Curvilinear mesh generation for boundary layer problems

D. Moxey, M. Hazan, S. J. Sherwin and J. Peiró

Abstract In this article, we give an overview of a new technique for unstructured curvilinear boundary layer grid generation, which uses the isoparametric mappings that define elements in an existing coarse prismatic grid to produce a refined mesh capable of resolving arbitrarily thin boundary layers. We demonstrate that the technique always produces valid grids given an initially valid coarse mesh, and additionally show how this can be extended to convert hybrid meshes to meshes containing only simplicial elements.

1 Introduction

As the popularity of high order methods continues to increase in academia and industry alike, there is an increasing demand for new robust mesh generation strategies which are capable of generating meshes for complex three dimensional geometries. Industrial aeronautics applications, such as the test cases considered in the IDIHOM project, are particularly demanding as they usually require the generation of boundary layer grids, where the mesh is refined near walls in order to adequately resolve high-shear regions of flow. The size of elements in the wall-normal direction is dictated by a wall unit

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

where y dnotes the coordinate normal to the wall, Re is the Reynolds number, C_f is the skin friction coefficient and L is a reference length. Industrial requirements typically use very high Reynolds numbers and require meshes with a resolution of $y^+ = 1$ so that the flow in the viscous sublayer is adequately resolved. Since the skin

David Moxey, Menashe Hazan, Spencer J. Sherwin, Joaquim Peiró
Department of Aeronautics, Imperial College London,
e-mail: {d.moxey, menashe.hazan10, s.sherwin, j.peiro}@imperial.ac.uk

friction coefficient scales as $O(\mathrm{Re}^{-2})$ or $O(\mathrm{Re}^{-3})$ [17], at the high Reynolds numbers which are required for aeronautical applications these grids are highly stretched in the wall-normal direction, so as to produce meshes which strike a balance between accuracy and computational efficiency. Such meshes typically have elements possessing stretching ratios of 1:1000 or even lower.

Whilst there are many well-established methods for generating linear boundary layer grids [12, 8, 15], the development of such techniques for unstructured curvilinear meshes has been limited due to the relatively recent nature of high order mesh generation strategies [2, 19, 5]. The most recent developments in curvilinear mesh generation methods rely on approaches that split a linear boundary conforming mesh into high-order straight-sided elements, project the boundary nodes onto the curved surface and then deform the mesh to accommodate them [14, 21] or alternatively undertake an optimization procedure to untangle it [20]. Some work has been undertaken to investigate the applicability of these methods to boundary layer meshes [4, 16]. However, the expensive nature of these techniques, particularly at high polynomial orders, and the uncertainty of the resolution that can be obtained when the mesh is deformed, means that their application to boundary layer grids remains to be investigated.

The common theme of these methods however is that they rely on the generation of a dense linear boundary layer grid, and then undertake an optimization procedure in order to deform it. In this chapter, we given an overview of an alternative method proposed in [10, 9] and developed during the IDIHOM project that aims 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 [22] or, in general, a transfinite interpolation [6] where a high-order coarse boundary-layer mesh is subdivided using the same mapping that define these high-order elements. The procedure is also very versatile as it permits meshes with different distributions of y^+ to be generated with ease and, further, the validity of these meshes is guaranteed if the initial mesh is valid and the polynomial space is chosen appropriately.

An overview of this process can be seen for a representative quadrilateral element in figure 1, where we assume the bottom edge of the element is attached to the wall, and therefore require extra resolution in the vertical direction. The top row shows how the standard quadrilateral element Ω_{st} is deformed under the mapping χ to produce a curved element Ω. To refine the element in physical space, we first split Ω_{st} into a series of smaller elements as shown in bottom left of the figure. Applying the mapping χ to these subelements of the standard region leads to the production of curved subelements of the physical element as desired.

This chapter is organized in three main sections. Section 2 describes the various procedures involved in generating arbitrarily high-order boundary layer meshes suitable for these flows in three dimensions: a process for generating the initial coarse mesh comprising of a prismatic boundary layer; the splitting of the prismatic elements into finer prisms; and the spacing distributions used to define the size of the finer subelements. We also demonstrate how the method can be adapted to produce a curvilinear mesh containing only simplicial elements. The procedure for splitting

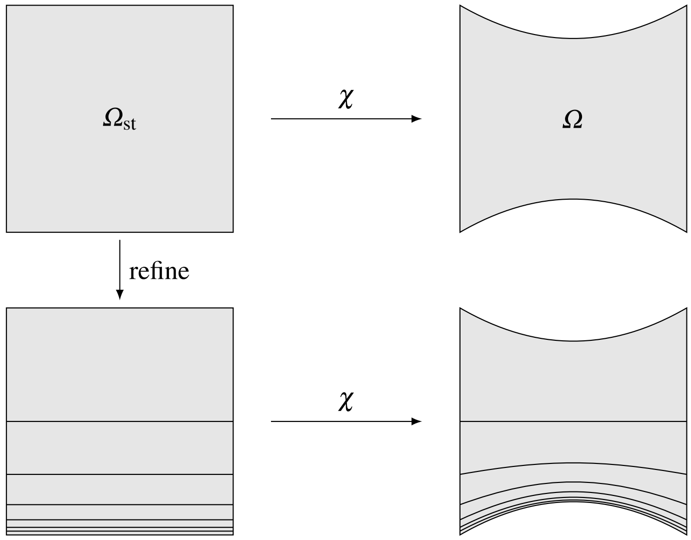


Fig. 1 Overview of boundary layer refinement technique.

the coarse boundary-layer mesh into a fine prismatic mesh whilst ensuring its validity and the mathematical framework for a generalisation of the method to other element types are described in section 3. Section 4 presents examples of application of the methodology to the generation of a high-order mesh for the ONERA M6 wing. A final section on conclusions follows.

