INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:79–108 (DOI: 10.1002/fld.669)

Improvements in the reliability and quality of unstructured hybrid mesh generation

Yasushi Ito*,†,‡ and Kazuhiro Nakahashi§

Department of Aeronautics and Space Engineering, Tohoku University, Aoba-yama 01, Sendai 980-8579, Japan

SUMMARY

This paper presents a reliable and automated approach to the generation of unstructured hybrid grids comprised of tetrahedra, prisms and pyramids for high Reynolds number viscous flow simulations. To enhance robustness, the hybrid mesh generation process starts with the formation of an isotropic tetrahedral grid. Prismatic layers are then added on no-slip walls fully automatically by obeying user-specified boundary conditions and three parameters: the number of the layers, an initial layer thickness normal to the walls, and a stretching factor. Topological modifications to the original isotropic tetrahedral elements are prohibited during the layer generation process. The tetrahedral elements near no-slip walls are shifted inward and the resulting gap between the tetrahedra and the walls is filled up with prismatic elements. To enhance the quality of the prismatic layers around sharp corners, two normals are evaluated for the marching process in these regions. The addition of prismatic elements is locally stopped if negative-volume elements are created or not enough space is left. An angle-based smoothing method ensures that the quality of the tetrahedral elements is retained for a reasonable computational cost. The method is demonstrated for two scaled experimental supersonic airplane models designed at the National Aerospace Laboratory of Japan (NAL). Numerical results are compared with wind tunnel experimental data. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: unstructured hybrid mesh; CFD; viscous flow simulations

1. INTRODUCTION

Computational fluid dynamics (CFD) has shown a prodigious growth over the last few decades. The generation of unstructured isotropic tetrahedral grids has been successfully demonstrated as a sound basis for solving inviscid and low Reynolds number flows, even around complex geometries. For high Reynolds number flows, on the other hand, such unstructured grids still have issues that need to be addressed. In order to resolve thin boundary

*Correspondence to: Y. Ito, Department of Mechanical Engineering, University of Alabama at Birmingham, 1530 3rd Avenue South, BEC 352, Birmingham, AL 35294-4461, U.S.A.
†E-mail: yito@uab.edu
‡Currently, Postdoctoral Fellow, Department of Mechanical Engineering, University of Alabama at Birmingham, 1530 3rd Avenue South, BEC 352, Birmingham, AL 35294-4461, U.S.A.
§Professor.

Received 29 March 2003
Revised 10 October 2003

layers on the solid surface accurately, the grid spacing normal to no-slip walls has to be very small. A solution is to create high aspect ratio elements near the surface. It may, however, cause a stiffness problem for the flow solver. Moreover, the generation of such anisotropic stretched grids near the wall is a demanding issue.

To treat 3D high Reynolds number flows using the unstructured grid approach, several methods have been proposed for generating the highly stretched grid near to the walls [1–13]. Some of them generate fully tetrahedral or prismatic grids and the others create hybrid grids consisting of a few types of elements such as prisms and tetrahedra. Although these methods have shown their capabilities in simulating viscous flows around 3D configurations, their reliability still needs to be improved in order to treat the sorts of configurations encountered in engineering applications [14].

Hybrid grid generation methods have been developed because they exhibit the advantages of both structured and unstructured grid generation approaches. To capture boundary layers near no-slip walls, prismatic layers are adopted, which realize good orthogonality and clustering capabilities. Tetrahedral elements are used to fill the rest of the computational domain. This strategy allows a single mesh to cover a domain even for complex geometries. The flexibility of the approaches enables automatic mesh generation to be achieved. Moreover, compared with an all-tetrahedral mesh of the same resolution, the prismatic section of a hybrid mesh reduces the memory requirements and runtimes for a flow solver.

Most hybrid grid generation methods start with the prismatic grid generation near the solid surface, and the rest of the domain is subsequently discretized by tetrahedral elements. This conventional approach may treat most of 3D complex geometries. There are, however, difficulties in deciding a marching direction for the prismatic layers, in generating grids in small gaps, in avoiding negative-volume elements and in guaranteeing the quality of the generated mesh. Because of these difficulties, reliability and quality issues for hybrid mesh generation still remain.

In contrast, the isotropic tetrahedral grid generation technique is very mature. Therefore, for viscous grid generation, it is better to utilize the tetrahedral grid as well. Currie [8] proposed an approach in which isotropic tetrahedral grids were modified for viscous flow simulations. From an initially isotropic tetrahedral mesh, tetrahedra on no-slip walls are subdivided into anisotropic ones in order to obtain the user-specified number of layers of high aspect ratio tetrahedra near the walls. The advantage of this approach is that it is easier to treat sharp corners and to check grid intersections or grid shapes. The approach may be utilized in a hybrid grid generation method. For complex geometries, however, the control of the heights of layers becomes difficult because the tetrahedra on walls have various sizes depending on the local mesh density of the surface mesh. In addition, collapsing small tetrahedra, to reduce the total number of elements in the generation of each layer, is time-consuming.

Löhner and Cebral [11] proposed a mesh generation method for high Reynolds number flows based on an isotropic tetrahedral mesh. First, all nodes where stretched tetrahedra are required are removed. Second, new nodes are then added to generate highly stretched tetrahedra. It is an efficient procedure for controlling the heights of tetrahedra on no-slip walls. However, the method sometimes cannot add stretched elements and a re-meshing process is required. Because of this limitation, the method may not be very reliable.

In the present approach, the main objective is to develop a reliable and robust algorithm for viscous mesh generation. For this purpose, topological modifications of originally isotropic tetrahedral elements, which are the main cause of reducing reliability, are avoided during the prismatic layer generation process. Tetrahedral elements near boundary walls are shifted to accommodate each prismatic layer. Prismatic layers are then generated near the walls by automatically monitoring the mesh validity. By using an isotropic tetrahedral mesh as a background grid for the hybrid mesh generation, mesh quality control becomes easier.

Another advantage of using isotropic tetrahedral meshes as the initial grids is that the validity of the grid can be checked by solving the Euler equations beforehand. The meshes are then used for generating hybrid meshes for viscous flow simulations. Required analysis time for a given configuration may be shortened.

The method for creating prisms around sharp corners is also a critical issue. Sharov *et al.* [13] proposed a reliable viscous mesh generation method by modifying the surface mesh generation to generate high-quality stretched tetrahedra near corners and ridges. Although its robustness is quite appealing, a large number of elements will be needed to represent complex geometries. Garimella and Shephard [10] employed multiple normal directions at sharp corners for creating anisotropic elements on no-slip walls. Although smoother transition between neighbouring elements near the corners may be required, far fewer elements are needed. This idea is implemented into the present approach in order to keep the required number of elements to a minimum.

In this paper, a new, robust and fully automated unstructured hybrid grid method based on isotropic tetrahedral grids is presented. A method to improve the quality of the hybrid mesh is also presented. The capability of the method is demonstrated for numerical flow simulations around two wind tunnel models of experimental supersonic airplanes designed at the National Aerospace Laboratory of Japan (NAL).

2. SURFACE GRID GENERATION

Figure 1 shows the outline of the mesh generation process. CAD stereolithography (STL) files are employed for the surface definition. The STL file format is *de facto* standard in rapid prototyping, and the tessellated model defined by the STL file is precise enough to be used for CFD grid generation. The direct advancing front method based on geometric features is employed to generate surface grids [15]. This method is efficient and effective for generating fine surface meshes and has been demonstrated for various complex geometries successfully [16–18].

3. VOLUME GRID GENERATION

3.1. Isotropic tetrahedral grid generation

The entire computational domain defined by a surface mesh is first tessellated by isotropic tetrahedra [19]. Nowadays, techniques to generate isotropic tetrahedral grid have been well developed and any computational field about complex shapes can be fully automatically filled up with tetrahedral cells.

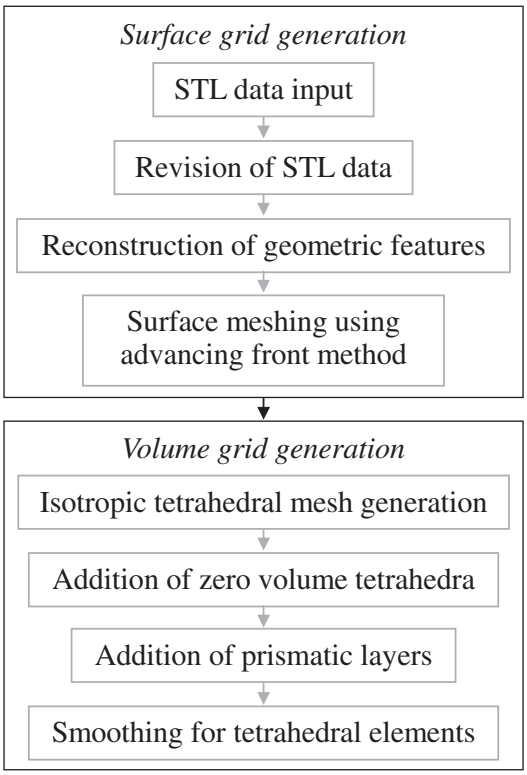

Figure 1. Outline of grid generation process.

3.2. Addition of zero volume tetrahedra

At first, a surface normal vector is calculated at each node on the no-slip walls based on the visibility criterion [2]. These normals are smoothed and are used as marching directions for the prismatic layer generation except at nodes on sharp corners (defined as corner nodes). If a folding angle at each edge on no-slip walls is more than 150°, the edge is considered as a part of a sharp corner. If an edge is bounded by a symmetry plane, the reference angle is halved. Two marching directions are given to all corner nodes.

To employ two marching directions, some pre-processing is required because the hybrid mesh generation method starts with isotropic tetrahedral mesh. Therefore virtual elements—overlapping nodes and edges, zero area triangles, and zero volume tetrahedra—are added on sharp corners. Figure 2 shows how to modify a surface mesh at a sharp corner. In Figure 2(a), edge AB is on a sharp corner and nodes A and B will be given two normals. In this case, nodes C and D, edges AC, BC, BD and CD, triangles ACB and BCD are added as shown in Figure 2(b). Nodes C and D have the same co-ordinates as A and B, respectively, accordingly the lengths of edges AC and BD are zero, and triangles ACB and BCD have zero areas. Two zero volume tetrahedra, which have triangles ACB and BCD, respectively, are added. Each of the tetrahedra exists merely as a triangular face at this stage. The angles between the face and the surface normal vectors of the connecting corner nodes must be small in order

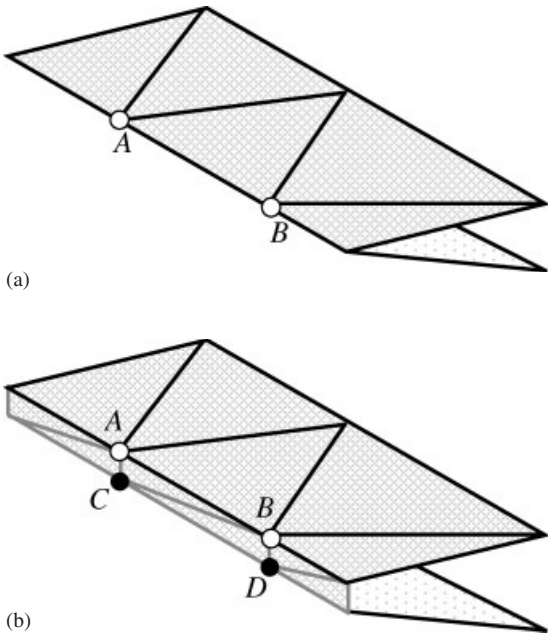

Figure 2. Addition of elements at a sharp corner: (a) original surface mesh and (b) modified surface mesh.

to add prismatic layers easily without creating negative-volume elements. The initially zero volume tetrahedral, which are considered as part of the unstructured tetrahedral mesh, will have increasingly positive volumes as each layer of prisms is added.

The normals at nodes A, B, C and D are modified as follows:

$$\bar{\mathbf{N}}_X = \frac{\bar{\mathbf{N}}_X^0 + \bar{\mathbf{N}}_{MX}}{|\bar{\mathbf{N}}_X^0 + \bar{\mathbf{N}}_{MX}|} \quad (X = A, B, C \text{ or } D), \quad \bar{\mathbf{N}}_C^0 = \bar{\mathbf{N}}_A^0, \quad \bar{\mathbf{N}}_D^0 = \bar{\mathbf{N}}_B^0 \tag{1}$$

where $\bar{\mathbf{N}}_X^0$ is the original surface normal, and $\bar{\mathbf{N}}_{MX}$ is calculated by use of an averaging of the face normals of the modified manifold, as shown in Figure 2(b). If $\bar{\mathbf{N}}_X$ does not satisfy the visibility criterion, it must be corrected.

