

OVERSET UNSTRUCTURED GRID METHOD FOR FLOW SIMULATION OF COMPLEX AND MULTIPLE BODY PROBLEMS

Kazuhiro Nakahashi, Fumiya Togashi
Tohoku University

Keywords: *CFD, Unstructured grid, overset grid*

Abstract

The use of the overset concept for the unstructured grid method is discussed for the numerical simulations of flows around complex and multiple bodies in relative motion. The intergrid boundaries in the overset grids are automatically localized using the wall distance as a basic parameter. The search for donor cells is efficiently performed by the neighbor-to-neighbor jump search on a modified convex domain utilizing a byproduct of the Delaunay triangulation method. The capability of the method is demonstrated by simulations of a rocket-booster separation from a supersonic experimental airplane. A grid around the rocket booster is overset on the airplane grid that covers the entire flow field. The complicated configuration of the rocket booster is faithfully reproduced by a single grid. This capability of the unstructured grid significantly simplifies the overset procedure as compared with the conventional structured grids approach. Comparisons of the computed results with the experiment show good agreements in the lift and pitching moment histories.

1 Introduction

CFD applications to real engineering problems require the numerical simulation of flows around complex geometries. The overset grid method [1] provides a powerful means of handling complex geometries by structured grid methods. Unstructured grid methods are also well suited to these kinds of simulation due to their ability in accurately discretizing complex computational domains and to their flexibility in

refining the grid in order to match the local flow features.

Although these two CFD approaches for complex geometry problems have become widespread, some difficulties still remain in each approach. In the overset structured grid method, the number of subgrids which overlap increases if the geometry of the computational models becomes complex. The large number of subgrids complicates the procedure to construct the intergrid communications among overset-structured grids. Unstructured grid approach has a difficulty in computing the unsteady flow around multiple bodies in relative motion. Although the unstructured grid is capable of treating moving bodies as well (for example, [2]), a part of the computational grid or in some cases the whole grid has to be regenerated at every time step. This procedure may become quite complex and computationally expensive.

In this paper, we discuss the application of the overset concept to unstructured grid methods. By use of unstructured meshes, the number of submeshes required for covering the flow field can be significantly reduced as compared with that needed in the overset structured grid. It can also extend the applicability of the unstructured grid method to multiple moving-body problems without much need for code development.

In this paper, an efficient and reliable algorithm to automatically localize the intergrid boundaries for the overset unstructured grid method is described. The method is tested for a separation simulation of a supersonic airplane and a rocket booster.

2 Intergrid Boundary Definition

There are two major steps to establish intergrid communications in the overset method:

- Hole-cutting, which involves dividing all points of each subgrid into two groups, active and non-active points. The intergrid-boundary points are identified as the active points next to non-active points.
- Identification of interpolation stencils, which involves a search of donor cells for all intergrid-boundary points.

The second step, identification of interpolation stencils, is straightforward for unstructured grids. Once a donor cell is identified, values on the point in this cell are interpolated from values on the vertex of the cell using the area co-ordinates for a triangle and the volume co-ordinates for a tetrahedral cell. In the present approach, the donor cell for the interpolation at each intergrid-boundary point is identified during the process of hole-cutting. Therefore, the first step is described here.

2.1 Automatic Hole-Cutting

The identification of the intergrid boundary must be performed completely automatically if unstructured grids are used for the overset approach. Manual creations or corrections of the hole-cutting for the overset unstructured grids are almost impossible because of the unstructured numbering of the node points. Here, the wall distance is used as a parameter to construct the intergrid boundary. In the mesh overlapping region, a node point having a shorter distance to the solid boundary which belongs to the same grid of the node is selected.

The procedure of the intergrid boundary definition is schematically shown in Fig. 1. Suppose that the dotted lines show a grid (Grid-A) generated around Body-A, and the solid lines show a grid (Grid-B) for Body-B. Before the hole-cutting, the minimum distance of each node point to its body surface is computed. Then, the hole-cutting procedure is divided into two steps. The first step is to designate all nodes points as active or non-active ones. The second