2 Mesh generation strategy

Generating meshes for simulations of turbulent viscous flows is very challenging due to the presence of a extremely thin boundary layer close to walls. In these regions the flow posses a high level of shear, and so the accurate representation of flow features typically requires far greater mesh resolution than other areas of the domain. However, since this level of accuracy is not usually required in transverse directions, computational efficiency necessarily dictates the use of elements possessing stretching ratios of the order of 1:1000 or below. This resolution is of critical importance for turbulence simulations, as many of the structures that lead to the development of turbulence form inside or near the boundary layer. This section describes a method to generate high-order meshes meeting this requirement. Given a high-order coarse mesh consisting of prisms in the boundary layer and tetrahedra elsewhere, a finer mesh is obtained by splitting the coarse prismatic elements (referred to in the following as ‘macro-elements’) into finer elements, utilising the

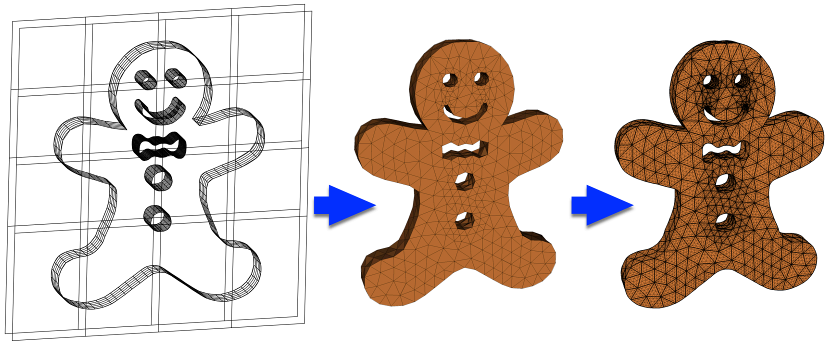


Fig. 2 High-order meshing strategy.

mapping that defines the high-order prisms to insert the necessary curvature into subelements. The various steps involved are described in the following sections.

2.1 Generation of the coarse hybrid mesh

We begin by outlining a procedure for generating the coarse hybrid mesh. This procedure is essentially the one presented in [19] and will be briefly summarized here for completion. 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 methodology is illustrated in figure 2. We emphasise however that this method is not the only one for generating a coarse grid, and the method we propose in the next section may equally well be applied to any valid coarse grid.

The geometry of the computational domain is represented by means of a CAD spline curves and surfaces to obtain a boundary representation (B-Rep) of the computational domain. The linear mesh is then generated using an implementation of the method of advancing normals [11] to generate a boundary-layer mesh of triangular prisms near the walls in the computational domain. The advancing front technique [13] 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. This follows the method presented in [19] where the generation of the additional degrees of freedom required to obtain a high-order mesh proceeds in a bottom-up fashion, starting with the edges, then the faces and finally the interior of the elements. One of the principal contributions of [19] is the development of a method for calculating the positions of the newly generated

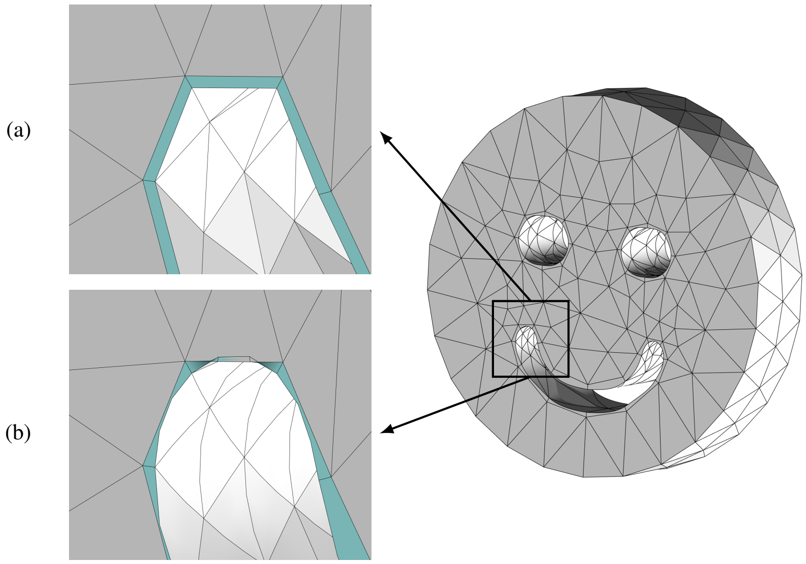


Fig. 3 Generation of invalid elements in concave regions of the computational domain. (a) depicts a valid linear mesh and (b) shows the self-intersection that often occurs when high-order information is introduced.

points by minimizing an energy of deformation that accounts for anisometries in the CAD surface mapping. 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. Figure 3 shows 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. In the following section, we describe a strategy to palliate this problem.

2.2 *Determining a suitable height of the macro-element*

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 curvature of the surface is obtained through interrogation of the B-Rep of the computational domain.

Following the notation of figure 4 and considering a two-dimensional domain, we approximate a curve by its osculating circle [1]. 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

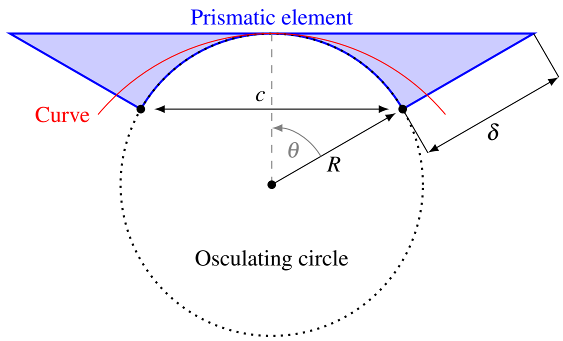


Fig. 4 Notation used for the calculation of the minimum height δ of a valid prismatic macro-element.

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

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. In the following section, we describe how a prismatic boundary layer mesh generated using this criterion is refined in an isoparametric sense.

2.3 The isoparametric approach

The main idea of the splitting method is to interpret the prismatic macro-element as a mapping, $\chi^e : \Omega_{\mathrm{st}} \to \Omega^e$, from a reference element Ω_{st} to obtain the physical element Ω^e. This mapping is illustrated in figure 5.

There are many ways to define this mapping. Here we adopt an approach in which χ^e isoparametrically projects coordinates $\xi = (\xi_1, \xi_2, \xi_3)$ in a reference element $\Omega_{\mathrm{st}} = \{(\xi_1, \xi_2, \xi_3) \mid \xi_1, \xi_2 \in [-1,1], \xi_1 + \xi_3 \leq 1\}$ onto the 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

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

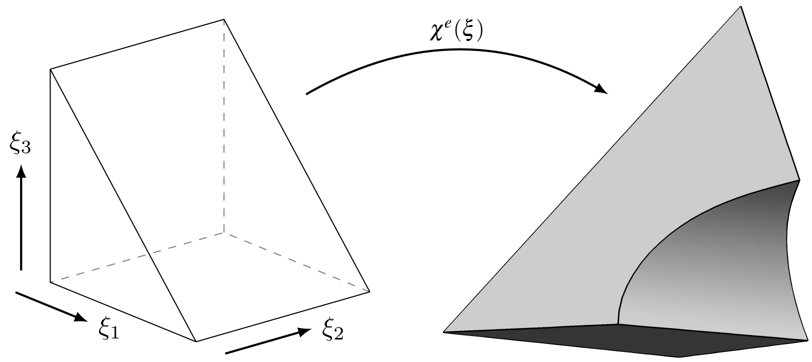


Fig. 5 High-order element mapping from a standard prism Ω_{st} to a corresponding prism Ω^e in Cartesian space.

