AIAA JOURNAL
Vol. 47, No. 5, May 2009

Efficient Computational Fluid Dynamics Evaluation of Small-Device Locations with Automatic Local Remeshing

Yasushi Ito*
University of Alabama at Birmingham, Birmingham, Alabama 35294
Mitsuhiro Murayama[†,‡] and Kazuomi Yamamoto[§]
Japan Aerospace Exploration Agency, Tokyo 182-8522, Japan
and
Alan M. Shih[¶] and Bharat K. Soni[**]
University of Alabama at Birmingham, Birmingham, Alabama 35294

DOI: 10.2514/1.40875

This paper describes a new efficient automatic remeshing method for three-dimensional hybrid meshes for viscous flow simulations to accommodate the meshes with changes of small devices quickly and easily. This remeshing method has two notable advantages. First, hybrid meshes can be generated automatically from a baseline mesh when the small devices are moved and/or deformed. This enables shape optimization techniques to find better models quickly, because tens or hundreds of meshes are usually required during the process. Second, the updated hybrid meshes are the same as the baseline mesh except for elements around the small devices. Their effect can be evaluated more accurately. Solution data from the baseline mesh can be shared with new hybrid meshes (e.g., as an initial condition to expedite the convergence of computational simulations). The remeshing method is applied to the Japan Aerospace Exploration Agency's high-lift-configuration standard model with a nacelle chine in different locations to demonstrate its capability. Computational simulations are also performed to further discuss the effectiveness of the remeshing method.

I. Introduction

COMPUTATIONAL fluid dynamics (CFD) has been widely used for engineering design purposes to optimize object shapes efficiently. Many geometrical models with small changes often need to be evaluated during the optimization process. The small changes can be the small geometrical change of components or the relocation of small devices, such as vortex generators, pitot tubes, and winglets. The Japan Aerospace Exploration Agency (JAXA) has been working on the research and development of civil transport aircraft [1–7]. The Civil Transport Team of JAXA has conducted computational and experimental research to develop design technologies for advanced high-lift devices. One of their interests is to find the best locations of nacelle chines (Fig. 1) [6,7] or other small devices to efficiently control the flows around aircraft models in takeoff and landing phases.

Because mesh generation is a crucial step for CFD simulations, meshes need to be generated in a reasonable amount of time for successful design optimization when updated models are produced with reshaped and/or relocated small devices. Cartesian-based meshes can accept those changes quickly because new meshes can be created in a short period of time [8–10]. However, tetrahedral and hybrid meshes still have an advantage in high-Reynolds-number viscous flow simulations for complex geometries over the Cartesian-based meshes if currently available computational resources and the maturity of computational algorithms are taken into account. Widely used unstructured mesh generation tools, however, have a limited capability of modifying an existing volume mesh when the shape of the geometry is changed even slightly. The most difficult steps are local surface remeshing around the small devices and local near-field volume remeshing if anisotropic elements are placed on no-slip walls. Volume meshes for updated configurations often have to be created from the very beginning of the process, from the modification of the original CAD model to the surface and volume mesh generation. It takes time to create a high-quality volume mesh for each of the many slightly changed models, especially when the mesh has more than millions of nodes. To minimize the turnaround time for the mesh generation, certain information on the baseline model must be kept.

Several approaches can be considered to create meshes quickly for models consisting of *a baseline geometry* and *small add-on devices* that are moved and reshaped during the optimization process. If the difference between the models is big but they are topologically the same, a topology matching method may be a good approach to keep the quality of meshes for the different models [11]. Although user interventions can be greatly reduced, CPU time required for the generation of each mesh is almost the same. This can be a big problem, because large meshes are usually required for practical CFD applications.

If the difference is small, moving-mesh methods can be used [12]. They can be applied to hybrid meshes while maintaining the quality of semistructured elements on no-slip walls. However, poor-quality or even negative-volume elements can be created near the small devices being moved or deformed if their displacement becomes large. A new mesh sometimes needs to be generated from scratch to overcome this issue.

Unstructured overset (or chimera) grid methods are widely used for moving-body problems, because complex geometries can be

Presented as Paper 7180 at the 26th AIAA Applied Aerodynamics Conference, Honolulu, HI, 18–21 August 2008; received 8 September 2008; revision received 26 December 2008; accepted for publication 12 January 2009.

*Research Assistant Professor, Department of Mechanical Engineering, 1530 Third Avenue South, Building BEC 356B. Senior Member AIAA.

[†]Researcher, Civil Transport Team, Aviation Program Group, 7-44-1 Jindaiji-Higashi, Chofu. Senior Member AIAA.

[‡]Currently Visiting Engineer, Aerodynamic Design Team, Aerodynamics and Performance Group, Engineering Department, Mitsubishi Aircraft Corporation, 2-15 Oye-cho, Minato-ku, Nagoya 455-8555, Japan.

[§]Senior Researcher, Civil Transport Team, Aviation Program Group, 7-44-1 Jindaiji-Higashi, Chofu. Member AIAA.

[¶]Research Professor, Department of Mechanical Engineering, 1530 Third Avenue South, Building BEC 356C. Associate Fellow AIAA.

**Professor and Chair, Department of Mechanical Engineering, 1530 Third Avenue South, Building BEC 257. Associate Fellow AIAA.

