An isoparametric approach to high-order curvilinear boundary-layer meshing

D. Moxey[*], M.D. Green, S.J. Sherwin, J. Peiró

Department of Aeronautics, South Kensington Campus, Imperial College London, SW7 2AZ, UK

Highlights

- We present a method to generate high-order 3D curvilinear boundary layer meshes.
- The method refines a valid coarse mesh using an isoparametric mapping.
- Resulting elements are of arbitrary order and can be sized to any desired thickness.
- We demonstrate the resulting meshes are valid given a valid coarse mesh.
- The method is modular and allows fast generation of meshes for convergence studies.

Abstract

The generation of high-order curvilinear meshes for complex three-dimensional geometries is presently a challenging topic, particularly for meshes used in simulations at high Reynolds numbers where a thin boundary layer exists near walls and elements are highly stretched in the direction normal to flow. In this paper, we present a conceptually simple but very effective and modular method to address this issue. We propose an isoparametric approach, whereby a mesh containing a valid coarse discretization comprising of high-order triangular prisms near walls is refined to obtain a finer prismatic or tetrahedral boundary-layer mesh. The validity of the prismatic mesh provides a suitable mapping that allows one to obtain very fine mesh resolutions across the thickness of the boundary layer. We describe the method in detail for a high-order approximation using modal basis functions, discuss the requirements for the splitting method to produce valid prismatic and tetrahedral meshes and provide a sufficient criterion of validity in both cases. By considering two complex aeronautical configurations, we demonstrate how highly stretched meshes with sufficient resolution within the laminar sublayer can be generated to enable the simulation of flows with Reynolds numbers of 10^6 and above.

Keywords: High-order mesh generation; Boundary-layer meshes; Isoparametric elements

1. Introduction

The use of high-order methods in complex aeronautical geometries and their associated flows is attracting attention due to their low dispersion and diffusion errors and their potential to achieve exponential convergence [1]. However,

* Corresponding author. Tel.: +44 20 7594 5129.
E-mail address: d.moxey@imperial.ac.uk (D. Moxey).

before these methods are adopted by industry, there are a number of issues that must be addressed in order to meet industrial requirements. One of the main bottlenecks preventing the widespread industrial use of such methods is the generation of high-order curvilinear meshes for complex geometries and high Reynolds numbers [2,3].

In modelling the flow in boundary layers at high Reynolds numbers, computational efficiency dictates the use of high aspect ratio elements in the near-wall region. The element size in the direction normal to the wall is usually expressed in terms of a wall unit y^+ that can be written as

$$y^+ = \frac{y}{L} \operatorname{Re} \sqrt{\frac{C_f}{2}}$$

where y is the coordinate normal to the wall, Re is the Reynolds number, C_f is the skin friction coefficient and L is the reference length used to calculate these non-dimensional coefficients. In order to adequately resolve the viscous sublayer near the wall, elements at the wall are typically required have a height no larger than $y^+ = 1$. Estimates of y^+ are usually obtained from empirical relations of the form $C_f = F(\operatorname{Re})$ [4], where $C_f = O(\operatorname{Re}^{-\alpha})$ with $2 \leq \alpha \leq 3$. When Re is high, this in turn leads to mesh stretching ratios of 1:1000 or even lower.

Current generation methods for linear unstructured meshes can be used to produce these stretched elements [5,6] but the quality of the generated elements is very difficult to control. An alternative method is the use of a hybrid approach in which the mesh near the solid surfaces is generated by a hyperbolic type mesh generator whilst the region outside the viscous layer is discretized using a conventional approach [7].

Techniques for the generation of unstructured high-order meshes, in which high-order information is projected onto a coarse linear grid in order to curve it, are a relatively new development [8–10]. The main challenge in generating such meshes is robustness, since near walls curvature must be introduced in such a way as to prevent self-intersecting elements. More recent approaches attempt to address this issue by projecting curvature into the interior of the domain, through the use of elastic analogies to deform the mesh where curvature is introduced [11,12], defining an optimization procedure to untangle invalid elements [13,14] or utilizing Bézier polynomials to monitor the appearance of invalid elements in adaptive simulations [15,16].

When applied to the problem of boundary layer mesh generation, these methods follow a similar procedure: a linear mesh is generated which contains a stack of boundary layer elements near walls; boundary nodes are introduced onto the surface of each element in order to curve it and finally; this curvature is propagated into the interior of the domain to eliminate invalid elements. There are however some issues with this approach, the most serious being that there is no guarantee of obtaining a valid mesh at the end of the procedure. Additionally, these methods are all relatively expensive, meaning that when either high polynomial orders or large numbers of elements are desired for very thin boundary layer resolution, long generation times are required. The success of these methods in the generation of meshes possessing an arbitrarily thin boundary layer has therefore been limited.

In this article we propose a method which adopts a different approach to the generation procedure outlined above and that helps to address the problem of generating high-order meshes for high Reynolds number flows. The method is conceptually simple, cheap to implement and does not require a dense linear boundary-layer mesh. It is based on the use of an isoparametric [17] or, in general, a transfinite interpolation [18] where a high-order coarse boundary-layer prismatic mesh is subdivided into either prisms or tetrahedra using the mapping that defines the coarse high-order prisms. The procedure is also very versatile as it permits meshes with different distributions of y^+ to be generated with ease and furthermore, the validity of these meshes is guaranteed if the initial mesh is valid and the polynomial space is chosen appropriately.

The paper is organized in two main sections. Section 2 describes the various procedures involved in generating arbitrarily high-order boundary layer meshes suitable for these flows: the generation of the initial coarse mesh, the splitting of the prismatic elements into finer prisms or tetrahedra and the spacing distributions used to define the size of the finer sub-elements. Section 3 presents examples of application of the methodology to the generation of a high-order mesh for some selected aeronautical configurations: the ONERA M6 wing, a NACA 0012 wing and the front section of a representative Formula 1 car. A final section on conclusions follows.

Section 2.1 describes some methods for generating a high-order hybrid mesh of prisms in the boundary layer and tetrahedra elsewhere starting from a straight-sided mesh. In particular we note that this procedure could lead to invalid elements in concave regions of the surface, and so a criterion to select a size of the prismatic elements in the boundary-layer based on the curvature of the wall is outlined. The idea of using a mapping provided by the high-order macro-elements of the initial coarse mesh to obtain a mesh suitable for high-Reynolds number flows is

B-Rep   Linear mesh   High-order mesh
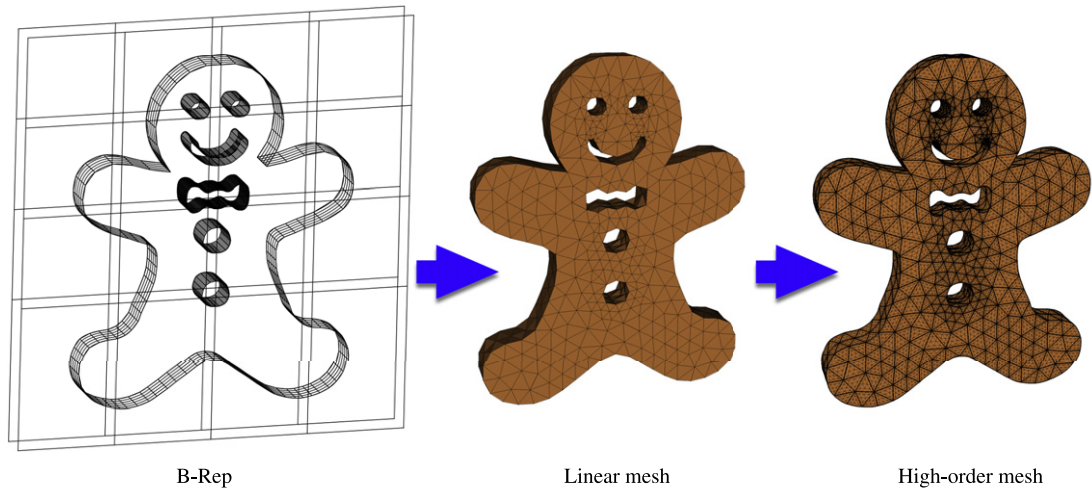
Fig. 1. High-order meshing strategy.

described in Section 2.2. The procedure for splitting the coarse boundary-layer mesh into a fine prismatic mesh whilst ensuring its validity is described in Section 2.3. The strategy for achieving fully valid tetrahedral meshes is presented in Section 2.4. Finally we outline the spacing function used to determine the sizes of sub-elements in Section 2.5, and show how this can be varied between elements to provide a spatially-aware refinement procedure.