The calculated node normals are smoothed by the following simple Laplacian smoothing:

$$\bar{\mathbf{N}}_i^{n+1} = \frac{\sum_j \bar{\mathbf{N}}_j^n}{\left|\sum_j \bar{\mathbf{N}}_j^n\right|} \tag{2}$$

where $\bar{\mathbf{N}}_i^n$ and $\bar{\mathbf{N}}_i^{n+1}$ are intermediate and modified node normals of node i, j represents all surrounding nodes belonging to the manifold of node i. This smoothing is not applied to corner nodes. The combination of using two normals at sharp corners and this smoothing method improve the quality of the prisms. To reduce the computational time, the smoothing starts from nodes near sharp corners. A final node normal $\bar{\mathbf{N}}_i$ is obtained from an initial node normal $\bar{\mathbf{N}}_i^0$ after the smoothing has been applied for twenty iterations.

In order to correct the height of the prismatic layers in the following section, a correction value is prepared at each node on no-slip walls.

$$c_i = \min\left(\frac{1}{\bar{\mathbf{N}}_i^0 \cdot \bar{\mathbf{N}}_i}, 2.0\right) \tag{3}$$

3.3. Generation of prismatic layers

The user inputs to generate the prismatic layers are boundary conditions and three parameters for the layers to be added: the number of the layers n_p, an initial layer thickness near boundary walls h_{min}, and a stretching factor f_s. The boundary conditions can be selected from the following three types:

- *No-slip wall*: prismatic layers are generated on these walls.
- *Far-field outer boundary*: nodes on the outer boundaries are fixed.
- *Cross section* (symmetry plane etc.): nodes can be moved on each plane and faces are added as the prismatic layers are grown up the plane.

No user intervention is required during the process. In this process, the addition of each prismatic layer is continued until the user-specified number of layers is obtained or the height of the top layer is compatible with neighbouring tetrahedra. Figure 3 illustrates the mesh generation process, which is summarized as follows:

1. Rank each node by the minimum number of edges that need to be transversed to reach a no-slip wall. Let R_{Wi} be the rank value at node i. For example, nodes on no-slip walls are ranked to be zero, and nodes next to the walls are ranked to be one.
2. Rank each node by the minimum number of edges that need to be transversed to reach an outer boundary or a 'stagnation' node. Let R_{Oi} be this rank value at node i. Nodes on an outer boundary or stagnation nodes are ranked to be zero. The stagnation nodes are defined as follows:
   i. Compare R_{Wi} with R_{Wj} at node i. Index j denotes all the neighbouring nodes of node i. If $R_{Wj} > R_{Wi}$, node i is not a stagnation node.
   ii. Shift no-slip walls by the hypothetical total height of the prismatic layers, and shift the inner nodes in accordance with the no-slip walls using a Laplacian-like method in order to check the nodes' tendencies to move. The grid moving method to be mentioned in Step 7 may be adopted for this purpose. Let $\mathbf{H}_i$ be the resulting moving vector at node i, then calculate the following angle:

$$\theta_{ij} = \cos^{-1}(\bar{\mathbf{H}}_i \cdot \bar{\mathbf{H}}_j) \tag{4}$$

   If the angle $\theta_{ij} > 70°$, node i is considered as a stagnation node. The stagnation nodes may appear around the centre of a tube, in small gaps and so on.
3. No-slip walls are initially considered as a front-line, and all nodes except those on outer boundaries are considered as movable ones. The surface normal at each node on the no-slip walls, $\bar{\mathbf{N}}_i$, corresponds to a marching direction for adding the prismatic layers.

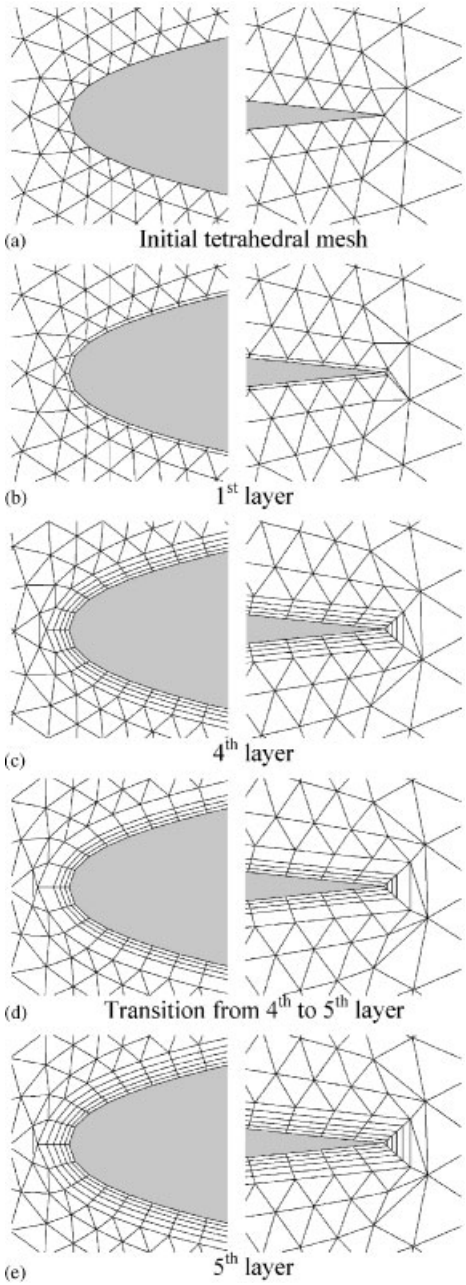

Figure 3. Coarse mesh generation for an NACA 64A010 wing; its leading and trailing edges on the symmetry plane.

4. The mth prismatic layer ($m=1$ to n_p) will be created under the front-line, and its height h_m is as follows:

$$h_m = f_s^{m-1} h_{\min} \tag{5}$$

5. Initialize a displacement vector at each node ($\mathbf{D}_i^0 = \mathbf{0}$).
6. Assign the following displacement vector $\mathbf{D}_i^0$ for each movable node on the front-line.

$$\begin{aligned} \mathbf{D}_i^0 &= \varepsilon \mathbf{D}_i \\ \mathbf{D}_i &= c_i h_m \bar{\mathbf{N}}_i \end{aligned} \tag{6}$$

where ε is a coefficient and c_i is calculated using Equation (3). This value is essential to create prisms that are approximately the same height from the front-line [7]. An example of the prismatic layer generation around a bolt head is shown in Figure 4. Near the no-slip wall, the height of each prismatic layer is constant. While each node on the front-line will be shifted by $\mathbf{D}_i$, $\mathbf{D}_i^0$ is assigned in order to shift interior nodes adequately. ε is initially set as 1.03, and can be locally assigned a larger number when shifting a node on the front-line walls is difficult.

Note that a displacement vector is first assigned at each node in Steps 5–8, and then the nodes will be actually shifted in Step 9 based on the assigned vectors.

7. Assign a displacement vector for each of the interior grid points employing the following weighted Laplacian-like method based on the displacement vectors on the front-line (Equation (6)).

$$\begin{aligned} \mathbf{D}_i^{n+1} &= \mathbf{D}_i^n + c\left(\frac{1}{\sum_j r_j}\sum_j r_j \mathbf{D}_j^n - \mathbf{D}_i^n\right) \\ r_j &= r_{Wj} + r_{Oj} \\ r_{Wj} &= \frac{1}{R_{Wj}+1}, \quad r_{Oj} = \frac{1}{R_{Oj}+1} \end{aligned} \tag{7}$$

where $\mathbf{D}_i^n$ and $\mathbf{D}_i^{n+1}$ are intermediate and modified vectors at node i, respectively, and c is a coefficient for convergence acceleration ($c=1.44$ in this case). The weighting value, r_j, is important to diffuse the assigned movements from nodes on the front-line adequately (r_{Wj}) and to keep the isotropic shape of tetrahedra inside tubes and within small gaps (r_{Oj}). The number of iterations of Equation (7), n, is less than 20 if the applied nodes are sorted by the value, R_{Wi}.

8. Assign the displacement vector $\mathbf{D}_i$ instead of $\mathbf{D}_i^0$ for each movable node on the front-line.
9. Shift each movable node based on the vector $\mathbf{D}_i$ (see Figure 3(d), for example). In order to avoid creating negative-volume or flat tetrahedra, new node positions should be restricted as follows. Figure 5 shows a node and its surrounding triangles in the 2D case. If the node is moved within the shaded zone, negative-volume and flat elements are not created. In 3D, the distance between a node and its surrounding triangular faces must be greater than a certain positive value ζ. If a small value was adopted, slivers would be created. If a large value was adopted, the prismatic layer generation would be locally difficult. In this case, $\zeta = 0.5h_m$ is selected.

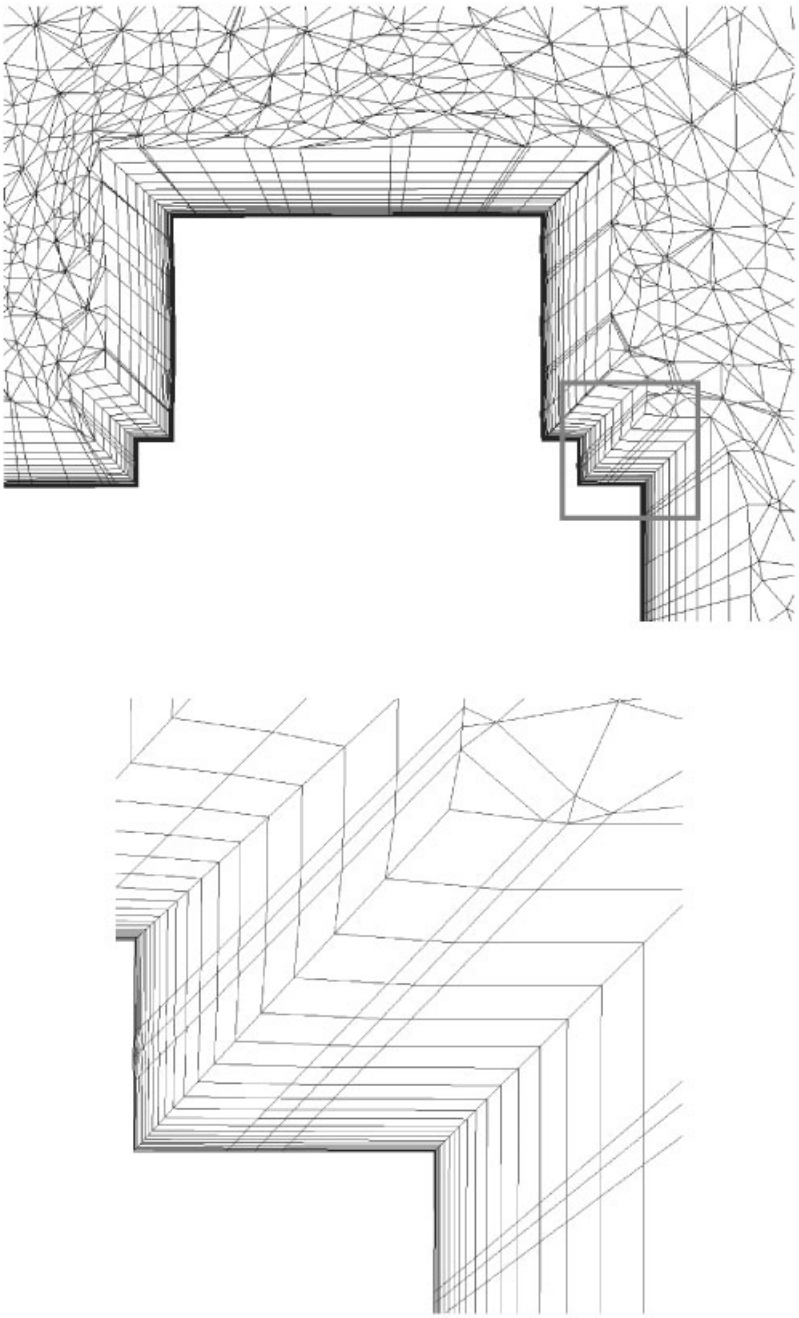

Figure 4. Prismatic layer generation around a bolt head.

