AIAA JOURNAL
Vol. 44, No. 7, July 2006

Overset Unstructured Grids Method for Viscous Flow Computations

Fumiya Togashi,* Yasushi Ito,† Kazuhiro Nakahashi,‡ and Shigeru Obayashi§
Tohoku University, Sendai 980-8579-01, Japan

The overset unstructured grid method is extended to viscous flow computations of multiple bodies in contact. Overset treatment of hybrid meshes composed of tetrahedral, prismatic, and pyramidal elements is discussed in detail when those elements are embedded in or in contact with the wall boundary of another grid. This method is applied to the ONERA M5 wing–body model and computed by the Navier–Stokes code. The computational results show good agreement with those obtained by a single grid and the experimental data. The boundary layer in the intersection region between two grids is well captured without a collar grid.

Introduction

THE overset-grid (Chimera) approach, in which several structured grids overlap and cover the flowfield to handle complicated configurations, was introduced by Steger et al.[1] and Benek et al.[2] Currently, this method is effectively used to solve both steady and unsteady viscous flow around complex geometries[3–5] and for moving-body problems.[6,7] This method is quite powerful for dealing with the moving bodies because remeshing is not required. Recently, owing to the progress in the field of computer technology, more detailed and complicated configurations can be treated. As the complexity of the geometry increases, however, the required number of grids that overlap increases. This causes difficulty in constructing the information necessary for communication among grids and requires more time-consuming work. These difficulties may have hindered wide applications of the method in industry.

The overset unstructured grid method has been proposed and successfully applied to several problems.[8–11] The approach has been also employed in other studies.[12,13] In those studies, the method showed good performance when used to solve complicated moving body problems. The unstructured grid has a great flexibility in handling complex geometries. When unstructured meshes are used for the overset concept, the number of submeshes required for covering the flowfield can be significantly reduced as compared with that needed with the overset structured grids. As a result, generating interpolation stencils between grids becomes simple. It can also extend the applicability of the unstructured grid method to multiple moving-body problems without serious effort in the code development for a deforming mesh and remeshing procedure.

The overset unstructured grid method can deal with multiple bodies in contact by a simple procedure.[11] In Ref. 11, the ONERA M5 body and wing case was used to validate the method by comparison of the overset-grid case and the single-grid case. This approach also shows the capability of the store separating simulation in which the store is in contact with or embedded in to the body at the beginning. With this method, flows around the National Aerospace Laboratory of Japan jet-powered experimental airplane with a small rocket booster under the fuselage have been computed by solving the Euler equations (see Ref. 14). In the cited study, the method also showed potential for application to aircraft design. For example, to find the optimum location for a nacelle–pylon on a wing, flow simulation around a complete airplane must be performed by changing the positions of the nacelle. This necessitates a troublesome procedure to generate a grid for each different position of the nacelle if a single grid is used to cover the entire configuration. Instead, this procedure can be simplified if overset grids are used. For this case, two grids, one for the wing–fuselage and the other for the nacelle–pylon, cover the entire flowfield, and a parametric study of the nacelle location for optimization can be performed easily and efficiently in a single flow computation without remeshing.

In this paper, an extension of the overset unstructured grid method to treat hybrid unstructured meshes for viscous flow computations is discussed. With such a problem, dealing with a thin boundary layer around the intersection region is very important. To deal with the problem, the overset structured mesh method often uses collar grids[15] for proper resolution of the intersection region and appropriate definition of the geometric surface with respect to both meshes (Fig. 1), especially when the thin-layer approximation is used to reduce the viscous terms to the direction normal to the body. However, overset unstructured meshes are capable of dealing with multiple bodies in contact and facilitate mesh refinement. These advantages can overcome such issues without use of a collar grid. In the previous code, only all-tetrahedral meshes could be treated because of the difficulty of searching for donor cells among the hybrid elements such as prismatic and pyramidal elements. To avoid the difficulty of modifying the code for the donor cell searching through the hybrid meshes, these hybrid elements are divided into tetrahedral elements. To evaluate the method in terms of its accuracy and capability, especially to validate the accuracy to deal with high Reynolds number viscous flows around the intersectional region, the ONERA M5 wing–fuselage configuration is considered as a test case of this approach because this configuration is a familiar example and the experimental data are available.[16]

Overset Procedure

Tetrahedral Elements

The identification of the intergrid boundary for multiple-body problems must be performed completely automatically to take fully advantage of the overset unstructured grid approach. The efficiency and robustness of the hole-cutting procedure are particularly important for moving-body problems. Here, the wall distance is used as a basic parameter to construct the intergrid boundary. Node points that are closer to the wall boundary of their own grid are defined as the computational field.[8,9]

Presented as Paper 2003-3405 at the AIAA 21st Applied Aerodynamics Conference, Orlando, FL, 23–26 June 2003; received 1 August 2003; revision received 16 December 2004; accepted for publication 12 January 2006.

*Graduate Student, Department of Aerospace Engineering, Aobaku Aramaki Aza Aoba 01; togashi@ad.mech.tohoku.ac.jp.