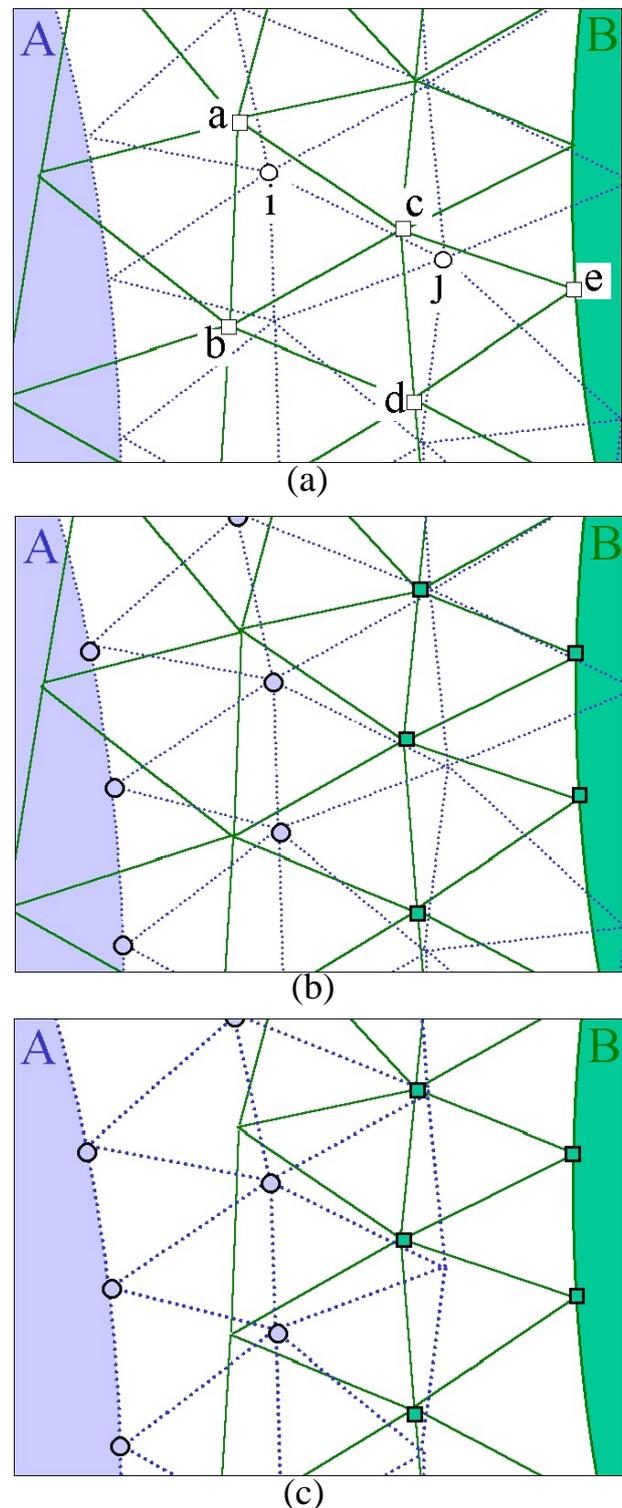


Fig. 1 Determination of intergrid-boundary: (a) Grid A (dotted lines) and Grid B (solid lines) are overlapped. (b) After the node identification. Circles are the active nodes in Grid A and squares are the ones in Grid B, (c) After removal of non-active cells.

step is to classify all cells into three groups: active cells, non-active cells and intergrid-boundary cells.

Let's consider node point i in Grid-A in Fig. 1(a). Suppose we know the donor cell in Grid-B for this node. In Fig. 1(a), the donor cell is indicated by abc . The distance to wall-B from position i in cell abc is then evaluated by a linear interpolation from its vertex values. This distance to wall-B of the donor cell is compared with the distance to Body-A of the node i . Since the distance of this node i to wall-A is shorter than that of the donor cell to wall-B, we select this node as an active node. In contrast, node j in Fig. 1(a) will be selected as a non-active node. This assignment procedure is repeated for all node points in both grids. In Fig. 1(b), nodes shown by circles are active nodes in Grid-A, and those shown by squares are active nodes in Grid-B. The remaining node points are non-active.

By designating all node points in the grids as active or non-active, the next step is to classify all cells into three groups: active cells, non-active cells and intergrid-boundary cells. An active cell is a cell whose vertex nodes are all active, while a non-active cell is the one whose vertexes are all non-active nodes. The remaining cells are the intergrid-boundary cells which construct the overlapping layers among subgrids for intergrid communications. Figure 1(c) shows the grids after removal of the non-active cells.

The above-mentioned procedure is very simple, yet it automatically defines the intergrid boundary and overlapping layer between grids. The overlapping layer has a width of mostly one or two cells as shown in Fig. 1(c).

2.2 Neighbor-to-neighbor search

The use of the wall distance for the automatic definition of the intergrid boundary is simple and very reliable. However, all node points must find their donor cells in the overset meshes. The number of searches easily surpasses one million for three-dimensional problems. Therefore, an efficient and reliable search algorithm must be developed.

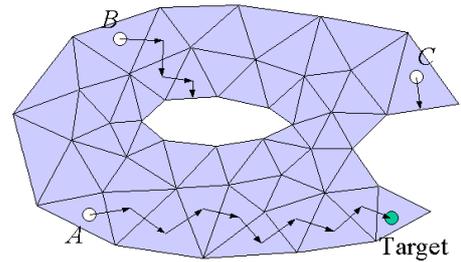


Fig. 2 Neighbor-to-neighbor search. The searches from B and C will fail.

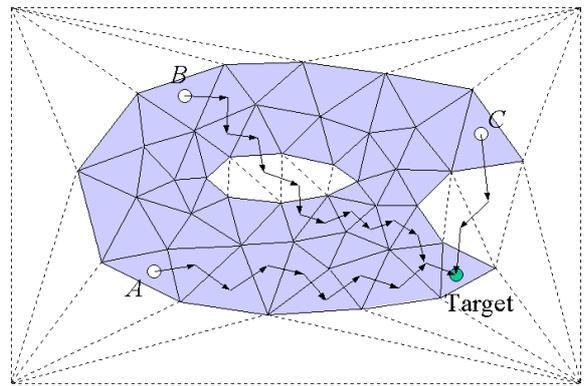


Fig. 3 Neighbor-to-neighbor search in a convex domain.