2. Mesh generation strategy

This section describes a method to generate high-order meshes comprising of highly stretched elements. We first outline some techniques from the literature which lead to the generation of a high-order coarse mesh consisting of prisms in the boundary layer and tetrahedra elsewhere. We then describe how a finer mesh can be obtained by splitting the coarse prismatic elements (referred to hereafter as 'macro-elements') into finer elements, utilizing the mapping that defines each macro-element to insert the necessary curvature into sub-elements. Finally we describe how this technique can be further adapted to produce meshes containing only tetrahedral elements. The various steps involved are described in the following sections.

2.1. Generation of the coarse hybrid mesh

The method that we propose in the following sections relies on the generation of a high-order coarse grid, which we then refine. In order to place our refinement technique in the context of the mesh generation process, we begin by outlining some available techniques for the generation of such meshes.

If CAD data is available, then we may adopt a procedure such as the one described in [9] which we briefly summarize here for completion and depict in Fig. 1. In broad terms, the high-order mesh generation steps are: the definition of a CAD boundary representation (B-Rep) of the computational domain; the generation of a hybrid mesh of linear prismatic and tetrahedral elements; and finally, the subdivision of this mesh into a mesh of high-order elements. The geometry of the computational domain is represented by means of CAD spline curves and surfaces to obtain a B-Rep of the computational domain. The linear mesh is then generated using an implementation of the method of advancing normals [19] to generate a boundary-layer mesh of triangular prisms near the walls in the computational domain. The advancing front technique [20] is then used to generate a mesh of regular tetrahedra for the rest of the domain.

The final step in the generation process is to insert high-order curvature information to produce the high-order mesh, which proceeds in a bottom-up fashion, starting with the edges, then the faces and finally the interior of the elements. This is the most troublesome aspect of the method as it could lead to the generation of invalid elements in the concave regions of the boundary of the computational domain. Fig. 2 illustrates that when the prismatic boundary layer is generated using a fixed width, some elements are not sufficiently large to prevent a self-intersection of the element once curvature information is introduced to the element.

To alleviate this problem, in this work we use a criterion based on the curvature of the surface to determine a lower bound for the boundary layer thickness that aims to prevent the intersection of the element with the surface. The
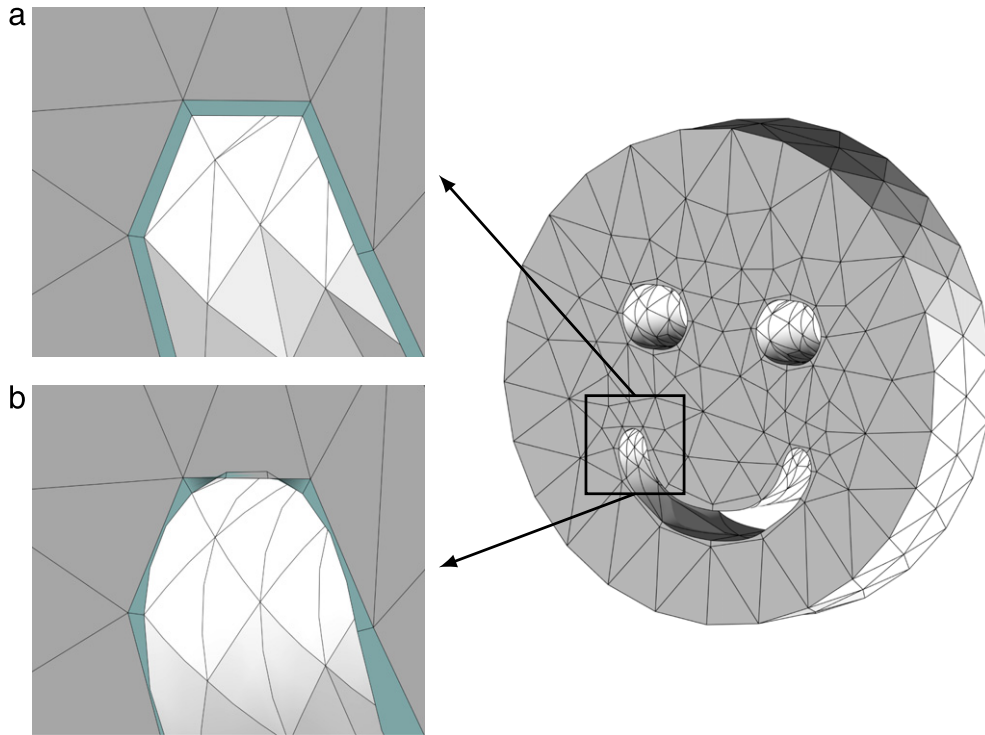
Fig. 2. Generation of invalid elements in concave regions of the computational domain. (a) A valid linear mesh and (b) the self-intersection that often occurs when high-order information is introduced.

curvature of the surface is obtained through interrogation of the B-Rep of the computational domain. Following the notation of Fig. 3 and considering a two-dimensional domain, we approximate a curve by its osculating circle [21]. Assuming that the radius of the osculating circle is R, and the size of the element along the curve is represented by the chord c, then the minimum length in the direction normal to the curve required to obtain a valid prismatic element, represented by δ, is given by

$$\left(\frac{c}{2R}\right)^2 + \left(\frac{R}{R+\delta}\right)^2 = 1 \Rightarrow \frac{\delta}{R} \geq -1 + \left[1 - \left(\frac{c}{2R}\right)^2\right]^{-\frac{1}{2}}.$$

If $c \ll R$, this requirement can be approximated by

$$\frac{\delta}{R} \geq \frac{c^2}{8R^2}.$$

To extend this method to surfaces in three dimensions, we simply interpret R as the smallest of the two principal radii of curvature of the surface and use the same criterion. Whilst this procedure does not necessarily guarantee the validity of the coarse mesh, it does frequently prevent the appearance of invalid elements.