†Graduate Student, Department of Aerospace Engineering, Aobaku Aramaki Aza Aoba 01.

‡Professor, Department of Aerospace Engineering, Aobaku Aramaki Aza Aoba 01. Associate Fellow AIAA.

§Professor, Department of Aerospace Engineering, Institute of Fluid Science, Aobaku Aramaki Aza Aoba 01. Associate Fellow AIAA.

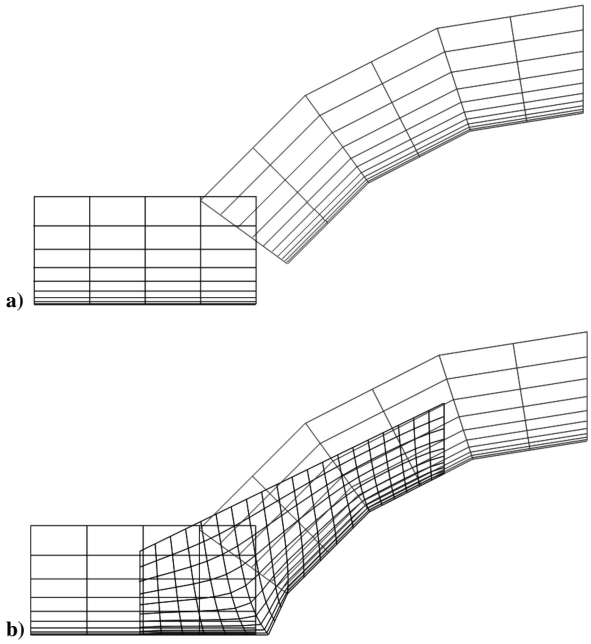

Fig. 1 Collar grid: a) structured meshes after hole cutting around intersection region and b) collar grid filling in intersection region (black mesh).

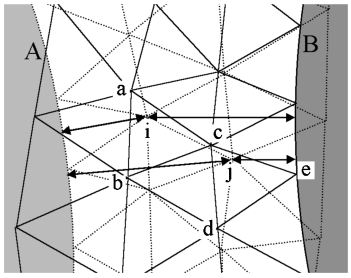

Fig. 2 Determination of intergrid boundary between grid A (dotted lines) and grid B (solid lines).

Consider an example of the hole-cutting procedure. In Fig. 2, suppose that the broken lines show the grid (grid A) generated around body A, and the solid lines show that (grid B) for body B. Node *i* belongs to grid A, and then the donor cell in grid B is indicated by *a–b–c* in Fig. 2. The minimum wall distance of this node *i* to body A is compared with the wall distance of the donor cell to body B. The distance of the donor cell is evaluated by a linear interpolation from its vertex values. Because the wall distance of node *i* to body A is shorter than that of donor cell *a–b–c* to body B, node *i* is assigned as an active node. (It belongs to the computational field.) In contrast, node *j* in Fig. 2 is selected as a nonactive node. In this way, all node points in the overlapping region are assigned as active (computational) or nonactive (noncomputational) nodes. All nonactive node points are interpolated from the partner cells. Because of this interpolation, the computational stability and the computational accuracy around the intergrid boundary are improved. A detailed discussion is presented in the following section. As just mentioned, all nonactive node points are interpolated. However, the cells including the nonactive nodes are not used as donor cells because the partner node point of such a cell must be an active node point. To judge whether the target point is located inside the current cell or not, the volume of the tetrahedron composed of the target point and the three vertexes of the current cell is calculated as shown in Fig. 3. If all of the volumes (where four tetrahedrons are composed of the target and each face of the current cell is calculated) are positive, the target point is located inside the current cell.

Nontetrahedral Elements

Other elements, such as prismatic and pyramidal elements, are divided into tetrahedral meshes (Fig. 4) to perform complete and automatic searching without modification of the searching algorithm.

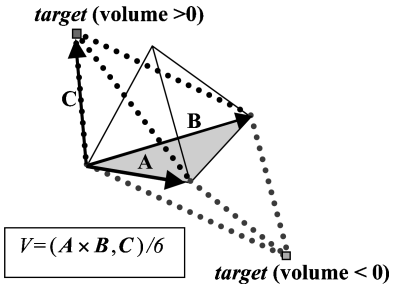

Fig. 3 Donor cell searching.

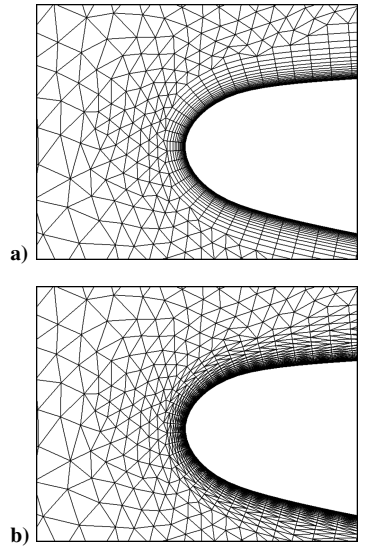

Fig. 4 Unstructured mesh a) including prismatic, pyramidal, and tetraheral cells and b) after dividing prismatic and pyramidal cells into tetrahedral mesh.