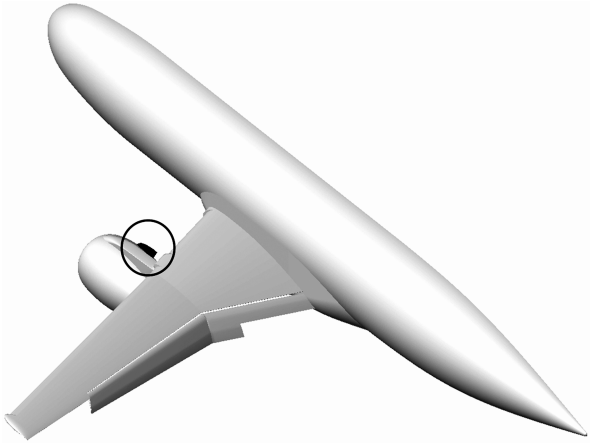


Fig. 1 JAXA JSM with a nacelle chine (circled).

represented easily with a set of meshes [13]. Three steps are usually required: 1) mesh generation only once for the baseline and add-on geometries separately, 2) hole cutting to identify nodes to be used, and 3) CFD simulations while solution data are interpolated at interface boundaries. Steps 2 and 3 are repeated when the meshes are relatively moved. Although mesh generation can be very simple, two issues need to be resolved. First, the add-on devices can be moved, but are not allowed to be deformed. The mesh around the add-on devices needs to be replaced, or a moving-mesh method is required to deform them automatically. Much more implementation efforts are needed, and the resulting approach will not accept large deformations because of the limitation of the moving-mesh method. Second, mesh generation for overset grids may not be so simple. In general, the sizes of elements in the overlapping volume of the overset grids should not be changed drastically, to avoid poor interpolation results.

Automatic local remeshing only around the small add-on devices looks attractive and can save a lot of time when they are inserted to a volume mesh for the baseline geometry. Because elements are always updated around the small add-on devices, the quality of the elements is easily controlled. No interpolations are needed during a computational simulation, because a single volume mesh is always created for each of the updated models. There are two challenges for the local remeshing. First, the small devices intersect with the baseline geometry. The surface around the intersection must be reconstructed and remeshed. Second, we are interested in three-dimensional (3-D) hybrid meshes. The remeshing is not very difficult if isotropic tetrahedral meshes are used for computational simulations. For high-Reynolds-number viscous flow simulations, stretched or anisotropic elements are needed on no-slip walls. If hybrid meshes are considered, semistructured elements must be placed on the inserted small devices first, so that the connectivity and quality of elements are well maintained at the interface between remeshing and nonremeshing domains.

In this paper, a novel automatic local remeshing method is proposed to accommodate hybrid meshes with changes of small devices. It is applied to the JAXA Standard Model (JSM), a high-lift configuration aircraft model, with a nacelle chine in different locations. CFD simulations are carried out to validate the quality of resulting meshes. The rest of the paper is organized as follows. Section II describes mesh generation methods for the generation of baseline hybrid meshes and local remeshing. Section III describes an unstructured flow solver to be used. In Sec. IV, the local remeshing method is applied to the JSM to demonstrate its capability, and computational results are discussed. Finally, conclusions are given in Sec. V.

II. Mesh Generation

The automatic local remeshing method proposed in this paper consists of four steps. First, a hybrid mesh is generated for a baseline geometry without *an add-on geometry* using a normal mesh generation method (*baseline hybrid mesh*). Second, the add-on geometry is added to the baseline hybrid mesh and is trimmed against the surface to which it is to be attached (Figs. 2a and 2b). Third, remeshing and nonremeshing domains are identified by removing elements of the baseline hybrid mesh around the add-on geometry (Fig. 2b). Fourth, local surface and volume remeshing is performed in the remeshing domain to create an updated hybrid mesh (Fig. 2c). The local remeshing method allows modifying elements only around the add-on geometry. There is no need to generate the entire volume mesh from scratch once the baseline hybrid mesh is obtained.

The input files required for the local remeshing method are as follows: 1) baseline hybrid mesh (Sec. II.A), 2) add-on geometry consisting of a number of boundary surfaces, 3) original geometry of the baseline hybrid mesh as a discrete surface (recommended, but optional), 4) node distribution for local surface meshing (Sec. II.D), and 5) parameters for local hybrid remeshing [14].

A. Generation of a Baseline Hybrid Mesh

Although the generation of a baseline hybrid mesh can be done using any existing mesh generator, our mesh generation methods are introduced here. They are available as the Mixed-Element Grid Generator in 3-D (MEGG3D) [15]. To generate surface meshes for CAD-based models, a direct advancing-front method is used [16,17]. A stereolithography (STL) file is used as a background mesh, on which a new surface mesh suitable for computational simulations is created. Geometrical features (ridges) are extracted based on a folding angle at each edge. A user specifies node distributions on the ridges, which form an initial front for the advancing-front method. The surface triangulation is performed in the physical 3-D space so that the quality of triangles can be evaluated easily.

