

Hybrid Grid Generation for Viscous Flow Simulations in Complex Geometries

Hongfei Ye

Zhejiang University

Yang Liu

China Aerodynamics Research and Development Center

Bo Chen

China Aerodynamics Research and Development Center

Zhiwei Liu

Zhejiang University

Jianjing Zheng

Zhejiang University

Yufei Pang

China Aerodynamics Research and Development Center

Jianjun Chen (✉ chenjj@zju.edu.cn)

Zhejiang University <https://orcid.org/0000-0002-0318-0997>

Research

Keywords: mesh generation, hybrid mesh, mesh quality, viscous flow

Posted Date: May 29th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-31698/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published on July 30th, 2020. See the published version at <https://doi.org/10.1186/s42774-020-00042-x>.

Hybrid Grid Generation for Viscous Flow Simulations in Complex Geometries

Hongfei Ye^{a,b}, Yang Liu^c, Bo Chen^{c*}, Zhiwei Liu^{a,b}, Jianjing Zheng^{a,b},
Yufei Pang^c, Jianjun Chen^{a,b,d†}

^a Center for Engineering and Scientific Computation, Zhejiang University, Hangzhou 310027, China

^b School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China

^c China Aerodynamics Research and Development Center, Mianyang, 621000, China.

^d State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, China

Abstract: In this paper, we present a hybrid grid generation approach for viscous flow simulations by marching a surface triangulation on viscous walls along certain directions. Focuses are on the computing strategies used to determine the marching directions and distances since these strategies determine the quality of the resulting elements and the reliability of the meshing procedure to a large extent. With respect to marching direction, three strategies featured with different levels of efficiencies and robustness performance are combined to compute the initial normals at front nodes to balance the trade-off between efficiency and robustness. A novel weighted strategy is used in the normal smoothing scheme, which evidently reduce the possibility of early stop of front generation at complex corners. With respect to marching distances, the distance settings at concave and/or convex corners are locally adjusted to smooth the front shape at first; a further adjustment is then conducted for front nodes in the neighbourhood of gaps between opposite viscous boundaries. These efforts, plus other special treatments such as multi-normal generation and fast detection of local/global intersection, as a whole enable the setup of a hybrid mesher that could generate qualified viscous grids for geometries with industry-level complexities.

Key words: mesh generation; hybrid mesh; mesh quality; viscous flow

1 Introduction

For RANS computations involving complex geometries, a challenging task is the generation of high-quality RANS meshes. Among the different mesh types, prismatic hybrid meshes are preferred in many applications because they represent a good compromise between solution accuracy and ease of use [1-4]. In a prismatic hybrid mesh, the near field of viscous walls (referred to as a *boundary layer* hereafter) is configured with layered prismatic elements to resolve high flow gradients normal to the walls, whereas the remaining domain is usually filled with an unstructured mesh. Thus, the generation of a hybrid prismatic mesh usually consists of two individual meshing steps: boundary layer mesh generation and unstructured mesh generation.

* Corresponding author. E-mail: chenbo01010401@163.com

† Corresponding author. E-mail: chenjj@zju.edu.cn

36 In general, the generation of boundary layer elements starts from the surface triangulation on
37 viscous walls. The initial front is defined on this triangulation, and a marching direction is
38 computed at each mesh point of the front. Each front point is then propagated to a new position
39 by adding a step value along the marching directions. As a result, a layer of prismatic elements
40 can be formed by connecting all the front points with their new neighbours. The entire boundary
41 layer mesh could then be created by repeating the above procedure a few times [2-7]. With
42 respect to the generation of unstructured meshes, either the advancing front technique (AFT)
43 based approach [8] or the Delaunay triangulation (DT) based approach [9-11] could be adopted.
44 In some studies, it has been suggested to first fill an axis-aligned Cartesian mesh in the far field
45 of viscous walls and then connect the boundary layer mesh and the Cartesian mesh with a few
46 transition layers of unstructured elements [12-15].

47 Compared with the now mature unstructured mesh generation and Cartesian mesh generation,
48 the generation of boundary layer meshes still gives rise to numerous difficulties and is therefore
49 the main challenge in generating the entire prismatic hybrid mesh. Among those difficulties,
50 those induced by the computations of marching directions and marching distances should be
51 highlighted because both computations determine the quality of the resulting elements and the
52 reliability of the meshing procedure to a large extent. A large portion of the efforts involved in
53 the development of a prismatic hybrid mesher were invested in tackling these issues [17-23].

54 In principle, a practically useful hybrid mesh generation scheme should take the quality of
55 the resulting elements and the reliability of the meshing procedure as the primary consideration.
56 Following this principle, we proposed several novel computing strategies for marching
57 directions and marching distances. With respect to the computation of marching directions,
58 three strategies featured with different levels of efficiencies and robustness performance are
59 combined to compute the initial normals at front nodes to balance the trade-off between
60 efficiency and robustness. After that, a revised smoothing scheme is proposed for these normals
61 to avoid the abrupt changes on lengths of neighbouring front lines. As a result, this smoothing
62 could evidently reduce the possibility of early stop of front generation at complex corners. With
63 respect to the computation of marching distances, the initial marching distances are computed
64 by the user-specified parameters, followed by a two-step adjustment: the distance settings at
65 concave and/or convex corners are locally adjusted to smooth the front shape at first; a further
66 adjustment is then conducted for front nodes in the neighbourhood of gaps between opposite
67 viscous boundaries. To support the second-step adjustment, an improved ray-casting algorithm
68 is developed for the automatic identification of the gaps. In the meantime, the cost of this
69 computation is reduced to a very low level at the aid of a background mesh.

70 In addition to the above novel strategies, other special treatments are developed to improve
71 the robustness and efficiency of the hybrid meshing procedure, multi-normal generation, fast
72 detection of intersections between front faces, remove of non-manifold fronts, to name a few.
73 These efforts, as a whole, enable the setup of a hybrid mesher that could generate qualified
74 viscous grids for geometries with industry-level complexities. Numerical experiments are
75 conducted including comparison with results by state-of-the-art commercial tools to verify the
76 effectiveness and efficiency of the mesher.

77 The remainder of this paper is organized as follows. Section 2 reviews the existing computing
78 strategies of marching directions and distances briefly. Section 3 presents an outline of the
79 hybrid meshing method. Section 4 introduces important implementation details involved in the

80 hybrid meshing method. Section 5 presents various numerical experiments. Section 6 concludes
81 with the outcomes of the study and points to some directions for future studies.

82 **2 Literature review**

83 **2.1 On computation of marching directions**

84 Presently, the most prevailing approaches for computing marching directions are those based
85 on the analysis of the *manifold* of a point [16-21]. The manifold of a point here refers to the set
86 of front faces adjacent to the point, and these front faces are thus named *manifold faces* of that
87 point. Intuitively, the marching direction at a point could be obtained by computing a weighted
88 average of the normal vectors of its front faces. However, this intuitive computation strategy
89 cannot ensure the resulting marching direction is always *visible* to all the manifold faces. As a
90 remedy, Kallinderis and Ward [16] presented the *visibility cone* concept, which refers to a subset
91 of the space depicted by the manifold of the point. To ensure the mesh validity, the computed
92 marching direction must be located within the visibility cone. Based on the visibility cone
93 concept, Aubry and Löhner [17, 18] recast the problem of computing a marching direction into
94 an optimization problem. The solution of that problem could result in the ‘best’ marching
95 direction at a point by providing an optimal angle property for the next layer of elements that
96 meet at that direction. Nevertheless, it was reported that if the marching direction at each point
97 of a front face was computed in a locally optimal fashion [20, 21], it still might not be optimal
98 for the prism carried by the face. Therefore, some kind of global smoothing must be performed
99 after the initial computation of the marching directions. To improve the effect, a front node
100 classification procedure is required before the execution of such smoothing techniques such
101 that the marching directions defined at different types of front nodes could be treated differently.
102 It is worth noting that the results of this node classification procedure are very sensitive to the
103 user-specified angle thresholds [20, 21]. There also exist some other approaches for computing
104 marching directions. For example, in the *face offsetting* method proposed by Jiao [22] faces are
105 directly propagated along their normals and the vertices are then reconstructed through an
106 eigenvalue analysis locally, and good resulting boundary layer meshes are presented [23] for
107 several biomedical models based on this method.