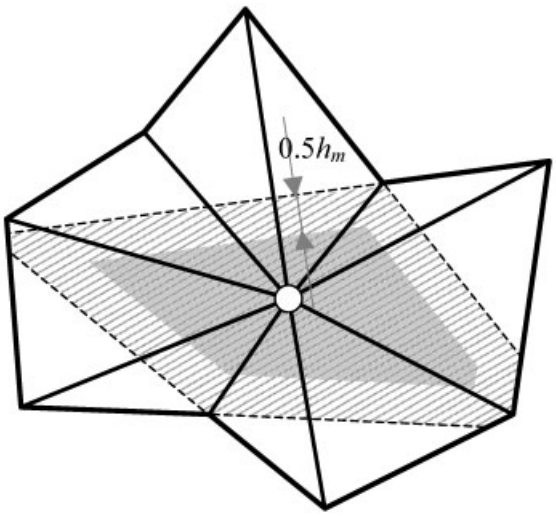

Figure 5. New node location: the node can be moved within the shaded zone of its surrounding triangles in 2D.

10. Check the actual moving distance d_{0i} ($\leqslant|\mathbf{D}_i|$ due to the above restriction) at each movable node on the front-line.
    i. If $0\leqslant d_{0i}\leqslant 0.7|\mathbf{D}_i|$, node i cannot be shifted and is labelled as immovable.
    ii. If $d_{0i}>0.7|\mathbf{D}_i|$, node i can be shifted to accommodate new elements. However, node i will be labelled as immovable after Step 13 if $0.7|\mathbf{D}_i|<d_{0i}\leqslant 0.95|\mathbf{D}_i|$.
11. Generate new elements under the front-line temporarily as shown in Figure 3(e). If a face on the front-line has
    i. one movable node, a tetrahedron is created.
    ii. two movable nodes, a pyramid is created.
    iii. three movable nodes, a prism is created.
12. To avoid creating negative-volume elements, a control volume is calculated at each movable node on the front-line by using projection planes through the node. Two different projection planes are employed to avoid computational errors. If the two computed volumes V_{1i} and V_{2i} at node i satisfy at least one of the following conditions, the control volume is considered negative.

$$V_{1i}<0, \quad V_{2i}<0, \quad \left|\frac{V_{2i}-V_{1i}}{V_{1i}}\right|>10^{-3} \tag{8}$$

    If negative control volumes are detected, the corresponding nodes are labelled as immovable. Then all newly created elements in Step 11 are removed and the procedure goes back to Step 11.
13. If the number of prismatic layers already created is greater than $2n_p/3$, compare height of a connecting tetrahedron, T_t, and prism, T_p, at each face on the front-line. If $T_t<1.7T_p$, the addition of prisms will be locally stopped here, i.e. the three nodes of the face are labelled immovable.

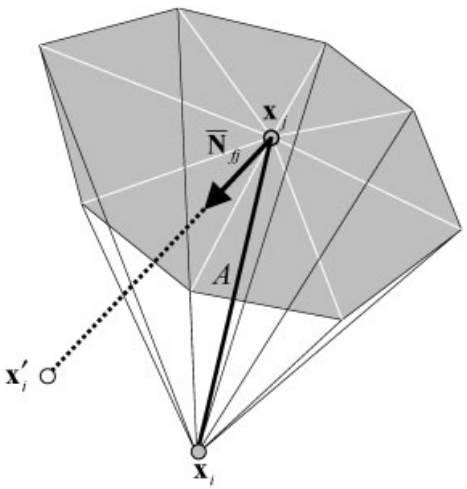

Figure 6. Node i and connecting tetrahedra that share edge A.

14. Update the front-line and continue Steps 4–13 until the user-specified number of layers is created or all the nodes on the front-line become immovable.

3.4. Smoothing for tetrahedral elements

Laplacian smoothing is often employed for enhancing mesh quality because it requires a low computational cost. However, it does not guarantee improving mesh quality and often creates lower quality or invalid elements. Zhou and Shimada [20] proposed an efficient angle-based smoothing method for 2D unstructured meshes. Mesh quality after this algorithm is much better than that after Laplacian smoothing. Moreover, this approach requires much lower computational time than optimization-based smoothing. In this subsection, this method is limited to tetrahedral elements because the prismatic layer generation algorithm in the previous subsection guarantees the quality of the prisms.

Let us consider each node that is only connected to tetrahedral elements. Node i, whose co-ordinate is $\mathbf{x}_i$, has n_{ti} tetrahedra and n_{ei} edges. Figure 6 shows node i and the connecting tetrahedra T_{jk} ($k=1$ to n_f) that shares edge A. Node j is the other end of edge A and its co-ordinate is $\mathbf{x}_j$. Let f_{jk} be the face of T_{jk} opposite to node i. In Figure 6, faces f_{jk} ($k=1$ to n_f) are shown as solid triangles. Face f_{jk} has an area a_{fjk} and a unit normal vector $\bar{\mathbf{N}}_{fjk}$ pointing toward node i. The average unit normal vector $\bar{\mathbf{N}}_{fj}$ and total area a_{fj} are defined as follows:

$$\bar{\mathbf{N}}_{fj} = \frac{\sum_{k=1}^{n_f} \bar{\mathbf{N}}_{fjk}}{\left|\sum_{k=1}^{n_f} \bar{\mathbf{N}}_{fjk}\right|}, \qquad a_{fj} = \sum_{k=1}^{n_f} a_{fjk} \tag{9}$$

With respect to the contribution of node j, the new location of node i is

$$\mathbf{x}'_i = \mathbf{x}_j + |\mathbf{x}_j - \mathbf{x}_i|\bar{\mathbf{N}}_{fj} \tag{10}$$

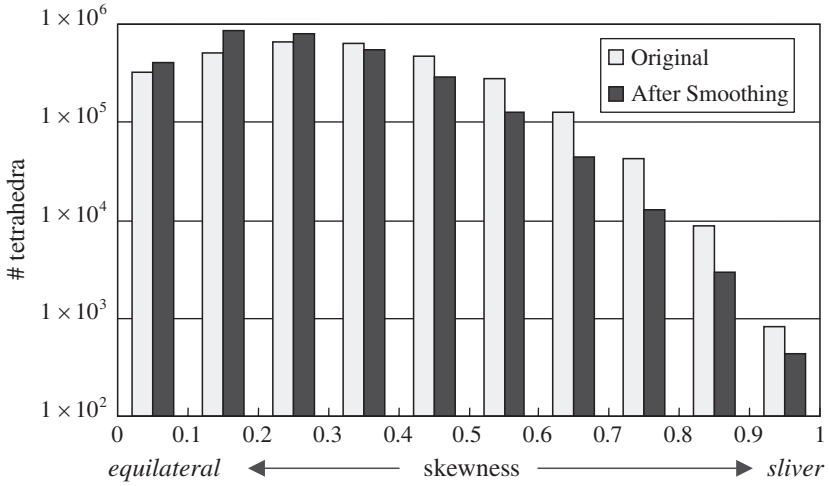

Figure 7. Node smoothing for an isotropic tetrahedral mesh around an airplane configuration.

where $|\mathbf{x}_j - \mathbf{x}_i|$ corresponds the length of edge A. There are n_{ei} neighbouring edges at node i, and each of their contributions is weighted by its area.

$$\mathbf{x}_i' = \frac{\sum_{j=1}^{n_{ei}} a_{fj}(\mathbf{x}_j + |\mathbf{x}_j - \mathbf{x}_i|\bar{\mathbf{N}}_{fj})}{\sum_{j=1}^{n_{ei}} a_{fj}} \tag{11}$$

This method is effective for smoothing tetrahedral elements. However, it does not guarantee the creation of valid elements. To evaluate the local quality, the shortest distance between node i and its surrounding faces is employed. The same idea was used in Section 3.3 and an example in 2D is illustrated in Figure 5. The shortest distances before and after applying Equation (11) are compared. If we obtain a larger value, the node location is updated.

To demonstrate its effectiveness, this smoothing algorithm is applied to a tetrahedral mesh around an airplane configuration having three million elements. The skewness of each tetrahedron i is defined as follows:

$$\varepsilon = \frac{V_{i\text{opt}} - V_i}{V_{i\text{opt}}} \tag{12}$$

where V_i is the volume of tetrahedron i and $V_{i\text{opt}}$ is the volume of an equilateral tetrahedron with the same circumradius. Figure 7 shows a histogram of the skewness of all the elements before and after the node smoothing. The mesh quality is improved significantly.

4. FLOW SOLVER

The full Reynolds-averaged Navier–Stokes equations that retain the unsteady form are solved by a finite volume cell-vertex scheme. The control volumes are non-overlapping dual cells constructed around each node. The Harten-Lax-van Leer-Einfeldt-Wada (HLLEW) Reimann

solver [21] was used for the numerical flux computations. The Lower-Upper-Symmetric Gauss-Seidel (LU-SGS) implicit method [22] was used for time integration. A one-equation turbulence model by Goldberg and Ramakrishnan [23] was implemented to treat turbulent boundary layers.

5. APPLICATIONS

The National Aerospace Laboratory of Japan (NAL) has been working on scaled experimental supersonic airplanes [24–27]. The project will develop two types of experimental airplanes: a rocket-launched unpowered airplane (NEXST-1 project), and a jet-engine-powered airplane (NEXST-2 project). In this section, the hybrid mesh generation applied to an NAL NEXST-1 airplane model is shown.

5.1. NAL NEXST-1 airplane model

In this section, the hybrid mesh generation for an NAL NEXST-1 wing/body/empennage model is presented. Wind tunnel tests have been conducted by NAL with an 8.5% scaled wing/body/empennage/sting model and a 23.3% scaled wing/body/sting model. Numerical results are compared with these experimental data in this section.

Figure 8 shows a surface mesh, which has 89 165 nodes, 178 326 faces. Figures 9 and 10 show a hybrid mesh, which has 2 644 204 nodes, 3 138 729 tetrahedra, 4 075 354 prisms and 49 262 pyramids. This mesh, referred as the intermediate or 2.6 M nodes grid, is mainly used for viscous flow simulations. The full length of the 8.5% scaled model is $L = 0.9775$ m. Parameters for adding the prismatic layers were as follows: the maximum number of layers of 40, the initial layer thickness of 5.0×10^{-7} based on L and the stretching factor of 1.25. At the sharp corners such as the wing trailing edge, two normal vectors were employed to enhance the mesh quality. A smooth transition from prismatic to tetrahedral elements is evident. Two additional meshes were generated for a grid sensitivity analysis. The information on the three grids is summarized in Table I.

The turnaround time for generating the intermediate mesh from an STL file is about 6h. Two hours are required for the surface mesh generation, which includes manual GUI operations for specifying node distributions on geometric features. Fifty minutes for automatic generation of an isotropic tetrahedral mesh on a workstation with a 500 MHz Alpha 21264 processor, having 4 GB of memory. Three hours for addition of the prismatic layers automatically on a PC with 1.7 GHz Pentium III Xeon dual processors and 2 GB of memory.

