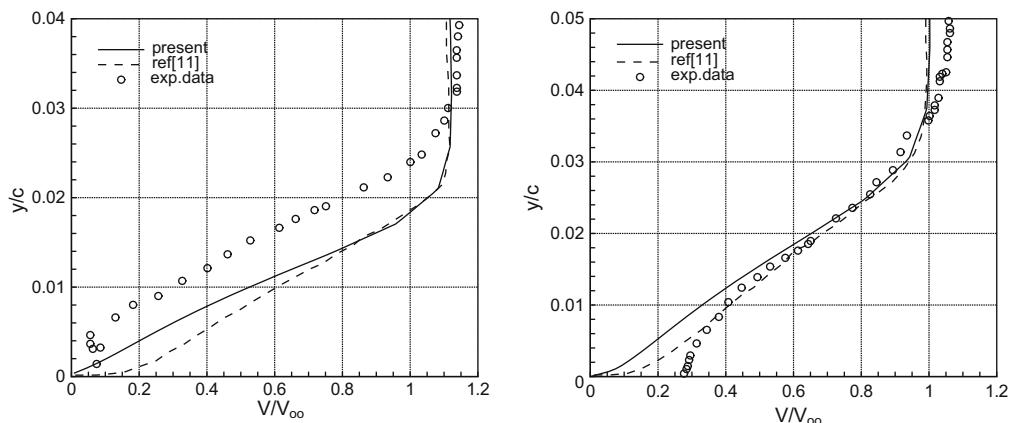


Fig. 18. Velocity profiles at 75% (left) and 90% (right) of the chord length for case T2 in Table 4. Experimental results are taken from Ref. [5].

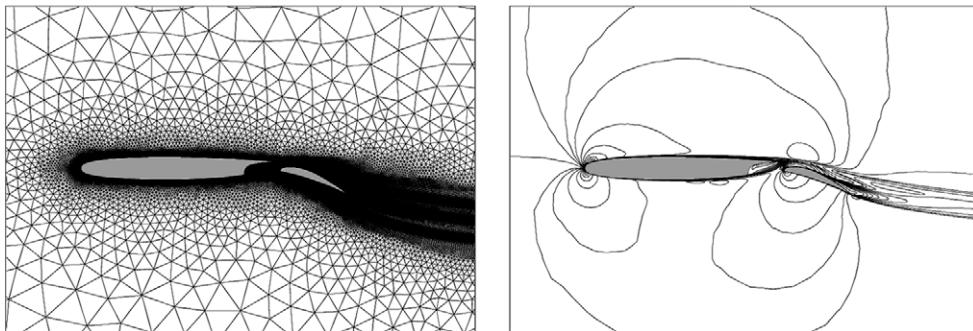


Fig. 19. Two-element airfoil. Grid arrangement and Mach number contours.

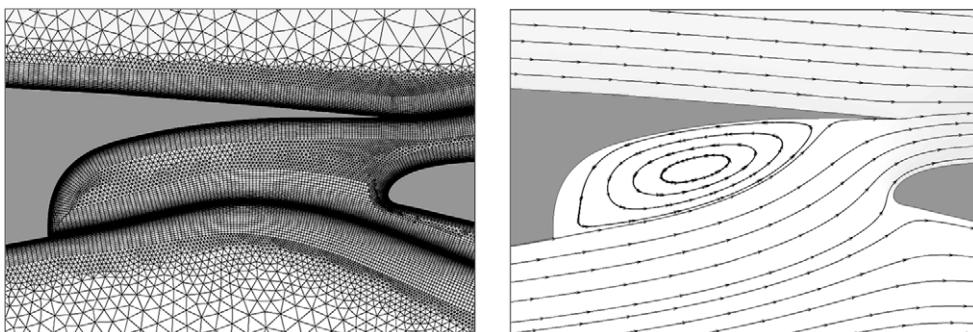


Fig. 20. Two-element airfoil. Detail of the flap well grid and stream lines.

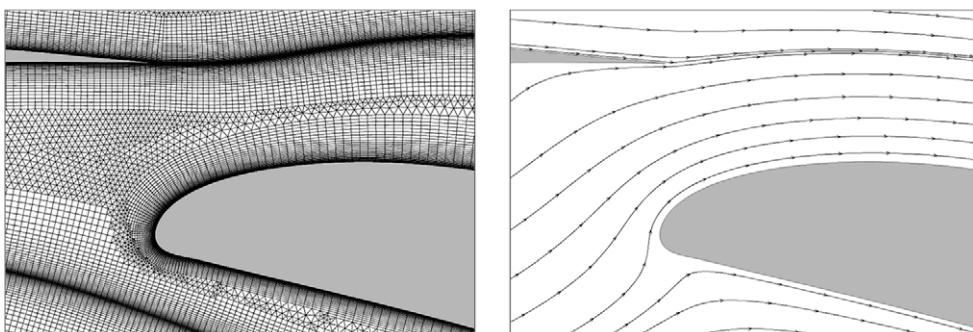


Fig. 21. Two-element airfoil. Wakes interaction and stream lines near the flap leading edge.