108 Recently, a few new techniques that rely on the solution of a partial differential equation
109 (PDE) have been investigated for boundary layer mesh generation [1, 14, 24-27]. Accordingly,
110 the computation of marching directions is defined in the solution space of the adopted governing
111 equation rather than in the geometric space. For example, the marching direction at a point
112 could be defined as the gradient vector of the solution at that point [25]. At present, a frequently
113 adopted governing equation is the Eikonal equation, and the adopted numerical schemes for the
114 solution include the fast-marching method [25, 28], fast sweeping method [29], the finite
115 element method and the finite difference method. To harness these numerical schemes, an
116 additional volume background mesh is always created [24-27]. Recently, the Laplacian
117 equation has been chosen as the governing PDE but changed from the scalar function to a vector
118 one [1]. Mathematically speaking, the solution of this vector form Laplacian equation could
119 smoothly propagate the marching vectors defined at initial fronts into the domain interior.
120 Meanwhile, the authors suggest the boundary element method (BEM) adopted as the numerical
121 scheme of the governing equation since two advantages of the BEM over other numerical
122 schemes. Firstly, the result of BEM is computed by boundary integration equations rather than

123 by interpolations; therefore, the computed directions are more accurate. Secondly, the BEM
124 only needs a surface mesh input rather than a volume counterpart.

125 In comparison with conventional approaches based on local geometric computations, the
126 PDE based approaches provide a new global angle to view the front propagation problem.
127 However, the present PDE-based approaches have some common issues as well. For example,
128 these approaches are usually much slower. Moreover, it remains a challenging issue on how to
129 combine these approaches with multi-normal generation schemes.

130 **2.2 On computation of marching distances**

131 The default marching distance at a front node could be computed by user-specified parameters.
132 This default value leads to viscous elements with same lateral edge lengths at each layer. For
133 viscous walls having complex corners and/or small gaps, local adjustment is necessary to avoid
134 intersections of front lines and generation of low-quality elements. Most existing algorithms
135 support increase marching distances slightly at concave corners and vice versa at convex
136 corners in order to smear concave and convex corners and facilitate the marching process. The
137 computation of local curvatures or angle values, based on either the discrete manifold or the
138 original CAD model, is commonly used to determine local marching distances [1, 3, 21].
139 Nevertheless, for some PDE-based approaches [1], the solution of PDEs could support a correct
140 adjustment of local marching distance as above. In this case, no local geometric computation is
141 needed any more.

142 In the neighbourhood of small gaps, reducing marching distances appropriately is an option
143 to avoid global intersection of viscous elements propagated from opposite viscous walls. Here,
144 the main issue is efficient computation of gap distances. Normally, an extra data structure (e.g.
145 quadtree in two-dimensional and octree in three dimensional) is required. In [21], an approach
146 relying on constrained DT is suggested.

147 The local adjustment of marching distances may lead to an abrupt change of marching
148 distances at neighbouring front nodes. If this issue happens, Laplacian-type smoothing
149 strategies are usually suggested to resolve it.

150 The more challenging issue is to adapt the mesh to flow solutions or boundary movements.
151 Since this issue is not involved in this study, the discussion is beyond the scope of this paper.
152 Interested readers are referred to [30, 31].

153 **3 Outline of the hybrid meshing method**

154 Figure 1 presents the main steps included in our hybrid meshing method. Given a valid CAD
155 model, the proposed method mainly takes the following steps to output a hybrid mesh.

156 **Step 1.** Apply the approach proposed in [32] to compute a sizing function for surface
157 mesh generation and define boundary conditions on surface patches of the model.

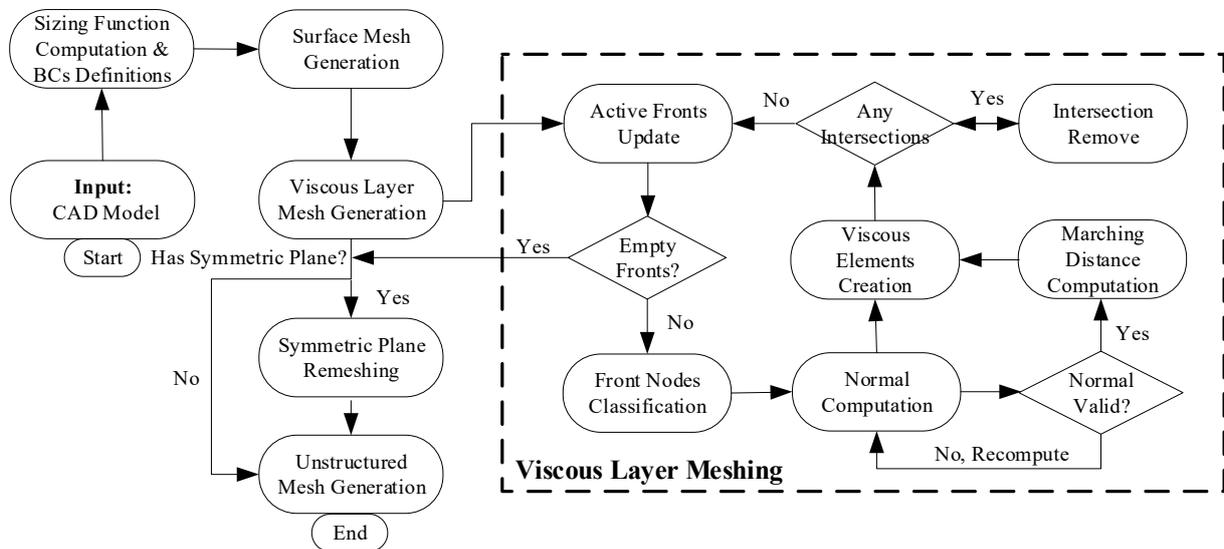
158 **Step 2.** Given the input model and the sizing function, create a surface triangulation by
159 an in-house advancing front mesher [33].

160 **Step 3.** The viscous layer meshing step needs three user parameters that indicate the
161 height of the first layer h_0 , the expansion ratio of neighbouring layers μ and the
162 allowed maximum number of layers n_l , respectively. According to these parameters,
163 we can compute the marching distance at each front node. In addition, the marching

164 direction at each front node can be computed by analyzing the manifold of the node.
 165 Once the marching directions and marching distances are determined at all the front
 166 nodes, a layer of prismatic elements can then be created by connecting the front nodes
 167 and their duals after propagating the front. Repeating this front propagation procedure
 168 for at most n_i times, we can then create semi-structured prismatic elements in the
 169 vicinity of the viscous walls.

170 **Step 4.** If a symmetry plane is defined on the domain boundary, layered quadrilateral
 171 elements should be created in the vicinity of the common curves of the symmetry plane
 172 and viscous walls after Step 3. Therefore, the surface mesh of the symmetry plane, which
 173 is initially composed of triangular elements only, need be updated to accommodate these
 174 quadrilateral elements.

