INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **43**:769–783 (DOI: 10.1002/fld.559)

Some challenges of realistic flow simulations by unstructured grid CFD

Kazuhiro Nakahashi*,†, Yasushi Ito and Fumiya Togashi

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

SUMMARY

This paper discusses the unstructured grid method to compute flows around geometrically complex bodies having relative motions. To enhance the capability to treat complex geometries, a surface triangulation method using the advancing front method coupled with geometric feature extraction technique is described. Stereolithography (STL) data are adopted as an interface between a CAD system and the surface grid generator. Moving bodies are treated by the overset unstructured grid method. The capability of the method is demonstrated for simulations of an airplane separation process from a rocket booster and a hornet in flight. In the hornet simulation, the detailed components such as antennas, legs and a sting are all included in the computational grid. The flapping wings are treated by the overset unstructured grid method where a grid around the wing is overlapped on a stationary grid around the body of a hornet and moves with time to simulate the flapping motion. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: unstructured grid; surface grid generation; overset grid; supersonic airplane; insect

1. INTRODUCTION

The computational fluid dynamics (CFD) has achieved a significant progress in its algorithms and the applications in the last 30 years. In these days the CFD is considered to be very close to its maturing stage. It is certainly true that the CFD is routinely applied to flows around airfoils and wings for the performance analysis and the design of airplanes. For geometrically complex and moving-body problems, however, the current CFD still needs painful efforts for the grid generation and enormous CPU times to obtain the desired accuracy level.

Difficulty in generating a grid around a complex geometry hinders the practical use of the CFD in engineering analysis and design. To overcome this, unstructured grids are becoming popular for complex geometry CFD because of the easiness of tetrahedral volume grid generation. However, a time-consuming procedure is still required for the surface grid gener-

*Correspondence to: K. Nakahashi, Department of Aeronautics and Space Engineering, Tohoku University, Sendai 980-8579-01, Japan.
†E-mail:naka@ad.mech.tohoku.ac.jp

Received 1 April 2002
Revised 12 March 2003

ation. This pre-process of the grid generation becomes more critical as the complexity of the geometry increases.

Unsteady flow computations around multiple bodies in relative motion have another difficulty in grid. In the unstructured grid method, moving bodies are treated by deforming and remeshing the grids. However, a part of the computational grid or in some cases the whole grid has to be regenerated at every time step for computation of unsteady flows with large movements of bodies. This procedure may become quite complex and computationally expensive. A simple and efficient approach should be developed to treat such a problem.

In this paper, approaches to treat complex and moving body problems are discussed. For the surface meshing around a complex geometry, the advancing front method coupled with the geometric feature extraction is directly applied to the stereolithography (STL) data that is an output of the CAD. An automatic revision procedure of ill-conditioned facets in STL data, reconstruction method of ridges, and the GUI interface to control the grid density significantly reduce the surface meshing work. Overset unstructured grid method is used for the moving body problems. This approach holds great promise for extending the applicability of the unstructured grid method for real engineering problems without much needed for code development.

The methods are applied to two flow problems in this paper to demonstrate the capability of the present approach. One is a separation simulation of an experimental supersonic airplane and a rocket booster. The other is a flow computation around an insect with flapping wings. These two problems are challenging subjects for the current CFD not only because of the geometrical complexity but also due to the moving bodies that complicates the flow physics.

2. GRID GENERATION

The grid generation procedure employed here is shown in Figure 1. The detailed procedure of the surface meshing is described in References [1, 2] and here we discuss its outline. The basic strategy is to use both automatic and GUI-based interactive procedures effectively. The automatic meshing is preferable to reduce the user works. At the same time, the controllability of the local grid density and quality by user is also very important for surface meshing because the surface mesh directly affect to the solution accuracy.

The surface meshing starts from the input of the STL file which is one of the CAD output formats. The advantage of the STL data is the simplicity since they contain only pure geometric information of the co-ordinates of each triangular facet and its corresponding unit normal vector. Owing to the simplicity of the STL file, other geometry data such as a DXF file and a structured surface grid file can be easily converted to this format.