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 [7], 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 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 $(\hat{\chi}_i)_{pqr}$.

2.4 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

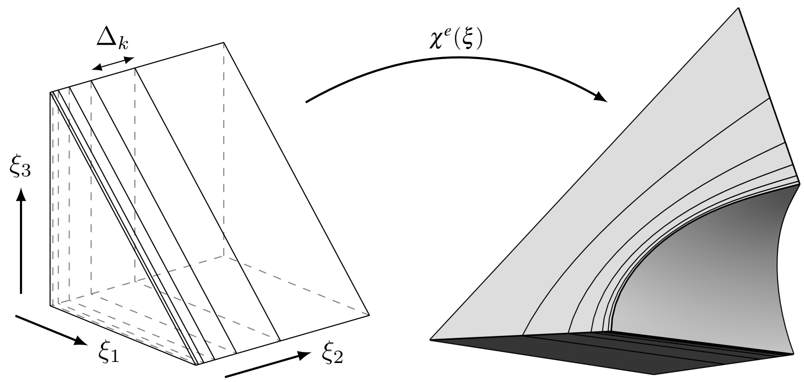


Fig. 6 Splitting $\Omega_{\rm st}$ and applying the mapping χ^e to obtain a high-order layer of prisms from the macro-element of figure 5.

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 utilise χ^e in order to determine curvature information for each sub-element of the refinement. This projects the subelements back to Cartesian space, thereby splitting the original prism into curved prismatic subelements. This is depicted in figure 6.

Assuming that we wish to produce n subelements 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 figure 6. 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. We describe in more detail a method for defining Δ_k in section 2.6, and how the distribution of points can be varied between elements if we relax the guarantee of mesh conformity.

Each prism is split by iterating over the number of subelements 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 $\Omega_{\rm st}$, and then apply the mapping 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 thus equivalent to an affine transformation that will preserve the sign of the Jacobian of the mapping and therefore the validity of the mesh. Further, the subelements span the same polynomial space as the original macro-element, therefore the use of the macro-element mapping to define the sub-element mappings means they span an identical space and so have identical properties to the original element.

2.5 Fully tetrahedral meshes

Many solvers do not have the capability of producing solutions on hybrid meshes, instead preferring to consider meshes containing only simplicial 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 [3] 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 figure 7.

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 figure 7. We begin by constructing the mapping 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 $\Omega_{\mathrm{st}}^{\mathrm{tet}} = \{(\xi_1,\xi_2,\xi_3) \mid \xi_1 \in [-1,1], \xi_1+\xi_2 \leq 1, \xi_1+\xi_2+\xi_3 \leq 1\}$ 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 utilise a collapsed coordinate system $(\eta_1,\eta_2,\eta_3) \in [-1,1]^3$ with

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

and the additional modified hierarchical basis function ψ^c is defined in [7].

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

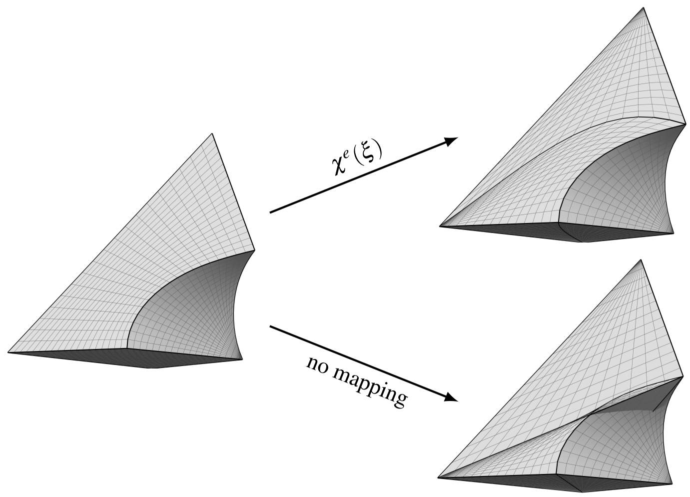


Fig. 7 Splitting a valid prismatic element at polynomial order $P = 20$ into tetrahedra. Applying the mapping χ^e leads to three valid tetrahedra (above), whereas preserving curvature on only a single face can lead to self-intersection of one or more tetrahedra (below).

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 [3]. 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.

An examination of the conditions under which the subdivision process produces valid elements, not only for the refinement procedure outlined here, but for more generic transformations of the standard element is presented in Section 3.

2.6 Normal 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 $1 \leq k \leq n$. To generate a spacing which gradually refines the elements towards the boundary lying at $\xi_2 = -1$, we utilise a geometric progression

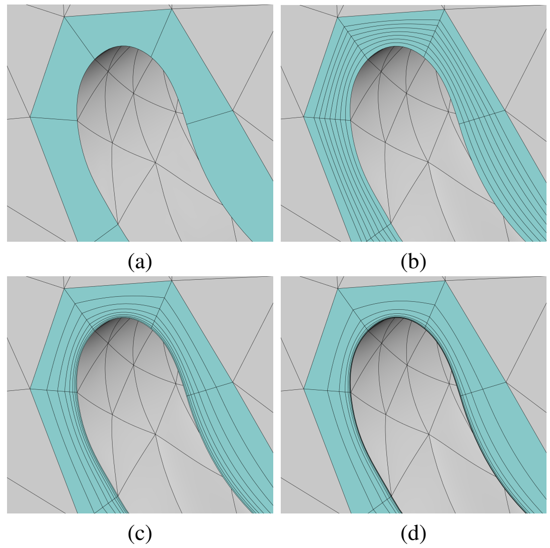


Fig. 8 A sequence of meshes obtained by splitting macro-elements into $n = 8$ elements using a geometric progression and various values of r. (a) The macro-element mesh; (b) $r = 1$; (c) $r = \frac{3}{2}$; (d) $r = 2$.

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

for $1 \leq k < n$ 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. Figure 8 illustrates a sequence 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 figure 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 $C_f = F(\mathrm{Re})$ 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 subelements which are produced connect

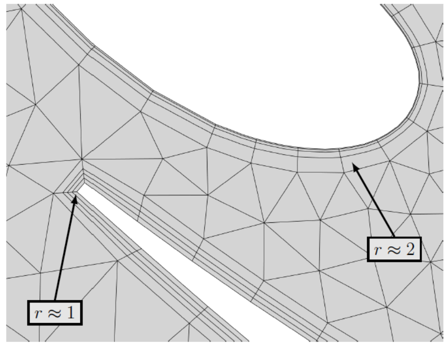


Fig. 9 Example of spatially-varying distribution of ratio of element size, r, for a staggered wing configuration

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 arbitraily 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 subelements to self-intersect.