175 **Step 5.** We can then collect the surface triangles that depict the remaining unmeshed
 176 volume region. These triangles include those located at the boundaries with the non-
 177 viscous wall types and those depicting the outmost boundary of the boundary layer
 178 elements. With these surface triangles as the input, we finally employ an in-house
 179 mesher to fill the unstructured tetrahedra in the domain enclosed by the input surface
 180 triangles [10, 11]. A feature of the employed mesher is its robust capability to create a
 181 boundary constrained tetrahedral mesh. This feature is a key for the success of this step,
 182 where a point-to-point conformity is required between the unstructured tetrahedra and
 183 boundary layer elements.



184
 185 **Figure 1.** Flowchart of hybrid mesh generation.
 186 The above discussion only sketches the main steps in our method. Nevertheless, to be concise,
 187 this discussion does not include a few non-trivial techniques incorporated in our method. These
 188 techniques are necessary to improve our method for application to real problems. In Sections 4
 189 to 6, we will discuss the important technical details involved in the three steps, respectively,
 190 with a particular focus on Step 3.

191 **4 Boundary layer mesh generation**

192 4.1 Outline of the method

193 The right part of Figure 1 presents the workflow of our boundary layer mesh generation method.
194 The inputs include the surface triangulation of the domain boundary and some user parameters
195 (such as h_0 , μ and n_l). A list of front faces L_f and a list of front nodes L_n are maintained
196 during the entire workflow. Accordingly, flags are attached to the active front nodes and faces
197 to distinguish them from others.

198 Initially, L_f and L_n are filled in with those input surface elements and surface nodes
199 located on the viscous walls, respectively. After that, four steps are consecutively followed to
200 create the boundary layer mesh: (1) computing marching directions, (2) computing marching
201 distances, (3) creating a layer of elements and (4) updating L_f and L_n . To ensure the
202 reliability of the algorithm and the validity and usability of the output mesh, the intermediate
203 outputs of the former three steps are checked carefully.

204 In the following subsections, we will present the algorithmic details of the four main steps.

205 4.2 Computing marching directions

206 The computation of marching directions is based on the classification of front nodes. A front
207 node is labelled as *flat* if the maximal angle is smaller than 5 degrees between any two normals
208 of the faces connected to the node. For those unlabelled nodes, a further classification is
209 conducted by computing the average of angles between neighbour face normals. Here, the
210 average angle is denoted by β , and approximately, front nodes are classified as *concave* or
211 *convex* ones by their β values.

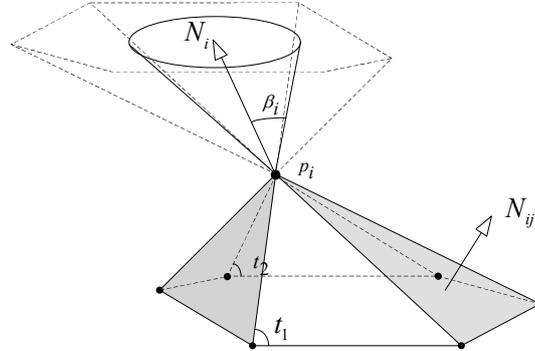
212 For flat nodes, its marching direction is computed by a simple average of all neighbouring
213 normals. For other nodes, three strategies are combined to set up a cost-effective scheme for
214 marching direction computation:

215 **Strategy I.** Compute the normals of faces connected to a given front node and classify
216 them into groups such that the number of groups is as small as possible under the
217 condition that the maximal angle is smaller than 25 degrees between any two normals
218 belonging to the same group. For each group, a *representative normal* is computed by
219 averaging all normals belonging to the group. The normal at the front node is exactly
220 the average of all representative normals.

221 **Strategy II.** Compute the marching vector lying on the bisection plane of the two faces
222 on the manifold forming the wedge with the smallest angle. The location of the marching
223 vector on that plane is evaluated by bisecting the visibility region on that plane [6]. As
224 is shown in Figure 2, the visibility region is represented by a polyhedral cone extending
225 outward from the point, and it can be simplified into a visibility cone with the circular
226 cross section and half-cone angle, which can also be called the visibility angle β_i .

227 **Strategy III.** This is an iterative algorithm aimed at finding the ‘most normal’ normal,
228 i.e., the normal that minimizes the maximal angle with the given set of normals [17].
229 Weights are given to each face normal depending on the angle created with the current
230 normal. If the angle is high, more weight is given to the normal. See [17] for a pseudo
231 code of this implementation.

232 To balance the trade-off between efficiency and robustness, the above strategies are
 233 conducted in the order listed above. The quality of the normal at the front node is evaluated by
 234 the maximal angle between the normal and normals of manifold faces. A hill-climbing scheme
 235 is used to ensure the optimal normal is kept always. Meanwhile, the next level of strategy gets
 236 no opportunity in order to save computing time when the quality of the present ‘optimal’ normal
 237 is less than 30 degrees.



238
 239 **Figure 2.** The visibility region and its subset (a cone region) at a front node

240 After computing the initial marching directions, a further smooth is executed to ensure a
 241 desirable variation across the front and facilitate the following marching process. Here, the
 242 smooth is performed by a weighted Laplacian approach, i.e.,

243
$$\mathbf{N}_i^n = (1 - \omega)\mathbf{N}_i^{n-1} + \frac{\omega}{\sum_j w_{ij}} \sum_j (w_{ij})\mathbf{N}_j^{n-1},$$

244 where \mathbf{N}_i^{n-1} and \mathbf{N}_i^n are normals at front node p_i after n and $n-1$ iterations, \mathbf{N}_j^{n-1} is the
 245 normal at neighbouring front node p_j after $n-1$ iterations, and w_{ij} is the weight defined at p_j .
 246 Note that it is beneficial to let normals at convex corners be closer to their neighbours, and
 247 vice versa for concave corners. To achieve this, w_{ij} is defined as below,

248
$$w_{ij} = \begin{cases} k_{ij}^2 & p_i \text{ is a convex point} \\ 1/k_{ij}^2 & p_i \text{ is a concave point,} \\ 1 & \text{otherwise} \end{cases}$$

249 where

250
$$k_{ij} = n \frac{d_{ij}}{\sum_{j=1,n} d_{ij}}$$

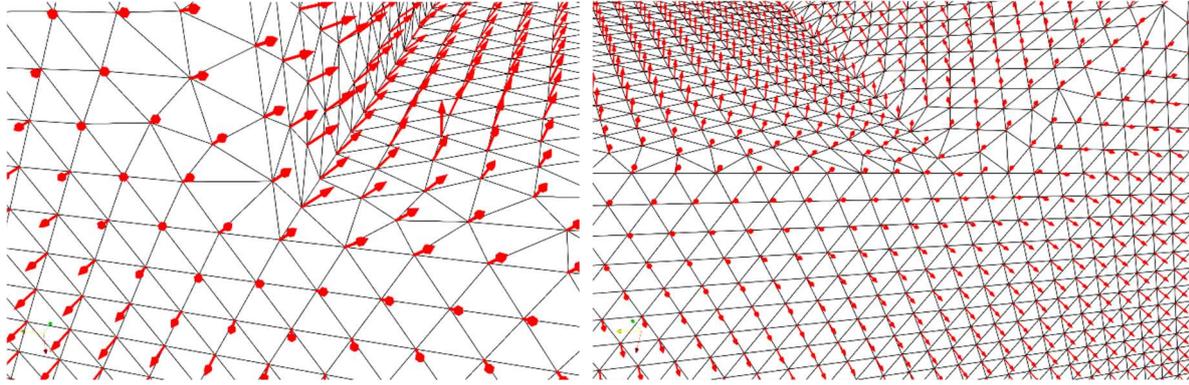
251 and d_{ij} are the distance between p_i and p_j .

252 There are cases where one single normal could not ensure the validity of visibility cone. A
 253 multi-normal strategy is adopted in these cases. Interested readers are referred to [34] for
 254 more details.