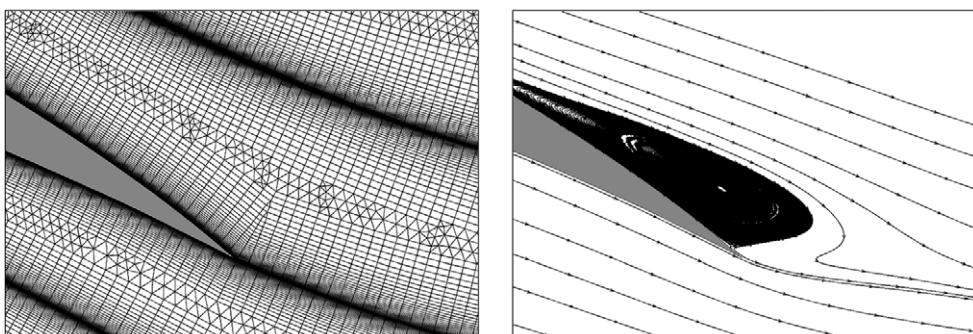


Fig. 22. Two-element airfoil. Flap trailing edge, wakes and stream lines.

quires almost no intervention from the user, which has to specify only the quality, i.e., the minimum angle, of triangles to be inserted. Polyline curves representing the wake inside the computa-

tional domain can be also prescribed. A front collision check algorithm is introduced to track the possible interference of grid front originating from different boundaries. The triangulation of

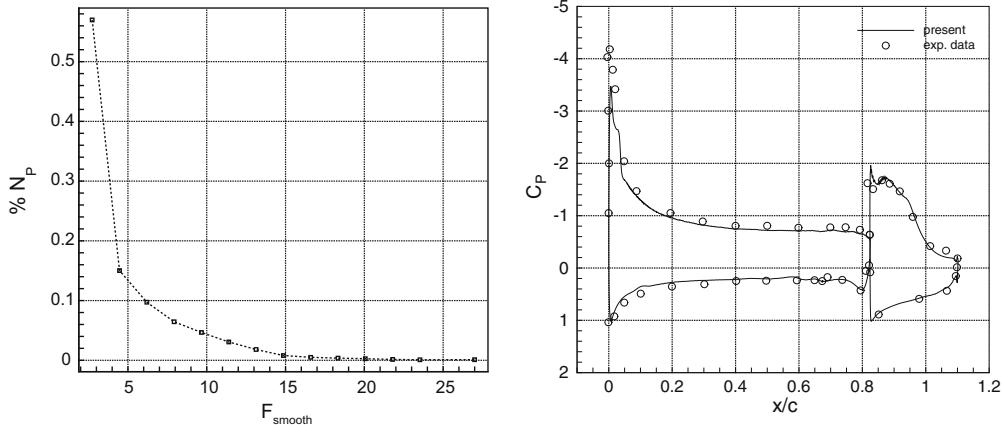


Fig. 23. Smoothness functional (left). Pressure coefficient (right). Experimental data are from Ref. [7].

the outer domain which is not covered by the boundary layer or wake grids is performed by means of a standard front-advancing/Delaunay software, which automatically infers the correct element size from the last layer of elements.

The results presented here have to be considered as a preliminary step towards the definition and implementation of a technique for the generation of complex unstructured hybrid three-dimensional grids. In three spatial dimensions, boundary layer or wake grids are made of either hexaedra (if the surface mesh is made of quadrilateral elements) or prisms (for surface meshes of triangles) or both. In principle once a suitable surface mesh is generated on each boundary/wake surface, the same approach of advancing layer can be employed to obtain boundary layer grids. In this case, in fact, the three-dimensional counterpart of the present two-dimensional algorithm could work in a similar fashion, having hexaedra and prisms collapsing into pyramids and tetrahedra, respectively, provided that a suitable quality constraint on the minimum angle of the elements to be inserted is violated.

Acknowledgement

The authors acknowledge the help of Stefano Cinquina, who implemented the turbulence model in the flow solver.