Alternatively, we may use a number of techniques which are defined elsewhere in the literature. In particular we note that the elastic deformation and untangling procedures defined in the introduction may prove extremely useful here, where the number of elements is smaller than we would find in a boundary layer grid and the stretching ratio of elements is low. These factors combined mean that successful grid generation is more likely, and also the presence of fewer elements may support the use of higher polynomial orders. Additionally, we note that in cases where CAD is not available, for example in biomedical applications where surface meshes are derived from medical imaging, various commercial generator packages can be used to generate a very coarse linear grid, which can then be smoothed using a variety of high-order techniques [22,23] and then combined with one of the aforementioned methods to produce a high-order grid.
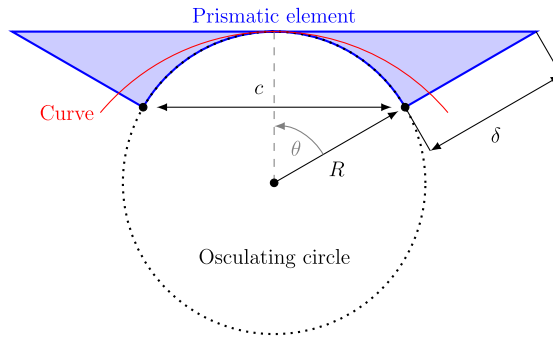
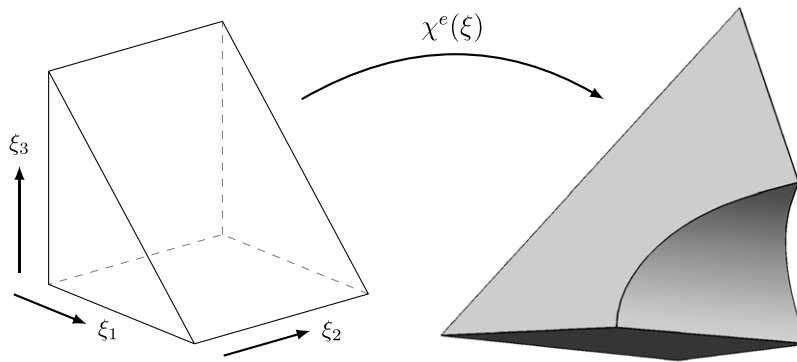Fig. 3. Notation used for the calculation of the minimum height δ of a valid prismatic macro-element.

Fig. 4. High-order element mapping from a standard prism Ω_{st} to a corresponding prism Ω^e in Cartesian space.

The rest of this section describes the main contribution of this work, which describes how the boundary layer mesh generated using any number of the techniques above can be refined to produce arbitrarily thin boundary layer meshes. The basis of this method is to use the isoparametric mapping which defines its curvature, which we describe in the next section. We then demonstrate how it can be used to generate either mixed prismatic–tetrahedral or fully tetrahedral meshes in the following sections.

2.2. The isoparametric approach

The refinement procedure relies on a definition of the coordinate mapping $\chi^e : \Omega_{st} \to \Omega^e$ which projects a reference element Ω_{st} to an element Ω^e in the Cartesian domain. This mapping is illustrated in Fig. 4. Although there are many ways to define χ^e, here we adopt an approach in which χ^e isoparametrically projects coordinates $\xi = (\xi_1, \xi_2, \xi_3)$ in $\Omega_{st} = \{(\xi_1, \xi_2, \xi_3) \mid \xi_1, \xi_2 \in [-1, 1], \xi_1 + \xi_3 \leq 1\}$ onto Cartesian coordinates $(x_1, x_2, x_3) \in \Omega^e$. Given polynomial orders P, Q and R for each component ξ_i, we represent each component of $\chi^e = (\chi_1^e, \chi_2^e, \chi_3^e)$ as a tensor product expansion of one-dimensional hierarchical modal functions, so that

$$\chi_i^e(\xi_1, \xi_2, \xi_3) = \sum_{p=0}^{P} \sum_{q=0}^{Q} \sum_{r=0}^{R-p} (\hat{\chi}_i)_{pqr} \psi_p^a(\eta_1) \psi_q^a(\xi_2) \psi_{pr}^b(\eta_3). \tag{1}$$

In this expansion we assume $P \leq R$, and map the standard prism Ω_{st} onto a reference hexahedron through a collapsed coordinate system $(\eta_1, \eta_3) \in [-1, 1]^2$ defined by

$$\eta_1 = 2\frac{1+\xi_1}{1-\xi_3} - 1, \qquad \eta_3 = \xi_3.$$

The use of the modified hierarchical expansion functions ψ^a and ψ^b, defined in [1], is useful in the mesh generation process as it permits a decomposition of the element into modes which have non-zero support only on designated
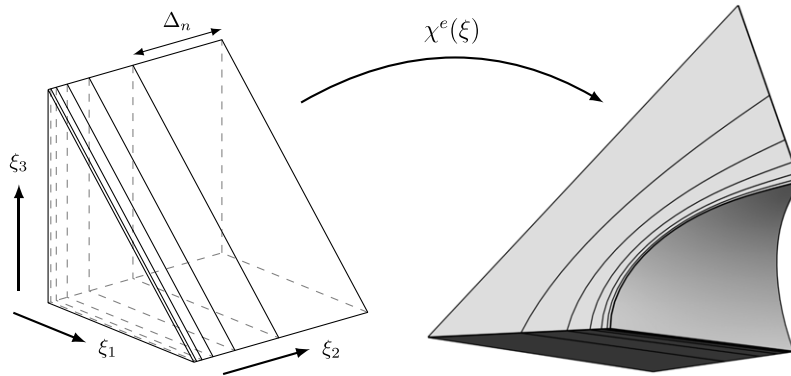
Fig. 5. Splitting Ω_{st} and applying the mapping χ^e to obtain a high-order layer of prisms from the macro-element of Fig. 4.

vertices, edges and faces of the element. If only edge-interior or face-interior curvature is provided for an element, then upon a transformation from the coefficient space $(\hat{\chi}_i)_{pqr}$ to an arbitrary set of quadrature points, the Cartesian positions of any volume-interior (or face-interior) points are blended in a linear manner from the curvature information that is supplied. Therefore, our existing mesh generation methodology only derives face-interior curvature information on a set of electrostatically-distributed triangular nodal points for faces of prisms representing the geometric surface. We assume without loss of generality that this face lies in the plane $\xi_2 = -1$ with respect to Ω_{st}. It should be noted therefore that in the ξ_2 direction, only a linear expansion is required so that $Q = 1$ in (1). This curvature information is transformed to the coefficient space through a Galerkin projection to obtain $(\hat{\chi}_i)_{pqr}$.

2.3. Splitting the boundary-layer mesh

The goal of the splitting process is to refine each macro-element in the wall-normal direction ξ_2, so that at the wall surface the flattest prism is correctly sized to resolve a desired boundary layer thickness. Where the Reynolds number is high, extremely thin elements are required in order to resolve the high shear of the flow, and therefore the method must be capable of generating valid elements of large aspect ratio, even where the curvature of the surface is large. To guarantee the validity of the refined elements, we pick a refinement strategy which focuses on first splitting the standard prismatic element into a series of n smaller elements distributed by a spacing Δ_k for $1 \leq k \leq n$. We then apply χ^e in order to determine curvature information for each sub-element of the refinement. This projects the sub-elements back to Cartesian space, thereby splitting the original prism into curved prismatic sub-elements. This is depicted in Fig. 5.

Assuming that we wish to produce n sub-elements for each boundary layer prism, we select a partition of $[-1, 1]$ defined as $-1 = x_0 < x_1 < \cdots < x_n = 1$, for every element to be refined. This gives rise to a spacing distribution $\Delta_k = x_k - x_{k-1}$ which denotes the width of each sub-element in the reference element as shown in Fig. 5. The spacing of these points is unimportant in the following method. In order for the mesh to be conformal, we assume that n and Δ_k are fixed across all of the prismatic macro-elements, although in Section 2.5 we describe a method for varying Δ_k in more detail, if we relax the guarantee of mesh conformity.