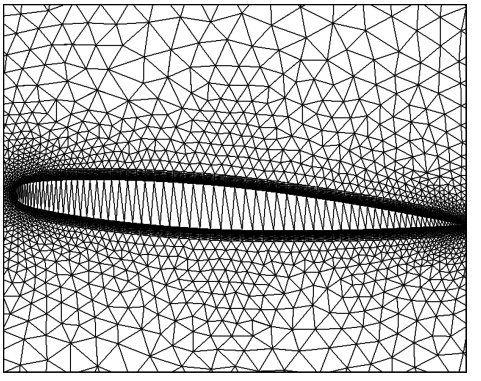

Fig. 5 Unstructured mesh generated inside body for quick and robust searching.

However, the aspect ratio of the divided elements will become very high, especially near the wall boundary. Such a tenuous cell could cause searching failure because too short an edge of the tenuous cell creates an error during the calculation of the tetrahedron, such that the volume of the tetrahedron is zero due to the cancellation of significant digits. To avoid searching failure, the relative position of the searching element and the target node point are normalized. At the volume calculation of tetrahedron, the shortest edge length is assumed as a unit and other edge lengths are multiplied by the normalized ratio. Inner meshes are also generated inside of the body for faster and more accurate searching (Fig. 5). The inner meshes are generated as a byproduct of the Delaunay triangulation method. These generated tetrahedral elements are never used for the flow solver because this treatment causes an increase of the edges and elements and results in heavy computational cost.

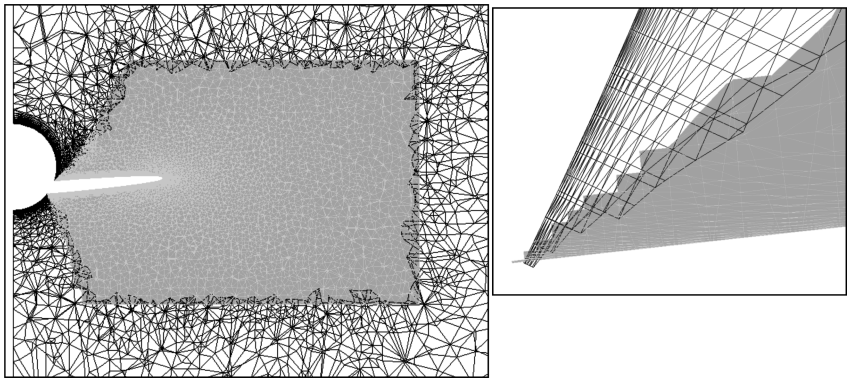

Fig. 6 Intermesh boundary for ONERA M5 wing–fuselage configuration and enlarged view in intersection region.

Intergrid Communications

Dividing the prismatic and pyramidal elements into tetrahedral elements realizes the generation of the intergrid boundary in the intersection region automatically by the use of the same method applied to all tetrahedral meshes[8–11] (Fig. 6). Suppose that the surface and the volume meshes around the boundary layer, comprising the collar grid,[15] are not necessary for the proper resolution of the intersection region and for appropriate definition of the geometric surface with respect to both meshes because of the aforementioned procedures. Mesh refinements (subdivision) can be applied to the intersection region if a more adequate resolution of the intersection region is needed.

The interpolation of the flow variables between the intergrid regions is performed using the volume coordinate method. In this method, the variables are averaged by using the ratio of each volume that is composed of the recipient point and each face of the current cell to the total volume of the current cell. However, the interpolation could also cause an error due to the cancellation of significant digits of the tenuous donor cells. Too short a length of the edge could create an improper donor cell whose volume is zero due to the cancellation of significant digits. To avoid interpolation failure, the relative positions of the donor cell and recipient node point are also normalized when the flow variables are interpolated.

Flow Solver

In the present method, the computational domain consists of several unstructured subgrids that may overlap. The unstructured subgrids generated around each component in the flowfield are put together to discretize the whole computational domain. A hybrid volume grid composed of tetrahedrons, prisms, and pyramids for high Reynolds number viscous flows is generated by the method described in Refs. 17–21. The Navier–Stokes equations are solved in each subgrid with the proper boundary conditions.

The Navier–Stokes equations for compressible viscous flows are written in an integral form, as follows:

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{Q}\,\mathrm{d}V + \int_{\partial\Omega}\left[\boldsymbol{F}(\boldsymbol{Q}) - \frac{M_\infty\sqrt{\gamma}}{Re}\boldsymbol{G}(\boldsymbol{Q})\right]\boldsymbol{n}\,\mathrm{d}S = 0 \qquad (1)$$

where $\boldsymbol{Q} = [\rho, \rho u, \rho v, \rho w, e]^T$ is the vector of conservative variables; ρ is the density; u, v, and w are the velocity components in the x, y and z directions; and e is the total energy. The vector $\boldsymbol{F}(\boldsymbol{Q})$ represents the inviscid flux vector and $\boldsymbol{n}$ is the outward normal of $\partial\Omega$, which is the boundary of the control volume Ω. This system of equations is closed by the perfect gas equation of state.