The neighbor-to-neighbor jump search algorithm [3] is efficiently utilized in the present method. The procedure is schematically shown in Fig. 2 for a triangular grid. Starting from an initial guess, cell-A in Fig. 2 for example, the method is to repeat a jump to the neighboring cell that is located on the target side of the current cell.

This search is very efficient because the search path is one dimensional even in a three-dimensional field. However, it easily fails depending on the starting point. As shown in Fig. 2, a search from point A succeeds in reaching the target. Searches starting from B and C, however, get stuck at the body boundary or the outer boundary. For these cases, the search has to be restarted by changing the cell from which the search is commenced.

To avoid such uncertainty of the search, the search domain is modified to be a convex hexahedron for any computational geometry. This is done so as to add subsidiary grids into the bodies and outside of the computational region as shown in Fig. 3. If we use the

Delaunay triangulation for the grid generation, the subsidiary grids can be obtained automatically as a byproduct of the grid generation procedure. By utilizing the subsidiary grids, the neighbor-to-neighbor search becomes reliable and more efficient.

Since the efficiency of the neighbor-to-neighbor search depends on the initial guess, the computational load for the second search after the initial hole-cutting becomes significantly small. Once the initial hole-cutting has been done, the search of the donor cells after a small relative movement of the subgrid can be limited to nodes around the current intergrid boundaries.

3 Solution Algorithm

3.1 Flow solver

The Euler equations for compressible inviscid flows are written in an integral form as follows:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{Q} dV + \int_{\partial\Omega} \mathbf{F}(\mathbf{Q}) \cdot \mathbf{n} dS = 0, \quad (1)$$

where $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, e]^T$ is the vector of conservative variables, ρ the density, u, v, w the velocity components in the x, y, z directions, and e the total energy. The vector $\mathbf{F}(\mathbf{Q})$ represents the inviscid flux vector and \mathbf{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 non-overlapping dual cell. For the control volume, Eq. (1) can be written in an algebraic form as follows:

$$\frac{\partial \mathbf{Q}_i}{\partial t} = -\frac{1}{V_i} \sum_{j(i)} \Delta S_{ij} \mathbf{h}(\mathbf{Q}_{ij}^+, \mathbf{Q}_{ij}^-, \mathbf{n}_{ij}), \quad (2)$$

where ΔS_{ij} is the segment area of the control volume boundary associated with the edge connecting points i and j . This segment area, ΔS_{ij} , as well as its unit normal, \mathbf{n}_{ij} , can be computed by summing up the contribution from

each tetrahedron sharing the edge. The term \mathbf{h} is an inviscid numerical flux vector normal to the control volume boundary, and \mathbf{Q}_{ij}^{\pm} are values on both sides of the control volume boundary. The subscript of summation, $j(i)$, means all node points connected to node i .

The numerical flux, \mathbf{h} , is computed using an approximate Riemann solver of Harten-Lax-van Leer-Einfeldt-Wada[4]. The second order spatial accuracy is realized by a linear reconstruction of the primitive gas dynamic variables with Venkatakrisnan's limiter [5].

The LU-SGS implicit method [6] is applied to integrate Eq. (2) in time. With $\Delta \mathbf{Q} = \mathbf{Q}^{n+1} - \mathbf{Q}^n$ and a linearization of the numerical flux term as $\mathbf{h}_{ij}^{n+1} = \mathbf{h}_{ij}^n + \mathbf{A}_i^+ \Delta \mathbf{Q}_i + \mathbf{A}_j^- \Delta \mathbf{Q}_j$, the final form of the LU-SGS method on an unstructured grid becomes,

Forward sweep:

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

Backward sweep:

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

where $\mathbf{R}_i = -\sum_{j(i)} \Delta S_{ij} \mathbf{h}_{ij}^n$, $\Delta \mathbf{h} = \mathbf{h}(\mathbf{Q} + \Delta \mathbf{Q}) - \mathbf{h}(\mathbf{Q})$ and

\mathbf{D} is a diagonal matrix derived by Jameson-Turkel approximation of Jacobian [7] as $\mathbf{A}^{\pm} = 0.5(\mathbf{A} \pm \rho_A \mathbf{I})$, where ρ_A is a spectral radius of Jacobian \mathbf{A} . And \mathbf{D} is given as follows:

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

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 [6] to improve the convergence and the vectorization.

3.2 Overset implementation

The above flow solver must be modified to account for the use of multiple meshes. In

addition to the boundaries of the computational domain, subgrids may have holes and intergrid boundaries with the neighboring donor-subgrids. The non-active cells must be excluded or blanked from the flow field solution.