Each prism is split by iterating over the number of sub-elements and constructing each one in a bottom-up fashion; that is, vertices are derived first, followed by edge-interior points and finally face-interior points. We calculate the position of these points inside Ω_{st}, and then apply χ^e at each point in order to compute the Cartesian coordinates. We note that edge and face interior points are only calculated for the two triangular faces of the sub-element, since the use of a linear expansion in the ξ_2 direction of the original prism means high-order information is not required.

If the high-order prismatic macro-element is valid, we can argue that the mesh is also valid after subdivision as follows. We will assume that the sides of the elements in the direction "normal" to the wall (ξ_2) are straight lines. For prismatic elements, the process of subdivision is equivalent to an orientation-preserving affine transformation which does not alter the sign of the Jacobian of the mapping and therefore guarantees the validity of the mesh. Additionally, the polynomial spaces which define both the sub-element and the macro-element in terms of the local coordinates ξ are identical, and therefore the use of the macro-element to define the sub-element mappings means they span an identical space and so have identical properties to the original element. This implies that even when the curvature on the surface is high, or indeed changes sign across a triangular face, the resulting sub-elements will still be valid.
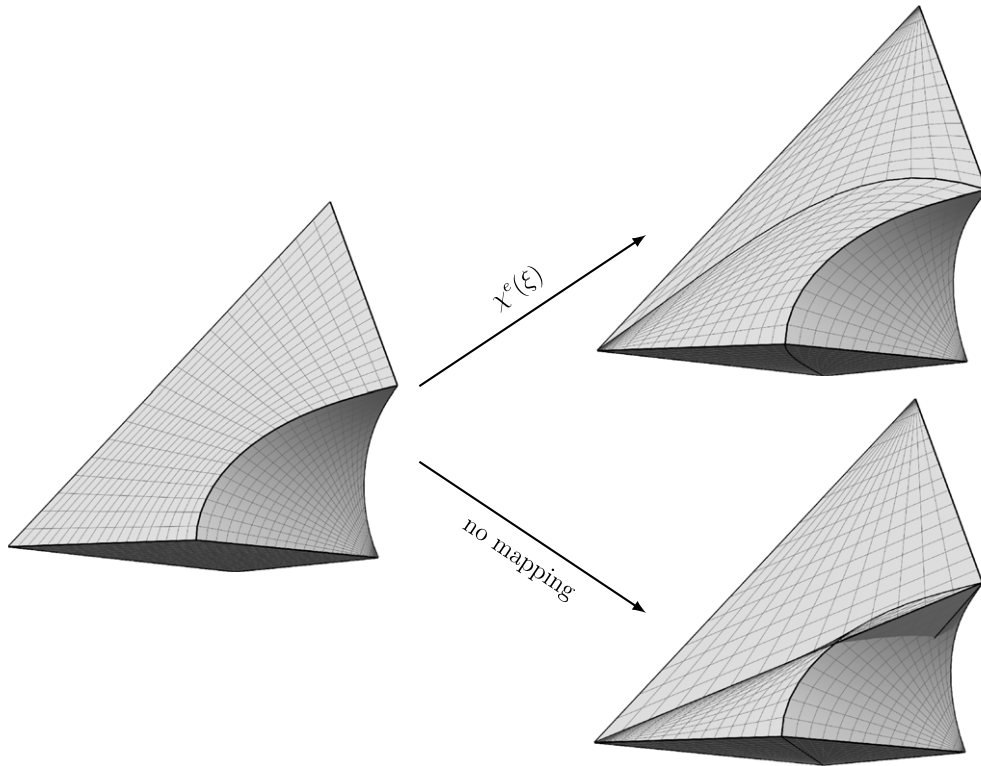
Fig. 6. Splitting a valid prismatic element at polynomial order $P = 20$ into tetrahedra. Applying the mapping χ^e leads to three valid tetrahedra (above), whereas preserving curvature on only a single face can lead to self-intersection of one or more tetrahedra (below).

2.4. Fully tetrahedral meshes

Many solvers do not have the capability of producing solutions on hybrid meshes, instead preferring to consider meshes containing only simplex elements (i.e. triangles and tetrahedra). This poses a considerable challenge for high-order mesh generation, particularly when extremely fine boundary layer elements are required.

One strategy is to adapt the prismatic boundary layer mesh by splitting each prism into three tetrahedra. Techniques such as the one presented in [24] are straightforward to implement and can be used to reliably split prismatic elements in a conformal manner. However these techniques are designed for linear finite element meshes. When curvature is introduced to the face of the tetrahedron lying on the surface of the geometry, and the prismatic element is sufficiently thin, then the lack of curvature on other faces causes the tetrahedron to self-intersect. This is illustrated in the lower half of Fig. 6.

To bypass this problem, we demonstrate how the refinement in the section above can be extended to introduce curvature into each of the tetrahedral elements in such a way that the elements are valid, as depicted in the upper half of Fig. 6. We begin by constructing the mappings χ^e defined above for each prism in the hybrid mesh. For the tetrahedral elements which are to be generated, we define a reference tetrahedron $\Omega_{\text{st}}^{\text{tet}} = \{(\xi_1, \xi_2, \xi_3) \mid \xi_1 \in [-1, 1], \xi_1 + \xi_2 \leq 1, \xi_1 + \xi_2 + \xi_3 \leq 1\}$ and a mapping ζ^e, which for each tetrahedron is represented by an expansion of modal functions

$$\zeta_i^e(\xi_1, \xi_2, \xi_3) = \sum_{p=0}^{P} \sum_{q=0}^{Q-p} \sum_{r=0}^{R-p-q} (\hat{\chi}_i)_{pqr} \psi_p^a(\eta_1) \psi_{pq}^b(\eta_2) \psi_{pqr}^c(\eta_3), \tag{2}$$

where $P \leq Q \leq R$. We again utilize a collapsed coordinate system $(\eta_1, \eta_2, \eta_3) \in [-1, 1]^3$ with

$$\eta_1 = \frac{-2(1 + \xi_1)}{\xi_2 + \xi_3} - 1, \qquad \eta_2 = \frac{2(1 + \xi_2)}{1 - \xi_3} - 1, \qquad \eta_3 = \xi_3,$$

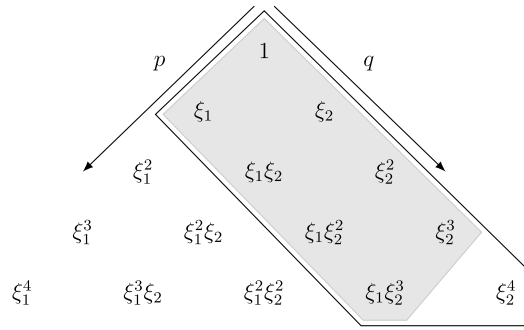and the additional modified hierarchical basis function ψ^c is defined in [1].

Fig. 7. Pascal's triangle of polynomial spaces in the (ξ_1, ξ_2) plane. It is used here to illustrate that to guarantee that the polynomial representation is correct and thus the mesh valid, the space of the tetrahedral element (black outline) must contain that of the prismatic element (shaded). We are assuming the element is linear ($p = 1$) in the ξ_1 direction.

We then proceed in a similar fashion to the prismatic splitting algorithm by first splitting the reference element of the prism, as opposed to the prism in Cartesian space. For each prismatic element in the domain, which we assume has been generated by the procedure in the previous section, we construct the isoparametric mapping χ^e. Assuming that each vertex in the mesh has a unique identifying number, we split $\Omega_{\mathrm{st}}^{\mathrm{prism}}$ into three tetrahedra according to the criterion defined in [24]. This ensures that where two prisms are connected by a quadrilateral face, the face is split in a consistent manner and a conformal mesh is produced. High-order information is then inserted into each tetrahedron in a bottom-up fashion by evaluating χ^e at positions in the standard prismatic element which correspond to the edge- and face-interior points. The insertion of this curvature information prevents each tetrahedron self-intersecting, leading to the generation of a valid mesh.