The Navier–Stokes equations were solved at a free-stream Mach number of 2.0, a Reynolds number based on the full length of the 8.5% scaled model, L, of 26.9×10^6 and angles of attack of $-2.0°$ to $6.0°$. The selected minimum spacing is adequate for solving the turbulent flow because almost all the surface nodes have dimensionless wall distances (y^+) of less than unity. Some nodes on the wing leading edge region had y^+ of around 2.0. Figure 11 shows convergence histories of residual norms, lift coefficients (C_L), and drag coefficients (C_D) for an angle of attack of $4.0°$. The difference, ε, based on the value at 2000 time steps, C_{2000}, is also plotted in each of the latter two graphs. ε is defined as follows:

$$\varepsilon = 100 \left| \frac{C_t - C_{2000}}{C_{2000}} \right| [\%] \quad (t \geqslant 2000) \tag{13}$$

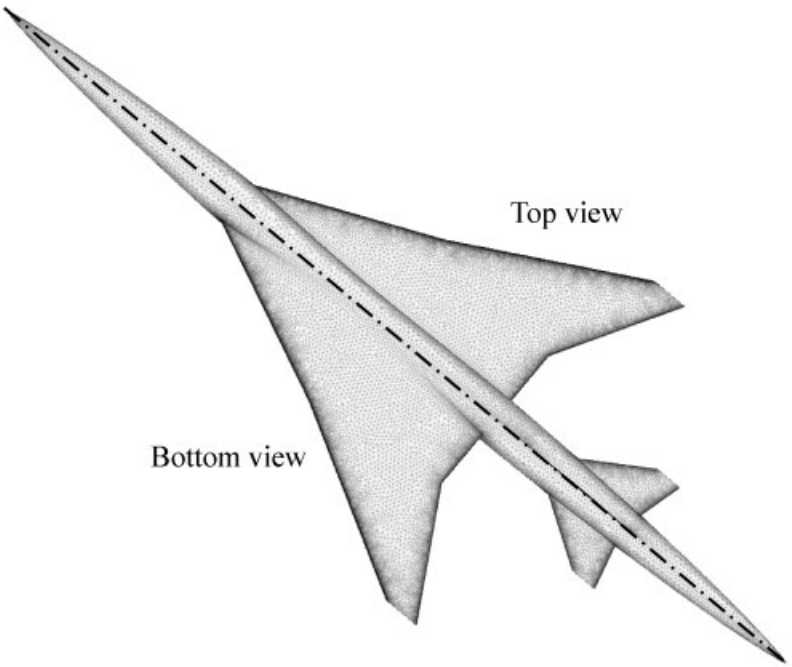

Figure 8. Intermediate surface mesh of a NEXST-1 model.

Although more than 6500 time steps are required to obtain the fully converged solution in this supersonic flow, 2000 time steps seem to be practically adequate, for a Courant–Friedrichs–Levy (CFL) number of 1.0×10^5. Note that fluctuations in the C_L and C_D lie within the range of $\pm 1.0 \times 10^{-5}$ by the same number of time steps. The other cases described later were also computed with the same CFL number for 2000 time steps.

Plate 1 shows a computed surface pressure distribution at an angle of attack of 4.0°. At an angle of attack of 2.0°, pressure contours at $x/L = 0.5$ are illustrated in Figure 12. The smooth transition from prismatic to tetrahedral elements enables smooth pressure distributions to be predicted. Figure 13 shows the chordwise pressure coefficient (C_p) distribution at 30, 50 and 70% semi-span stations at an angle of attack of 2.0°. The wind tunnel data for the two scaled models are plotted for comparison. The C_p distributions of the two numerical results using the intermediate and the fine meshes are approximately consistent. At the wing trailing edge, the C_p distributions seem to be correct in most of the cases due to employing two normal directions at corners when adding prismatic layers. At 30 and 50% semi-span stations, the two wind tunnel models do not give the same results on the upper surface, near the leading edge. This discrepancy is due to using the small-scaled models, which correspond to the original CAD models within a specified tolerance [28]. Except in these regions, the numerical result agrees with experiment well.

C_L, C_D and pitching moment coefficients (C_M) are plotted versus angle of attack in Figure 14 to compare the numerical result and the wind tunnel test data for the 8.5% scaled

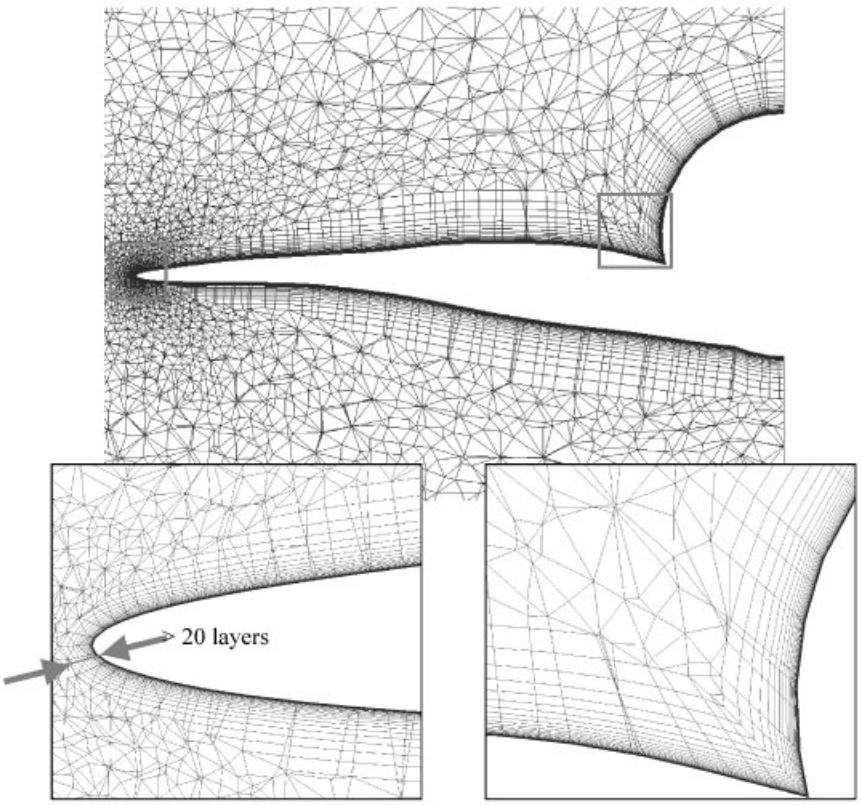

Figure 9. Intermediate hybrid mesh for a NEXST-1 model ($x/L = 0.5$).

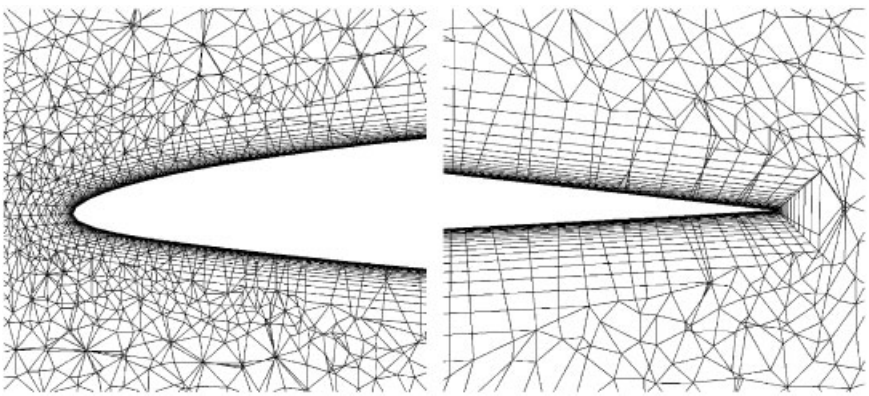

Figure 10. Intermediate hybrid mesh for a NEXST-1 model: wing leading edge and trailing edge at 30% semi-span station.

Table I. Grid information of a NEXST-1 model.

		Grids		
		Coarse (1.4M nodes)	Intermediate (2.6M nodes)	Fine (4.6M nodes)
Surface mesh	# nodes	61 095	89 165	147 730
	# triangles	122 186	178 326	295 456
Initial tet mesh	# nodes	327 481	562 727	927 843
	# tetrehedra	1 768 146	3 085 014	5 083 723
Hybrid mesh	# layers	30	40	42
	Initial thickness	1.0×10^{-5}	5.0×10^{-7}	2.5×10^{-7}
	Stretching factor	1.20	1.25	1.25
	# nodes	1 356 482	2 644 204	4 591 418
	# tetrehedra	1 825 409	3 138 729	5 167 255
	# prisms	1 984 774	4 075 354	7 207 519
	# pyramids	56 169	49 262	77 461

model. Computational results using the coarse and fine hybrid meshes are also presented for reference. Although the initial layer spacing of the coarse mesh is too thick in regard to the y^+ on the viscous walls, the three grids used predict almost the same result. The computational results, however, do not agree with experiment data, especially at higher angles of attack:

- C_L: The gradient of the numerical results are steeper than that of the wind tunnel result.
- C_D: The numerical results are overpredicted at high angles of attack.
- C_M: The gradient of the wind tunnel result is steeper than that of the numerical results.

In the experiment, the measured data seems to be affected by the wind tunnel wall. To validate this hypothesis, C_L and C_M are plotted versus C_D as shown in Figure 15 (drag polar) and Figure 16 (C_M–C_D). The experimental C_D's are plotted with an accuracy of $\pm 3.0 \times 10^{-4}$ (3 counts). Although the numerical results underpredict the pitching moment at high angles of attack, good agreement is achieved with each case.

5.2. NAL NEXST-2 airplane model

In this section, the hybrid mesh generation was applied to an 8.3% scaled wind tunnel model of the NAL NEXST-2 (No. 08 configuration) as shown in Figure 17 [29]. Its geometry is completely redesigned from NEXST-1. Twin engine nacelles are installed through diverters. Figure 18 shows a surface mesh, which has 167 448 nodes and 334 896 triangles. To represent the small gap between the wing lower surface and the upper intake lip correctly, adequate nodes were clustered there. An isotropic tetrahedral mesh, which has 1 304 880 nodes and 7 284 547 tetrahedra, was created based on the surface mesh and prismatic layers were automatically added to obtain a viscous mesh as shown in Figures 19 and 20, which has 5 465 951 nodes, 7 769 306 tetrahedra, 8 052 167 prisms and 90 801 pyramids. Since 32-bit Windows PC's cannot have enough memory for generating such a large hybrid mesh, an SGI Origin 2000 with

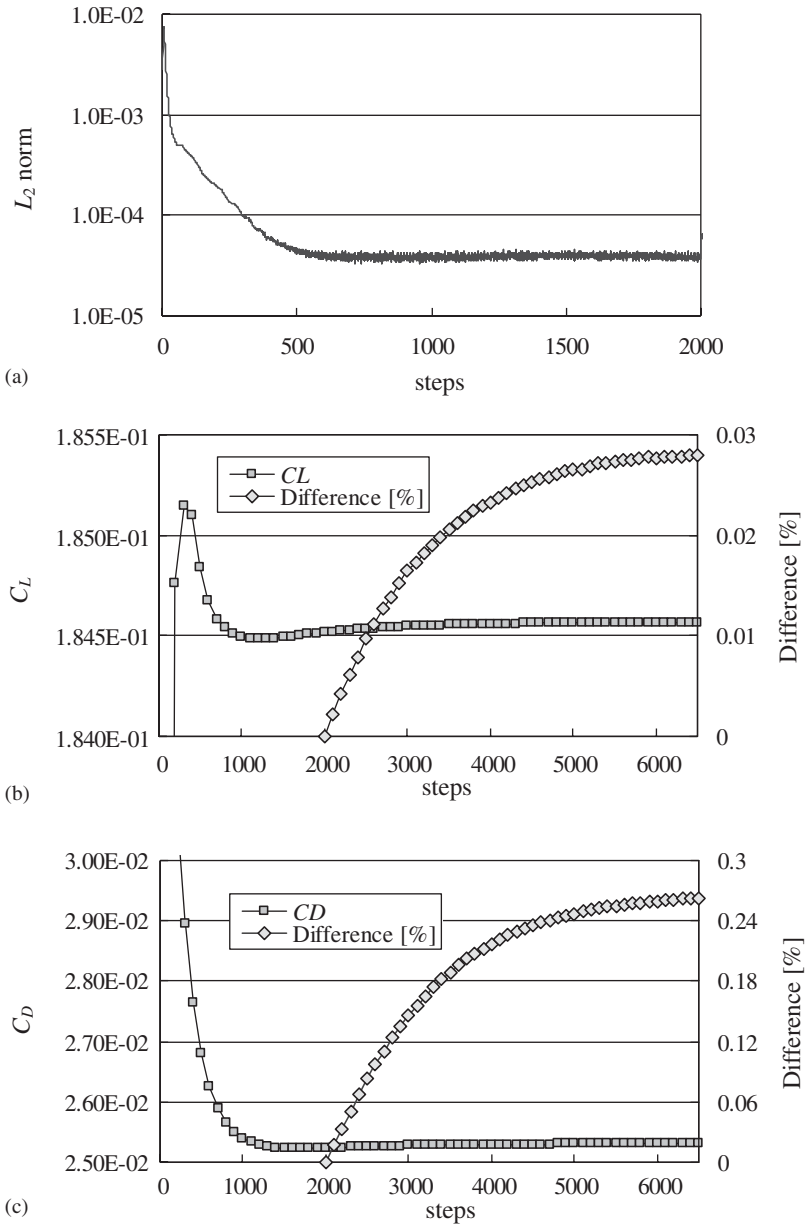

Figure 11. Convergence history of a viscous flow computation for a NEXST-1 model (2.6 M nodes, $M_\infty = 2.0$, $\alpha = 4.0^\circ$): (a) residual norm; (b) C_L and (c) C_D.