All node points have information as to whether they belong to the active or non-active cells. Namely,

$$IBLANK = \begin{cases} 1, & \text{if a point isn't blanked;} \\ 0, & \text{if a point is blanked.} \end{cases} \quad (5)$$

This value is 1 or 0 depending on the area inside or outside of the computational subregion. In the flow solver, the right-hand side vector \mathbf{R}_i in Eq. (3) is multiplied by the value $IBLANK(i)$. Namely the $f_{\epsilon Q}$ in the outside region (hole region) is set to be zero.

4 Computational Results

The National Aerospace Laboratory (NAL) of Japan is currently working on a project to develop experimental supersonic airplanes [8] as a basic study for the next generation supersonic transport. The first model of the experimental airplanes is unpowered and a solid rocket booster will be used to launch it to a high altitude at a speed of about Mach 2.5. Figure 4 shows the configuration for launch. The fuselage length of the airplane is 11.5 m, and the wingspan is 4.718 m.

The present method was applied to the numerical simulation of this experimental supersonic airplane's separation from a rocket booster. The geometry shown in Fig. 4 was produced by NAL using the CATIA CAD software. It includes detailed components, such as the attachments of the airplane/booster, fittings for the launcher, and fringes on the rocket booster. To compute the flow around this complex geometry by the conventional overset structured grid method will need a time consuming work for generating the grid due to the small components attached on the model. Thus this is a good test case for the present method in order to evaluate its capability for a multiple moving-body problem with complex geometry.

At the beginning, the unstructured surface grids on the airplane and the rocket booster were generated separately by the direct advancing front method combined with the geometrical feature reconstruction [9] on the STL format file produced by the CATIA. This surface grid generation approach significantly reduces the surface meshing burden for complex geometry. Figure 4(b) is an enlarged view of the surface grid at the wing trailing edge region of the airplane. The unstructured surface grid appropriately covers the detailed components of the airplane-booster connection parts as well as the fringe and the wire cover on the rocket booster.

After the surface meshing, two unstructured volume grids, each of which covers the airplane and the rocket booster, respectively triangulations as shown in Fig. 5, were generated using the Delaunay method [10]. The

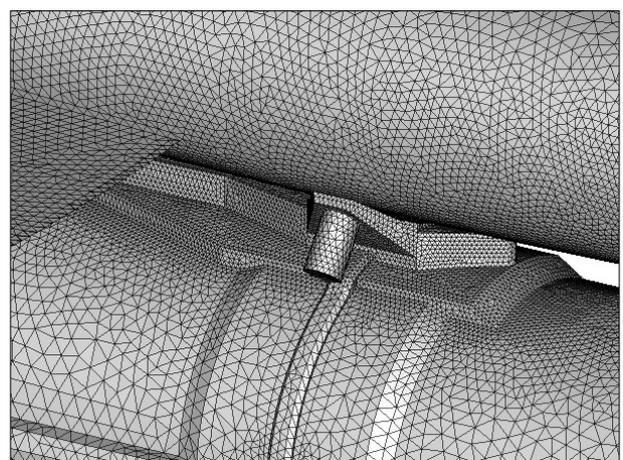
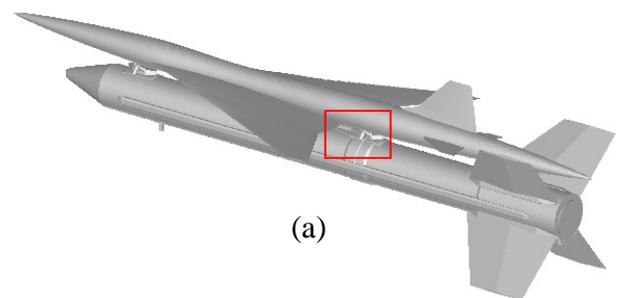


Fig. 4 NAL's experimental supersonic airplane with rocket booster for launch. (a) Entire view, (b) Enlarged view showing the surface grid.

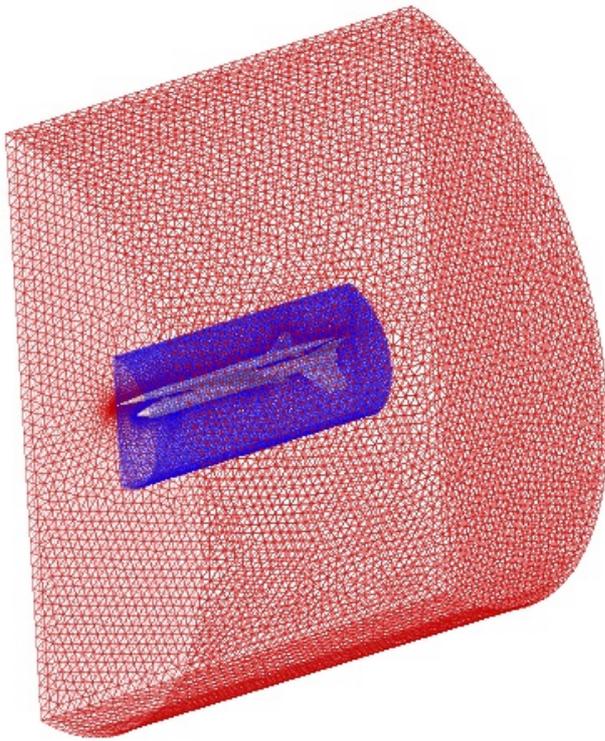


Fig. 5 Overset grids for the supersonic airplane (outer cylindrical region) and rocket booster (inner cylindrical region).

outer cylindrical grid was generated for the airplane and the inner cylindrical grid for the rocket booster. The number of node points and cells of these grids are shown in Table 1. For a simulation of the airplane/rocket booster separation, the inner grid moves with the rocket booster in the stationary outer cylindrical grid.