We must note here that if the prismatic mapping is linear in the ξ_2 direction (which is a typical situation for many meshes) we can also guarantee the curved tetrahedral elements have the same mapping properties as the original prismatic elements if the polynomial space is enriched. The requirement for obtaining a valid tetrahedral mesh after subdivision is that the polynomial space of the tetrahedral element must contain that of the prismatic element. If not, then there are terms in the mapping χ^e which cannot be represented in the expansion ζ^e, which may result in an invalid mesh.

In Fig. 7 we demonstrate this by representing the expansion spaces of representative quadrilateral and triangular faces, defined for $P \le Q$ as

$$\mathcal{Q} = \{\xi^p \xi^q \mid 0 \le p \le P,\ 0 \le q \le Q\},$$
$$\mathcal{T} = \{\xi^p \xi^q \mid 0 \le p \le P,\ 0 \le p + q \le Q\},$$

in the form of a Pascal's triangle. We see that in order for the condition on polynomial spaces to be satisfied, we require the triangle to have an expansion order of one greater than the quadrilateral, otherwise the cross-term $\xi_1 \xi_2^3$ is not contained within the triangular space.

2.5. Mesh spacing following a geometric progression

In this section we describe the form of the spacing function which determines the height Δ_k of each sub-element of the refined mesh. Recall that given a partition $-1 = x_0 < \cdots < x_n = 1$ of $[-1, 1]$, we set $\Delta_k = x_k - x_{k-1}$ for $1 \le k \le n$. To generate a spacing which gradually refines the elements towards the boundary lying at $\xi_2 = -1$, we utilize a geometric progression

$$x_k = x_{k-1} + ar^k, \quad a = \frac{2(1 - r)}{1 - r^{n+1}}$$

for $1 \le k < n$ with $x_0 = -1$ and fix $x_n = 1$. The parameter r controls the relative sizes of elements, and $\Delta_1 = ar - 1$ is the height of the element closest to the boundary. Since in this formulation a is a function of r, we note that given a desired physical thickness (for example, a requirement for the element closest to the wall to have a thickness corresponding to $y^+ = 1$), one may derive the necessary value of r given n or vice versa. Fig. 8 illustrates a sequence
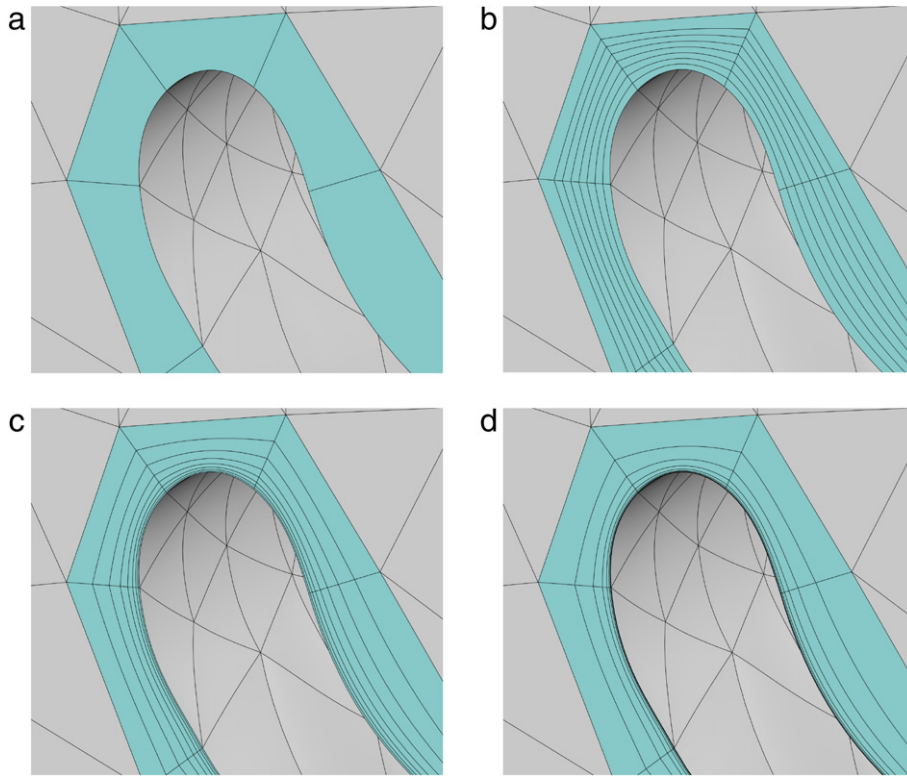
Fig. 8. A sequence of meshes obtained by splitting macro-elements into $n = 8$ elements using a geometric progression and various values of r. (a) The macro-element mesh; (b) $r = 1$; (c) $r = \frac{3}{2}$; (d) $r = 2$.

of meshes with various values of r (and thus y^+) that can be obtained using this process. Additionally we note that if $r = 1$ as in Fig. 8(b), then a is ill-defined. Since $r = 1$ corresponds to the case where all elements are of an equal size, we assume a uniform spacing so that $\Delta_k = \frac{1}{n}$.

A further point to note is that a skin friction obtained from an empirical relation $C_f = F(\text{Re})$ is usually derived using flat-plate calculations or regression based on experimental measurements. However, for complex geometries, there is usually no way to predetermine an exact value for C_f, and therefore a simulation is required to calculate an accurate value. An empirical relation gives an estimate for an initial simulation, and based on a series of experiments where the mesh is further refined (or coarsened) an accurate value of C_f can be obtained.

In addition, one can also smoothly vary the geometric factor as a function of spatial position so that different areas of the mesh receive varying levels of refinement. For each macro-element Ω^e, we consider an elementally-varying spacing distribution Δ_k^e. In order for the mesh to be conformal, it is clear that the splitting procedure must be altered slightly, so that the sub-elements which are produced connect correctly through quadrilateral faces. We consider a simple procedure where, for each of the three edges which connect the two triangular faces of a prism, we arbitrarily choose a spacing function from any of the prismatic macro-elements elements which share this edge. When each sub-element is constructed, we use the distribution of points previously defined on these three edges. The method then proceeds in an identical fashion. By adopting this simple technique we assume that Δ_k^e varies smoothly so that the jump between different spacing distributions is not 'too large', or else the addition of high-order curvature may cause sub-elements to self-intersect.