We consider an example of this spatially-varying distribution in figure 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. This problem can be solved without varying the spacing distribution and by increasing the height δ of the macro-elements, but the close proximity of the two wings can cause issues when generating the volume mesh of tetrahedra in the space between the wings. In 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 $r \approx 1$.

3 Subdivision strategies and analysis of element validity

The purpose of this section is to frame the subdivision technique in the context of a more general mathematical framework, to investigate the range of conditions under which the resulting meshes are valid, and to demonstrate how it can be utilised to subdivide a broader range of elemental types in both two and three dimensions. We note that in general, the subdivision of elements in this manner often requires the enrichment of the polynomial space so that the subdivided elements capture all curvature of the original element.

3.1 Mathematical framework

We begin by considering a finite element Ω^e, which in general belongs to a mesh arising from the tesselation $\mathscr{T}(\Theta) = \{\Omega^1, \dots, \Omega^{N_{\mathrm{el}}}\}$ of some domain $\Theta \subset \mathbb{R}^n$ with $n = 2,3$ of N_{el} elements, so that

$$\Theta = \bigcup_{e=1}^{N_{\mathrm{el}}} \Omega^e, \qquad \Omega^e \cap \Omega^f = \emptyset \text{ if } e \neq f.$$

In two dimensions, we consider quadrilateral and triangular elements, and in three dimensions tetrahedral, prismatic and hexahedral elements. In order to introduce curvature into an element Ω (where we drop the superscript e for convenience), we assume there exists a mapping $\chi : \Omega_{\mathrm{st}} \to \Omega$ which projects a canonical standard element Ω_{st} into the Cartesian coordinates defining an element. In this work we define reference elements to be

$$\begin{aligned}
\Omega_{\mathrm{st}}^{\mathrm{quad}} &= \{(\xi_1, \xi_2) \quad\ \ \mid -1 \leq \xi_1, \xi_2 \leq 1\},\\
\Omega_{\mathrm{st}}^{\mathrm{tri}} &= \{(\xi_1, \xi_2) \quad\ \ \mid -1 \leq \xi_1 + \xi_2 \leq 1\},\\
\Omega_{\mathrm{st}}^{\mathrm{hex}} &= \{(\xi_1, \xi_2, \xi_3) \mid -1 \leq \xi_1, \xi_2, \xi_3 \leq 1\},\\
\Omega_{\mathrm{st}}^{\mathrm{pri}} &= \{(\xi_1, \xi_2, \xi_3) \mid -1 \leq \xi_1 + \xi_3 \leq 1, -1 \leq \xi_2 \leq 1\},\\
\Omega_{\mathrm{st}}^{\mathrm{tet}} &= \{(\xi_1, \xi_2, \xi_3) \mid -1 \leq \xi_1 + \xi_2 + \xi_3 \leq 1\},
\end{aligned}$$

respectively. Inside the standard elements we define a polynomial space in terms of the reference coordinates $\xi = (\xi_1, \xi_2, \xi_3)$ from which an expansion basis is selected. Assuming that we select a polynomial order P, Q and R for each coordinate direction, the polynomial spaces take the form

$$\mathscr{P}(\Omega_{\mathrm{st}}) = \mathrm{span}\{\xi_1^p \xi_2^q \xi_3^r \mid (pqr) \in \mathscr{I}\}$$

where $\mathscr{I}$ represents an indexing set, defined for each element as

$$\begin{aligned}
\mathscr{I}^{\mathrm{quad}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq q \leq Q,\ r = 0\}\\
\mathscr{I}^{\mathrm{tri}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq p+q \leq Q,\ r = 0,\ P \leq Q\}\\
\mathscr{I}^{\mathrm{hex}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq q \leq Q,\ 0 \leq r \leq R\}\\
\mathscr{I}^{\mathrm{pri}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq q \leq Q,\ 0 \leq p+r \leq P,\ P \leq R\}\\
\mathscr{I}^{\mathrm{tet}} &= \{(pqr) \mid 0 \leq p \leq P,\ 0 \leq p+q \leq Q,\ 0 \leq p+q+r \leq R,\ P \leq Q \leq R\}
\end{aligned}$$

In order to preserve the positivity of discretised spatial operators, we insist that given the components of $\chi = (\chi_1, \dots, \chi_n)$ the determinant of the Jacobian matrix

$$[J_\chi(\xi)]_{ij} = \frac{\partial \chi_i(\xi)}{\partial \xi_j}, \qquad i, j = 1, \dots, n$$

is positive for all $\xi \in \Omega_{\mathrm{st}}$, so that χ preserves orientation and is invertible. Furthermore we consider an isoparametric representation of χ in terms of a set of shape functions ϕ_{pqr}, so that

$$\chi_i(\xi) = \sum_{(pqr)\in\mathscr{I}} (\hat{\chi}_i)_{pqr}\phi_{pqr}(\xi)$$

In the numerical examples below we consider an expansion in terms of a tensor product of modified hierarchical modal functions which permits a boundary-interior decomposition [7]. We note however that in this setting the choice of shape function is relatively unimportant, so long as they span the polynomial space of the element. However, as we will demonstrate later, this choice of basis is useful for certain types of elemental subdivisions as it permits fewer restrictions on the resulting subelement polynomial spaces.

3.2 Subdivision into the same element type

In this section we demonstrate how the isoparametric mapping χ, which we assume has positive Jacobian for all $\xi \in \Omega_{\mathrm{st}}$, can be used to subdivide an element into smaller elements of the same type. The goal of the subdivision process is to obtain a mapping $\zeta : \Omega_{\mathrm{st}} \to \widetilde{\Omega}$ where $\widetilde{\Omega} \subset \Omega$ and $\det J_\zeta(\xi) > 0$ for all $\xi \in \Omega_{\mathrm{st}}$.

In the isoparametric approach we adopt here, instead of attempting to determine the exact subdomain $\widetilde{\Omega}$ of the physical element Ω, we select a subdomain of the standard region, $\widetilde{\Omega_{\mathrm{st}}}$, and construct an invertible mapping $f : \Omega_{\mathrm{st}} \to \widetilde{\Omega_{\mathrm{st}}}$ with $\det J_f(\xi) > 0$. Initially, we also assume that the polynomial expansion in each direction is equal so that $P = Q = R$. Setting ζ as the composition $\chi \circ f$ we then obtain a subelement $\widetilde{\Omega} = \zeta(\Omega_{\mathrm{st}})$.