STL data are used as a background grid for the surface meshing. For this purpose, a set of STL facets must cover the original surface without overlaps or gaps. STL data, however, often contain some ill-conditioned facets due to errors in the tessellation process because the STL file format does not provide for consistency and completeness tests. These facets often cause some conflicts in the following steps, for example, when ridges are automatically reconstructed from the original STL data, or when the local direction of surface meshing is calculated. For this problem, a cleanup procedure of the STL data was developed to automatically correct or remove those ill-conditioned facets. Commonly encountered ill-conditioned

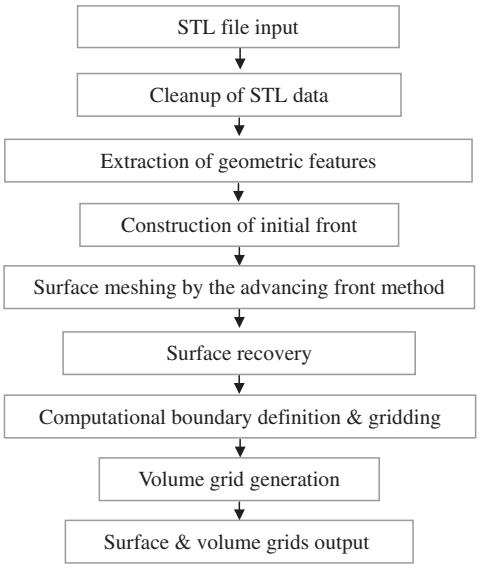


Figure 1. Flowchart of grid generation.

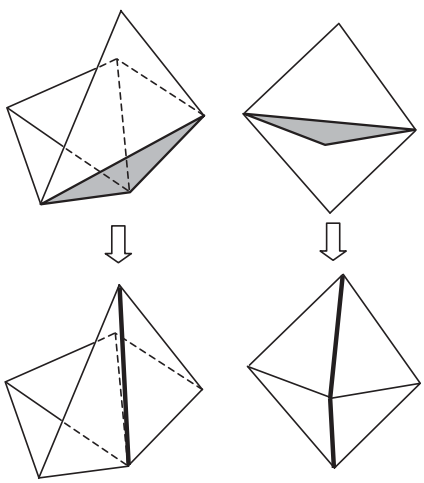


Figure 2. Examples of sticking facets in STL data and their revisions.

facets, such as shown in Figure 2 for examples, are automatically detected and fixed by a specified procedure for each pattern. The remaining ill-conditioned facets that do not meet the commonly encountered patterns will be revised by means of the GUI tool developed here.

For the surface triangulation, the direct advancing front method [3] is applied to the background grid defined by the revised STL data. The set-up of the initial front for the advancing

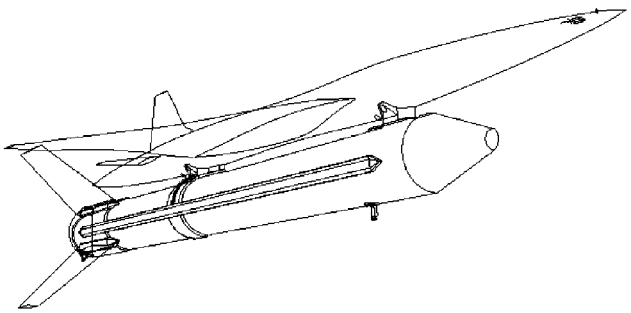


Figure 3. Extracted geometric features for the NALs experimental airplane with rocket booster for launch.

front method, however, required a time-consuming procedure for complex geometries. Here, a geometric feature extraction technique is utilized to simplify the initial front set-up. Typical geometric features are ridges and corners as shown in Figure 3 and the initial fronts of the advancing front method are set-up on these curves. The extraction of these featuring curves is important to accurately represent the original configuration [4]. They are also utilized to control the local grid density.

The geometric features are detected by the folding angles between two neighbouring STL facets. The ridges detected by the folding angle, however, have not yet been connected to each other in this stage, but may be isolated, and do not represent the model properly. The automatic eliminations or reconnections of ridges are performed by checking the distances and angles to the adjacent ridges. Some manual treatments of ridges may be required to recover the proper features accordingly.