255 To ensure the validity of the computed marching directions, for each marching direction,
 256 we check whether it is visible to all the front faces adjacent to the front nodes. If no valid
 257 marching direction can be defined at a front node, we stop propagating the front node and
 258 clear the ‘front’ flag attached to that node.

259 Figure 3 presents the computed marching directions in the neighbourhood of two typical

260 corners. It can be seen that the computed directions are reasonable and no abrupt changes
 261 occur between neighbouring marching directions.



262 (a)

262 (b)

263 **Figure 3.** Marching directions in the neighbourhood of two different types of corners.

264 (a) A concave corner. (b) A convex corner.

266 4.3 Computing marching distances

267 Suppose p_i is a front node, \mathbf{u}_i is the marching direction at p_i , and m is the present layer
 268 number. The marching distance at p_i can be computed by

$$269 \quad h_i = \mu^{m-1} h_0.$$

270 If simply applying the above Equation in all front nodes, the computed marching sizes would
 271 be the same. As a result, the front will conduct a so-called *advective motion* at expansion. As
 272 illustrated in Figure 4, *wave-frontal motion* may be more desirable than advective motion
 273 because it would smooth out convex or concave corners, whereas the advective motion would
 274 preserve them. To implement the effect of a wave-frontal motion, a coefficient α can be added
 275 such that the distances at the convex corners need to be shortened and the marching distances
 276 at the concave corners need to be lengthened.

$$277 \quad h_i = \alpha \mu^{m-1} h_0,$$

278 where

$$279 \quad \alpha = \begin{cases} 1 + |\cos \beta_i| & \beta < 180^\circ \\ 1 - |\cos \beta_i| & \beta \geq 180^\circ \end{cases}.$$

280 Another geometric factor that impacts the computation of marching distances is the gap
 281 between opposite viscous boundaries. Given a front node p_i and a direction starting from p_i to
 282 domain interiors, a minimal distance between p_i with viscous walls is computed first, denoted
 283 by d_i^G . Denote the total height of viscous boundary at p_i by

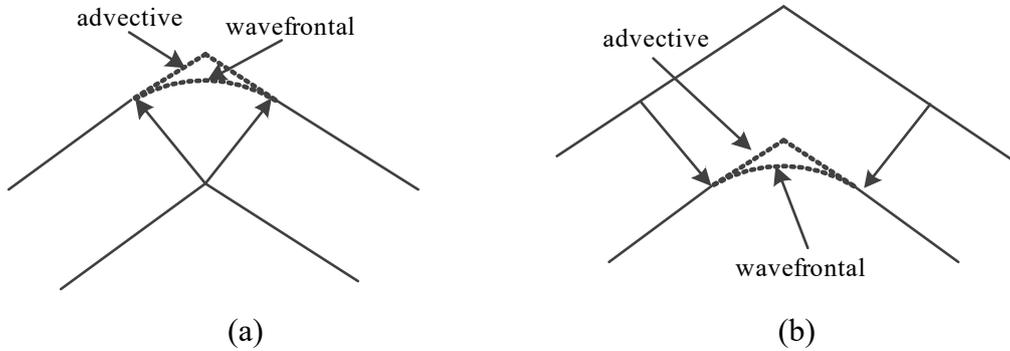
$$284 \quad l_i = \sum_{m=1, n_i} \alpha \mu^{m-1} h_0.$$

285 If $d_i^G > 2l_i$, no further modification is required on the marching distance at p_i ; otherwise, the
 286 height of the first viscous layer at p_i is computed by

287

$$h_{i,0} = \frac{\varepsilon_1 \times d_i^G}{\sum_{m=1, n_i} \mu^{m-1}},$$

288 where $\varepsilon_1 \in [0, 0.5]$ is a user parameter that determines how large the space is left for
 289 unstructured elements (less is more).



290

291

292 **Figure 4.** Advection vs. wave-frontal propagation at expansion at a (a) convex corner and (b)
 293 concave corner.

294 Apparently, the computation of d_i^G should be speeded up at the aid of a spatial data
 295 structure. Here, we reuse the octree grid for speeding up front intersection checks to compute
 296 d_i^G . Algorithm 1 presents the setup procedure of this octree grid, and Algorithm 2 presents the
 297 computation of d_i^G based on the grid.

298 Finally, a smoothing procedure is executed to avoid the abrupt changes of neighbouring
 299 marching directions. A simple Laplacian smoothing is presently employed.

Algorithm 1 Setup procedure of this octree grid

Input: Octree root node ocT , Maximum depth d_{max} , Maximum number of triangles in one cube
 T_{max} , Boundary triangle Mesh topology

Output: Octree ocT

- 1: **for** triangles T_i in boundary mesh topology **do**
 - 2: Find all the leaf node $\{l_{node}\}$ in ocT with T_i may intersect it by recursive
 - 3: **for** node n in $\{l_{node}\}$ **do**
 - 4: **if** current depth $d_n < d_{max}$ and the number of triangles in n $T_n \geq T_{max}$ **then**
 - 5: Split n into 8 cube
 - 6: move the triangles in n to the subnode of n
 - 7: **else**
 - 8: Insert T_i to n
 - 9: **end if**
 - 10: **end for**
 - 11: **end for**
-

300

Algorithm 2 Computation of gap distances

Input: Octree ocT , Node normal N_i , Boundary triangle Mesh topology, Node Coordinate C_i , Minimum distance D_{min} ;

Output: Distance field D_i

```
1: Insert all the boundary triangle mesh to  $ocT$  by Algorithm 1.
2: Initialize the distance field  $D_i = 1$ 
3: for node  $n_i$  in boundary mesh do
4:   Caculate the average neighbor front size  $S_i$  .
5:   Caculate the expected length  $H_i = \frac{S_i \mu - h_0}{\mu - 1}$ 
6:   for Node normal  $N_j$  in neighbor node normal of  $n_i$  do
7:     Get the expected edge  $P_s = C_i$  and  $P_e = P_s + H_i N_j$ 
8:     Insert the edge  $(P_s, P_e)$  to  $ocT$  and check intersection
9:     if no inserection between edge and triangles then
10:       $D = 1$ 
11:     else
12:      Find the first intersection  $P_{in}$ 
13:       $D = \max(D_{min}, \frac{P_{in} - P_s}{P_e - P_s})$ 
14:     end if
15:      $D_i = \min(D, D_i)$ 
16:   end for
17: end for
```

301

302 **4.4 Creating a layer of elements**

303 For each front node qualified for propagation, we can compute its dual by marching it along the
304 marching direction. After that, we can create a layer of prismatic elements by connecting all the
305 front nodes qualified for propagation and their duals.

306 Low-quality elements may be created in this step, in particular in the vicinity of concave
307 corners. In this study, we selected *scaled aspect ratio* [23] to evaluate the quality of a prism.
308 This quality measure in effect combines the measures of triangle shapes and edge orthogonality.
309 For a given prism τ , denote the scaled aspect ratio of this element by $\rho(\tau)$ ($\rho(\tau) \in [-1, 1]$). $\rho=1$
310 indicates an ideal prism, and $\rho < 0$ indicates an inverted element.

311 After creating a layer of elements, we pick those elements whose quality values are below
312 0.1 for removal. Meanwhile, we stop propagating the front faces that carry those elements.