References

- [1] Babuska I, Aziz AK. On the angle condition in the finite-element method. *Siam J Numer Anal* 1976;13(6):214–26.
- [2] Barth TJ. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. AGARD AR 787-6 on unstructured grid methods for advection dominated flows, NATO; 1992.
- [3] Bowyer A. Computing Dirichlet tessellations. *Comput J* 1981;24:162–6.
- [4] Briseau MO, Glowinski R, Periaux J, Viviand H. Numerical simulation of compressible Navier-Stokes flows. Number 18 in notes on numerical fluid mechanics. Vieweg; 1987.
- [5] Cook PH, McDonald MA, Firmin MCP. Airfoil RAE 2822 pressure distributions, boundary layer and wake measurements. AGARD AR 138, NATO; 1979.
- [6] Debize C, Dervieux A, Mer K, Nkonga B. Computation of unsteady flows with mixed finite volume/finite element upwind methods. *Int J Numer Methods Fluids* 1998;27:193–206.
- [7] Han M, Szapiro E, Omar E, Ziertzen T, Mahal A. Two-dimensional wind tunnel tests of a nasa supercritical airfoil with various high-lift systems. Technical Report CR 2215, NASA; 1977.
- [8] Forsyth PA, Jiang H. Nonlinear iteration methods for high speed laminar compressible Navier-Stokes equations. *Comput Fluids* 1997;26:249–68.
- [9] Garimella RV, Shepard MS. Boundary layer mesh generation for viscous flow simulations. In: 7th international meshing roundtable, Sandia National Lab, September; 2000.
- [10] George PL. Delaunay triangulation. Theoretical aspects, element quality and robustness. Lecture Series 05, von Karman Institute of Fluid Dynamics; 2000.
- [11] Hellsten A. Implementation of a one-equation turbulence model into the FINFLO flow solver. Report B-49, Helsinki University of Technology, Laboratory of Aerodynamics; 1996.
- [12] Kallinderis YK, McMorris H. Hybrid prismatic/tetrahedral grid generation for complex geometries, AIAA Paper 1995-0211; 1995.
- [13] Kallinderis YK, Nakajima K. Finite element method for incompressible viscous flows with adaptive hybrid grids. *AIAA J* 1994;32(8):1617–25.
- [14] Khawaja A, Kallinderis Y. Hybrid grid generation for turbomachinery and aerospace applications. *Int J Numer Methods Eng* 2000;49:145–66.
- [15] Knupp PM. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. Part i – dua framework for surface mesh optimization. *Int J Numer Methods Eng* 2000;48:401–20.
- [16] Delanaye M, Afotsmis MJ, Berger MJ, Liu Y, Pulliam TH. Automatic hybrid-cartesian grid generation for high Reynolds number flows around complex geometries, AIAA Paper 1999-0777; 1999.
- [17] Mavriplis DJ. Unstructured and adaptive mesh generation for high-Reynolds number viscous flows. Report 187534, NACA ICASE; 2005.
- [18] Mavriplis MJ, Venkatakrishnan V. A unified multigrid solver for the Navier-Stokes equations on mixed element meshes, AIAA paper 1991-1666; 1991.
- [19] Merriam ML. Unstructured viscous grid generation by the advancing layers method, AIAA Paper 1991-0792; 1991.
- [20] Pirzadeh S. Unstructured viscous grid generation by the advancing layers method. *AIAA J* 1994;32(8):1735–7.
- [21] Quarteroni A, Valli A. Numerical approximation of partial differential equations. Number 23 in Springer series in computational mathematics. Springer-Verlag; 1994.
- [22] Rebay S. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. *J Comput Phys* 1993;106:125–38.
- [23] Rebay S, Guardone A, Dussin D. UMESH2D user manual. Technical Report, Politecnico di Milano; 2003.
- [24] Ryskin G, Leal LG. Orthogonal mapping. *J Comput Phys* 1983;50:71–100.
- [25] Selmin V. A multistage method for the solution of the Euler equations on unstructured grids. In: Gruber R, Priaux J, Shaw RP, editors. Proceedings of the fifth international symposium on numerical methods in engineering, vol. 2. Berlin: Springer-Verlag; 1989. p. 449–54.
- [26] Selmin V. The node-centred finite volume approach: bridge between finite differences and finite elements. *Comp Methods Appl Mech Eng* 1993;102:107–38.
- [27] Selmin V, Formaggia L. Unified construction of finite element and finite volume discretizations for compressible flows. *Int J Numer Methods Eng* 1996;1:3–32.
- [28] Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. *Rech Aerosp* 1994;1:5–21.
- [29] Spitaleri RM, Manzi C, Formaggia L. Hyb2d (a hybrid grid generation in 2d) algorithm overview and description. In: Proceedings of the 6th International Conference, University of Greenwich; 1998.
- [30] Steinbrenner JP, Noack RW. Three-dimensional hybrid grid generation using advancing front techniques. In: Surface modeling, grid generation and related issues in computational fluid dynamics workshop, NASA Lewis Research Center; 1995.
- [31] Thompson JF. Numerical grid generation: foundations and applications. Elsevier Science Publishing; 1985.
- [32] Watson DF. Computing the n -dimensional Delaunay tesselation with application to Voronoi polytopes. *Comput J* 1981;24:167–72.
- [33] Zhang Y, Jia Y, Wang SY. Structured mesh generation with smoothness control. *Int J Numer Methods Fluids* 2006;51(11):1255–76.