Table 1 Grid data and average numbers of cell jumps for search

Airplane grid	599,203 nodes points, 3,268,529 cells
Booster grid	262,513 node points, 1,436,487 cells
Donor cell search from airplane to booster	100 cells/node (initial) 0.039 cells/node (the following steps)
Donor cell search from booster to airplane	192 cells/node (initial) 0.13 cells/node (the following steps)

Figure 6 shows the intergrid boundaries of the booster grid for several relative locations between the airplane and the booster. The relative positions and angles of attack were

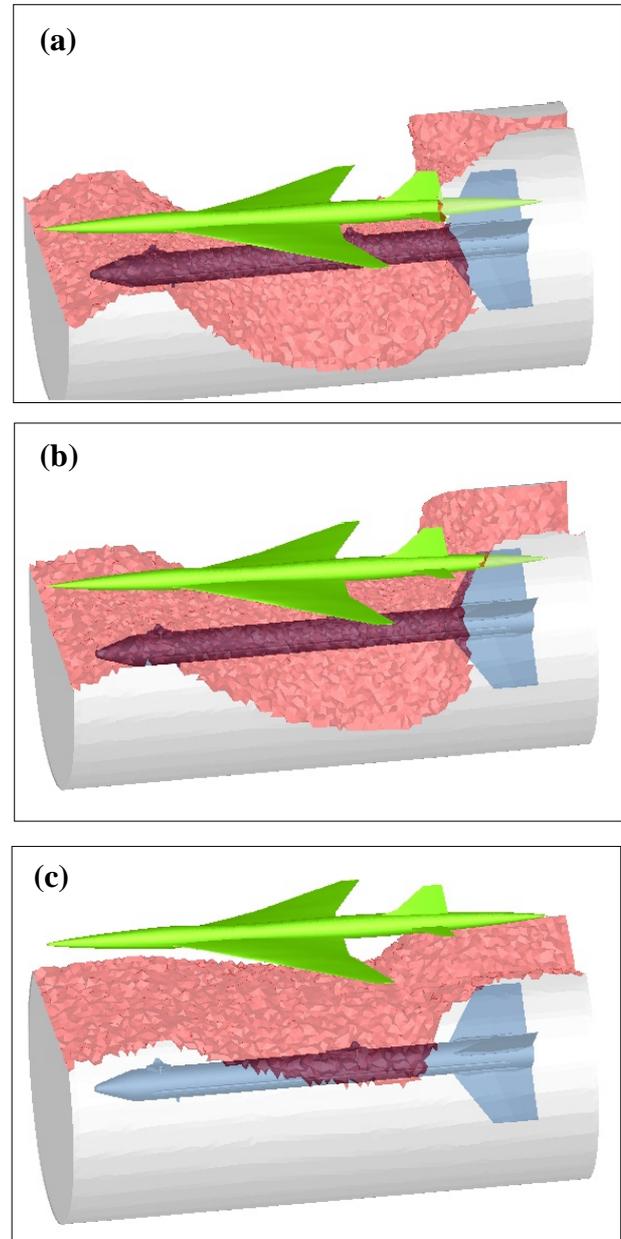
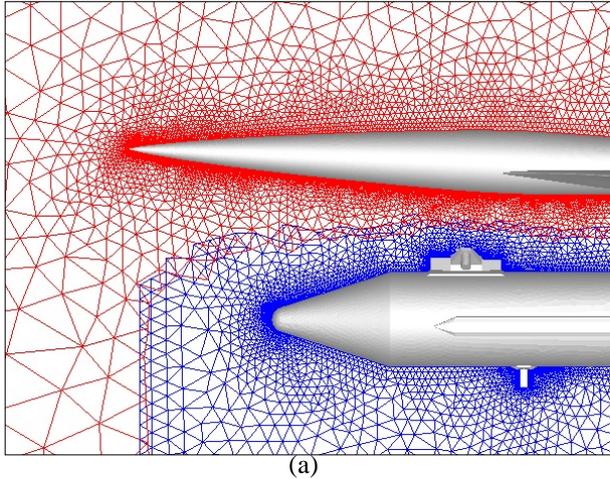
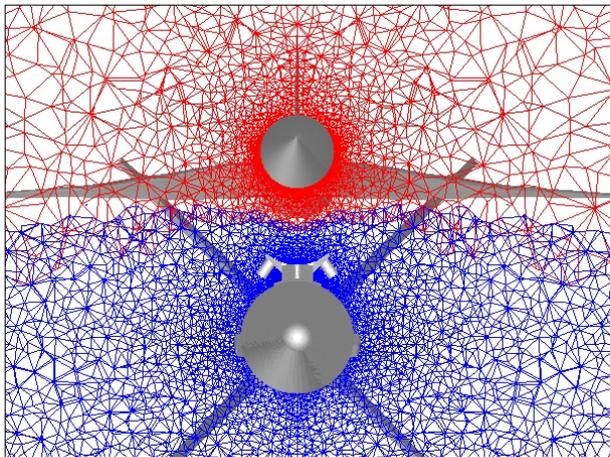