Along each featuring curve, the initial front for the advancing front method is constructed by distributing node points by the Vinokur's stretching function [5]. The number of points and the clustering parameters must be specified using a GUI command. At this stage, local grid density is also controlled by adding point sources and line sources on the user-specified region.

During the surface triangulation, the new node position of the surface grid is temporarily determined with a first-order correction so as not to fail in the triangulation. The generated surface grid may not, however, represent the original configuration accurately if the STL-defined original data is not fine enough. In order to improve the grid quality, a surface recovery algorithm using the second-order interpolation [4] is applied after the surface triangulation.

Figure 4 shows a generated surface grid for the NAL experimental airplane piggybacked on the rocket booster. The surface grid covers faithfully the CAD defined original configuration including such as the attachments of the airplane/booster, fittings for the launcher, and fringes on the rocket booster. Total number of the surface triangles is 200 084 and the required CPU time on Pentium III (900 MHz) PC is about 10 min for the surface grid generation process. Additional 30–60 min was required for the interactive operations on GUI screen, such as creating initial fronts in this case.

After the surface meshing, other computational boundaries are defined and the tetrahedral volume grid is generated by a Delaunay-type generation method [6].

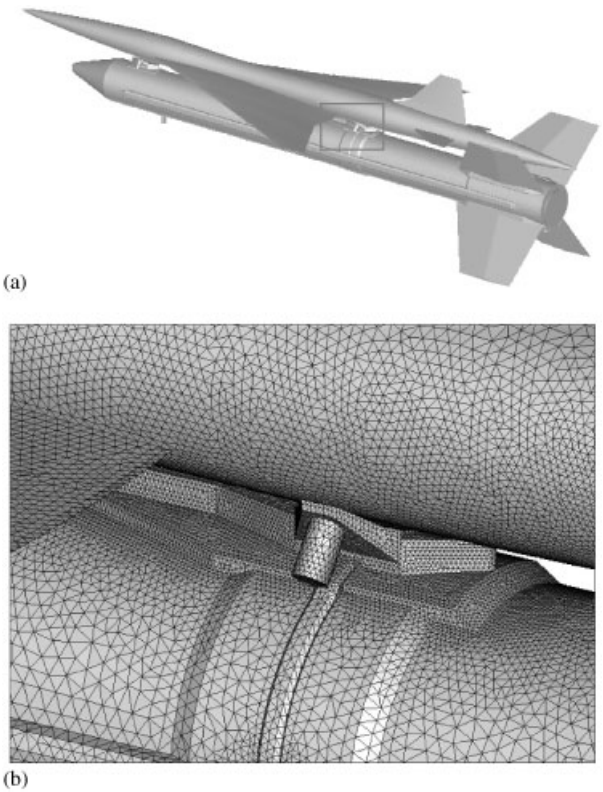


Figure 4. NALs experimental supersonic airplane with rocket booster for launch. (a) Entire view and, (b) enlarged view showing the surface grid.

3. OVERSET UNSTRUCTURED GRID

In structured grids, moving body problems have been successfully simulated by the overset grid approach or the so-called Chimera method [7]. However, the approach still requires technical expertise and considerable man-hours to construct the intergrid communications among overlapping grids. One of the reasons for this difficulty is that, as the complexity of the geometry increases, the number of grids which must be overlap increases so far as the structured grids are used.

It is a reasonable idea to use unstructured grids for the overset concept [8]. By use of unstructured meshes, the number of submeshes required for covering the flow filed around complex geometries 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 the Chimera concept, there are two major steps to establish intergrid communications among overlapping unstructured grids:

Step 1: 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.

Step 2: 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 volume co-ordinates for a tetrahedral cell. This treatment of the interface between meshes is first-order and non-conservative. It is simple and practically enough for the accuracy of computed large-scale values such as pressure loading on surfaces [9]. However, a conservative treatment of the interface should be developed when the oversetting is used near critical regions such as shear layers for the Navier–Stokes computations.

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. The detailed description can be found in Reference [8].

The hole-cutting 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 as the active point in that grid. This 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. The wall distance is calculated only once at the beginning of the computation. It is computed by a method similar to the advancing front method, where the computing front of the wall distance is marched from the wall boundary to the outer boundary.