The justification for the validity of ζ, and moreover the resulting element $\widetilde{\Omega}$ under the restriction of equal polynomial order is as follows. Firstly, it is clear that the determinant of the Jacobian of ζ is positive for any $\xi \in \Omega_{\mathrm{st}}$, since through an application of the chain rule we have that

$$\det J_\zeta(\xi) = \det J_\chi(f(\xi)) \det J_f(\xi) > 0. \tag{3}$$

Let us assume that each component of χ lies in the polynomial space $\mathscr{P}(\Omega_{\mathrm{st}})$. In order for ζ to retain the isoparametric representation of the subelements, we note in turn that each of its components must be defined in a polynomial space $\mathscr{P}'(\Omega_{\mathrm{st}})$ where in the most general case, $\mathscr{P}(\Omega_{\mathrm{st}}) \subset \mathscr{P}'(\Omega_{\mathrm{st}})$. A consequence of subdivision

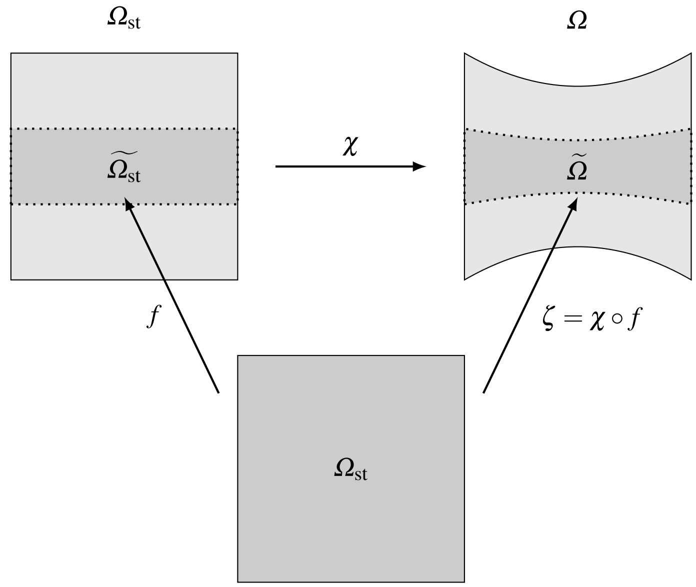


Fig. 10 Construction of the mapping ζ for the subdivision of a quadrilateral element.

therefore is that the subdivided elements may have a higher polynomial order than the parent element depending on the choice of f.

Figure 10 shows a simple application of this subdivision strategy for a quadrilateral element. Here we choose for example an affine mapping $f(\xi_1, \xi_2) = (\xi_1, c\xi_2)$ for some $c \in (0,1)$ so that the standard element is scaled in the ξ_2 direction. Applying the original χ mapping we obtain a new element $\widetilde{\Omega}$ which is appropriately scaled, and naturally introduces curvature into the resulting subelement. In this case, any polynomial term $\xi_1^p \xi_2^q$ is mapped under f to the term $c^q \xi_1^p \xi_2^q$ which clearly lies in $\mathscr{P}(\Omega_{\text{st}}^{\text{quad}})$, and indeed it is clear that by equation 3 that $\det_\zeta(\xi)$ is simply a scalar multiple of $\det_\chi(\xi)$. We may therefore choose $\mathscr{P}'(\Omega_{\text{st}}) = \mathscr{P}(\Omega_{\text{st}})$ and the order of the subelements may be the same as the parent element.

Since the restriction of equal polynomial order is somewhat restrictive, we now consider the case where the polynomial order in each direction is not equal. Whilst a similar argument to the previous explanation can be used in this case, more care must be taken either in the choice of the mapping f or in the order of the resulting subelements to ensure that the polynomial space is correctly spanned. For example, consider a quadrilateral element with expansion orders $P = 2$ and $Q = 1$ which has corresponding polynomial space $\mathscr{P}$, and suppose we choose to produce a trivially subdivided element by applying the transformation $f(\xi_1, \xi_2) = (-\xi_2, \xi_1)$. This map has positive Jacobian determinant and indeed is affine, as in the previous example. However, since ξ_1 and ξ_2 are permuted in the composition with f, the expansion has polynomial terms which lie outside of $\mathscr{P}$ leading to unpredictable element generation.

There are two solutions in this case. Firstly we may choose to obey the general condition $\mathscr{P}(\Omega_{\rm st}) \subset \mathscr{P}'(\Omega_{\rm st})$, and enrich the polynomial order of the subelement so that $P = Q = 2$. Alternatively however, we may permute the polynomial orders of the resulting subelements, so that $P = 1$ and $Q = 2$, to form a space $\mathscr{Q}$. We see in this instance that the resulting subelement is still valid as all of the terms of the original χ expansion are represented in ζ, but the previous condition is not held since $\mathscr{P} \not\subset \mathscr{Q}$. We therefore note that $\mathscr{P}(\Omega_{\rm st}) \subset \mathscr{P}'(\Omega_{\rm st})$ represents a sufficient, but not necessary condition on the validity of subelements in this case.

A similar warning also applies to the other element types, and in particular triangles, prisms and tetrahedra since additional conditions are placed on the summation of mode indices which must be observed. In the next section, we discuss a similar enrichment strategy to permit the subdivision of elements into different element types.

3.3 Subdivision into different element types

Another possible strategy one may adopt when subdividing elements is to consider their division into elements of a different type; for instance, we may subdivide a quadrilateral into triangles in two dimensions, or alternatively hexahedra into prisms or prisms into tetrahedra in three dimensions. Such techniques are well understood for linear finite elements [3] but for curvilinear elements self-intersection may occur, as outlined in the previous section. In this section we demonstrate how the technique introduced in the previous section can be adapted in a more general way than simply prismatic to tetrahedral splitting in order to introduce curvature into the subelements in such a way as to prevent them becoming invalid.

We must adapt the previous argument above since now $f : \Omega'_{\rm st} \to \widetilde{\Omega}$ where $\Omega'_{\rm st} \subsetneq \Omega_{\rm st}$ and so the polynomial spaces which span these standard elements obey the relation $\mathscr{P}(\Omega'_{\rm st}) \subsetneq \mathscr{P}(\Omega_{\rm st})$. In the same way that the technique needs an enrichment of the polynomial space if direction-dependent polynomial orders are used, if we naively apply the method then the polynomial expansion χ can contain terms which are not contained inside $\mathscr{P}(\Omega'_{\rm st})$, and so the resulting mapping ζ may not produce valid elements.