The equations are solved by a finite volume cell-vertex scheme. The control volume is a nonoverlapping dual cell. For the control volume, Eq. (1) can be written in an algebraic form as follows:

$$\frac{\partial \boldsymbol{Q}_i}{\partial t} = -\frac{1}{V_i}\left[\sum_{j(i)} \Delta S_{ij}\boldsymbol{h}\left(\boldsymbol{Q}_{ij}^{+}, \boldsymbol{Q}_{ij}^{-}, \boldsymbol{n}_{ij}\right) - \sum_{j(i)} \Delta S_{ij} G(Q, n_{ij})\right] \qquad (2)$$

where ΔS_{ij} is the segment area of the control volume boundary associated with the edge connecting points i and j. ΔS_{ij}, as well as its unit normal $\boldsymbol{n}_{ij}$, can be computed by summing up the contribution from each tetrahedron sharing the edge. The term $\boldsymbol{h}$ is an inviscid numerical flux vector normal to the control volume boundary, and $\boldsymbol{Q}_{ij}^{\pm}$ are values on both sides of the control volume boundary. The subscript of summation $\boldsymbol{j(i)}$ represents all node points connected to node $\boldsymbol{i}$.

The numerical flux $\boldsymbol{h}$ is computed using the approximate Riemann solver of Harten–Lax–van Leer–Einfeldt–Wada (see Ref. 22). Second-order spatial accuracy is realized by a linear reconstruction of the primitive gasdynamic variables with Venkatakrishnan's limiter.[23]

The lower–upper symmetric Gauss–Seidel (LU-SGS) implicit method (see Ref. 24) is applied to integrate Eq. (2) in time. With $\Delta\boldsymbol{Q} = \boldsymbol{Q}^{n+1} - \boldsymbol{Q}^n$ and a linealization of the numerical flux term as $\boldsymbol{h}_{ij}^{n+1} = \boldsymbol{h}_{ij}^n + \boldsymbol{A}_i^+\Delta\boldsymbol{Q}_i + \boldsymbol{A}_j^-\Delta\boldsymbol{Q}_j$, the final form of the LU-SGS method on an unstructured grid becomes, for the forward sweep,

$$\Delta\boldsymbol{Q}_i^* = \boldsymbol{D}^{-1}\left[\boldsymbol{R}_i - 0.5\sum_{j\in L(i)} \Delta S_{ij}\left(\Delta\boldsymbol{h}_j^* - \rho_A\Delta\boldsymbol{Q}_j^*\right)\right] \qquad (3a)$$

and for the backward sweep,

$$\Delta\boldsymbol{Q}_i = \Delta\boldsymbol{Q}_i^* - 0.5\boldsymbol{D}^{-1}\sum_{j\in U(i)} \Delta S_{ij}(\Delta\boldsymbol{h}_j - \rho_A\Delta\boldsymbol{Q}_j) \qquad (3b)$$

where

$$\boldsymbol{R}_i = -\sum_{j(i)} \Delta S_{ij}\boldsymbol{h}_{ij}^n$$

$\Delta\boldsymbol{h} = \boldsymbol{h}(\boldsymbol{Q} + \Delta\boldsymbol{Q}) - \boldsymbol{h}(\boldsymbol{Q})$, and $\boldsymbol{D}$ is a diagonal matrix derived by Jameson–Turkel approximation of Jacobian[25] as $\boldsymbol{A}^{\pm} = 0.5(\boldsymbol{A} \pm \rho_A\boldsymbol{I})$, where ρ_A is the spectral radius of Jacobian $\boldsymbol{A}$. $\boldsymbol{D}$ is given as follows:

$$\boldsymbol{D} = \left(\frac{V_j}{\Delta t} + 0.5\sum_{j(i)} \Delta S_{ij}\rho_A\right)\boldsymbol{I} \qquad (4)$$

The lower/upper splitting of Eq. (3), namely, $j \in L(i)$ and $j \in U(i)$, for the unstructured grid is realized by using a grid reordering technique[24] to improve the convergence and the vectorization.

A one-equation turbulence model by Goldberg and Ramakrishnan[26] is implemented to treat turbulent boundary layers for computations of viscous flows.

Overset Implementation

In addition to the boundaries of the computational domain, subgrids may have holes and intergrid boundaries with the neighboring donor subgrids. The node points belonging to the noncomputational field must be excluded or blanked out of the flowfield solution. To decide whether the node points should be computed in the flow

solver, all node points have information as to whether they belong to the calculating field. Namely,

$$\text{Iblank} = \begin{cases} 1, \text{ a point is not blanked out (active node)} \\ 0, \text{ a point is blanked out (nonactive node)} \end{cases}$$

This value is 1 or 0 depending on the area inside or outside the computational subregion. In the flow solver, the right-hand-side vector $\boldsymbol{R}_i$ in Eq. (3) is multiplied by the value iblank(i); namely, the variables of the nonactive node points are temporarily set to be zero during the computation of each grid. Then intergrid data are interpolated between the overlapped grids. If a nonactive node point has a donor cell, flow variables on the node are interpolated by the donor cell. Although the variables on the nonactive node points are never used for the computation, they are indirectly used for the estimation of the gradients of the flow variables. This ensures the robustness of the overset unstructured grid method for moving bodies.