Reverse-engineering approaches are also common to generate 3-D models. Commercial 3-D laser scanners can produce triangulated surfaces as output. Moreover, triangulated surfaces can be extracted from image data. The quality of these triangulated surfaces, however, is usually poor and cannot be used as computational meshes. A mesh-decimation method combined with a node-smoothing method and an edge-swapping method based on the Delaunay property enables high-quality surface mesh generation semi-automatically [18]. To

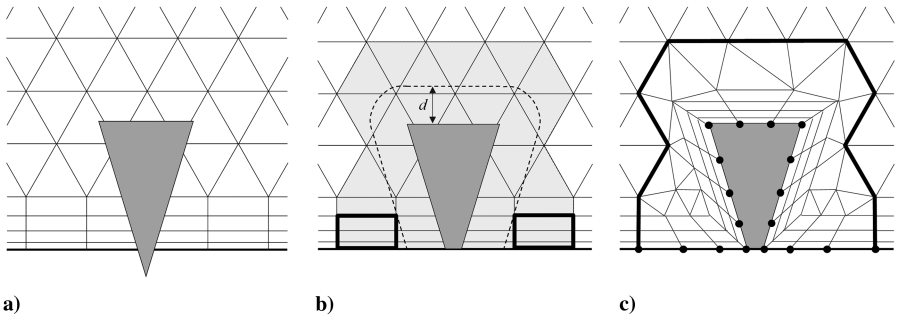


Fig. 2 Local remeshing process in 2-D: a) baseline hybrid mesh (background) and an add-on geometry added (gray), b) trimmed add-on geometry and marked elements to be removed (*d* is smaller than the actual value, to simplify the illustration), and c) remeshing on the boundary surface and in the remeshing domain.

control the mesh density, local surface curvature and volume thickness (i.e., the distance between two surfaces) can be considered.

To create hybrid meshes from surface meshes, a multiple-marching-direction approach is used for the near-field mesh generation to improve the quality of elements around sharp convex corners [14,19]. For the far-field mesh generation, an advancing-front method is used to easily control the mesh density around the boundary surfaces. The following six boundary surface types are provided for hybrid mesh generation:

1) A no-slip wall is a surface on which semistructured elements are created.

2) An adjacent surface is a flat surface that is not required to be advanced, but that is adjacent to no-slip walls (e.g., inlets, outlets, symmetry planes, and periodic boundaries). When semistructured elements are placed on the neighboring no-slip walls, quadrilaterals are created to maintain the connectivity.

3) A fixed adjacent surface is like the adjacent surface, but no modification is allowed during the hybrid mesh generation. This is a new boundary surface type for the interface between remeshing and nonremeshing domains (e.g., the bold line in Fig. 2c).

4) A fixed surface is a surface on which tetrahedra are created, such as outer boundaries.

5) The inner surface for tetrahedra is a surface embedded in the meshing domain to control internal mesh density. Tetrahedra are created around the surface, which can be automatically trimmed to avoid creating low-quality or overlapping elements between the end of the near-field mesh generation and the beginning of the far-field mesh generation [20].

6) The inner surface for semistructured elements is also an embedded surface, but semistructured elements are created around it [20].

B. Addition and Trimming of the Add-on Geometry

The add-on geometry is prepared as a discrete surface, such as an STL file from a CAD system (input file B). Note that the add-on geometry is not supposed to be trimmed at the beginning, for user-friendliness (Fig. 2a). A Boolean operation is applied to the intersection between the add-on geometry and the surface mesh of the baseline hybrid mesh to get a two-manifold surface [17].

C. Definition of Remeshing and Nonremeshing Domains

Elements of the baseline hybrid mesh close to or inside the add-on geometry are removed so that remeshing and nonremeshing domains can be defined. An element of the baseline hybrid mesh is considered to be close to the add-on geometry if $d < \alpha h_{\text{tot}}$, where d is the shortest distance from the add-on geometry at one of the nodes of the element, α is a user-specified parameter (default value is 2), and h_{tot} is the estimated height of the near-field mesh to be created on no-slip walls (Fig. 2b). If any of the semistructured elements that are extruded from the same face (triangle or quadrilateral) on the boundary surface are removed, all of the elements are removed to make the near-field remeshing process easier (e.g., quadrilaterals enclosed by bold lines in Fig. 2b).

In the actual implementation, elements cannot be removed based only on the distance. For example, suppose the add-on geometry is added on a thin wing. If only the distance is considered, elements on the opposite side of the wing can be removed accidentally. Therefore, three steps are required to make the element removal process robust.

1) Check each edge of the baseline hybrid mesh close to the add-on geometry to see if it intersects with the add-on geometry. If so, mark the two nodes of the edge to be removed.

2) If an element of the baseline hybrid mesh has any marked nodes, remove it. Also, mark each of nonmarked nodes of the element if $d < \alpha h_{\text{tot}}$. Continue this step until no elements are removed.

3) Remove elements inside the add-on geometry.

Table 1 Example of node distribution: input file D contains this information, except the ridge indices in the first column.

Ridges	Neighboring zones		Additional information	No. of nodes
A	0	1	None	20
B	1	3	$x1$	15
C	1	2	None	30
D	2	3	None	15
E	1	3	$x2$	8
F	0	3	None	60

D. Local Remeshing of the Baseline Hybrid Mesh

Surface remeshing is performed first on the trimmed add-on geometry and the boundary surface of the baseline hybrid mesh for which the neighboring elements are removed in Sec. II.C. The direct advancing-front method is used for this purpose [16,17]. It needs a background mesh, on which a new surface mesh is created. In Sec. II.B, the Boolean operation is applied to the add-on geometry (surface A) and the surface mesh of the baseline hybrid mesh (surface B) to get a new triangulated surface. It is used as the background mesh. Nodes of the new surface mesh on surface B are projected to the original geometry of the baseline hybrid mesh (input file C) if provided.