To demonstrate this point, we first examine the problem of figure 11, which depicts an example where a quadrilateral is split along a diagonal edge in order to obtain two triangles. We may again utilise an affine mapping $f(\xi) = -\xi$ in order to map $\Omega^{\rm tri}_{\rm st}$ onto a subdomain $\widetilde{\Omega_{\rm st}}$ of $\Omega^{\rm quad}_{\rm st}$. From our previous argument we see that each component of $\zeta = \chi \circ f$ has degree $2P$ in general if the original quadrilateral is of order P.

Since $\zeta \in [\mathscr{P}(\Omega^{\rm tri}_{\rm st})]^2$ we must select a sufficiently large polynomial order for the triangular space so that all terms of the expansion are represented in the resulting expansion. To guarantee this for a general quadrilateral-to-triangle split, given a

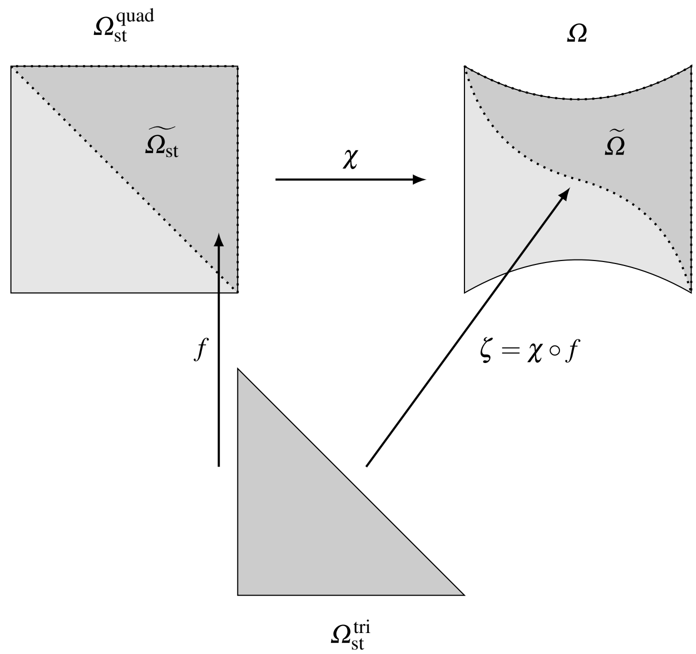


Fig. 11 Construction of the map ζ in the case of a quadrilateral being split into two triangles.

quadrilateral of order P we must generate triangles of order $2P$. Then the space $\mathscr{P}(\Omega_{st}^{quad}) \subset \mathscr{P}(\Omega_{st}^{tri})$ and thus ζ captures all curvature of the original mapping. For a visual illustration of this, we may represent the polynomial spaces of the triangular and quadrilateral elements in the form of a Pascal's triangle as shown in figure 12.

Figure 13 illustrates the problem of using triangular elements which are not sufficiently enriched. On the left, a second-order ($\mathbb{P}^2$) quadrilateral is split into two second-order triangles. Splitting the quadrilateral into two $\mathbb{P}^2$ triangles leads to the generation of degenerate elements. In this case, the symmetry of the deformed element coupled with the quadratic order of the triangles means that the diagonal edge which bisects the quadrilateral is forced to remain straight and thus causes a self-intersection. We note that in this example, the interior quadrilateral mode $\xi_1^2\xi_2^2$ is not energised since curvature is only introduced in one coordinate direction. We additionally note that this can be intuitively achieved by the choice of a boundary-interior hierarchical expansion in which edge and vertex degrees of freedom are decoupled from the interior. Other basis types, such as a nodal Lagrange scheme, will not in general have this property, although the use of the classical Gordon-Hall blending does have this property. Consulting the Pascal triangle of polynomial spaces we therefore see that only a $\mathbb{P}^3$ expansion is required for the triangular elements.

The same logic can be used in the splitting of prismatic and hexahedra elements into tetrahedra. In general an order P prismatic or hexahedral element also requires

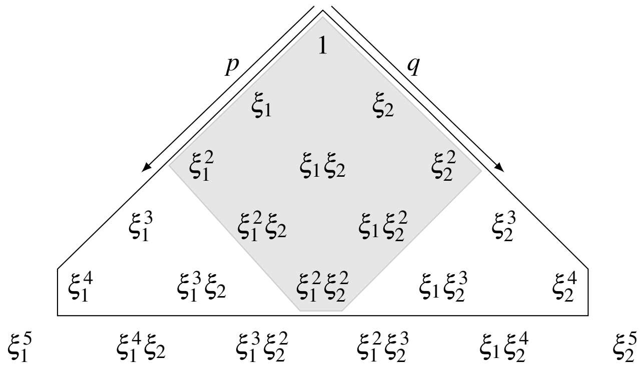


Fig. 12 Pascal's triangle representing the polynomial spaces of $\mathbb{P}^2$ quadrilateral (shaded grey) and $\mathbb{P}^4$ triangular (black outline) elements. The triangle shows that in order to split a general $\mathbb{P}^2$ quadrilateral we require $\mathbb{P}^4$ triangles so that all terms can be represented in the resulting mapping.

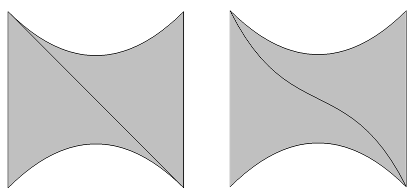


Fig. 13 Qualitative example of the necessary condition for subdivision. A $\mathbb{P}^2$ quadrilateral is split into $\mathbb{P}^2$ (left) and $\mathbb{P}^3$ (right) triangles. Since a $\mathbb{P}^2$ triangular expansion does not capture some of the terms of the original mapping, an additional order is required to produce valid elements.

enrichment so that the resulting tetrahedra have order $2P$ and $3P$ tetrahedra. However by applying the logic above, if curvature is introduced only into the triangular faces of the prisms, then it is only necessary to produce order $P+1$ tetrahedra. Since visualisation of the Pascal's triangle structure is more difficult in three dimensions, this can alternatively be seen from a brief analysis of the prismatic and tetrahedral spaces. If a linear expansion is used in the homogeneous direction of the prismatic element (i.e. $Q=1$) and $P=R$ then the resulting polynomial space is

$$\mathscr{P}^{\mathrm{pri}}(\Omega_{\mathrm{st}}) = \{\xi_1^p \xi_2^q \xi_3^r \mid 0 \le p+r \le P,\ q=0,1\}.$$

A tetrahedron with equal polynomial order P in each direction has the restriction on a triple (pqr) that $0 \le p+q+r \le P$. If $q=1$ then we obtain the restriction $0 \le p+r \le P-1$, and so the tetrahedral space at order P does not contain the prismatic space, leading to possible invalid elements. In order to guarantee validity of elements we therefore require tetrahedra of order $P+1$.