1 CPU (0.6 GFLOPS nominal computing power) was employed. The required CPU time was 35 h and 1.9 GB of memory was used at the maximum. The mesh generation process was fully automatic and the input parameters were: the maximum number of layers of 40, the initial layer thickness of 5.0×10^{-7} based on the full length of the wind tunnel model $L = 0.9545$ m

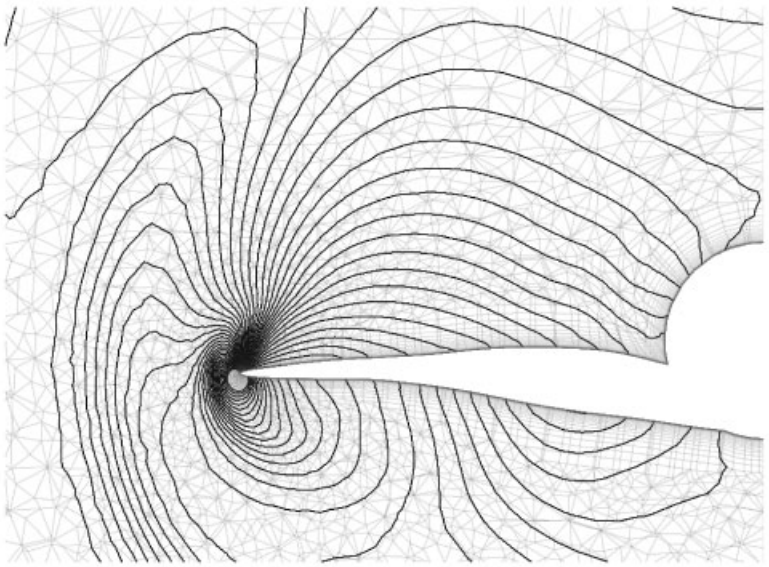

Figure 12. Pressure contours at $x/L = 0.5$ of a NEXST-1 model ($M_\infty = 2.0$, $\alpha = 2.0^\circ$).

and the stretching factor of 1.25. The actual maximum number of layers is 39 because the top of the layers already had enough height. A coarse mesh is also prepared for reference. Their grid information is summarized in Table II.

The Navier–Stokes equations were solved at a free-stream Mach number of 1.7, a Reynolds number based on L of 24.8×10^6 and angles of attack of -2.0° to 4.0°. The selected initial layer thickness of the fine grid was adequate to solve the turbulent flow because it ensured $y^+ < 1.0$ on most of the viscous walls (maximum of 2.0). Figure 21 shows convergence histories of residual norms, C_L and C_D at an angle of attack of 2.0°. This simulation required 79 CPU hours and 7.5 GB of memory using an NEC SX-5 single processor (8 GFLOPS nominal computing power) for 3000 time steps, for a CFL number of 1.0×10^5. Compared with the flow computation, the required computational resources for the prismatic layer generation were much less. 2000 time steps were adequate to obtain the converged solutions in this supersonic flow. The other cases are also computed with the same CFL number for 2000 time steps.

Plate 2 shows a computed surface pressure distribution, and Figure 22 an oil flow pattern at an angle of attack of 2.0°. Compared to the NEXST-1 model, the flow physics become more complicated due to the diverter and the nacelle, which produce strong shock waves. Pressure contours are illustrated in Figure 23 at $x/L = 0.6$.

The objective of the wind tunnel tests was to evaluate the performance of the airframe, not of the propulsion system, accordingly the contribution of the internal flow of the nacelles was subtracted. On the CFD side, the mesh inside the nacelle was made coarse and its surface was removed from the integral domain when the aerodynamic coefficients were estimated. C_L, C_D

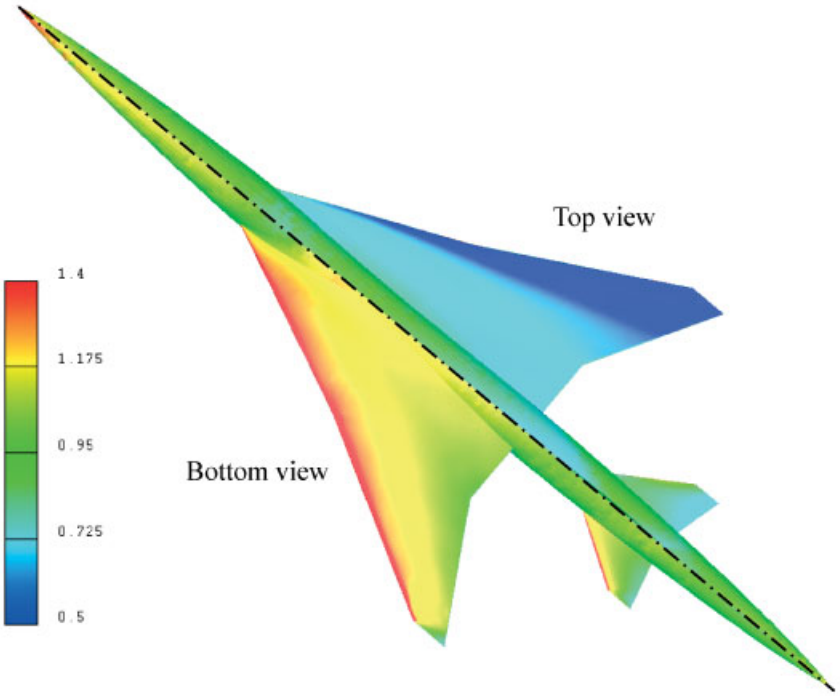

Plate 1. Surface pressure distribution of a NEXST-1 model (2.6M nodes, $M_\infty = 2.0$, $\alpha = 4.0^\circ$).

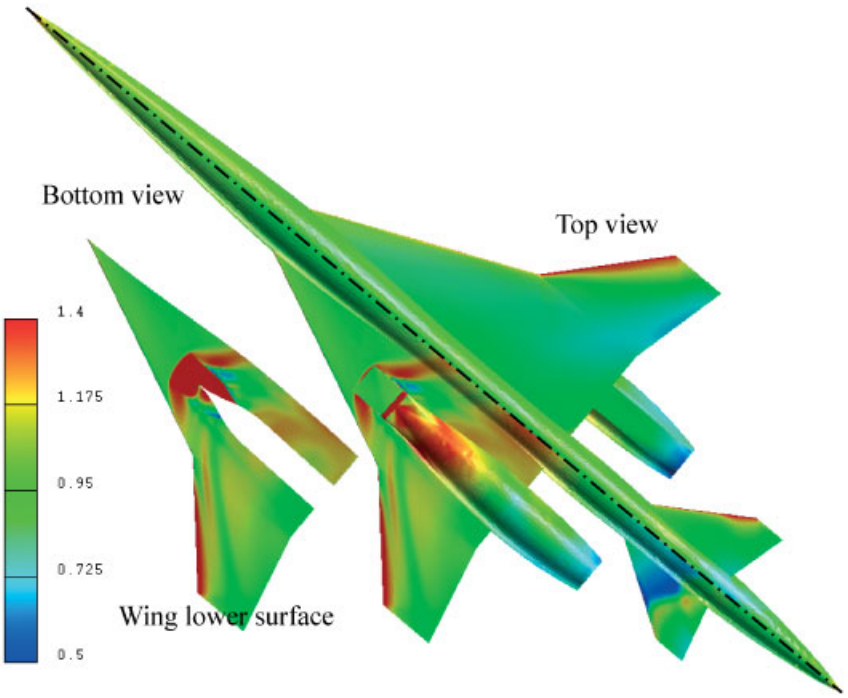

Plate 2. Surface pressure distribution of a NEXST-2 model ($M_\infty = 1.7$, $\alpha = 2.0^\circ$).

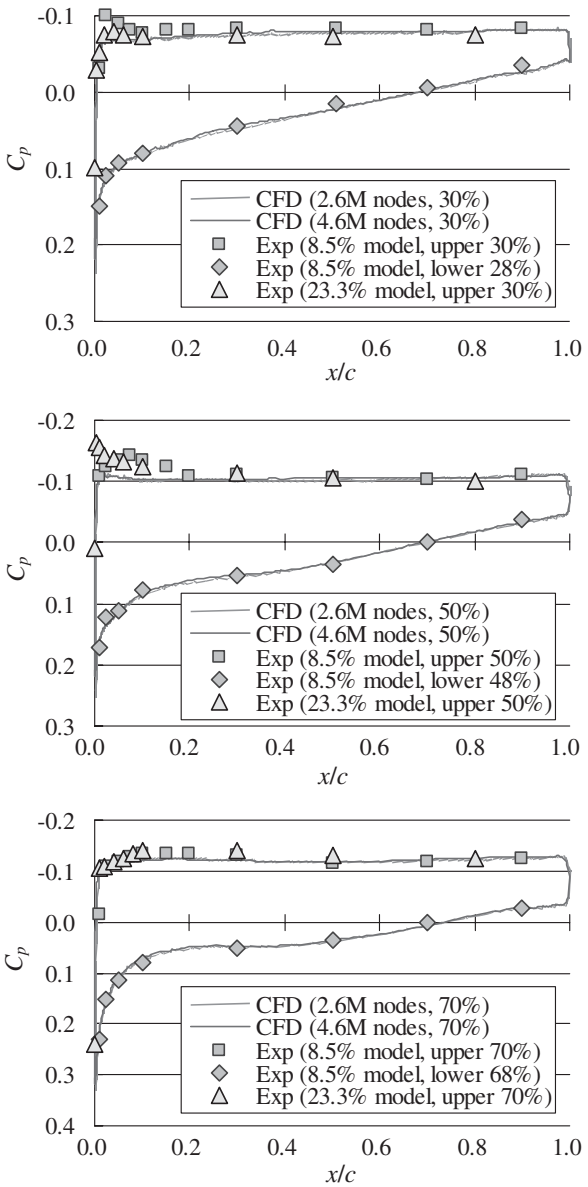

Figure 13. C_p distributions at 30%, 50% and 70% semi-span stations of a NEXST-1 model ($M_\infty = 2.0$, $\alpha = 2.0^\circ$).

and pitching moment coefficients (C_M) are plotted versus angle of attack in Figure 24 that compares the numerical result and the wind tunnel test data. The computational result using the coarse mesh is also presented for reference. Because this mesh has much fewer nodes around the engine intake, the physical phenomena due to the strong shock wave generated

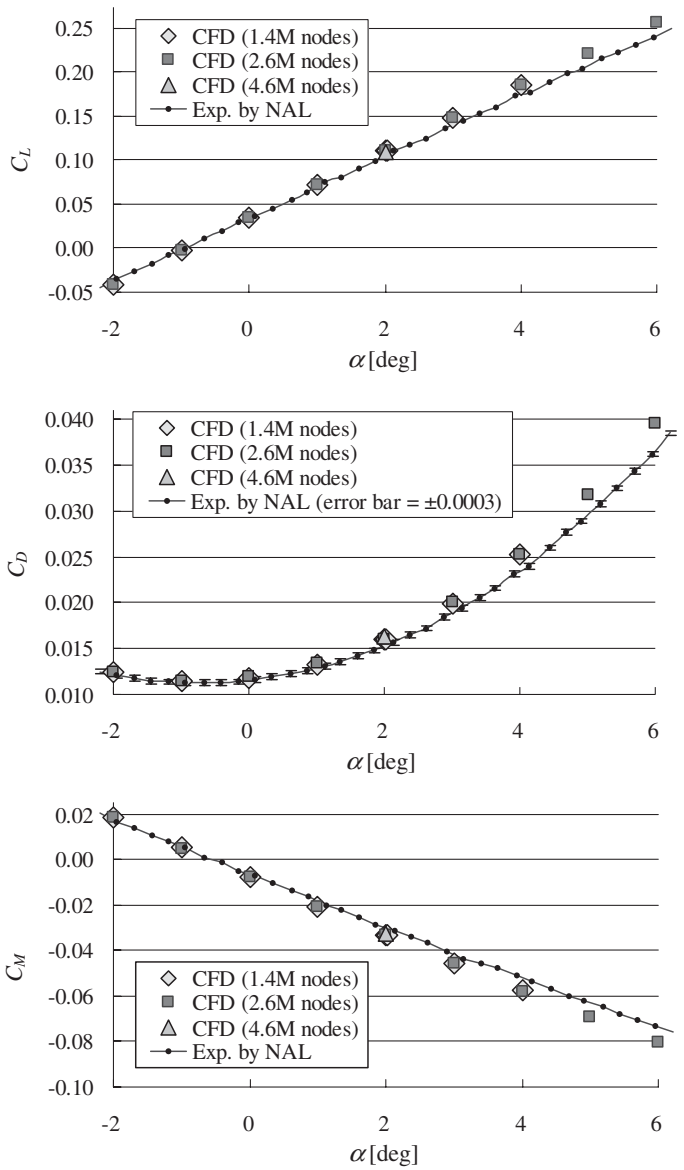

Figure 14. C_L-α, C_D-α and C_M-α of a NEXST-1 model ($M_\infty = 2.0$, $\alpha = -2.0$ to 6.0°).

there cannot be captured correctly. Moreover, the initial prismatic layer spacing normal to the viscous walls is $y^+ \cong 30$. Consequently, the coarse mesh underpredicts the drag in the entire region by 19 counts, although the fine mesh also underpredicts the drag by as much as 15 counts. The wind tunnel wall seems to affect the flow as with the NEXST-1 model case.