Fig. 6 Intergrid boundaries of the booster subgrid: angles of attack of the airplane at 2 deg and the booster at 0 deg and (a) $\Delta Z=0.4\text{m}$, (b) $\Delta Z=1.0\text{m}$, (c) $\Delta Z=3.0\text{m}$, where ΔZ is the relative distance between the airplane and the booster.

prescribed. In the figure, the dark rough surfaces are the cut surfaces (intergrid boundaries) of the booster grid due to the existence of the airplane. As shown in the figure, the intergrid boundary becomes complex because of the large fins attached to the end of the rocket booster. Figure 7 shows the grids on symmetrical plane and a cut plane perpendicular to the axis of the



(a)

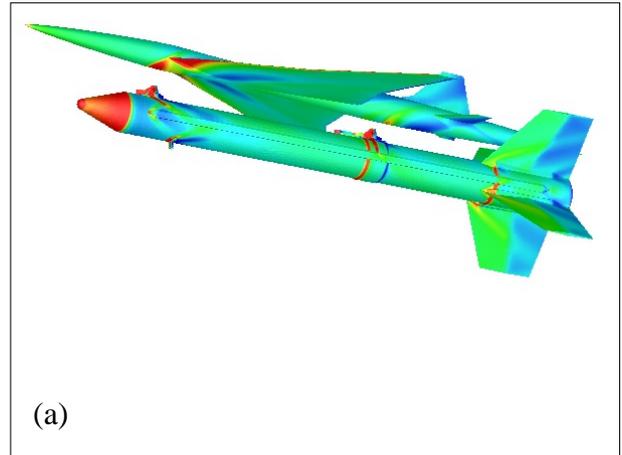


(b)

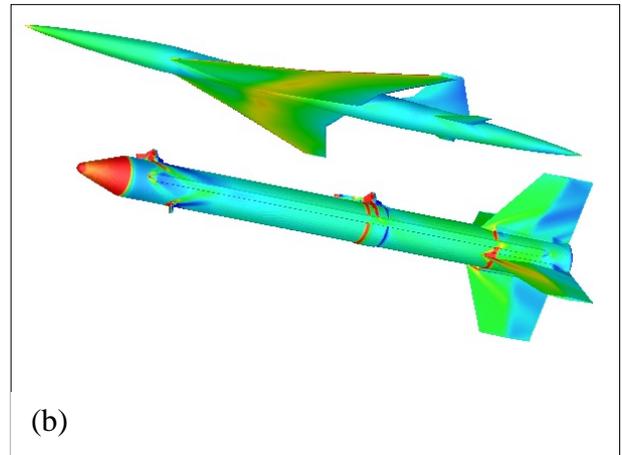
Fig. 7 Overset grids on: (a) symmetric plane, (b) a cross-sectional plane.

airplane fuselage. Two subgrids, one for the airplane and one for the booster, overlap each other. The overlapping layer between two grids has a width of mostly one or two cells.

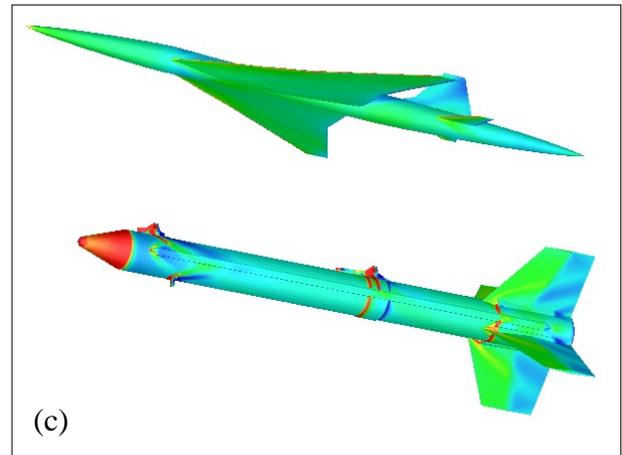
Table 1 shows the average numbers of cell jumps required for one search path measured for the configuration of zero relative angle of attack. The search from the booster grid to the airplane grid is to find a donor cell in about 3.3 million cells. Even with this large number of cells, one search path required 192 cell jumps in average. The initial guess for each search uses the search result of the previous node. Therefore, this search path length may be reduced some more if the node numbering is well ordered. The searches after the initial intergrid setup become significantly small as



(a)



(b)



(c)

Fig. 8 Computed pressure contours of supersonic airplane/rocket booster separation at $M_\infty = 2.5$, angles of attack of the airplane at 2 deg and the booster at 0 deg, and (a) $\Delta Z = 0.4\text{m}$, (b) $\Delta Z = 2.4\text{m}$, (c) $\Delta Z = 5.0\text{m}$

shown in the Table 1.