In the following section, we give a demonstration of this prism-to-tetrahedron splitting and also highlight the application of the refinement method in boundary-layer problems.

4 Applications

This section demonstrates the usefulness of the subdivision method by showing how it can be used to generate three-dimensional meshes for challenging applications. Firstly we consider the subdivision of a coarse prismatic boundary-layer mesh into a series of progressively thinner elements as the distance to the wall decreases. We then show how the prismatic elements can be subdivided to obtain a boundary-layer mesh comprising only tetrahedra for use by solvers supporting only simplicial elements. Finally we produce boundary layer meshes for the ONERA M6 wing in order to demonstrate the applicability of the technique in aeronautical problems. For each case, we consider a high-order coarse discretisation at polynomial orders of $P \geq 10$ to show the viability of the method even at very high polynomial orders.

4.1 Boundary layer mesh generation

In order to generate a sequence of n subelements which gradually become more slender towards the surface of the domain, we define a spacing distribution Δ_k for $1 \leq k \leq n$ following the procedure described in Section 2.6. Under the framework of section 3.1 then, we define a straightforward affine scaling function similar to that used in figure 10 which obeys the necessary conditions in order to generate valid subelements. We additionally note that as long as the same spacing distribution is used for all prismatic elements, the resulting mesh is conformal. One of the major advantages of this method for the generation of boundary layer meshes is that the resulting subelements are guaranteed to be valid, as shown in section 3.1, and thus we are able to produce boundary layers of arbitrary thickness. Figure 14 shows how the subdivision technique can be used to generate a boundary layer mesh for a simple geometry: a cylindrical duct.

Certain solvers only have support for meshes which are composed only of simplicial elements. For problems where boundary layers are required, this poses an additional problem for mesh generation software. In figure 14, we show how the same method can be used to split the prismatic elements of figure 14 into three tetrahedra. Firstly we note that in order for the resulting mesh to be conformal, we must employ a strategy so that the quadrilateral faces which connect prismatic elements are split in a consistent fashion, such as the one outlined by Dompierre et al. [3].

Once this strategy is applied, we may utilise the subdivision strategy to split the standard prismatic element into three tetrahedra by using an affine transformation

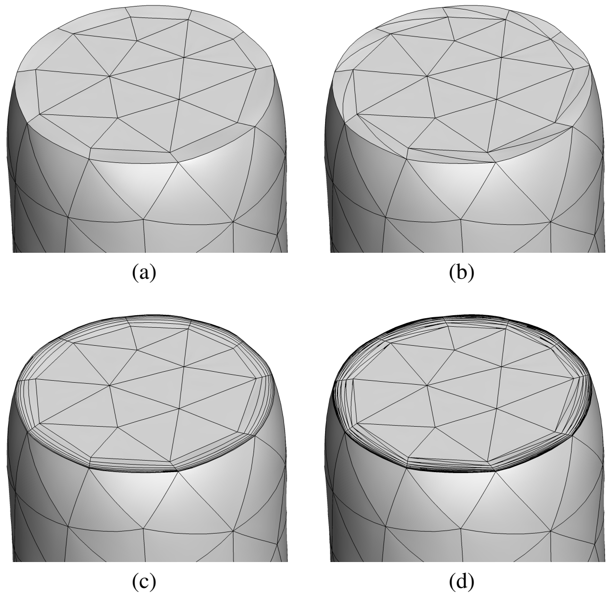


Fig. 14 Duct geometry: (a) Initial coarse mixed mesh with prisms in the boundary layer; (b) prism to tetrahedron split of the original mesh; (c) boundary layer refinement; and (d) we apply the prism to tetrahedron splitting after the boundary layer refinement has been performed.

similar to that used in figure 11. We note that in the specific case of figure 14, since the curvature of the original prisms is only imposed on the triangular surface, we may obtain valid tetrahedra by enriching the polynomial space by one order.

An important point to note is that whilst the validity of the resulting tetrahedra is guaranteed through our previous arguments, this method may lead to the production of tetrahedra which have suboptimal quality in terms of interior angles, depending on the curvature of the original prismatic elements. However, when tetrahedral boundary layers are required this is often unavoidable since the elements exhibit a large stretching ratio. In the very worst cases, the use of these meshes as a starting point for a mesh deformation procedure may lead to better quality elements. We suggest that the validity of the meshes produced here may lead to improved convergence speeds in such methods.

4.2 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 [18]. Here we use this case as an illustration of the methodology for generating high-order meshes for high Reynolds

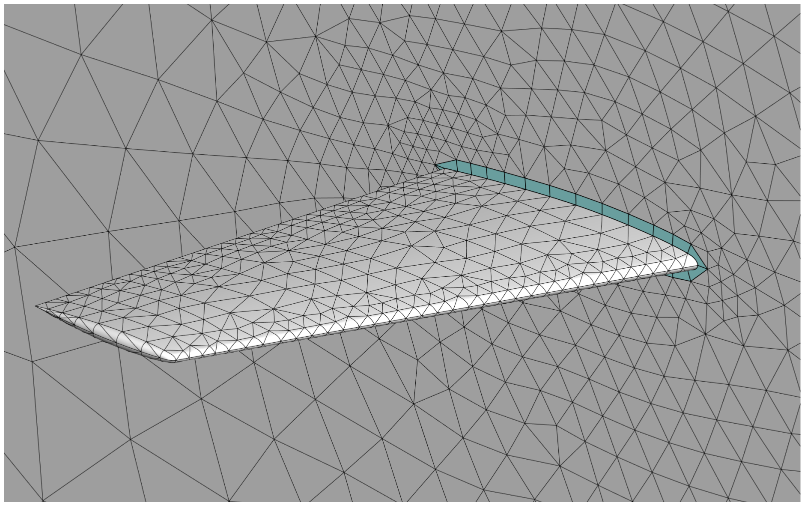


Fig. 15 Boundary of the initial coarse straight-sided mesh with prisms in the boundary layer (highlighted) and tetrahedra in the interior.

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.