We consider an example of this spatially-varying distribution in Fig. 9, which depicts a staggered wing configuration. The thin trailing edge of each blade dictates that extremely small elements are required. When the mesh is split with a constant distribution of points, any solvers which have a CFL constraint are therefore forced to use a prohibitively small timestep in order to maintain a stable simulation. Whilst this problem could be solved without varying the spacing distribution by instead increasing the height δ of the macro-elements, here the close proximity of the two wings can cause issues when generating the volume mesh of tetrahedra in the space between the wings. In
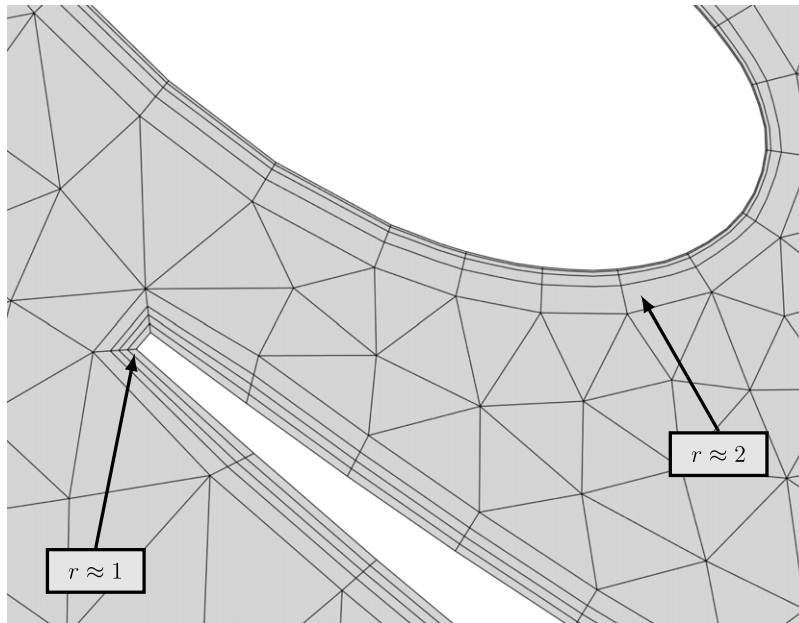
Fig. 9. Example of a two-wing configuration in which the geometric progression r is varied as a function of space.

this case then, the use of a spatially-varying height distribution is essential, and we smoothly vary the ratio of element sizes so that at the leading edge $r \approx 2$ whereas at the trailing edge $r \approx 1$.

2.6. A remark on mesh quality

One of the most important aspects of mesh generation software is the ability to produce meshes containing not only valid but good quality elements. Poor quality elements generally translate to stability restrictions in the case of explicit hyperbolic problems, or to poor conditioning of matrices in the case of implicit elliptic and parabolic problems. Metrics for assessing the quality of high-order elements have only been recently investigated, and most measures of quality proposed so far quantify it by means of distortion with respect to an ideal element [25,26].

We note however that boundary layer meshes are intentionally highly stretched as there is an underlying desire to refine the mesh only where is necessary. Ideally, one should obtain a mesh with a number of elements which is sufficiently large to resolve the shear in the flow, but small enough to make it viable for computation. Any metric based solely on measuring the geometric distortion of elements will therefore undoubtedly show the presence of 'poor quality' elements; however, since the gradient of the flow is far smaller in the transverse direction of velocity compared to the normal component, these meshes are in fact well-suited to the fluid problems they are applied to [27].

Nevertheless, it is certainly important to maximize the Jacobian of near-wall elements under the distortion constraint of the previous paragraph. In the context of our method, we note that for prismatic boundary layers, the spacing distribution gives rise to an orientation-preserving affine mapping between a given sub-element of Ω_{st} and Ω_{st}. The Jacobian of each sub-element of Ω^e is therefore a scalar multiple of Ω^e determined by the derivative of the affine mapping through an application of the chain rule. This calculation can be performed in a similar fashion for tetrahedral boundary layers, and we give a more detailed description of this process, together with more general mappings on a variety of element types, in other work [28]. In particular, this implies that the Jacobian of sub-elements can be maximized by applying an optimization technique to the coarse grid, since the resulting Jacobian of sub-elements is a function of the original coarse grid and the desired boundary layer thickness.

3. Applications to aeronautical geometries

In this section, we utilize the splitting techniques of Section 2 to produce boundary layer meshes for three complex aeronautical geometries in order to demonstrate the applicability of the technique in industrial problems. For each
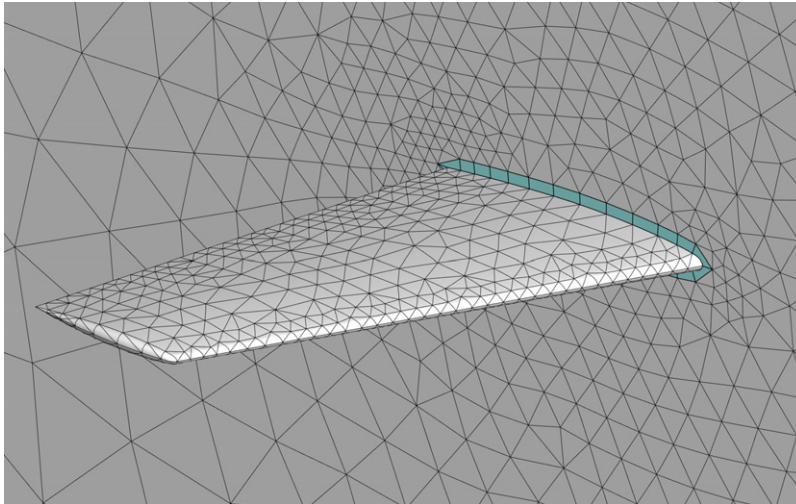
Fig. 10. Boundary of the initial coarse straight-sided mesh with prisms in the boundary layer (highlighted) and tetrahedra in the interior.

case, we consider a high-order coarse discretization at polynomial orders of $5 \leq P \leq 15$ to show the viability of the method even at very high polynomial orders.

3.1. ONERA M6 wing

The ONERA M6 wing is a classic CFD validation case for external transonic flow. The description of the geometry is given in [29]. Here we use this case as an illustration of the methodology for generating high-order meshes for high Reynolds number flows. In a typical simulation of transonic flow, the Reynolds number has a value of around 10^5, and therefore an extremely thin boundary layer is required in the wall-normal direction. However, in the transverse directions such high resolution is not necessarily required.

Fig. 10 shows the initial coarse high-order surface mesh of the wing geometry coupled with a symmetry plane. The intersection of the symmetry plane with the boundary layer of the wing, composed of the prismatic macro-elements, is highlighted, with the remainder of the mesh comprising tetrahedral elements. In Fig. 11 we apply the splitting strategies from Section 2 to obtain a mesh with $y^+ \approx 5$ at $\text{Re} = 10^5$, where the skin friction C_f is approximated using the Schlichting formula [4]

$$C_f \approx (2 \log(\text{Re}) - 0.65)^{-2.3} .$$

Fig. 11(a) shows the hybrid prismatic–tetrahedral mesh that is obtained after the splitting algorithm is applied. In Fig. 11(b) we apply the prism to tetrahedron splitting technique of Section 2.4 to obtain a fully tetrahedral mesh. In all cases a valid mesh is generated, with no elements having negative Jacobian.

3.2. High Reynolds number flow over a NACA 0012 wing

To further demonstrate the applicability of our technique to our aeronautical simulations, we present meshes and qualitative results from simulations of high Reynolds number flows over a NACA 0012 wing with a $\alpha = 10°$ angle of attack. This test case is fairly common but is challenging in terms of simulation due to the presence of highly transient flow dynamics, due to the formation of a strong wingtip vortex.

In this instance, although CAD data is available, we choose to highlight an alternative method of coarse grid generation. We utilize a typical commercial grid generator package (in this case STAR-CCM+) in order to generate a coarse linear grid comprising of prismatic elements in the boundary layer and tetrahedra elsewhere. To smooth the surface, we combine vertex normals which are calculated using the CAD surface information together with the spherigon patch technique [23] to produce a high order surface. For this test case, this approach is both efficient, straightforward and relatively robust so long as the prismatic boundary layer is sufficiently thick.