A user specifies the mesh density of the new surface mesh as the number of nodes on each ridge (input file D). Ridges are created automatically as the borders of boundary surfaces (zones) of the background mesh; Fig. 3 shows an example in which there are three zones (1 to 3), and six ridges (A to F) are created. Because it is difficult for the user to call those automatically created ridges by their indices, each of the ridges is identified mainly by its two neighboring zones, as shown in Table 1. Ridge A is identified by zones 0 and 1. Zone 0 means that there is no neighboring boundary surface of the add-on geometry, but that the surface mesh of the baseline hybrid mesh will be connected. Ridges B and E have the same two neighboring zones (1 and 3). In this case, the user needs to specify additional information. The ridges can be numbered in ascending order, either in the x, y, or z direction. The additional information for ridges B and E is $x1$ and $x2$, respectively. Figure 3b shows nodes on the ridges distributed based on the specification in Table 1. To distribute nodes smoothly, the node spacings at each of intersections P to S are automatically corrected based on the smallest spacing. For example, at intersections P and S, the smallest spacings are the node spacings on ridge B and on ridge F, respectively. The

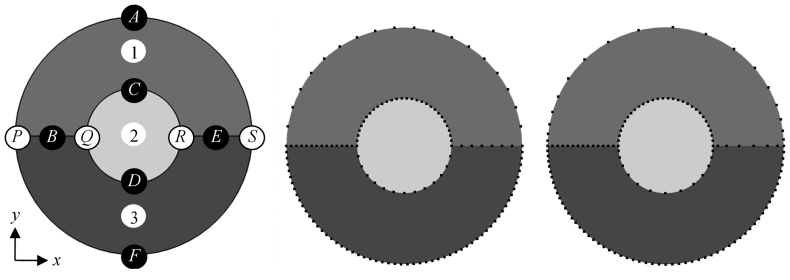


Fig. 3 Node distribution on ridges in 2-D: a) zones 1 to 3 and ridges A to F intersecting at points P to S, b) node distribution based on the number of nodes on each ridge in Table 1, and c) corrected distribution based on 1-D stretching functions.

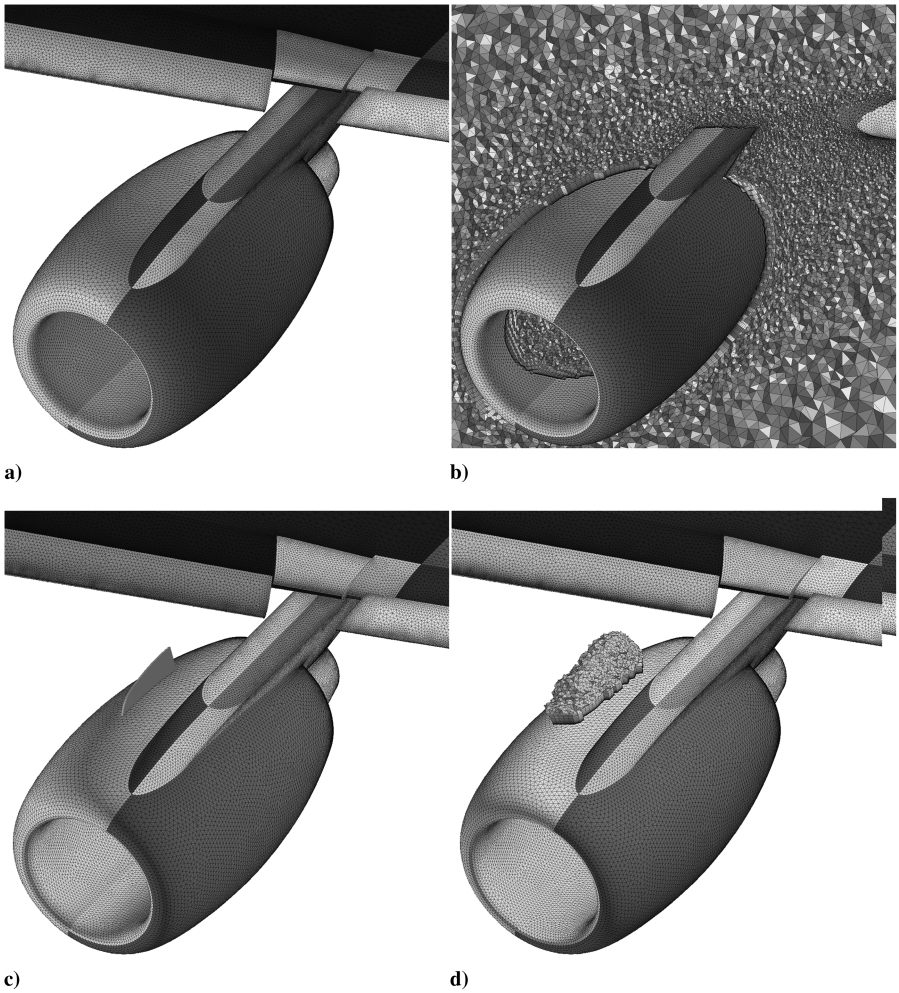


Fig. 4 Engine nacelle of the JAXA JSM: a) surface mesh for the baseline geometry, b) cross section of the baseline volume mesh, c) chine added (position A), and d) remeshing domain around the chine.

internal node distribution of each ridge is determined based on node spacings at its two ends by 1-D stretching functions [21]. Figure 3c shows the final node distributions on the ridges. This approach enables the user to easily and robustly specify smooth node distributions on the ridges, even if their lengths are changed as the result of the trimming in Sec. II.B.