Treatment of Multiple Bodies in Contact

A similar procedure can be applied to multiple bodies in contact. The procedure is shown in Fig. 7 and is explained as follows. In Fig. 7, node point *A* (circular-shaped points) belongs to body 1 (light gray zone) and node points *a*, *b*, *c*, *d*, *e*, *f*, and *g* (square-shaped and diamond-shaped points) belong to the body 2 (dark gray zone). Solid lines show the computational mesh generated around body 2 (mesh 2), and dotted lines show the mesh of the wall boundary of body 2 (mesh 2). Here, both lines are the mesh of body 2, not body 1. In Fig. 7a, the dotted lines show the inner meshes generated inside body 1. Inner meshes are generated inside the body for faster and more accurate searching as is mentioned in the preceding section and shown in Fig. 5, leading to the following:

1) The edge that straddles the wall boundary of the other grid is searched. If the partner cell of a node point of the edge is a computational volume cell, for example, node points *a* and *b*, and that of the other node point is an inner cell, for example, node points *c*, *d* and *e*, the edge straddles the wall boundary. (Repeat for each edge.)

2) The surface element that the edge straddles is identified by dropping perpendicularly from the node point to the surface and identifying the element it touches. (Repeat for each edge that straddles the wall boundary.)

3) The node points whose neighboring node points are located in the computational field, for example, the node point ***c*** whose neighboring node points *a*, *b*, and *h* are located in the computational field in Fig. 7, have the averaged density and pressure of the neighboring node points, for example, $\rho_c = (\rho_a + \rho_b + \rho_h)/3$. The velocities of the node points located inside the body are defined to satisfy the zero-velocity condition among the neighboring node points located in the computational field and themselves, as shown in Fig. 7c. To obtain more precise information, not only the node point that is the closest to the wall boundary, but also neighboring node points in the computational field, are extrapolated. (As node *c* is extrapolated from nodes *a*, *b*, and *h*.) Suppose the extrapolated velocity from *a* to *c* is V_{ac}, from *b* to *c* is V_{bc}, and from *h* to *c* is V_{hc}. The extrapolated velocity then becomes the averaged value $V_c = (V_{ac} + V_{bc} + V_{hc})/3$.

4) Because of this procedure, the node points on the wall boundary of another grid can have proper variables when flow variables are interpolated. (Node *A* can have proper variables interpolated from nodes *a*, *b*, and *c*.)

5) If a node point is inside the other grid body and all neighboring node points are also located inside the body (such as node *f*), the node points are interpolated from the inner mesh. This procedure prevents instability of the solver code resulting from the difference of the flow variables between the connected node points.

6) If the node points that are inside the other grid body do not have adequate partner cells or node points, such as node *g* (because the donor node point of the partner cell is also located inside the body), the nodes have the same flow variables of the nodes that are next to them and nearer to the flowfield.

The foregoing procedure is also important when this approach is applied to the separating simulation. The node points that are inside the body have proper variables so that they can be computed even when they enter the computational field.

Computational Results

Unstructured Meshes

The ONERA M5 wing–fuselage configuration was employed to validate this approach. First, a single hemispherical grid that covers the entire flowfield was generated. For a test of the overset grid, a box-shaped subregion that covers only the near field of the wing was defined as shown in Fig. 8, and a relatively fine grid was generated in it. To resolve the boundary layer around the intersection region, the surface mesh around the intersection region had to be relatively