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.

The neighbour-to-neighbour jump search algorithm [4] is efficiently utilized in the present method. In this method, the searching is performed by judging which side the target point locates from the current cell and moving to the neighbouring cell of the direction. This search is very efficient because the search path becomes one-dimensional even in a three-dimensional field.

Grids generated inside the bodies are also used to improve the reliability of the search by making the searching domain to be convex. These subsidiary grids are generated as a byproduct of the Delaunay triangulation method. The internal grids are also effective to get the in–out information that is important for the reliability of the donor-cell search.

4. FLOW SOLVER

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

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}\,\mathrm{d}V+\int_{\partial\Omega}\mathbf{F}(\mathbf{Q})\cdot\mathbf{n}\,\mathrm{d}S=0 \tag{1}$$

where $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, e]^{\mathrm{T}}$ 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. For the control volume of a non-overlapping dual cell, Equation (1) can be written in an algebraic form as follows:

$$\frac{\partial \mathbf{Q}_i}{\partial t} = \frac{1}{V_i}\mathbf{R}_i \tag{2}$$

and

$$\mathbf{R}_i = -\sum_{j(i)} \Delta S_{ij}\mathbf{h}(\mathbf{Q}_{ij}^{+}, \mathbf{Q}_{ij}^{-}, \mathbf{n}_{ij}) \tag{3}$$

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 subscript of summation, $j(i)$, means all node points connected to node i.

The term $\mathbf{h}$ in Equation (3) is an inviscid numerical flux vector normal to the control volume boundary. It is computed using an approximate Riemann solver of Harten–Lax-van Leer–Einfeldt–Wada [10]. The values on both sides of the control volume boundary, $\mathbf{Q}_{ij}^{\pm}$, are evaluated by a linear reconstruction of the primitive gas dynamic variables with Venkatakrishnan's limiter [11] for the second-order spatial accuracy.

With a time increment, $\Delta\mathbf{Q} = \mathbf{Q}^{n+1} - \mathbf{Q}^{n}$, and a linearization of the numerical flux term as $\mathbf{h}_{ij}^{n+1} = \mathbf{h}_{ij}^{n} + \mathbf{A}_i^{+}\Delta\mathbf{Q}_i + \mathbf{A}_j^{-}\Delta\mathbf{Q}_j$, a procedure similar to the conventional derivation of the LU-SGS on structured grids [12] leads to the following equations:

$$\left(\frac{V_i}{\Delta t}\mathbf{I} + \sum_{j(i)} \Delta S_{ij}\mathbf{A}_i^{+}\right)\Delta\mathbf{Q}_i + \sum_{j(i)} \Delta S_{ij}\mathbf{A}_j^{-}\Delta\mathbf{Q}_j = \mathbf{R}_i \tag{4}$$

The LU-SGS method on unstructured mesh can be derived by splitting the node points $j(i)$ into two groups, $j \in L(i)$ and $j \in U(i)$, for the second summation in LHS of Equation (4). The final form of the LU-SGS method for the unstructured grid becomes:

Forward sweep:

$$\Delta\mathbf{Q}_i^{*} = \mathbf{D}^{-1}\left[\mathbf{R}_i - 0.5\sum_{j\in L(i)} \Delta S_{ij}(\Delta\mathbf{h}_j^{*} - \rho_A\Delta\mathbf{Q}_j^{*})\right] \tag{5a}$$

Backward sweep:

$$\Delta\mathbf{Q}_i = \Delta\mathbf{Q}_i^{*} - 0.5\mathbf{D}^{-1}\sum_{j\in U(i)} \Delta S_{ij}(\Delta\mathbf{h}_j - \rho_A\Delta\mathbf{Q}_j) \tag{5b}$$

where $\Delta\mathbf{h} = \mathbf{h}(\mathbf{Q} + \Delta\mathbf{Q}) - \mathbf{h}(\mathbf{Q})$ and $\mathbf{D}$ is a diagonal matrix derived by Jameson–Turkel approximation of Jacobian [12] as, $\mathbf{A}^{\pm} = 0.5(\mathbf{A} \pm \rho_A\mathbf{I})$ where ρ_A is a spectral radius of Jacobian $\mathbf{A}$.

$$\mathbf{D} = \left(\frac{V_j}{\Delta t} + 0.5\sum_{j(i)} \Delta S_{ij}\rho_A\right)\mathbf{I} \tag{6}$$

The lower/upper splitting of Equation (5) for the unstructured grid is realized by using a grid reordering technique [13] to improve the convergence and the vectorization. The extension to the Navier–Stokes equations is described in Reference [14] with discussions of its accuracy and efficiency for various flow problems.