The valid coarse grid is visualized in Fig. 12, with the boundary layer highlighted in blue together with the symmetry plane and bottom surface of the computational domain. In Fig. 13 we give a closer view of the prismatic boundary
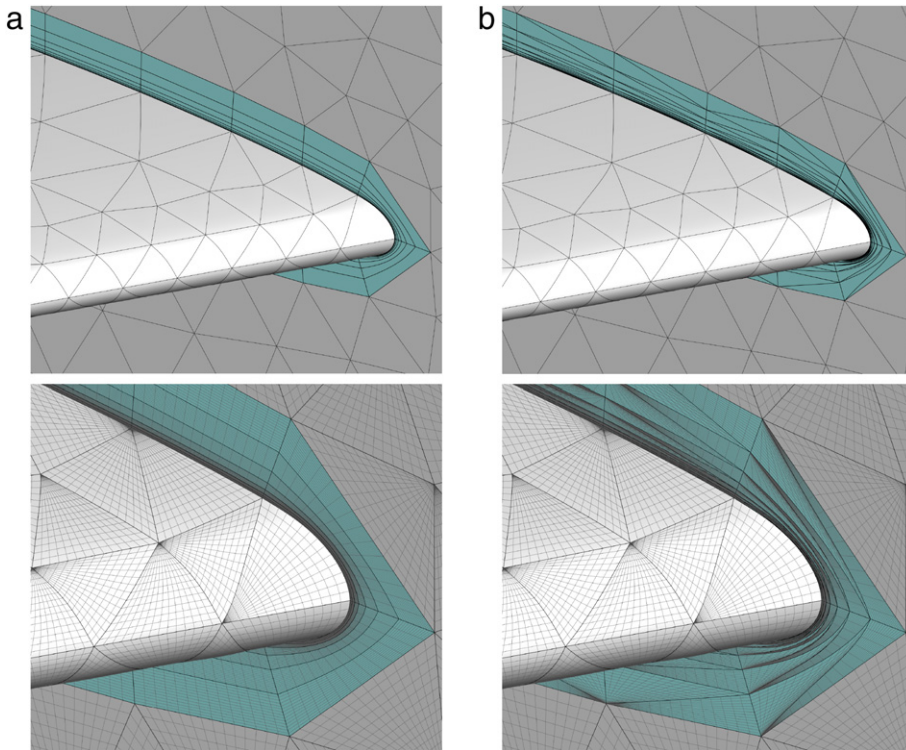
Fig. 11. Enlargements of the initial coarse mesh demonstrating the splitting of the boundary layer into (a) prismatic and (b) tetrahedral elements near the leading edge at polynomial order $P = 15$. The lower picture highlights interior quadrature points of the elements.
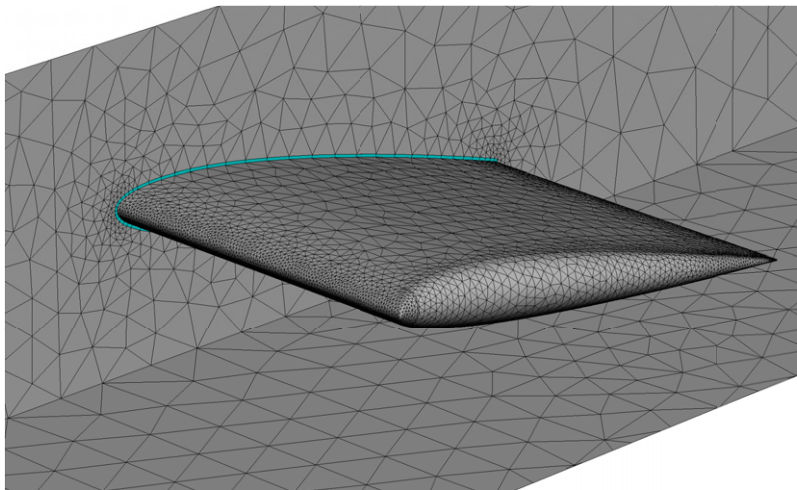

Fig. 12. Surface mesh of a NACA 0012 wing geometry at polynomial order $P = 5$. The coarse prismatic boundary layer is highlighted in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

layer on the left, and on the right demonstrate how it can be refined using the technique of the previous section. In this case we use $n = 3$ layers with a geometric progression of $r = 3$ to generate a much finer grid in the wall-normal direction at a polynomial order of $P = 5$.

As a demonstration of both the validity of the refinement procedure, and the suitability of such meshes for high-Re flows, Fig. 14 shows a representative snapshot from a LES simulation at polynomial order $P = 5$ at a Reynolds number of $\text{Re} = 5 \times 10^5$. In the visualization we show streamlines of the velocity field which show that the vortex generated at the wingtip is successfully captured.
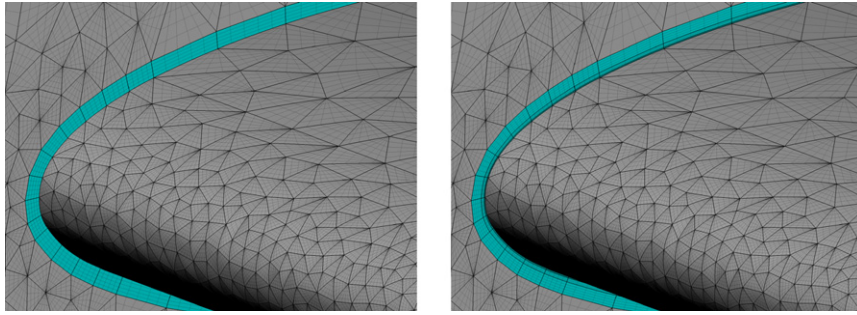
Fig. 13. Closer view of the NACA 0012 wing geometry near the leading edge of the wing where curvature of the surface is at its highest. The right hand figure shows the refined boundary layer which uses three layers and a progression ratio $r = 3$.
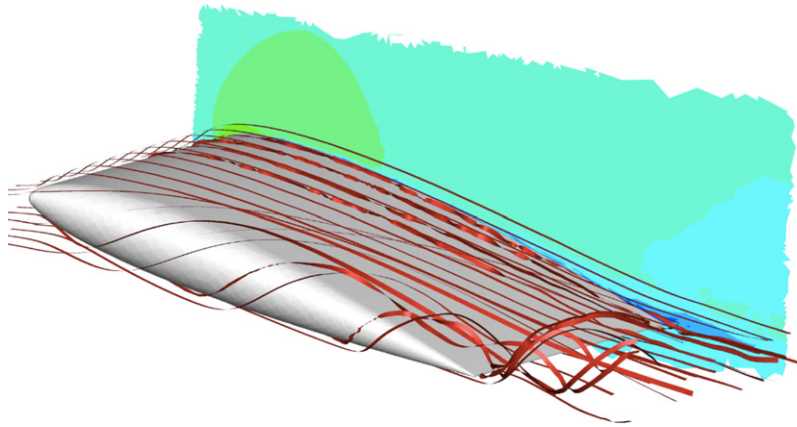

Fig. 14. A representative snapshot of a LES simulation at $P = 5$ at Re $= 5 \times 10^5$ on the NACA 0012 wing geometry showing streamlines of the velocity field.
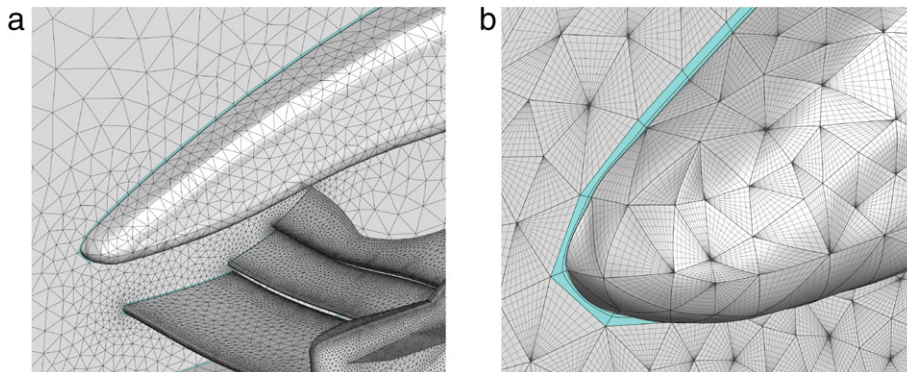

Fig. 15. Front section of a Formula 1 car at polynomial order $P = 10$. (a) The high-order surface mesh of the geometry; (b) a closer view of the curvature of elements in the boundary layer (highlighted) near the nose of the car.