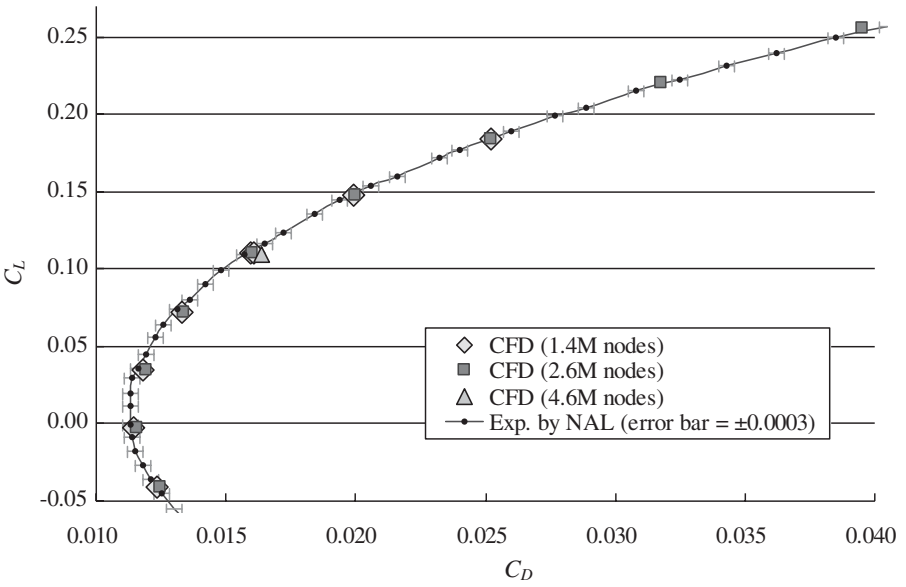

Figure 15. C_L–C_D curve of a NEXST-1 model ($M_\infty = 2.0$, $\alpha = -2.0$ to 6.0°).

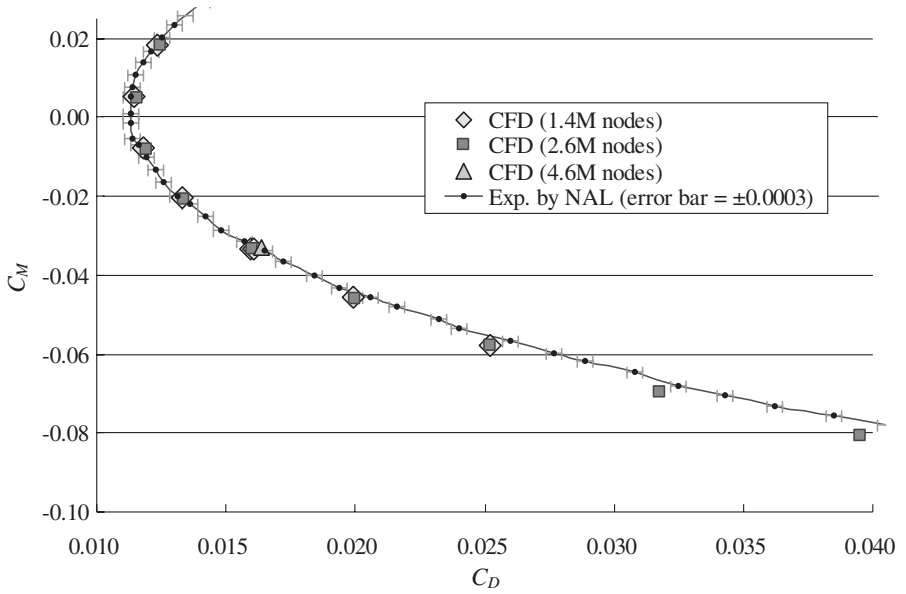

Figure 16. C_M–C_D curve of a NEXST-1 model ($M_\infty = 2.0$, $\alpha = -2.0$ to 6.0°).

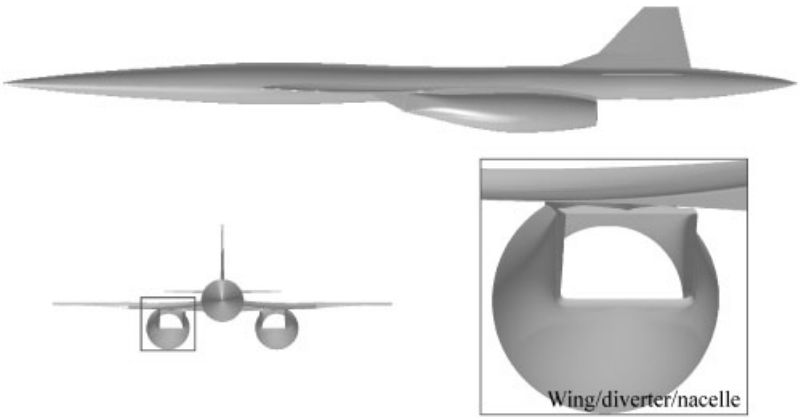

Figure 17. NEXST-2 08 model.

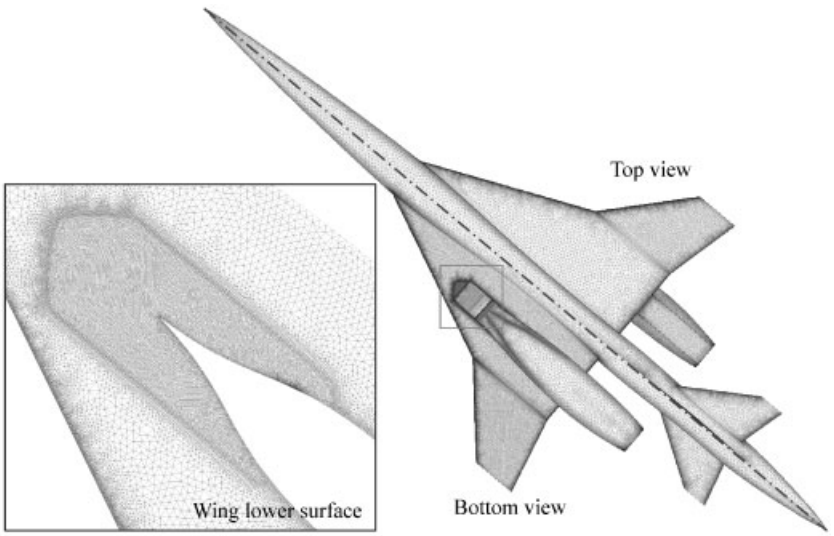

Figure 18. Surface mesh of a NEXST-2 model.

To exclude the influence of the wind tunnel wall, the relations between C_L, C_D and C_M are shown in Figure 25 (C_L-C_D) and Figure 26 (C_M–C_D). There are two discrepancies between the finer mesh result and the wind tunnel data:

1. Although the numerical result agrees qualitatively well with the experiment data, it underpredicts the drag by 15 counts.
2. Compared with the offset wind tunnel data, the numerical result underpredicts the pitching moment at high angles of attack.

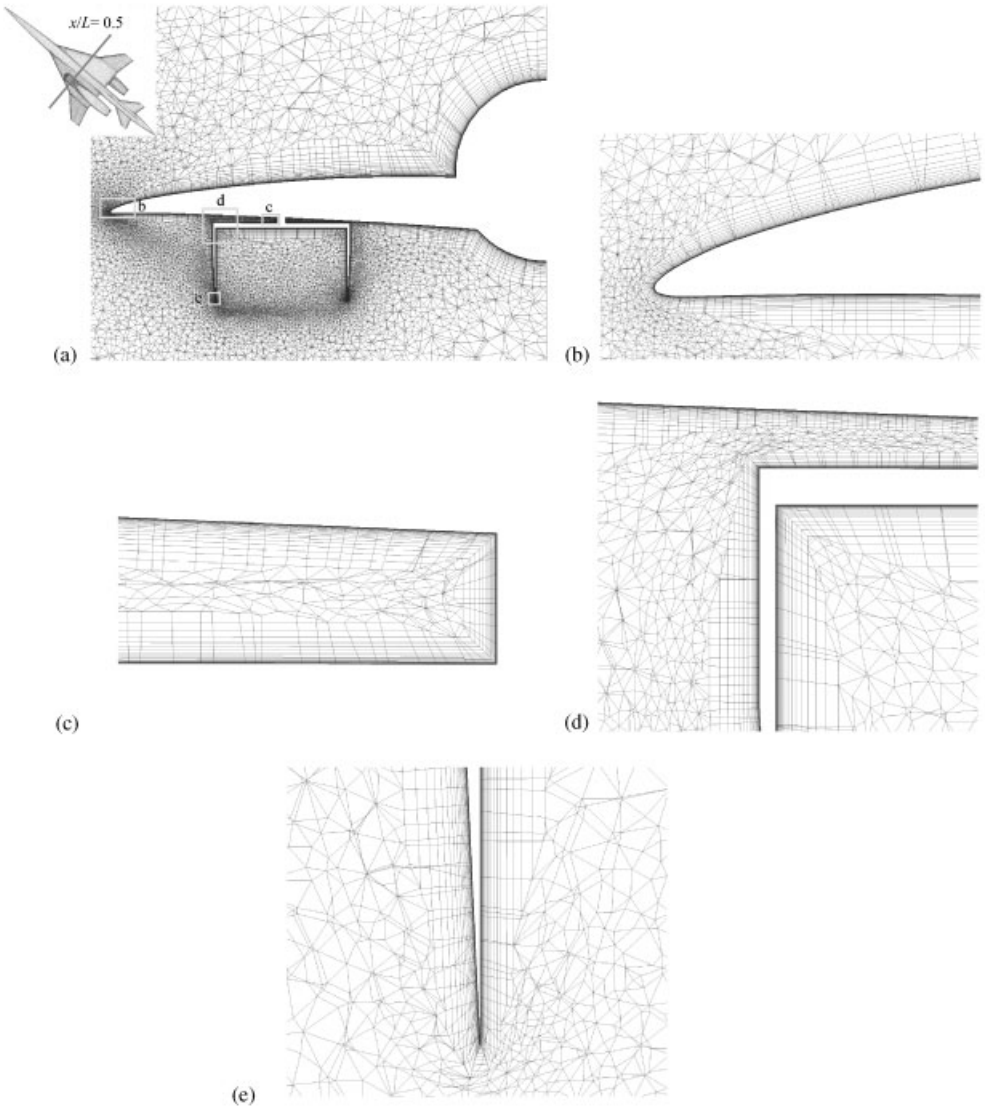

Figure 19. Hybrid mesh of a NEXST-2 model ($x/L = 0.5$): (a) wing/body/intake; (b) wing leading edge; (c) gap between wing and intake; (d) intake corner and (e) outboard intake lip.

With regard to point 1, adaptive mesh refinement may be needed to capture the shock waves around the engine intake more clearly. However, considering the coarse mesh, the big disagreement is not mainly due to the grid sensitivity. The prime reason for the disagreement of the drag probably originates in the selected turbulence model not being suitable for modelling such complicated flow. Thus, further validation is required.

With regard to 2, the tendency is the same as presented for the NEXST-1 model. The wind tunnel data seem to include a certain error.