The next step is local volume remeshing. A closed surface mesh enclosing the remeshing domain is defined by the new surface mesh

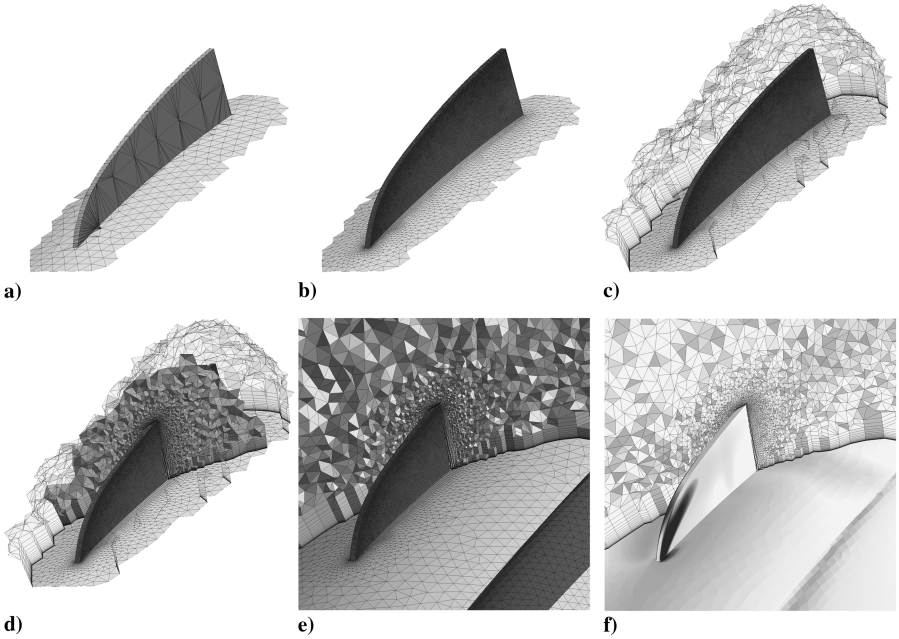
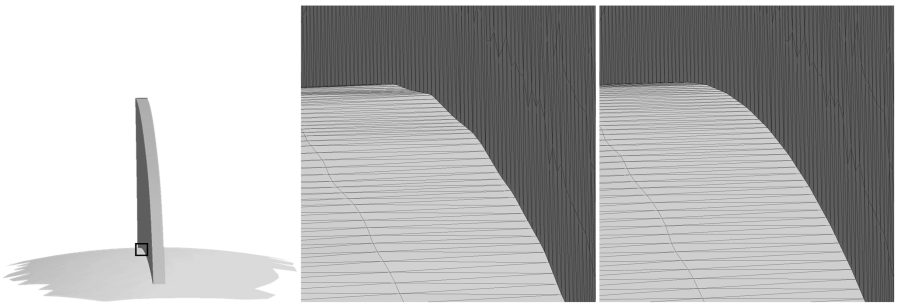


Fig. 5 Remeshing domain around the nacelle chine: a) surface remeshing area, b) new local surface mesh, c) remeshing domain represented as a closed surface, d) new local volume mesh and a cross section, e) new volume mesh, and f) pressure distribution.



a) b) c)

Fig. 6 Projection of a local surface mesh to the original geometry: a) surface remeshing area and a square to be enlarged, b) before the projection, and c) after the projection.

on the no-slip or slip walls and the interface between the remeshing and nonremeshing domains. If the domain contains no-slip walls, semistructured elements must be placed on them first [14]. The boundary surface type for the interface is set to the fixed adjacent surface automatically. Semistructured elements created in the remeshing domain are aligned with the interface so that the connectivity of elements in the remeshing and nonremeshing domains is well maintained. The rest of the remeshing domain is filled with tetrahedral elements. Basically, the mesh generation parameters specified in input file E are the same as those used for the generation of the baseline hybrid mesh.

III. Flow Solver

TAS (Tohoku University Aerodynamic Simulation) code [22,23] is used for computational simulations. The Navier–Stokes equations are solved on an unstructured mesh by a cell-vertex finite volume method. The Harten–Lax–van Leer–Einfeldt–Wada method [24] is used for the numerical flux computations. Second-order spatial accuracy is achieved by the unstructured MUSCL scheme [25]. The lower/upper symmetric Gauss–Seidel implicit method [26] is used for time integration.