313 **4.5 Updating the front**

314 The mesh nodes on viscous walls are regarded as the initial front nodes. Correspondingly, those
315 fronts composed of the initial front nodes are regarded as the initial propagating fronts. As the
316 propagation of nodes continues layer by layer, the propagating fronts are updated according to
317 the propagating behaviour of their forming nodes. In this study, three stopping criteria are
318 applied to each front node:

319 (1) The current layer number of the node is equal to the prescribed maximum number of
320 boundary layers, i.e., $i_l = n_l$;

321 (2) At each node of a newly generated element e , the scaled aspect ratio (denoted by $\rho(p)$
322 hereafter) [23] is computed, and the propagation will stop when $\rho(p) < 0$;

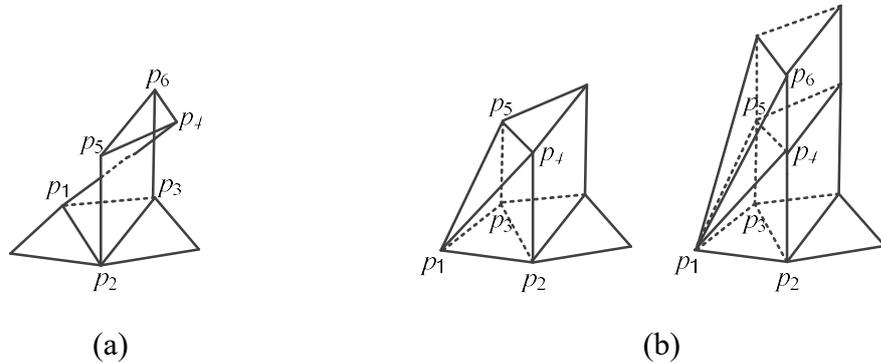
323 (3) The new front faces starting from the front node are involved in global intersections;

324 (4) If one of the neighbouring nodes of the current node in the previous layer is set to stop
325 propagating, then the propagation of that node is stopped.

326 Criterion 1 is straightforward and ensures the termination of the boundary layer mesh
327 generation procedure. Criterion 2 avoids the generation of elements with negative signed

328 volumes. Figure 5a presents a node p_1 with $\rho(p_1) < 0$, which leads to an inverted element.
 329 Criterion 3 is used to avoid global intersection. Criterion 4 is only executed for 3D cases and
 330 ensures the difference in the layer numbers of the neighbouring nodes does not exceed one. In
 331 2D problems, the exposed segments after boundary layer mesh generation are sent to a
 332 triangular mesh generator, and the difference in the layer numbers of the neighbouring nodes
 333 will not affect the resulting mesh quality. However, in 3D problems, the exposed faces include
 334 triangles and quadrilaterals, and some transition elements should be added to hide the
 335 quadrilateral faces before the exposed faces are sent to a tetrahedra mesh generator. If the
 336 difference in the layer numbers of the neighbouring nodes is allowed to be larger than one,
 337 stretched pyramids will be added as the transition elements. Figure 5b presents the added
 338 transition elements for two different cases.

339 Let F be one of the current propagating fronts and p_i ($i=0, 1, 2$) be the forming nodes of that
 340 front. If p_i ($i=0, 1, 2$) are all propagated to new positions, i.e., p'_i ($i=0, 1, 2$), then F is propagated
 341 to F' , with p'_i as its forming nodes. In addition, F' will replace F as a new front in the next layer.
 342 However, if at least one node of p_i ($i=0, 1, 2$) is not allowed to propagate to the next layer, F
 343 will also be allowed to propagate to the next layer.



344
 345 (a) (b)
 346 **Figure 5.** (a) An invalid element with $\rho(p_1) < 0$. (b) Transition elements in two different cases.

347 5 Unstructured mesh generation

348 If a symmetry plane is defined on the domain boundary, layered quadrilateral elements should
 349 have been created in the vicinity of the common curves of the symmetry plane and viscous
 350 walls after boundary layer mesh generation. To accommodate those quadrilateral elements, we
 351 first remove the original surface mesh of the symmetry plane, then obtain the boundary
 352 description of the unmeshed region of the symmetry plane, and finally mesh this region using
 353 an advancing front surface mesher [34].

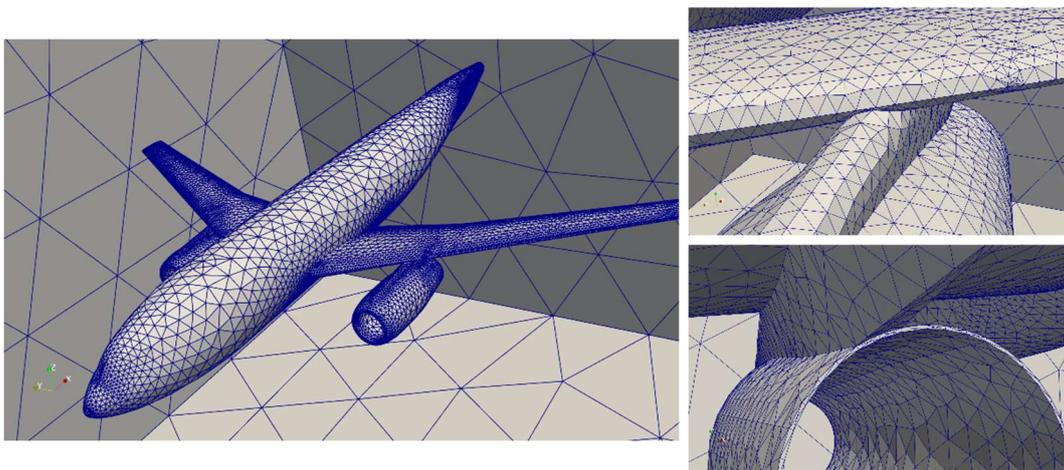
354 We next employ an in-house DT mesher to fill the unstructured tetrahedra in the domain
 355 enclosed by the input surface. Note that the DT criterion provides a reasonable method to link
 356 a given point set; however, it cannot ensure the existence of boundary constraints in the resulting
 357 tetrahedralisation. A boundary recovery procedure is thus required to ensure the boundary
 358 integrity of the resulting mesh. For the hybrid meshing problem focused on in this study, one
 359 part of the surface input to the DT mesher is composed of the exposed faces of the boundary
 360 layer elements. Some of those faces could be rather stretched and thus lead to a more
 361 challenging task for the boundary recovery procedure. By incorporating a few novel techniques
 362 for boundary recovery [10, 11], a feature of our in-house mesher is its capability to robustly
 363 create a boundary constrained tetrahedral mesh. This feature is a key for the success of the

364 unstructured mesh generation step because point-to-point conformity is required between the
365 unstructured tetrahedra and boundary layer elements.

366 6 Numerical experiments

367 6.1 DLR F6 aircraft model

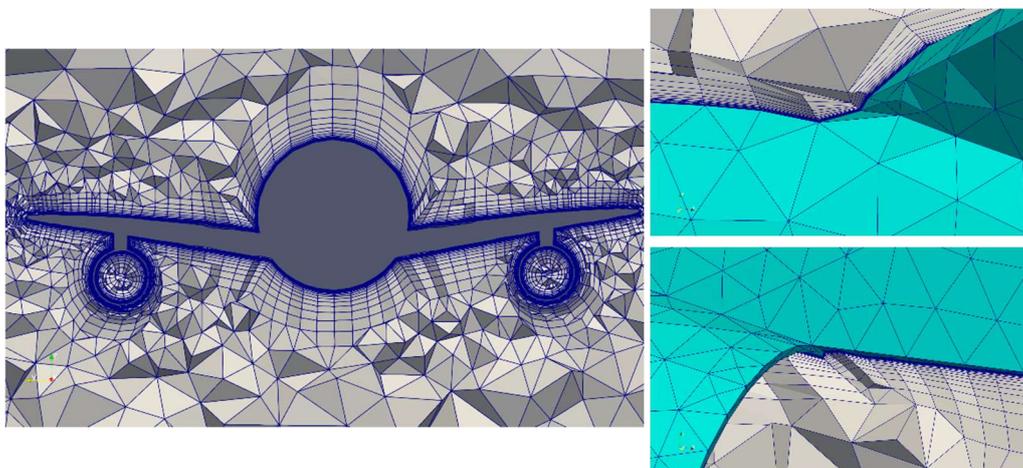
368 The surface mesh of the F6 model is presented in Figure 6a; it contains 14,866 nodes and 29,732
369 triangle elements. Many complex concave regions and corner nodes are involved in this model,
370 e.g., a complex corner node at the tail of the engine and several concave regions near the joint
371 of the engine and wing (see close-up views in Figure 6a). In the process of boundary layer mesh
372 generation, the surface mesh of the F6 model was set as the viscous wall boundary condition.
373 Figure 10b presents a cut view of the hybrid mesh for exterior flow simulations, in which
374 689,281 prisms, 160,452 tetrahedron and 9,441 pyramids are consisted. A cut-out view of the
375 boundary layer mesh and local mesh details around two complex corners are presented in Figure
376 10b. As can be seen, a valid normal vector was obtained by the proposed method in both corners.
377 If using the average normal vectors of the neighbouring surfaces, it was difficult to determine
378 suitable normals here because the normal vectors of several faces around both corners are nearly
379 in opposite directions.