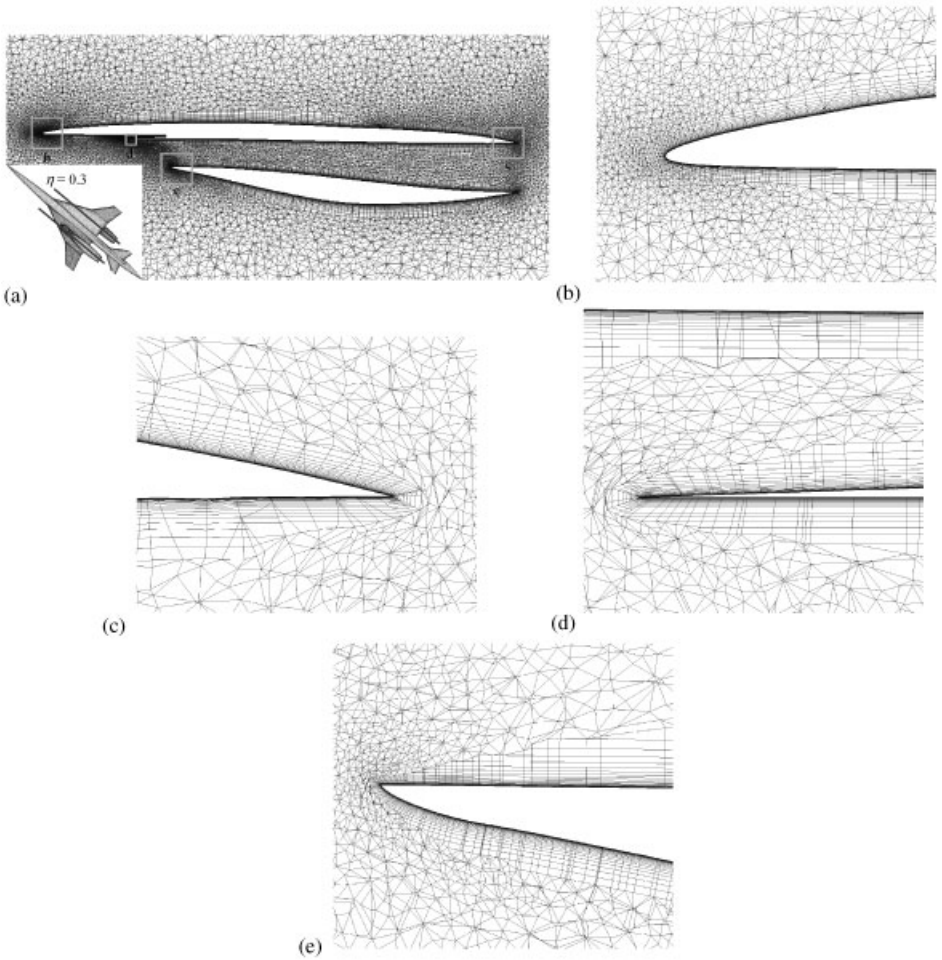

Figure 20. Hybrid mesh of a NEXST-2 model ($\eta = 0.3$): (a) wing and engine nacelle; (b) wing leading edge; (c) wing trailing edge (engine outlet); (d) wing lower surface and upper intake lip and (e) lower intake lip.

6. CONCLUSIONS

An automated hybrid grid generation method has been developed, which has the following features:

- The hybrid grid generation method starts from an isotropic tetrahedral grid that can be generated for any complex geometries to enhance the robustness of the generation procedure.
- The prismatic layer generation is carried out fully-automatically by obeying user-specified boundary conditions and three parameters: the number of the layers, an initial layer thickness near boundary walls, and a stretching factor.

Table II. Grid information of a NEXST-2 model.

		Grids	
		Coarse (2.0M nodes)	Fine (5.5M nodes)
Surface mesh	# nodes	105 097	167 448
	# triangles	210 194	334 896
Initial tet mesh	# nodes	588 022	1 372 195
	# tetrehedra	3 186 750	7 667 927
Hybrid mesh	# layers	22	39
	Initial thickness	1.0×10^{-5}	5.0×10^{-7}
	Stretching factor	1.30	1.25
	# nodes	1 961 068	5 465 951
	# tetrehedra	3 263 446	7 769 306
	# prisms	2 659 695	8 052 167
	# pyramids	75 398	90 801

- Each prismatic layer is then added step by step near no-slip wall boundaries automatically, while overall mesh validity is retained. Topological changes for original isotropic tetrahedral elements are prohibited during the addition process.
- The use of two normals improves grid quality around sharp corners.
- The angle-based smoothing method, which is applied to a tetrahedral domain, improves mesh quality significantly.

The method was applied to two wind tunnel models of the NAL experimental supersonic airplanes in order to check the quality of the generated meshes. Two or three grids of different sizes were prepared for each model in order to check the grid dependencies of the solutions. The numerical results of the viscous flow simulations were compared with experiment.

NEXST-1 wing/body/empennage model ($M_\infty = 2.0$, $\alpha = -2.0$ *to* 6.0°):

- Lift and drag coefficients agreed well with experiment.
- Pitching moment coefficients were slightly underpredicted at high angles of attack.

NEXST-2 wing/body/empennage/diverter/nacelle model ($M_\infty = 1.7$, $\alpha = -2.0$ *to* 4.0°):

- Lift coefficients agreed well with experiment.
- Drag coefficients were underpredicted in the entire region by 15 counts.
- Pitching moment coefficients were also slightly underpredicted at high angles of attack compared with the offset wind tunnel data.

The disagreement of the pitching moment coefficients at high angles of attack probably originated due to the presence of the wind tunnel wall. On the NEXST-2 model, drag coefficients of the finer mesh were lower than experimental values by 15 counts. Further validation of the turbulence model used in the flow solver may be required. The computational results certified that the proposed unstructured mesh generation method generated well-qualified grid distributions for high Reynolds number flow computations.

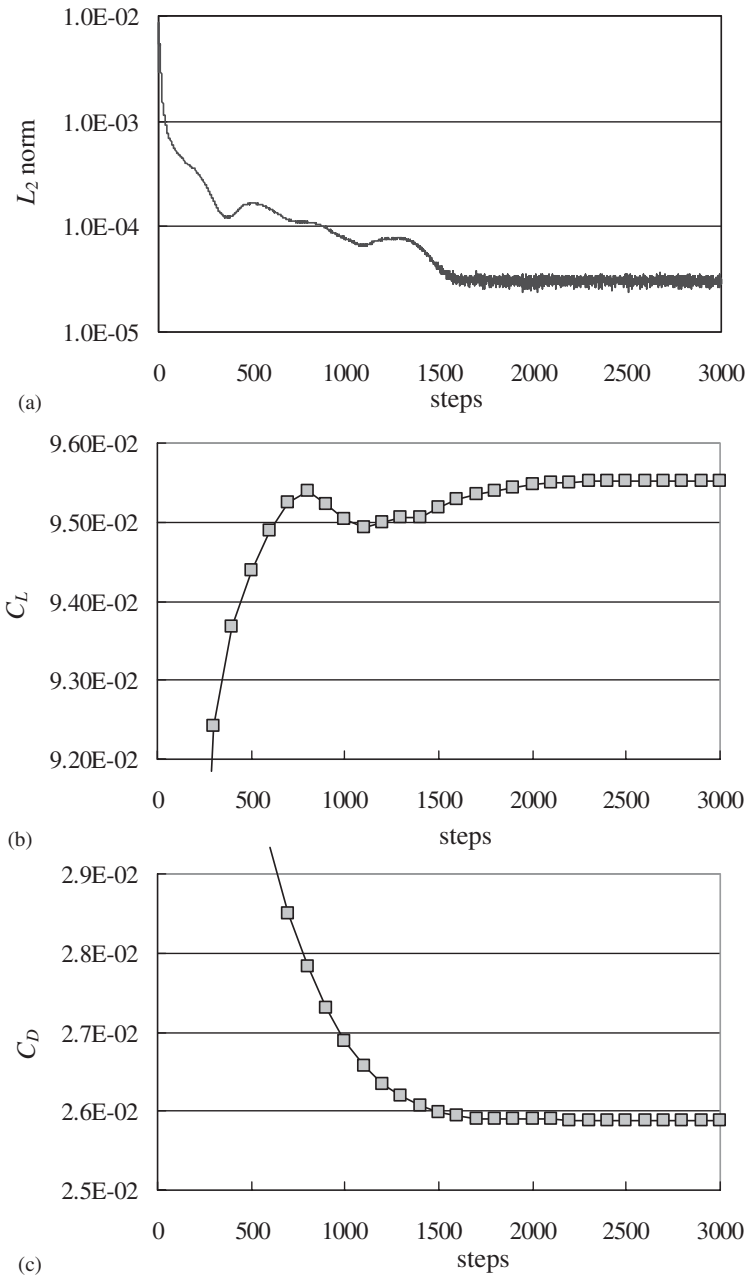

Figure 21. Convergence history of a viscous flow computation for a NEXST-2 model ($M_\infty = 1.7$, $\alpha = 2.0^\circ$): (a) residual norm; (b) C_L and (c) C_D.

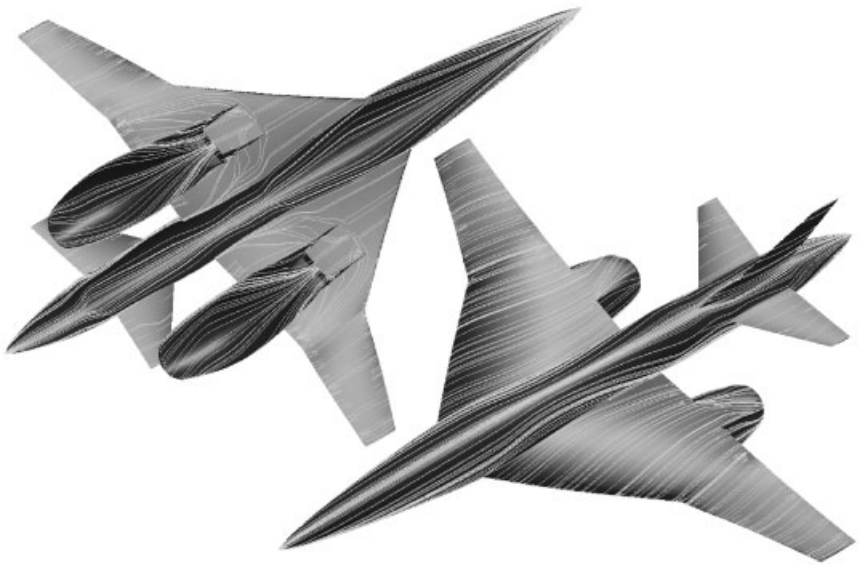

Figure 22. Oil flow pattern of a NEXST-2 model ($M_\infty = 1.7$, $\alpha = 2.0^\circ$).

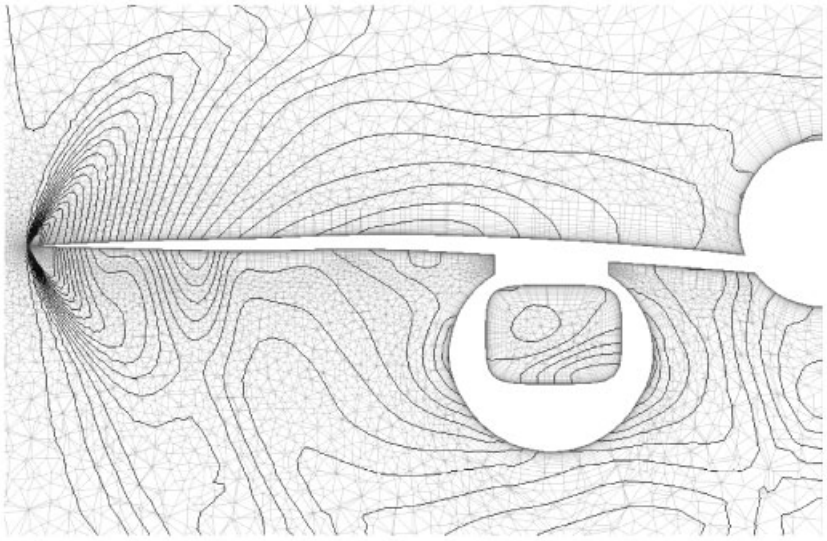

Figure 23. Pressure contours at $x/L = 0.6$ of a NEXST-2 model ($M_\infty = 1.7$, $\alpha = 2.0^\circ$).