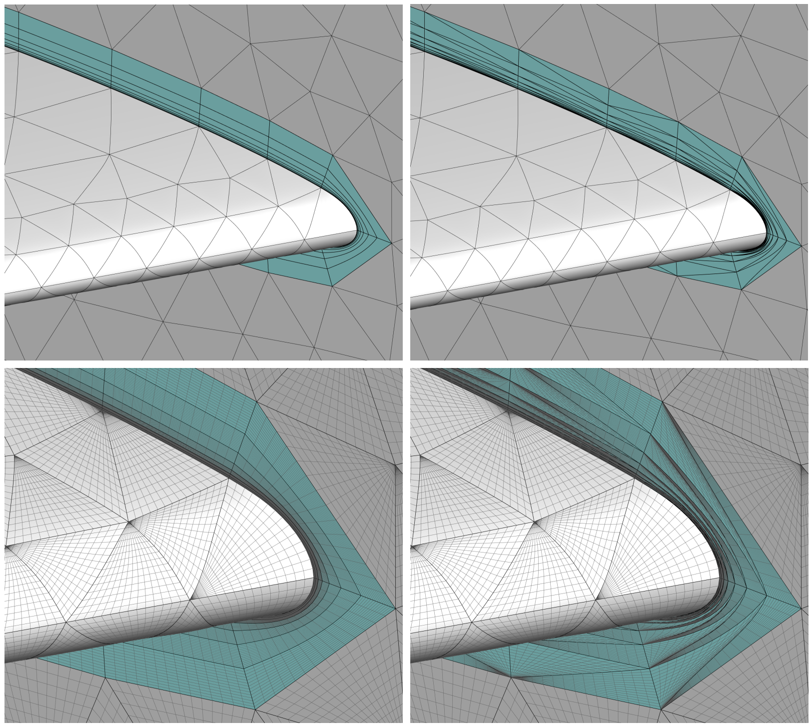
Figure 15 shows the initial coarse high-order surface mesh of the wing geometry coupled with a symmetry plane. The intersection of the symmetry plane with the boundary layer of the wing, composed of the prismatic macro-elements, is highlighted, with the remainder of the mesh comprising tetrahedral elements. In figure 16 we apply the splitting strategies from section 2 to obtain a mesh with $y^+ \approx 5$ at $\mathrm{Re} = 10^5$, where the skin friction C_f is approximated using the Schlichting formula [17]

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

Figure 16(a) shows the hybrid prismatic-tetrahedral mesh that is obtained after the splitting algorithm is applied. In figure 16(b) we apply the prism to tetrahedron splitting technique of section 2.5 to obtain a fully tetrahedral mesh. In all cases a valid mesh is generated, with no elements having negative Jacobian.

5 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. Starting from a valid coarse prismatic boundary-layer mesh,



(a) (b)

Fig. 16 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.

our isoparametric approach permits the generation of a sequence of meshes with increased resolution with very little additional cost. This should prove very 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.

We have derived the mathematical conditions that are necessary for the subdivision of high-order isoparametric elements, and show how this technique can be applied to tackle challenges in high-order mesh generation. We posit that the simplicity of the method outlined here will prove to be a valuable tool in improving both the efficiency and robustness of curvilinear mesh generation software, and particularly for the generation of meshes for high Reynolds number computational fluid dynamics problems or high Peclet number advection-diffusion problems.

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. Extending the method to other cases is not a difficult technical issue, but it will require the use of transition elements, such as pyramids. However, such implementation is beyond the scope of the work within the IDIHOM project.

References

1. Carmo, M.D.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)
2. Dey, S., O'Bara, R., Shephard, M.: Curvilinear mesh generation in 3D. In: Proceedings of the 8th International Meshing Roundtable (1999). South Lake Tahoe, California
3. Dompierre, J., Labbé, P., Vallet, M.G., Camarero, R.: How to subdivide pyramids, prisms and hexahedra into tetrahedra. In: 8th International Meshing Roundtable. Lake Tahoe, California (1999)
4. Gargallo-Peiró, A., Roca, X., Sarrate, J., Peraire, J.: Inserting curved boundary layers for viscous flow simulation with high-order tetrahedra. In: 22nd International Meshing Roundtable (2013)
5. Geuzaine, C., Remacle, J.F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering **79**(11), 1309–1331 (2009)
6. Gordon, W., Hall, C.: Construction of curvilinear co-ordinate systems and applications to mesh generation. International Journal for Numerical Methods in Engineering **7**(4), 461–477 (1973)
7. Karniadakis, G., Sherwin, S.: Spectral/*hp* Element Methods for Computational Fluid Dynamics. Oxford University Press (2005). Second edition
8. Mavriplis, D.: Unstructured grid techniques. Annual Review of Fluid Mechanics **29**, 473–514 (1997)
9. Moxey, D., Hazan, M., Peiró, J., Sherwin, S.J.: On the generation of curvilinear meshes through subdivision of isoparametric elements (2014). To appear in proceedings of Tetrahedron IV
10. Moxey, D., Hazan, M., Sherwin, S.J., Peiró, J.: An isoparametric approach to high-order curvilinear boundary-layer meshing (2014). Under review in Comp. Meth. App. Mech. Eng.
11. Peiró, J., Sayma, A.: A 3-D unstructured multigrid Navier-Stokes solver. In: K. Morton, M. Baines (eds.) Numerical Methods for Fluid Dynamics V. Oxford University Press (1995)
12. Peraire, J., Morgan, K.: Unstructured mesh generation including directional refinement for aerodynamic flow simulation. Finite Elements in Analysis and Design **25**, 343–356 (1997)
13. Peraire, J., Peiró, J., Morgan, K.: Multigrid solution of the 3-D compressible Euler equations on unstructured tetrahedral grids. International Journal for Numerical Methods in Engineering **36**, 1029–1044 (1993)
14. Persson, P.O., Peraire, J.: 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
15. Pirzadeh, S.: Three-dimensional unstructured viscous grids by the advancing-layers method. AIAA Journal **34**(1), 43–49 (1996)
16. Sahni, O., Luo, X.J., Jansen, K.E., Shepard, M.S.: Curved boundary layer meshing for adaptive viscous flow simulations. Finite Elements in Analysis and Design **46**(1-2), 132–139 (2010)
17. Schlichting, H.: Boundary-layer theory, Seventh edn. McGraw-Hill (1979)
18. Schmitt, V., Charpin, F.: 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
19. Sherwin, S., Peiró, J.: Mesh generation in curvilinear domains using high-order elements. International Journal for Numerical Methods in Engineering **53**(1), 207–223 (2002)
20. Toulorge, T., Geuzaine, C., Remacle, J.F., Lambrechts, J.: Robust untangling of curvilinear meshes. Journal of Computational Physics **254**, 8–26 (2013)
21. Xie, Z., Sevilla, R., Hassan, O., Morgan, K.: The generation of arbitrary order curved meshes for 3D finite element analysis. Computational Mechanics **51**(3), 361–374 (2013)
22. Zienkiewicz, O., Phillips, D.: An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co-ordinates. International Journal for Numerical Methods in Engineering **3**(4), 519–528 (1971)