To simulate turbulent flows, the Spalart–Allmaras one-equation turbulence model [27] with several modifications is used without the trip term for transition and f_{t2} function that suppresses the production of eddy viscosity due to numerical error. The production of eddy viscosity starts with the freestream value. In addition, a variation of the model that reduces the eddy viscosity in the domains of high vorticity is used [28,29]. The production term is modified as follows using the vorticity $\Omega = \sqrt{2\Omega_{ij}\Omega_{ij}}$ and strain rate $\hat{S} = \sqrt{2S_{ij}S_{ij}}$:

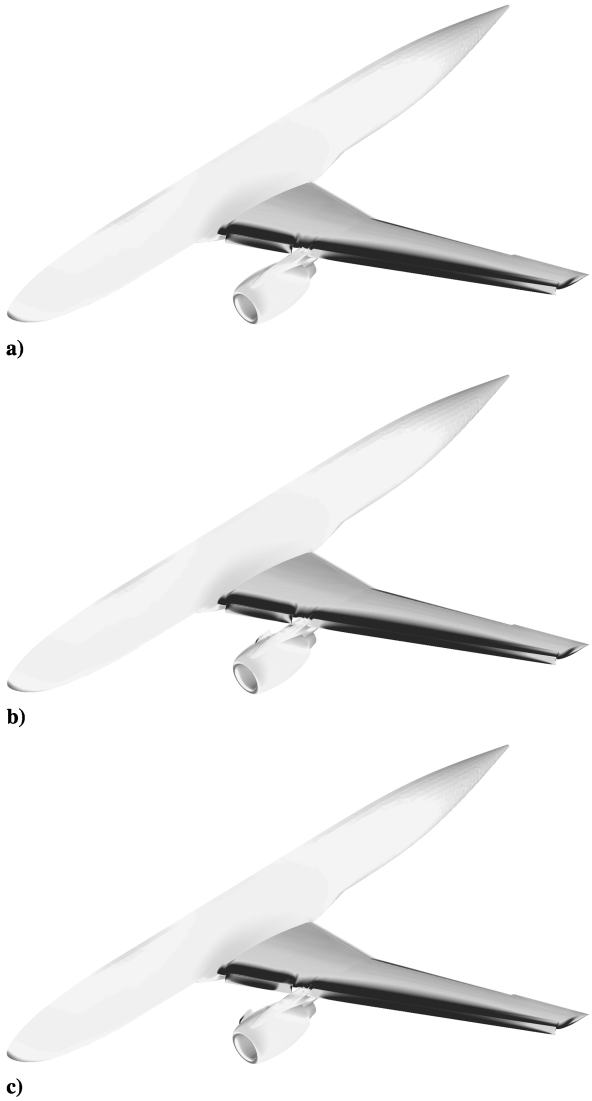
$$S = \Omega + \min(0, \hat{S} - \Omega) \tag{1}$$

The modified model computes turbulent vortical flows without adding much dissipation to vortex cores. If some laminar domains are specified to simulate boundary-layer transition in computations, the magnitude of the vorticity is set to zero in the domains.

IV. Applications

In this section, the automatic local remeshing method is applied to the JAXA JSM (Fig. 1) [1,2]. A chine is moved on the engine nacelle. This is a good example to demonstrate the capability of the local remeshing method, because the baseline geometry is complicated (with a pylon-mounted engine nacelle, leading-edge slats, and trailing-edge flaps) and the add-on geometry is small. Figures 4a and 4b show surface and volume meshes for the baseline geometry around the engine nacelle. The baseline hybrid mesh has 8.69 million nodes. A nacelle chine shown in Fig. 4c was added to the baseline hybrid mesh (position A). A remeshing domain around the chine was created after elements of the baseline hybrid mesh close to the chine were removed (Fig. 4d). The remeshing process is illustrated in Fig. 5. Figure 5a shows the surface remeshing area. The chine was represented as a triangulated surface from a CAD system. A Boolean operation was applied to the chine (surface A) and part of the surface mesh of the baseline hybrid mesh (surface B) to get a two-manifold surface. A new local surface mesh was then created on the surface remeshing area (Fig. 5b). Because nodes on surface B were initially projected to surface B as the coarse surface mesh (Fig. 5a), the resulting surface was not very accurate, as shown in Fig. 6b. At the end of the surface remeshing process, the nodes were projected back to the original representation of surface B, as shown in Fig. 6c.

A closed surface mesh was defined as the new local surface mesh and the interface between the remeshing and nonremeshing domains



a) b) c)

Fig. 7 Pressure distribution on JSM with the nacelle chine in different locations: a) baseline geometry without the chine, b) chine position A, and c) chine position B.

(Fig. 5c). A local hybrid mesh was created in the closed surface mesh (Figs. 5d and 5e), and an updated hybrid mesh with 9.65 million nodes was obtained. The updated hybrid mesh looks like the one created from scratch without the local remeshing method.

A computational simulation was performed using the new volume mesh at a freestream Mach number of 0.175, a Reynolds number of 2.1×10^6, a temperature of 293 K, and an angle of attack of 16 deg. Solution data from the baseline hybrid mesh (Figs. 7a and 8a) were used as the initial conditions to expedite the convergence of the simulation. Figures 5f, 7b, and 8b show a pressure distribution on the JSM with the chine in position A in the range of 0.92 to 1.03. Note that all pressure distributions in this paper have the same range. No big difference was observed between the computational result from the hybrid mesh using the local remeshing method and that of a hybrid mesh with 9.83 million nodes created from scratch (Fig. 9). Let us compare the pressure distribution on the inboard slat shown in Figs. 8b and 9b. The vortex created at the chine is captured slightly better using the mesh-from-scratch version, because it has more nodes near the chine (cf. the cross sections shown in Figs. 5f and 9a). If this difference is critical to optimize the chine location, then a larger α can be specified to make the remeshing domain larger. As a result, the size of elements around the chine is changed more smoothly. However, it seems that solution adaptive mesh generation is required, even for the mesh-from-scratch version, to precisely understand the details of the flow around the chine.