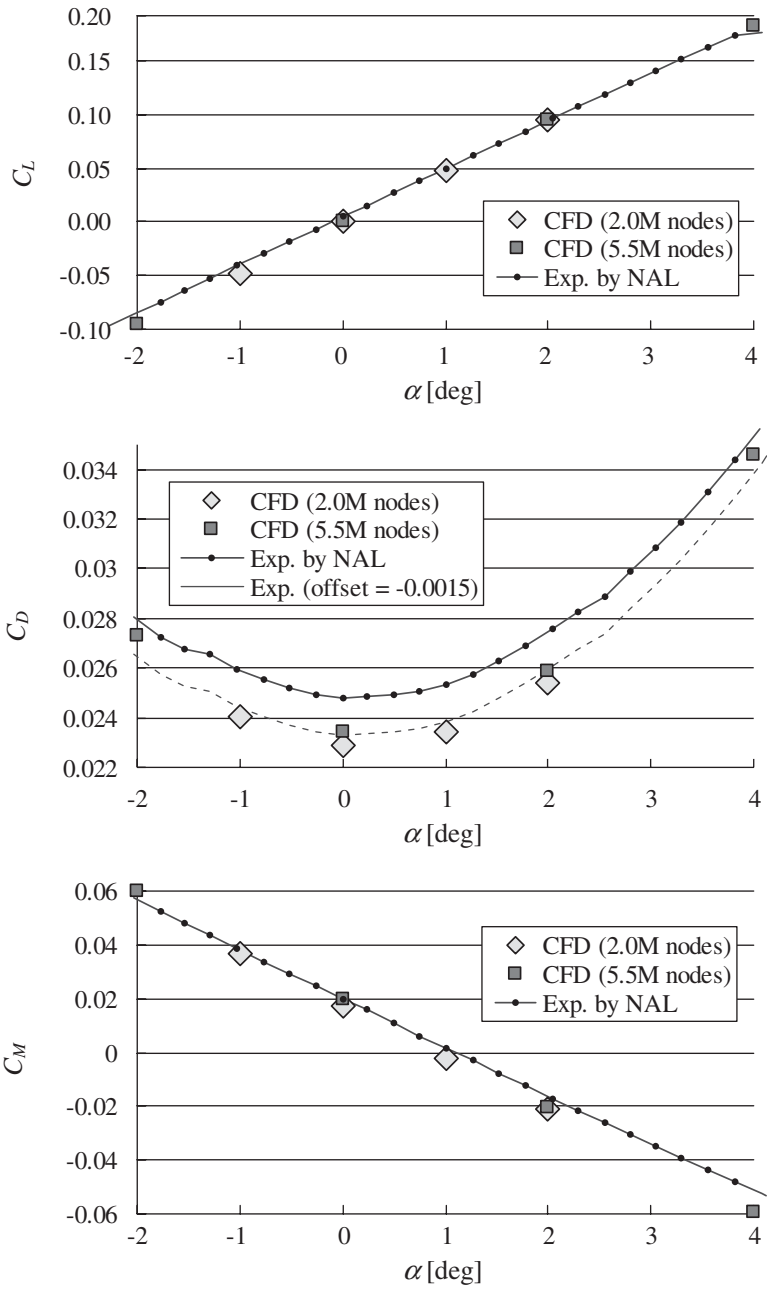

Figure 24. C_L-α, C_D-α and C_M-α of a NEXST-2 model ($M_\infty = 1.7$, $\alpha = -2.0$ to 4.0°).

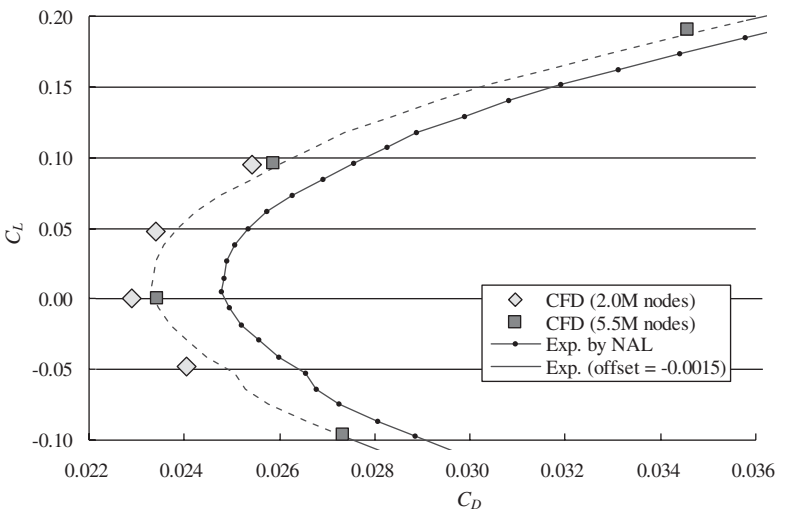

Figure 25. C_{L}–C_{D} curve of a NEXST-2 model ($M_\infty = 1.7$, $\alpha = -2.0$ to 4.0°).

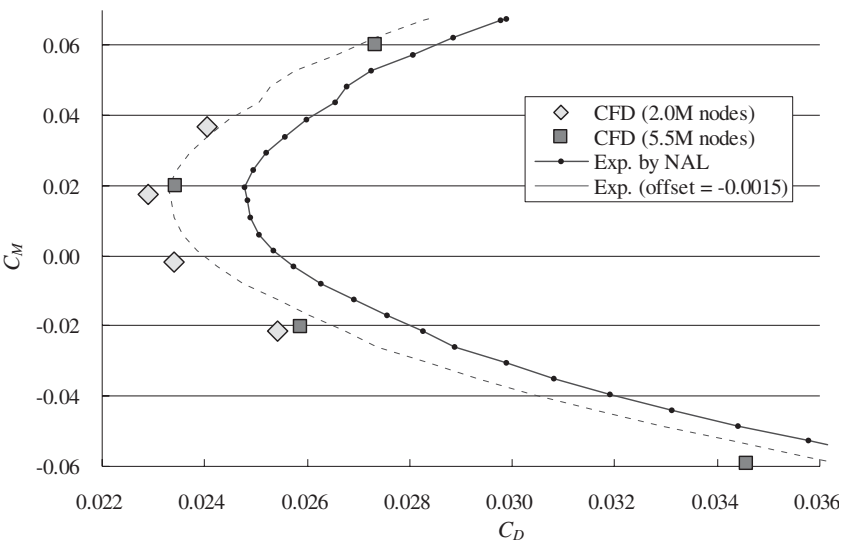

Figure 26. C_{M}–C_{D} curve of a NEXST-2 model ($M_\infty = 1.7$, $\alpha = -2.0$ to 4.0°).

ACKNOWLEDGEMENTS

The geometries and wind tunnel data of the experimental supersonic airplanes were provided by Aerodynamic Design Group, Next Generation Supersonic Transport Project Center, National Aerospace Laboratory of Japan. The authors would like to thank T. Fujita and M. Morita, graduate students of Tohoku University for generating the STL files. The Navier-Stokes equations were solved using NEC SX-5's in the Institute of Fluid Science, Tohoku University, and NEC SX-7's in Super-Computing System Information Synergy Center, Tohoku University.

REFERENCES

1. Nakahashi K. Marching grid generation for external viscous flow problems. *Transactions of Japan Society for Aeronautical and Space Sciences* 1992; **35**(108):88–102.
2. Kallinderis Y, Ward S. Prismatic grid generation for three-dimensional complex geometries. *AIAA Journal* 1993; **31**(10):1850–1856.
3. Marcum DL. Generation of unstructured grids for viscous flow applications. *AIAA Paper 95-0212*, 1995.
4. Connell SD, Braaten ME. Semistructured mesh generation for three-dimensional Navier-Stokes calculations. *AIAA Journal* 1995; **33**(6):1017–1024.
5. Pirzadeh S. Three-dimensional unstructured viscous grids by the advancing-layers method. *AIAA Journal* 1996; **34**(1):43–49.
6. Hassan O, Morgan K, Probert EJ, Peraire J. Unstructured tetrahedral mesh generation for three-dimensional viscous flows. *International Journal for Numerical Methods in Engineering* 1996; **39**:549–567.
7. Sharov D, Nakahashi K. Hybrid prismatic/tetrahedral grid generation for viscous flow applications. *AIAA Journal* 1998; **36**(2):157–162.
8. Currie TC. Modifying isotropic tetrahedral grids for viscous simulations. *Proceedings of the 7th Annual Conference of the Computational Fluid Dynamics Society of Canada*, 1999.
9. Khawaja A, Kallinderis Y. Hybrid grid generation for turbomachinery and aerospace applications. *International Journal for Numerical Methods in Engineering* 2000; **49**:145–166.
10. Garimella RV, Shephard MS. Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering* 2000; **49**:193–218.
11. Löhner R, Cebral J. Generation of non-isotropic unstructured grids via directional enrichment. *International Journal for Numerical Methods in Engineering* 2000; **49**:219–232.
12. Michal T, Cary A. Unstructured grid extrusion for viscous flow simulations. *AIAA Paper 2001-0444*, 2001.
13. Sharov D, Luo H, Baum JD, Löhner R. Unstructured Navier-Stokes grid generation at corners and ridges. *AIAA Paper 2001-2600*, 2001.
14. Wurtzler KE, Tomaro RF, Witzeman FC. Analysis of three unstructured grid techniques from a user's perspective. *Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations*, 2000; 357–366.
15. Ito Y, Nakahashi K. Direct surface triangulation using stereolithography data. *AIAA Journal* 2002; **40**(3):490–496.
16. Togashi F, Nakahashi K, Ito Y, Iwamiya T, Shimbo Y. Flow simulation of NAL experimental supersonic airplane/booster separation using overset unstructured grids. *Computers & Fluids* 2001; **30**(6):673–688.
17. Fujita T, Ito Y, Nakahashi K, Iwamiya T. Computational fluid dynamics evaluation of national aerospace laboratory experimental supersonic airplane in ascent. *Journal of Aircraft* 2002; **39**(2):359–364.
18. Ito Y, Nakahashi K. Surface triangulation for polygonal models based on CAD data. *International Journal for Numerical Methods in Fluids* 2002; **39**(1):75–96.
19. Sharov D, Nakahashi K. A boundary recovery algorithm for Delaunay tetrahedral meshing. *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, 1996; 229–238.
20. Zhou T, Shimada K. An angle-based approach to two-dimensional mesh smoothing. *Proceedings of the 9th International Meshing Roundtable*, 2000; 373–384.
21. Obayashi S, Guruswamy GP. Convergence acceleration of an aeroelastic Navier-Stokes solver. *AIAA Journal* 1995; **33**(6):1134–1141.
22. Sharov D, Nakahashi K. Reordering of hybrid unstructured grids for lower-upper symmetric Gauss-Seidel computations. *AIAA Journal* 1998; **36**(3):484–486.
23. Goldberg UC, Ramakrishnan SV. A pointwise version of Baldwin-Barth turbulence model. *Computational Fluid Dynamics* 1993; **1**:321–338.
24. Iwamiya T. NAL SST project and aerodynamic design of experimental aircraft. *Proceedings of the 4th ECCOMAS Computational Fluid Dynamics Conference*, Athens. Wiley: Chichester, UK, 1998; 580–585.
25. Sakata K. Supersonic experimental airplane (NEXST) for next generation SST technology. *AIAA Paper 2002-0527*, 2002.
26. Iwamiya T. A computational study on unmanned scaled supersonic experimental airplane. *AIAA Paper 2002-2841*, 2002.
27. Yoshida K, Makino Y, Shimbo Y. An experimental study on unmanned scaled supersonic experimental airplane. *AIAA Paper 2002-2842*, 2002.
28. Jeong S. Application of inverse method to mutual verification of CFD-EFD. Special Publication of National Aerospace Laboratory SP-57 (*Proceedings of the Aerospace Numerical Simulation Symposium 2002*), Tokyo, Japan, 2002; 89–93.
29. Yoshida K, Noguchi M, Shimbo Y, Kuroda F. Comparison of wind tunnel test and CFD analysis on an aircraft/nacelle configuration of the scaled supersonic experimental airplane. *Proceedings of the 39th Aircraft Symposium*, Gifu, Japan, 2001 (in Japanese).