380

381

(a)

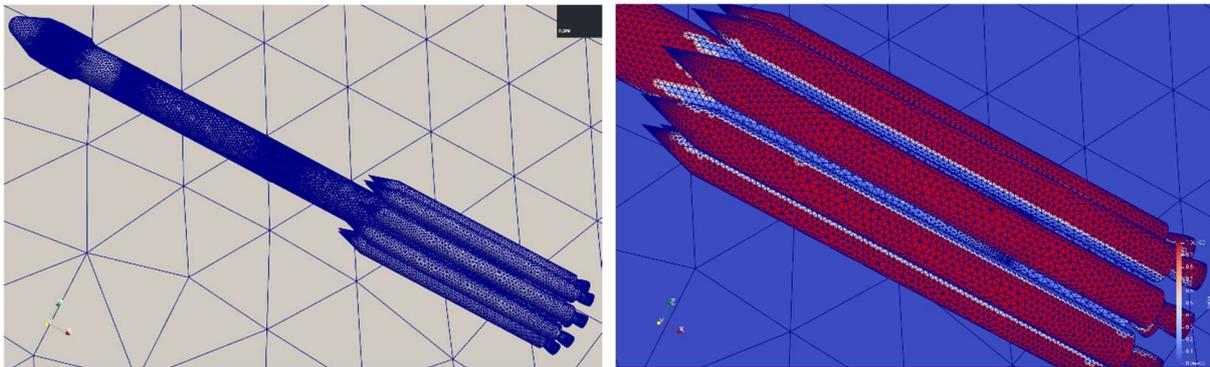


382

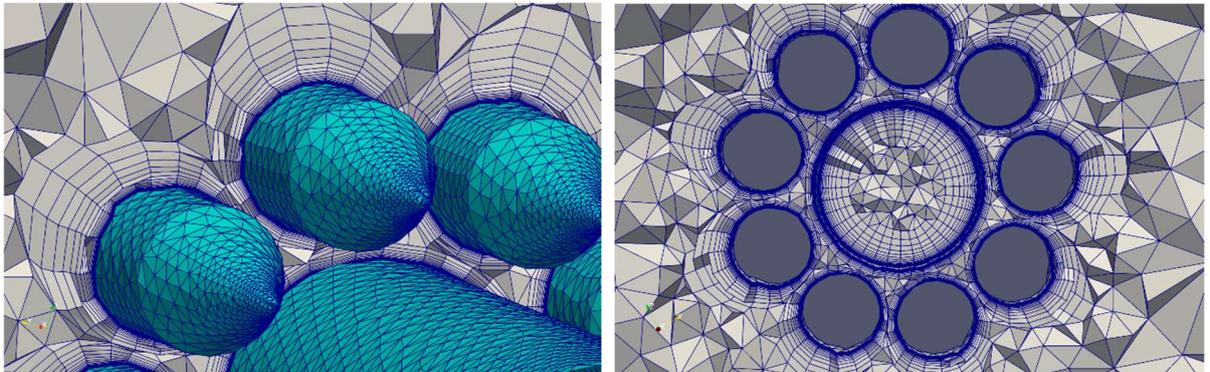
383 **Figure 6.** The hybrid mesh of the F6 model: (a) The surface mesh; (b) Cut-out views of the
384 hybrid mesh.

385 **6.2 Rocket model**

386 To demonstrate that the proposed method could avoid global intersections by identifying small
387 gaps and reducing marching distances locally, a rocket model is selected in which a few volume
388 proximities exist between the main body and 9 boosters. This model contains 445 surface
389 patches, on which 73,759 surface nodes and 147,470 triangles are generated. The resulting
390 hybrid mesh contained 3,891,188 prisms, 520,245 tetrahedrons and 42,862 pyramids. Figure 7
391 presents the surface mesh and the close-up views for the color map of gap distances, in which
392 regions with small gap distances are rendered in blue. As can be seen, the volume proximities
393 between the main body and boosters are all correctly identified. Figure 8 presents
394 two cut-out views of the hybrid mesh. Narrow gap can be clearly observed and the global
395 intersection there was effectively avoided with the proposed method.



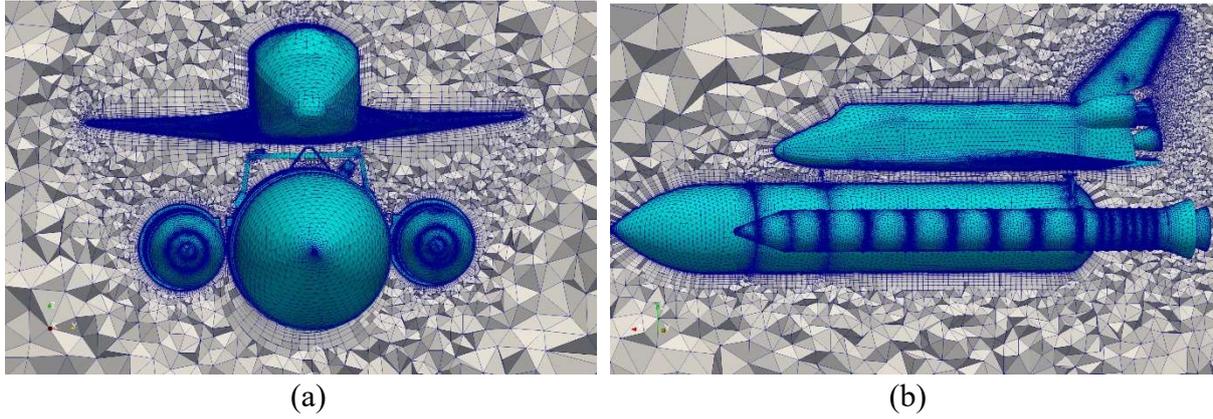
396
397 **Figure 7.** The hybrid mesh of a rocket model: (a) The surface mesh; (b) Close-up views of the
398 color map of gap distances (d_i^G).