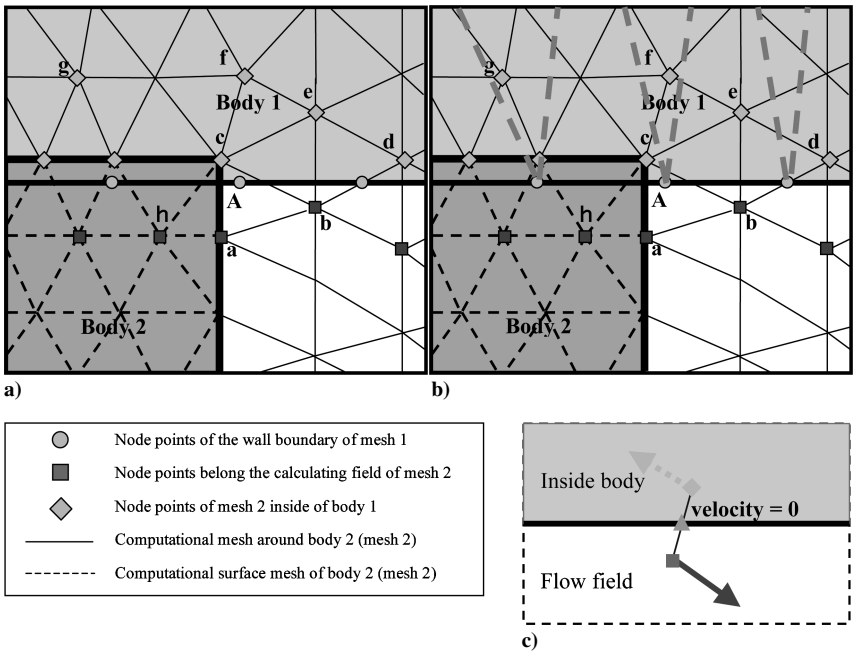

Fig. 7 Treatment of multiple bodies in contact for a) meshes generated around body 2, b) added dotted lines are inner meshes generated inside of body 1, and c) velocity of node point locating inside body is defined to satisfy zero-velocity condition among two node points.

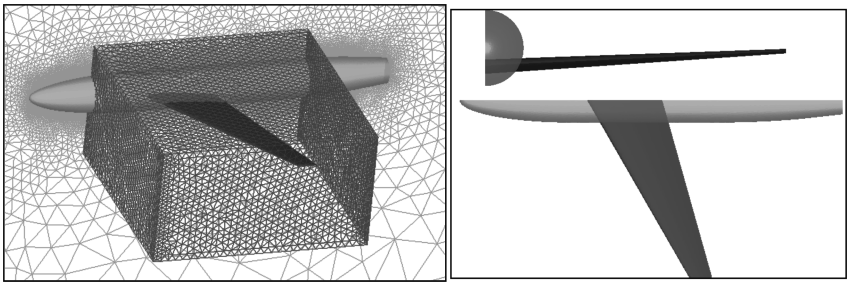

Fig. 8 Overset unstructured meshes for ONERA M5 wing–fuselage.

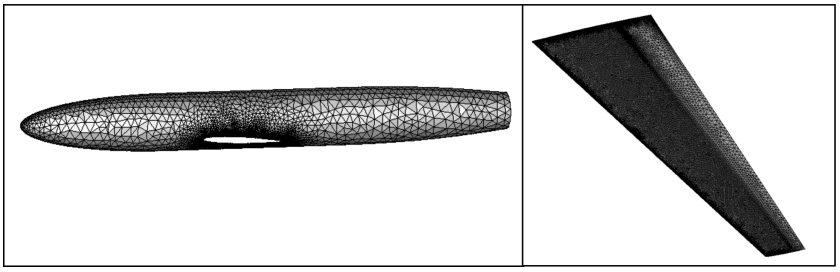

Fig. 9 Surface mesh generated around intersection region of wing–fuselage.

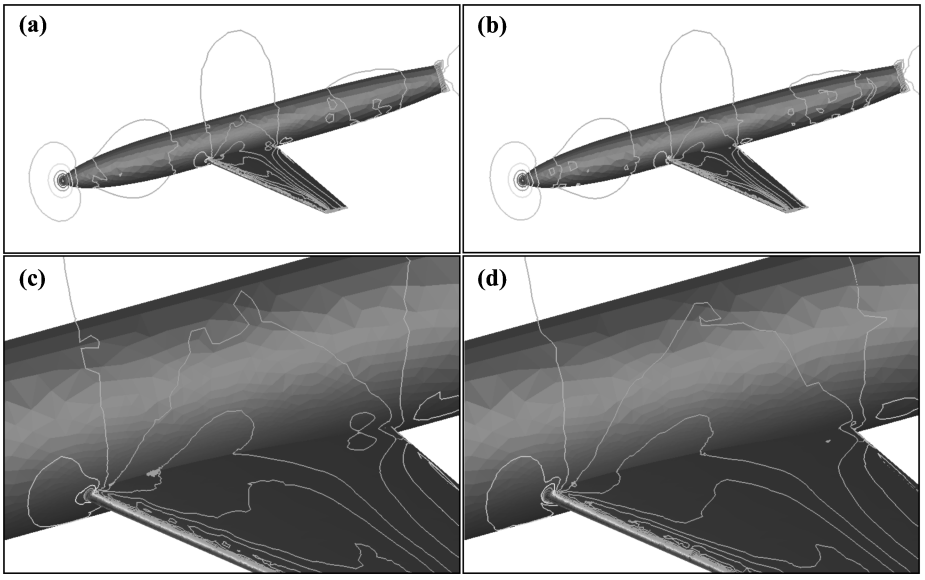

Fig. 10 Computed pressure contours on surface and symmetrical plane for ONERA M5 wing–fuselage $M_\infty = 0.8$ and $\alpha = -1.0$ deg: a) and c) single-mesh case and b) and d) overset-mesh case.