Computations were performed at a Mach number of 2.5 with an assumption of the quasi-

steady flow. The angle of attack of the airplane was fixed at 2.0 degrees and the angle of attack of the rocket booster relative to the airplane was at -2.0 degrees (0 degree to freestream). The relative horizontal location of the airplane and rocket was fixed to 0, the relative vertical distances between the airplane and the booster (ΔZ) were increased in a prescribed motion.

Figure 8 shows the computed pressure contours around the airplane and booster. Shock waves generated at the noses of the airplane and the booster create a complex reflection pattern in the narrow region between the bodies. At the beginning of the booster separation, the shock wave from the booster nose hits the forward part of the lower surface of the airplane wing. This initially causes an increase in the pitching moment of the airplane. This pitching moment then decreases to a negative value as the impinging point of the booster-nose shock on the airplane wing moves downward.

Figure 9 is an enlarged picture of the computed pressure contours at $\Delta Z = 0.1\text{m}$. A strong shock wave generated at the booster nose hits the lower surface of the airplane fuselage and the reflecting shock return to the booster surface. The airplane/booster attachment part also generates a strong shock wave which interacts with the reflecting nose shock.

Figure 10 shows comparisons of the lift and pitching moment coefficients between

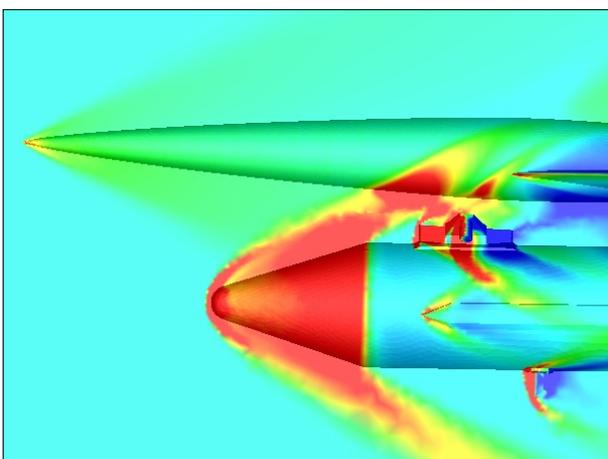


Fig. 9 Computed pressure contours of supersonic airplane/rocket booster separation at $M_\infty = 2.5$, angles of attack of the airplane at 2 deg and the booster at 0 deg, and $\Delta Z = 0.1\text{m}$

computational and wind tunnel results. In these figures, ΔC_L denotes the difference between the lift coefficient of the multiple body case and the one of the isolated body case. The value ΔC_M denotes the same difference of the pitching moment coefficients.

At the beginning of the booster separation from the airplane, the shock wave from the booster nose hits the forward part of the lower surface of the airplane wing. This causes an increase in the pitching moment and lift of the airplane. Then, the pitching moment of the airplane moves to be negative as the impinging point of the booster-nose shock on

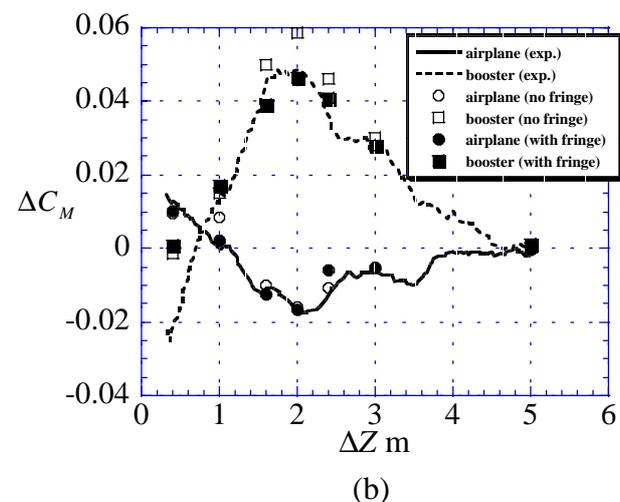
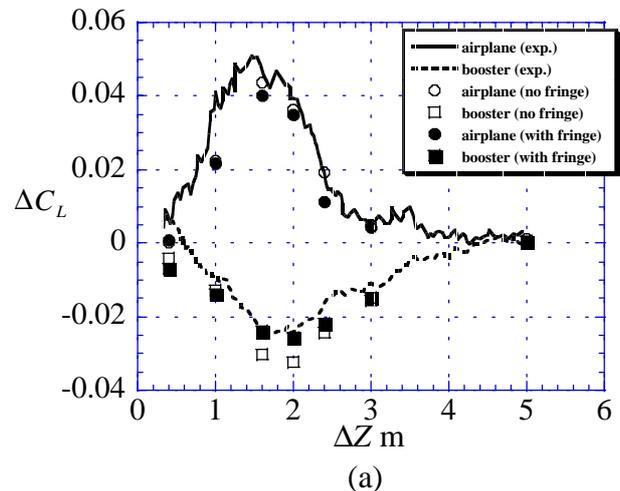