399
400 **Figure 8.** Cut-out views of the hybrid mesh of the rocket model

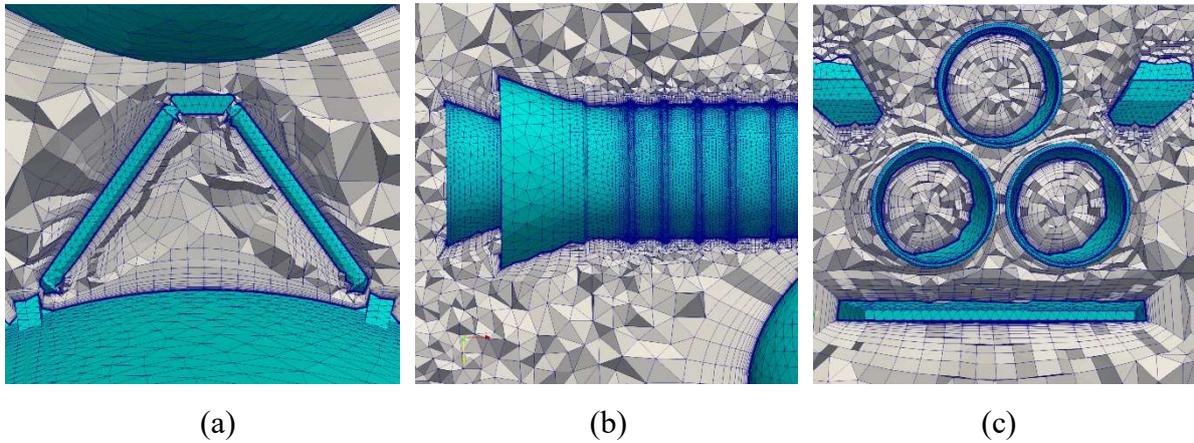
401 **6.3 Space shuttle model**

402 To demonstrate the proposed method for configurations with industry-level complexity, we
403 chose to generate the hybrid mesh of a space shuttle model. This model contains 595 surface
404 patches, on which 170,933 surface nodes and 341,846 triangles are generated. Note that this
405 model contains abundant geometric details near the joints of different parts, the volume
406 proximities between the main bodies of boosters and fuel tanks, etc. The resulting hybrid mesh
407 contained 6,176,455 prisms, 2,111,679 tetrahedrons and 114,465 pyramids. Figure 9 presents
408 two cut-out views of the hybrid mesh. Figure 10 presents close-up views of three local details
409 of the hybrid mesh.



410
411
412
413

Figure 9. Cut-out views of the hybrid mesh of the space shuttle model: (a) Front view; (b) Side view.



414
415
416
417
418

Figure 10. Local details of boundary layer meshes of the space shuttle aircraft. (a) Boundary layer meshes around the support between the plane and rocket; (b) Boundary layer meshes around the top view of rocket; (c) Boundary layer meshes around space shuttle tail.

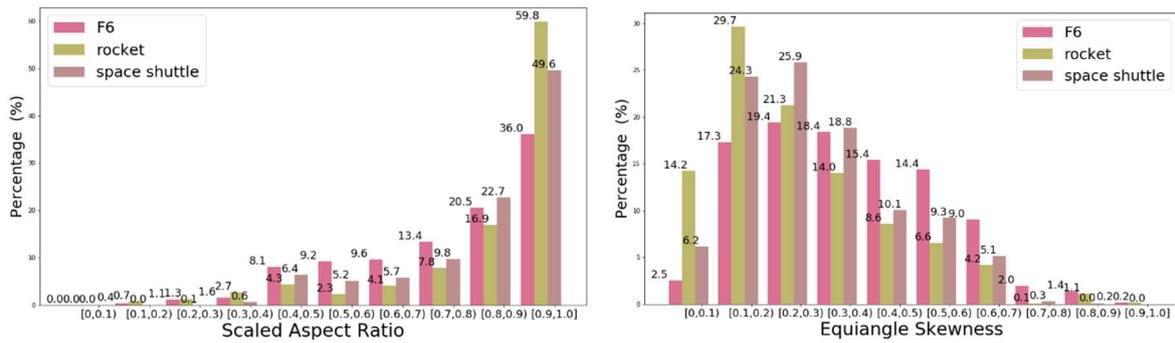
419 **6.4 Mesh quality**

420 Quality of prismatic elements is our major concern. to evaluate the quality of the generated
421 prismatic elements, the *scaled aspect ratio* quality measure was first adopted in this study. In
422 this study, inverted elements are not allowed, and we refer to elements with $\rho(\tau) < 0.2$ as low-
423 quality elements. The distributions of scaled aspect ratios of prismatic elements for the F6,
424 rocket and space shuttle models are presented in Figure 11. The ratio of low-quality elements
425 accounts for 0.4%, 0.7%, 0.03% of the total numbers of prism elements for the three models,
426 respectively.

427 The equiangle skewness is another commonly used quality measure for various types of
428 elements. For a prism, it is represented as the maximum ratio of the element faces' included
429 angles to the angles of equilateral faces. Its value varies between 0 (good) and 1 (bad). It is
430 recommended this skewness measure be kept below 0.8 for a good grid; values below 0.9 are
431 acceptable, depending on the solver. Therefore, we refer to elements with skewness values
432 larger than 0.9 as low-quality elements. Under this new standard, the ratio of low-quality
433 elements accounts for 0.17%, 0.18%, 0.026% of the total numbers of prism elements for the
434 three models, respectively.

435 Overall, the above data reveals that the boundary layer elements created by the proposed

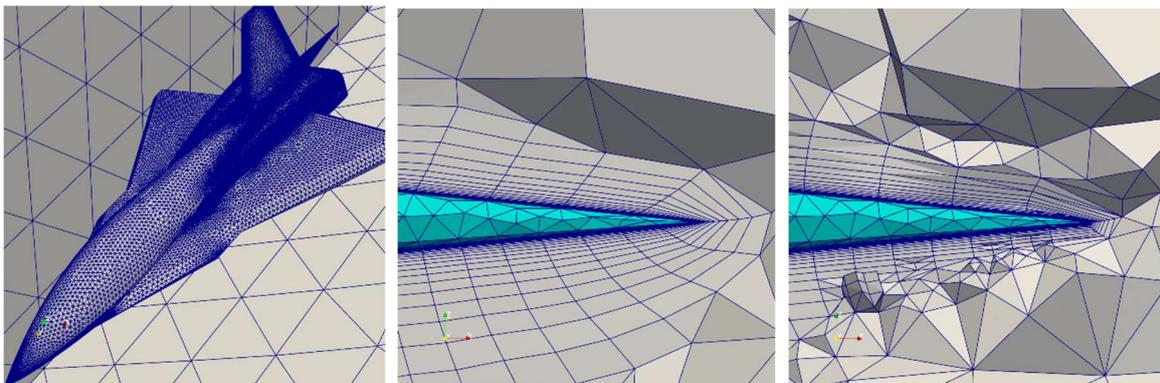
436 method have acceptable shape and quality.



437
438 (a) (b)
439 **Figure 11.** Quality histogram of prisms. (a) Scaled aspect ratio; (b) Equiangle skewness.

440 **6.5 Comparison with commercial tools**

441 In most tests we conducted, the proposed method achieved the similar level of reliability and
442 element quality as commercial mesh tools, such as Pointwise. However, it was also observed in
443 some complex corners, the proposed method creates boundary layer elements with more
444 desirable quality. Figure 12a presented the surface input used for comparison. With the same
445 surface input and same user settings, we create two hybrid meshes by using the proposed
446 method and Pointwise, respectively. Figures 12b and 12c enlarge the details of two meshers
447 near the tail of the aircraft, in which a convex corner with a very small angle exist. Although
448 the multi-normal technique was employed by Pointwise, the resulting boundary mesh by
449 Pointwise stop its propagation much earlier than its counterpart by our method.



450
451 (a) (b) (c)
452 **Figure 12.** Comparison with Pointwise. (a) Surface input. (b) Mesh by our method. (c) Mesh
453 by Pointwise.

454 **7 Concluding remarks**

455 A prismatic hybrid mesh configured with layered prismatic elements in the near field of viscous
456 walls and an unstructured mesh in the rest of the domain is preferred in many applications
457 because it represents a good compromise between solution accuracy and ease of use. Several
458 novel computing strategies for marching directions and marching distances are implemented
459 by taking the quality of the resulting elements and the reliability of the meshing procedure as
460 the primary consideration. These efforts enable the setup of a hybrid mesher that could generate

461 qualified viscous grids for geometries with industry-level complexities. Numerical experiments
462 including academic cases, benchmark cases and cases from industry-level simulations are
463 presented to verify its effectiveness and efficiency.