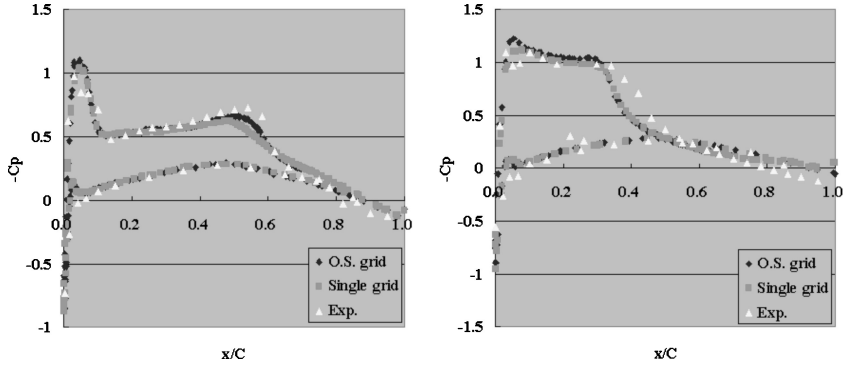

Fig. 11 Comparison of C_p distributions for ONERA M5 at semispan locations of 20%: a) 85% and b) $M_\infty = 0.8$ and $\alpha = -1.0$ deg.

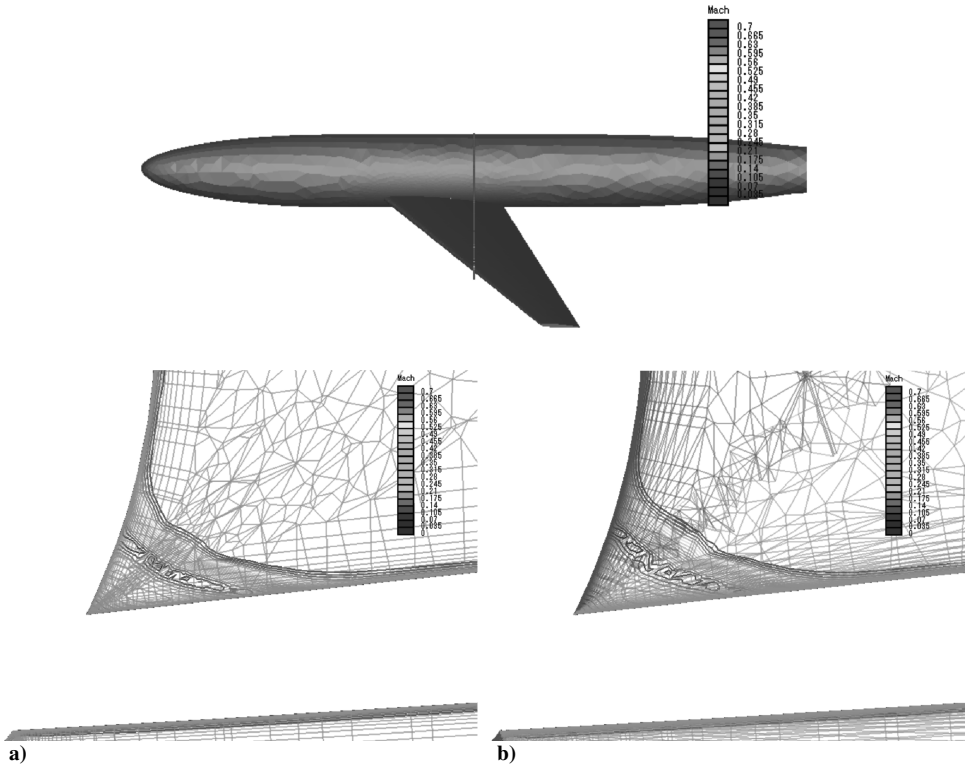

Fig. 12 Cross-sectional views of Mach number contours at 75% of root chord length: a) single mesh and b) overset mesh.

refined (with the edge length being at most 10 times the minimum mesh width of the prismatic layer for proper interpolation) as seen in Fig. 9. The intergrid boundaries between the fuselage grid and the wing grid were identified using the distance from the wall boundary. Figure 6 shows the cut view around the intersection region of the two grids after the overset implementation.

Computational Results

The computations were performed using both the single hemispherical grid and the overset grids for a freestream Mach number of 0.84, an angle of attack of -1 deg, a Reynolds number of 6×10^7, and full turbulence on the body.

Figure 10 shows the computed pressure contours for the case of a single grid (Figs. 10a and 10c) and that of overset grids (Figs. 10b and 10d). In Figs. 10b and 10d, smooth transitions of the contour lines between the wing and the fuselage boundary can be observed.

Figure 11 shows comparisons of the C_p distributions in the case of overset grids, that of a single grid and experimental data on the wing at 20% and at 85% semispan locations.[27] The results show good agreement between the overset-grid case and the single-grid case.

Figure 12 shows cross-sectional views of the Mach number contours at 75% of the root chord length. Smooth transition of the boundary layer between the body and wing grids is observed. The results show good agreement between the overset-grid case and the single-grid case without the collar grid, which resolves the intersection region and connects the related components grid.