The local remeshing method is robust as long as input files A to E are correct. It was also applied to the same baseline hybrid mesh with the same chine in a different location (position B) without any problems. The resulting hybrid mesh has 9.72 million nodes. Figures 7c and 8c show a pressure distribution on the JSM with the chine in position B.

The biggest advantage of the local remeshing method is turnaround time. The turnaround time for creating the baseline hybrid mesh with 8.69 million nodes was approximately one day from a CAD model, including the user operation time and computational time. If the local remeshing method is not available, one day is needed every time the chine is moved or reshaped. With the local remeshing method, each of the updated hybrid meshes for chine positions A and B was created in 30 min, *fully automatically* from the baseline hybrid mesh. The size of the add-on geometry is an important factor to reduce the turnaround time, because it directly relates to the size of the remeshing domain. The local remeshing method is very powerful if the add-on geometry is small compared with the baseline geometry, as in this case.

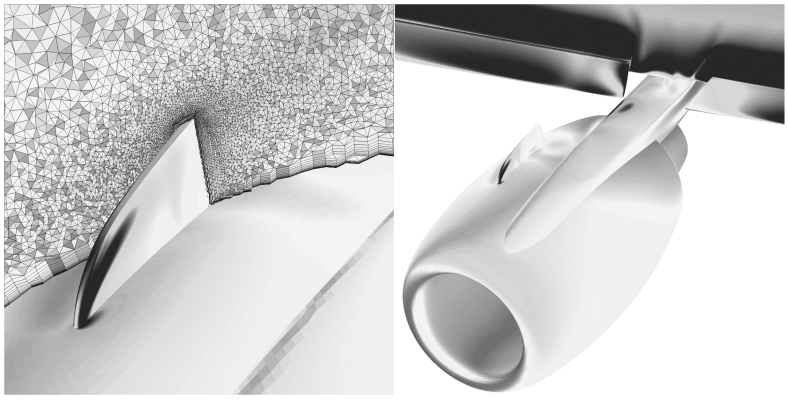
Another advantage of the local remeshing method is that all the elements and nodes except those around the add-on geometry are never changed during the remeshing process. The effect of the add-on geometry can be evaluated more accurately with the local remeshing method than with a standard mesh generation method that creates meshes from scratch. Figure 7 also shows that only the flow around the chine is changed. Solution data can be easily interpolated from the baseline hybrid mesh if needed.

V. Conclusions

An efficient automatic local remeshing method for three-dimensional hybrid meshes is proposed. A hybrid mesh is prepared for a baseline geometry without small add-on devices to be moved or reshaped. The small devices are inserted to the baseline hybrid mesh, and then elements around those are automatically updated. The local remeshing method has two notable advantages. First, turnaround time for creating a new hybrid mesh for each configuration can be greatly reduced. This is especially true if the baseline geometry is very complex and the add-on devices are small. Second, updated hybrid meshes are the same as the baseline hybrid mesh, except for elements around the small add-on devices. The effect of the add-on devices can be evaluated more accurately with the local remeshing method than with a standard mesh generation method that creates meshes from scratch. The local remeshing method was applied to the JAXA JSM, a high-lift configuration aircraft model, with a nacelle

a) b) c)

Fig. 8 Pressure distribution around the nacelle: a) baseline geometry without the chine, b) chine position A, and c) chine position B.



a) b)

Fig. 9 Hybrid mesh created from scratch for the baseline geometry with the chine in position A (pressure distribution): a) chine and a cross section (cf. Fig. 5f) and b) nacelle (cf. Fig. 8b).

chine in different locations. A dramatic reduction in turnaround time for creating hybrid meshes was observed. Computational simulations were performed without any problems using the resulting meshes.

References

[1] Ito, T., Ura, H., Yokokawa, Y., Kato, H., Mitsuo, K., and Yamamoto, K., "High-Lift Device Testing in JAXA 6.5 m × 5.5 m Low-Speed Wind Tunnel," AIAA Paper 2006-3643, 2006.

[2] Yokokawa, Y., Murayama, M., Ito, T., and Yamamoto, K., "Experimental and CFD of a High-Lift Configuration Civil Transport Aircraft Model," AIAA Paper 2006-3452, 2006.

[3] Yokokawa, Y., Murayama, M., Kanazaki, M., Murota, K., Ito, T., and Yamamoto, K, "Investigation and Improvement of High-Lift Aerodynamic Performances in Low-Speed Wind Tunnel Testing," AIAA Paper 2008-350, 2007.

[4] Murayama, M., Yokokawa, Y., Yamamoto, K., and Ueda, Y., "CFD Validation Study for a High-Lift Configuration of a Civil Aircraft Model," AIAA Paper 2007-3924, 2007.

[5] Murayama, M., Yokokawa, Y., Tanaka, K., Yamamoto, K., and Ito, T., "Numerical Simulation of Half-span Aircraft Model with High-Lift Devices in Wind Tunnel," AIAA Paper 2008-0333, 2008.

[6] Kanazaki, M., Yokokawa, Y., Murayama, M., and Ito, T., "Efficient Design Exploration of Nacelle Chine Installation in Wind Tunnel Testing," AIAA Paper 2008-0155, 2008.

[7] Kato, H., Watanabe, S., Murayama, M., Yokokawa, Y., and Ito, T., "PIV Investigation of Nacelle Chine Effects on High-Lift System Performance," AIAA Paper 2008-240, 2008.