3.3. Front section of a Formula 1 car

Finally, we consider a more complex example: the front section of a representative Formula 1 car. In this case, the goal is to refine a thin boundary layer around the main body of the car, as well as the wing and flap arrangement. Fig. 15(a) shows the high-order surface mesh of the full geometry. In Fig. 15(b) we show the extremely thin prismatic elements generated near the surface of the nose, which arise from splitting the macro-elements using a geometric progression ratio of $r = 3$. Once more we obtain a valid mesh.

4. Conclusions

We have presented a technique for generating highly stretched high-order boundary-layer meshes as required by current CFD solvers. The proposed technique is very effective, and modular in the sense that it can readily be combined with a number of existing coarse grid generation techniques in order to produce a mesh which is well-suited to boundary layer flows. Starting from a valid coarse prismatic boundary-layer mesh, our isoparametric approach permits the generation of a sequence of meshes with increased resolution with very little additional cost. This should prove valuable for mesh convergence studies at high Reynolds numbers. We have established requirements of validity for modal elemental shape functions, but the same arguments are also applicable to guarantee the validity of the mesh when using nodal shape functions.

The main limitation of the technique, as presented here, is the requirement that the subdivision of the prismatic mesh should be accomplished without affecting the rest of the mesh. Although extending the method to other cases is not a difficult technical issue, it will require the use of transition elements such as pyramids. Whilst this case is beyond the scope of this article, in [28] we give a more detailed description as to how one may split different element types. This goes some way towards solving this problem, and may offer the chance to extend this technique with new functionality, such as introducing variable numbers of boundary layer elements around the profile of an aerofoil.

Acknowledgements

References

[1] G.E. Karniadakis, S.J. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, second ed., Oxford University Press, 2005.

[2] P.E. Vincent, A. Jameson, Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists, Math. Model. Nat. Phenom. 6 (3) (2011) 97–140.

[3] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, Internat. J. Numer. Methods Fluids 72 (8) (2013) 811–845.

[4] H. Schlichting, Boundary-layer Theory, seventh ed., McGraw-Hill, 1979.

[5] J. Peraire, K. Morgan, Unstructured mesh generation including directional refinement for aerodynamic flow simulation, Finite Elem. Anal. Des. 25 (1997) 343–356.

[6] D.J. Mavriplis, Unstructured grid techniques, Annu. Rev. Fluid Mech. 29 (1997) 473–514.

[7] S. Pirzadeh, Three-dimensional unstructured viscous grids by the advancing-layers method, AIAA J. 34 (1) (1996) 43–49.

[8] S. Dey, R.M. O'Bara, M.S. Shephard, Curvilinear mesh generation in 3D, in: Proceedings of the 8th International Meshing Roundtable, 1999, lake Tahoe, California.

[9] S.J. Sherwin, J. Peiró, Mesh generation in curvilinear domains using high-order elements, Internat. J. Numer. Methods Engrg. 53 (1) (2002) 207–223.

[10] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, Internat. J. Numer. Methods Engrg. 79 (11) (2009) 1309–1331.

[11] P.-O. Persson, J. Peraire, Curved mesh generation and mesh refinement using Lagrangian solid mechanics, in: 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando (FL), USA, 2009, AIAA paper 2009–949.

[12] Z.Q. Xie, R. Sevilla, O. Hassan, K. Morgan, The generation of arbitrary order curved meshes for 3D finite element analysis, Comput. Mech. 51 (3) (2013) 361–374.

[13] T. Toulorge, C. Geuzaine, J.-F. Remacle, J. Lambrechts, Robust untangling of curvilinear meshes, J. Comput. Phys. 254 (2013) 8–26.

[14] A. Gargallo-Peiró, X. Roca, J. Sarrate, J. Peraire, Inserting curved boundary layers for viscous flow simulation with high-order tetrahedra, in: 22nd International Meshing Roundtable, Orlando, Florida, 2013.

[15] X.-J. Luo, M.S. Shephard, L.-Z. Yin, R.M. OBara, R. Nastasi, M.W. Beall, Construction of near optimal meshes for 3D curved domains with thin sections and singularities for p-version method, Eng. Comput. 26 (3) (2010) 215–229.

[16] O. Sahni, X.J. Luo, K.E. Jansen, M.S. Shepard, Curved boundary layer meshing for adaptive viscous flow simulations, Finite Elem. Anal. Des. 46 (1–2) (2010) 132–139.

[17] O.C. Zienkiewicz, D.V. Phillips, An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co-ordinates, Internat. J. Numer. Methods Engrg. 3 (4) (1971) 519–528.

[18] W.J. Gordon, C.A. Hall, Construction of curvilinear co-ordinate systems and applications to mesh generation, Internat. J. Numer. Methods Engrg. 7 (4) (1973) 461–477.

[19] J. Peiró, A.I. Sayma, A 3-D unstructured multigrid Navier–Stokes solver, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics V, Oxford University Press, 1995.

[20] J. Peraire, J. Peiró, K. Morgan, Multigrid solution of the 3-D compressible Euler equations on unstructured tetrahedral grids, Internat. J. Numer. Methods Engrg. 36 (1993) 1029–1044.

[21] M.P. Do Carmo, Differential Geometry of Curves and Surfaces, Prentice-Hall, 1976.

[22] K. Bock, J. Stiller, Generation of high-order polynomial patches from scattered data, in: Spectral and High Order Methods for Partial Differential Equations, ICOSAHOM 2012, Springer, 2014, pp. 157–167.

[23] P. Volino, N.M. Thalmann, The spherigon: a simple polygon patch for smoothing quickly your polygonal meshes, in: Computer Animation 98. Proceedings, IEEE, 1998, pp. 72–78.

[24] J. Dompierre, P. Labbé, M.-G. Vallet, R. Camarero, How to subdivide pyramids, prisms and hexahedra into tetrahedra, in: 8th International Meshing Roundtable, Lake Tahoe, California, 1999.

[25] A. Johnen, J.-F. Remacle, C. Geuzaine, Geometrical validity of high-order triangular finite elements, Eng. Comput. (2014) 375–382.

[26] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Defining quality measures for validation and generation of high-order tetrahedral meshes, in: 22nd International Meshing Roundtable, Orlando, Florida, 2013.

[27] H. Sun, D.L. Darmofal, R. Haimes, On the impact of triangle shapes for boundary layer problems using high-order finite element discretization, J. Comput. Phys. 231 (2012) 541–557.

[28] D. Moxey, M. Hazan, J. Peiró, S.J. Sherwin, On the generation of curvilinear meshes through subdivision of isoparametric elements, in: Proceedings of Tetrahedron IV, 2014, in press.

[29] V. Schmitt, F. Charpin, Pressure Distributions on the ONERA-M6 Wing at Transonic Mach Numbers, in: Experimental Data Base for Computer Program Assessment, Vol. AGARD AR 138, 1979, report of the Fluid Dynamics Panel Working Group 04.