Conclusions

The overset unstructured grid method was extended to treat hybrid meshes including prismatic, pyramidal, and tetrahedral cells for computations of viscous flows around multiple bodies in contact. The method was applied to the ONERA M5 body and wing configuration. To perform complete and automatic searching without modifying the searching algorithm, prism and pyramid elements were divided into tetrahedral elements. Flow variables were also interpolated to the node points located inside another body to realize more reliable interpolation in the intersectional region. Computational results showed good agreement with the single-grid computational results and experimental data. The boundary layers in the intersection region were well captured without using extra grids such as the collar grid. The authors believe that this method is effective not only for the store separation problems, but also for design problems such as when designers try to determine the optimum location of a junction of a wing or a nacelle on a wing.

References

[1]Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," American Society of Mechanical Engineers, 1982.

[2]Benek, J. A., Buning, P. G., and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper 85-1523, 1985.

[3]Meakin, R. L., "On Adaptive Refinement and Overset Structured Grids," AIAA Paper 97-1858, 1997.

[4]Cao, H. V., and Su, T. Y., "Navier–Stokes Analyses of a 747 High Lift Configuration," AIAA Paper 98-2623, 1998.

[5]Rogers, S. E., Cao, H. V., and Su, T. Y., "Grid Generation for Complex High-Lift Configurations," AIAA Paper 98-3011, June 1998.

[6]Nichols, R. H., and Tramel, R. W., "Applications of a Highly Efficient Numerical Method for Overset-Mesh Moving Body Problems," AIAA Paper 97-2255, 1997.

[7]Lee, Y., and Baeder, J. D., "High-Order Overset Method for Blade Vortex Interaction," AIAA Paper 2002-0559, 2002.

[8]Nakahashi, K., Togashi, F., and Sharov, D., "An Intergrid-Boundary Definition Method for Overset Unstructured Grid Approach," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2077–2084.

[9]Togashi, F., Nakahashi, K., Ito, Y., Shinbo, Y., and Iwamiya, T., "Flow Simulation of NAL Experimental Supersonic Airplane/Booster Separation," *Computers and Fluids*, Vol. 30, No. 6, 2001, pp. 673–688.

[10]Togashi, F., Ito, Y., Murayama, M., Nakahashi, K., and Kato, T., "Flow Simulation of Flapping Wings of an Insect Using Overset Unstructured Grid," *CFD Journal*, Vol. 12, No. 1, 2003, pp. 98–106.

[11]Togashi, F., Ito, Y., Nakahashi, K., and Obayashi, S., "Extensions of Overset Unstructured Grids to Multiple Bodies in Contact," *Journal of Aircraft*, Vol. 43, No. 1, 2006, pp. 52–57.

[12]Lohner, R., Sharov, D., Luo, H., and Ramamurti, R., "Overlapping Unstructured Grids," AIAA Paper 2001-0439, 2001.

[13]Luo, H., Sharov, D., and Baum, J. D., "An Overlapping Unstructured Grid Method for Viscous Flows," AIAA Paper 2001-2603, 2001.

[14]Togashi, F., Fujita, T., Ito, Y., Nakahashi, K., and Makino, Y., "CFD Evaluation of NAL Jet-Powered Experimental Airplane with Small Rocket

Booster," *International Journal for Numerical Methods in Fluids*, Vol. 48, March 2005, pp. 801–818.

[15]Parks, S. J., Buning, P. G., Steger, J. L., and Chan, W. M., "Collar Grids for Intersecting Geometric Components Within the Chimera Overlapped Grid Scheme," AIAA Paper 91-1587, June 1991.

[16]"Results of the Test on ONERA Calibration Model M5 in NAL 2m × 2m Transonic Wind Tunnel," National Aerospace Lab. Japan, Second Aerodynamics Div., TR-774T, Tokyo, 1983.

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

[18]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.

[19]Sharov, D., and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 157–162.

[20]Ito, Y., and Nakahashi, K., "Unstructured Hybrid Grid Generation Based on Isotropic Tetrahedral Grids," AIAA Paper 2002-0861, 2002.

[21]Ito, Y., and Nakahashi, K., "Unstructured Mesh Generation for Viscous Flow Computations," *Proceedings of the 11th International Meshing Roundtable*, Sandia National Labs., 2002, pp. 367–376.

[22]Obayashi, S., and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier–Stokes Solver," AIAA Paper 94-2268, 1994.

[23]Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, Jan. 1993.

[24]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.

[25]Jameson, A., and Turkel, E., "Implicit Schemes and LU Decompositions," *Mathematics of Computation*, Vol. 37, No. 156, 1981, pp. 385–397.

[26]Goldberg, U. C., and Ramakrishnan, S. V., "A Pointwise Version of Baldwin–Barth Turbulence Model," *Computational Fluid Dynamics*, Vol. 1, 1993, pp. 321–338.

[27]Proceedings of the 10th NAL Symposium On Aircraft Comp. Aerodynamics, CFD Workshop on GK Airfoil and ONERA M5 Geometry, 1992.

K. Ghia
Associate Editor