[8] Ishida, T., Takahashi, S., and Nakahashi, K., "Fast Cartesian Mesh Generation for Building-Cube Method using Multi-Core PC," AIAA Paper 2008-0919, 2008.

[9] Takahashi, S., Ishida, T., and Nakahashi, K., "Dynamic Load Balancing for Flow Simulation Using Adaptive Refinement," AIAA Paper 2008-0920, 2008.

[10] Dawes, W., Favaretto, C., Harvey, S., Fellows, S., and Richardson, G., "Towards Topology-Free Optimization–An Application to Turbine Internal Cooling Geometries," AIAA Paper 2008-0925, 2008.

[11] Ross, D. H., Dorothy, F. W., Ito, Y., Shih, A. M., and Soni, B. K., "Mesh Generation Transfer Based on Topology Matching," 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2008-0926, Jan. 2008.

[12] Murayama, M., Ito, Y., Nakahashi, K., Matsushima, K., and Iwamiya, T., "Viscous Flow Computations of Aircraft with Changing Control Surface Deflection Using Unstructured Dynamic Meshes," *International Journal for Numerical Methods in Fluids*, Vol. 52, No. 8, 2006, pp. 925–940.
doi:10.1002/fld.1212

[13] Togashi, F., Ito, Y., Nakahashi, K., and Obayashi, S., "Overset Unstructured Grids Method for Viscous Flow Computations," *AIAA Journal*, Vol. 44, No. 7, 2006, pp. 1617–1623.
doi:10.2514/1.4292

[14] Ito, Y., Shih, A. M., Soni, B. K., and Nakahashi, K., "Multiple Marching Direction Approach to Generate High Quality Hybrid Meshes," *AIAA Journal*, Vol. 45, No. 1, 2007, pp. 162–167.
doi:10.2514/1.23260

[15] Ito, Y., Shih, A. M., and Soni, B. K., "Unstructured Mesh Generation Using MEGG3D—Mixed-Element Grid Generator in Three Dimensions," *Proceedings of the International Conference on Numerical Geometry, Grid Generation and Scientific Computing (NUMGRID2008)*, Moscow, Russia, 2008, pp. 5–11.

[16] Ito, Y., and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490–496.
doi:10.2514/2.1672

[17] Ito, Y., and Nakahashi, K., "Surface Triangulation for Polygonal Models Based on CAD Data," *International Journal for Numerical Methods in Fluids*, Vol. 39, No. 1, 2002, pp. 75–96.
doi:10.1002/fld.281

[18] Ito, Y., Shum, P. C., Shih, A. M., Soni, B. K., and Nakahashi, K., "Robust Generation of High-Quality Unstructured Meshes on Realistic Biomedical Geometry," *International Journal for Numerical Methods in Engineering*, Vol. 65, No. 6 , 2006, pp. 943–973.
doi:10.1002/nme.1482

[19] Ito, Y., and Nakahashi, K., "Improvements in the Reliability and Quality of Unstructured Hybrid Mesh Generation," *International Journal for Numerical Methods in Fluids*, Vol. 45, No. 1, 2004, pp. 79–108.
doi:10.1002/fld.669

[20] Ito, Y., Shih, A. M., and Soni, B. K., "Hybrid Mesh Generation with Embedded Surfaces Using a Multiple Marching Direction Approach," *International Journal for Numerical Methods in Fluids*, DOI: 10.1002/fld.1962, in press.

[21] Vinokur, M., "On One-Dimensional Stretching Functions for Finite-Difference Calculations," *Journal of Computational Physics*, Vol. 50, No. 2, 1983, pp. 215–234.
doi:10.1016/0021-9991(83)90065-7

[22] Nakahashi, K., Togashi, F., Fujita, T., and Ito, Y., "Numerical Simulations on Separation of Scaled Supersonic Experimental Airplane from Rocket Booster at Supersonic Speed," AIAA Paper 2002-2843, June 2002.

[23] Murayama, M., and Yamamoto, K., "Comparison Study of Drag Prediction for the 3rd CFD Drag Prediction Workshop by Structured and Unstructured Mesh Method," AIAA Paper 2007-0258, 2007.

[24] Obayashi, S., and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier–Stokes Solver," *AIAA Journal*, Vol. 33, No. 6, 1995, pp. 1134–1141.
doi:10.2514/3.12533

[25] Burg, C., "Higher Order Variable Extrapolation for Unstructured Finite Volume RANS Flow Solvers," AIAA Paper 2005-4999, 2005.

[26] Sharov, D., and Nakahashi, K., "Reordering of Hybrid Unstructured Grids for Lower-Upper Symmetric Gauss-Seidel Computations," *AIAA Journal*, Vol. 36, No. 3, 1998, pp. 484–486.
doi:10.2514/2.392

[27] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, 1992.

[28] Dacles-Mariani, J., Zilliac, G. G., Chow, J. S., and Bradshaw, P., "Numerical/Experimental Study of a Wingtip Vortex in the Near Field," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1561–1568.
doi:10.2514/3.12826

[29] Lei, Z., "Effect of RANS Turbulence Models on Computational of Separated Flows over a Wing-Body Configuration," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 48, No. 161, 2005, pp. 152–160.
doi:10.2322/tjsass.48.152

Z. Wang
Associate Editor