464 **Declarations**

465 **Acknowledgements**

466 This research is funded by the National Numerical Wind Tunnel Project of China (Grant No.
467 NNW2018-ZT5B10).

468 **Authors' contribution**

469 The research output is coming from joint effort. All authors read and approved the final
470 manuscript.

471 **Funding**

472 National Numerical Wind Tunnel Project of China (Grant No. NNW2018-ZT5B10).

473 **Availability of data and materials**

474 The datasets used and/or analysed during the current study are available from the corresponding
475 author upon reasonable requests.

476 **Competing interests**

477 The authors declare that they have no competing interests.

478 **Author detail**

479 See the first page of the manuscript.

480 **References**

- 481 1. Zheng Y, Xiao Z, Chen J, Zhang J. Novel Methodology for Viscous-Layer Meshing by the Boundary-Element Method.
482 AIAA Journal 2018; 56: 209-221.
- 483 2. Ito Y, Murayama M, Yamamoto K, et al. Efficient hybrid surface and volume mesh generation for viscous flow simulations.
484 AIAA paper, 2011, 3539: 2011.
- 485 3. Sharov D, Nakahashi K. Hybrid prismatic/tetrahedral grid generation for viscous flow applications. AIAA journal, 1998,
486 36(2): 157-162.
- 487 4. Hassan O, Morgan K, Probert E J, et al. Unstructured tetrahedral mesh generation for three-dimensional viscous flows.
488 International Journal for Numerical Methods in Engineering, 1996, 39(4): 549-567.
- 489 5. Marcum DL. Efficient generation of high-quality unstructured surface and volume grids. Engineering with Computers,
490 2001, 17(3): 211-233.
- 491 6. Pirzadeh S. Three-dimensional unstructured viscous grids by the advancing-layers method. AIAA journal, 1996, 34(1):
492 43-49.
- 493 7. Garimella R V, Shephard M S. Boundary layer mesh generation for viscous flow simulations. International Journal for
494 Numerical Methods in Engineering, 2000, 49(1): 193-218.
- 495 8. Löhner R, Parikh P. Generation of three-dimensional unstructured grids by the advancing front method. International
496 Journal for Numerical Methods in Fluids, 1988, 8(10): 1135-1149.
- 497 9. Bowyer A. Computing dirichlet tessellations. The Computer Journal, 1981, 24(2): 162-166.
- 498 10. Chen J, Zhao D, Huang Z, Zheng Y, Gao S. Three-Dimensional Constrained Boundary Recovery with an Enhanced Steiner
499 Point Suppression Procedure. Computers and Structures 2011; 89: 455-466.
- 500 11. Chen J, Zheng J, Zheng Y, Si H, Hassan O, Morgan K. Improved Boundary Constrained Tetrahedral Mesh Generation by
501 Shell Transformation. Applied Mathematical Modelling 2017; 51: 764-790.
- 502 12. Zheng Y, Liou MS. A Novel Approach of Three-Dimensional Hybrid Grid Methodology: Part 1. Grid Generation.
503 Computer Methods in Applied Mechanics and Engineering 2003; 192:4147-4171.
- 504 13. Liou MS, Zheng Y. A Novel Approach of Three-Dimensional Hybrid Grid Methodology: Part 2. Flow Solution. Computer
505 Methods in Applied Mechanics and Engineering 2003; 192: 4173-4193.

- 506 14. Park S, Jeong B, Lee J G, et al. Hybrid grid generation for viscous flow analysis. *International Journal for Numerical*
507 *Methods in Fluids*, 2013, 71(7): 891-909.
- 508 15. Zhang L, Zhao Z, Chang X, He X. A 3D Hybrid Grid Generation Technique and a Multigrid/Parallel Algorithm Based on
509 Anisotropic Agglomeration approach. *Chinese Journal of Aeronautics* 2013; 26:47-62.
- 510 16. Kallinderis Y, Ward S. Prismatic grid generation for three-dimensional complex geometries. *AIAA Journal*, 1993, 31(10):
511 1850-1856.
- 512 17. Aubry R, Löhner R. On the ‘most normal’normal. *Communications in Numerical Methods in Engineering*, 2008, 24(12):
513 1641-1652.
- 514 18. Aubry R, Mestreau E L, Dey S, et al. On the ‘most normal’normal—Part 2. *Finite Elements in Analysis and Design*, 2015,
515 97: 54-63.
- 516 19. Loseille A, Löhner R. Robust Boundary Layer Mesh Generation. In: 21st International Meshing Roundtable, Orlando,
517 Florida, USA; 2013. p.457-474.
- 518 20. Ito Y, Shih A M, Soni B K, et al. Multiple marching direction approach to generate high quality hybrid meshes. *AIAA*
519 *Journal*, 2007, 45(1): 162-167.
- 520 21. Wang F, di Mare L. Hybrid Meshing Using Constrained Delaunay Triangulation for Viscous Flow Simulations.
521 *International Journal for Numerical Methods in Engineering* 2016; 108:1667-1685.
- 522 22. Jiao X. Face offsetting: A unified approach for explicit moving interfaces. *Journal of computational physics*, 2007, 220(2):
523 612-625.
- 524 23. Dyedov V, Einstein D R, Jiao X, et al. Variational generation of prismatic boundary layer meshes for biomedical
525 computing. *International Journal for Numerical Methods in Engineering*, 2009, 79(8): 907-945.
- 526 24. Wang Y. Eikonal equation based front propagation technique and its applications. *AIAA Paper*, 2009, 375.
- 527 25. Wang Y, Guibault F, Camarero R. Eikonal equation based front propagation for arbitrary complex configurations.
528 *International Journal for Numerical Methods in Engineering*, 2008, 73(2): 226-247.
- 529 26. Xia H, Tucker P G, Dawes W N. Level sets for CFD in aerospace engineering. *Progress in Aerospace Sciences*, 2010,
530 46(7): 274-283.
- 531 27. Dawes W N, Harvey S A, Fellows S, et al. Viscous layer meshes from level sets on cartesian meshes[C]//45th AIAA
532 *Aerospace Sciences Meeting & Exhibit*. 2007: 8-11.
- 533 28. Sethian JA. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid*
534 *Mechanics, Computer Vision, and Materials Science*. Cambridge University Press: Cambridge, 1999.
- 535 29. Zhao HK. A fast sweeping method for Eikonal equations. *Mathematics of Computation* 2005; 74(250):603–627.
- 536 30. Zheng J, Chen J, Zheng Y, et al. An improved local remeshing algorithm for moving boundary problems. *Engineering*
537 *Applications of Computational Fluid Mechanics*, 2016, 10(1): 405-428.
- 538 31. Hassan O, Sørensen KA, Morgan, K, Weatherill NP. A method for time accurate turbulent compressible fluid flow
539 simulation with moving boundary components employing local remeshing. *International Journal for Numerical Methods*
540 *in Fluids*, 2007, 53(8), 1243-1266.
- 541 32. Chen J, Xiao Z, Zheng Y, et al. Automatic Sizing Functions for Unstructured Surface Mesh Generation. *International Zhao*
542 *D, Chen J, Zheng Y, Huang Z, Zheng Z. Fine-Grained Parallel Algorithm for Unstructured Surface Mesh Generation.*
543 *Computers & Structures* 2015; 154:177-191.
- 544 33. *Journal for Numerical Methods in Engineering*, 2017; 109:577-608.
- 545 34. Aubry R, Löhner R. Generation of viscous grids at ridges and corners[J]. *International Journal for Numerical Methods in*
546 *Engineering*, 2009, 77(9): 1247.