Figure 10 Comparisons of (a) lift and (b) pitching moment coefficients between the experimental and computational results at $M_\infty = 2.5$, angles of attack of the airplane at 2deg and the booster at 0deg.

the lower surface of the airplane moves downward. At $\Delta Z = 2\text{m}$, the pitch-down moment of the airplane becomes the maximum value, and then it decreases temporarily when the impinging point of the booster shock moves downstream of the airplane wing. However the pitch-down tendency of the airplane occurred again when the booster shock hits the tail wing of the airplane. The pitching moment and lift of the booster are decreased due to the effect of not only the shock wave from the airplane but also the reflecting shock wave of itself as shown in Fig. 8 and Fig. 9. Then, the pitching moment of the booster moves to be positive as the impinging point of the shock wave from the airplane moves downward.

The computations were executed in two cases: one for the full detailed configuration shown in Fig. 4, and one for a clean configuration which does not have any small components such as the attachments of the airplane/booster, fittings for the launcher, and fringes on the rocket booster. In the conventional structured grid CFD, these small parts are often neglected because of the difficulty of the grid generation. It is also thought that the effect of those small components to the aerodynamic coefficients is negligible. In the present overset unstructured grid approach, to include the small components into the computation is straightforward. The wind tunnel tests were conducted for the full detailed configuration.

By comparing two computed results, with and without the fringes, the effect of the connecting fringes is relatively small for the airplane aerodynamic coefficients. However, as shown in Fig.10, the peak values of ΔC_L and ΔC_M of the booster are apparently affected by the fringes and the computed results with the detailed configuration agree better with the experiment.

5 Conclusions

The use of the overset concept for the unstructured grid method was discussed for the numerical simulations of a booster separation

from a supersonic experimental airplane. A grid around the rocket booster was overset on the airplane grid that covers the entire flow field. Owing to the unstructured grid, a single grid can faithfully reproduce detailed components of the rocket booster. This capability of the unstructured grid significantly simplifies the overset procedure as compared with the conventional approach to overlap a number of structured grids. An efficient and robust algorithm to localize the intergrid boundaries for the overset unstructured grid method has been developed using the wall distance as a basic parameter. The use of subsidiary grids, which are generated as a byproduct of the Delaunay triangulation method, makes the neighbor-to-neighbor jump search reliable and efficient. The computed result of the airplane/booster separation clearly reproduced the complex reflection shock wave patterns between two bodies. Comparisons with the experimental results showed good agreements in the lift and pitching moment histories.

The use of the overset concept with the unstructured grid methods holds great promise for extending the applicability of the unstructured grid method for real engineering problems without much needed for code development.

6 Acknowledgements

The authors wish to thank Dr. Iwamiya of National Aerospace Laboratory and Dr. Shinbo of Mitsubishi Heavy Industry for providing us with the airplane and rocket booster geometry data used for the wind tunnel experiment. The authors also wish to thank to Mr. Fujita and Mr. Ito, graduate students of Tohoku University, of their helps in the CAD operations and the grid generations.

References

- [1] Steger, J. L., Dougherty, F. C., and Benek, J. A. A Chimera grid scheme. *ASME Mini-Symposium on Advances in Grid Generation*, 1982.
- [2] Lohner, R. and Baum, J. D. Three-Dimensional Store Separation Using a Finite Element Solver and Adaptive Remeshing. *AIAA Paper 91-0602*, 1991.

- [3] Lohner, R. Robust, Vectorized Search Algorithms for Interpolation on Unstructured Grids. *Journal of Computational Physics*, Vol.118, pp.380-387, 1995.
- [4] Obayashi, S., and Guruswamy, G. P. Convergence acceleration of an aeroelastic Navier-Stokes solver. AIAA Paper 94-2268, 1994.
- [5] Venkatakrishnan, V. On the accuracy of limiters and convergence to steady state solutions. AIAA Paper 93-0880, January 1993.
- [6] Sharov, D. and Nakahashi, K. Reordering of Hybrid Unstructured Grids for Lower-Upper Symmetric Gauss-Seidel Computations. *AIAA Journal*, Vol.36, No.3, pp.484-486, 1998.
- [7] Jameson, A., and Turkel, E. Implicit schemes and LU decompositions. *Mathematics of Computation*, Vol.37, No.156, pp.385-397, 1981.
- [8] Iwamiya, T. NAL SST project and aerodynamic design of experimental aircraft. *Proc. Computational Fluid Dynamics '98*, Vol. 2, ECCOMAS 98, John Wiley & Sons, Ltd., pp. 580-585, 1998.
- [9] Ito, Y. and Nakahashi, K. Direct surface triangulation using stereolithography (STL) data. *AIAA Paper* 2000-0924, 2000.
- [10] Sharov, D. and Nakahashi, K. A Boundary Recovery Algorithm for Delaunay Tetrahedral Meshing. *5th Int. Conf. on Numerical Grid Generation in Computational Field Simulations*, pp.229-238, 1996.