The flow solver must be modified to account for the use of multiple meshes. In addition to the boundaries of the computational domain, subgrids have intergrid boundaries with the neighbouring donor subgrids and may also contain holes. 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,

$$IBLANK = \begin{cases} 1 & \text{if a point is not blanked} \\ 0 & \text{if a point is blanked.} \end{cases} \tag{7}$$

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 Equation (5) is multiplied by the value $IBLANK(i)$. Namely the $\Delta\mathbf{Q}$ in the outside region (hole region) is set to be zero.

5. NUMERICAL RESULTS

5.1. Simulation of a supersonic airplane/rocket booster separation

National Aerospace Laboratory (NAL) of Japan is currently working on a project to develop experimental supersonic airplanes [15] as a basic study for the next generation supersonic transport. The experimental airplane 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.

The geometry shown in Figure 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. Then, two unstructured volume grids, each of which covers the airplane and the rocket booster, respectively, as shown in Figure 5, were generated using the Delaunay method. The outer cylindrical grid was generated for the airplane and the inner cylindrical grid for the rocket booster. The numbers of tetrahedral cells for the airplane and booster grids are 3 268 529 and 1 436 487, respectively. For a simulation of the airplane/rocket booster separation, the inner grid moves with the rocket booster in the stationary outer cylindrical grid.

Plate 1 shows the grids on symmetrical plane and a cut plane perpendicular to the axis of the airplane fuselage after the hole cutting. 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.

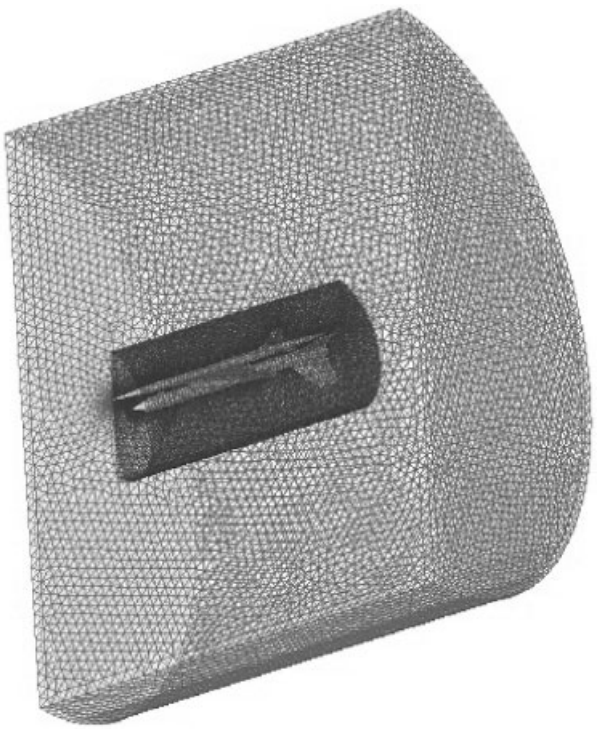


Figure 5. Overset grids for the supersonic airplane (outer cylindrical region) and rocket booster (inner cylindrical region).

Computations were performed at a Mach number of 2.5 with an assumption of the inviscid and quasi-steady flow. The angle of attack of the airplane was fixed at 2.0° and the angle of attack of the rocket booster relative to the airplane was at −2.0° (0° 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.

Plate 2 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.

Figure 6 [16] shows comparisons of the lift and pitching moment coefficients between 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 the lower surface of the airplane moves downward. At $\Delta Z = 2\,\mathrm{m}$, the pitch-down moment of the

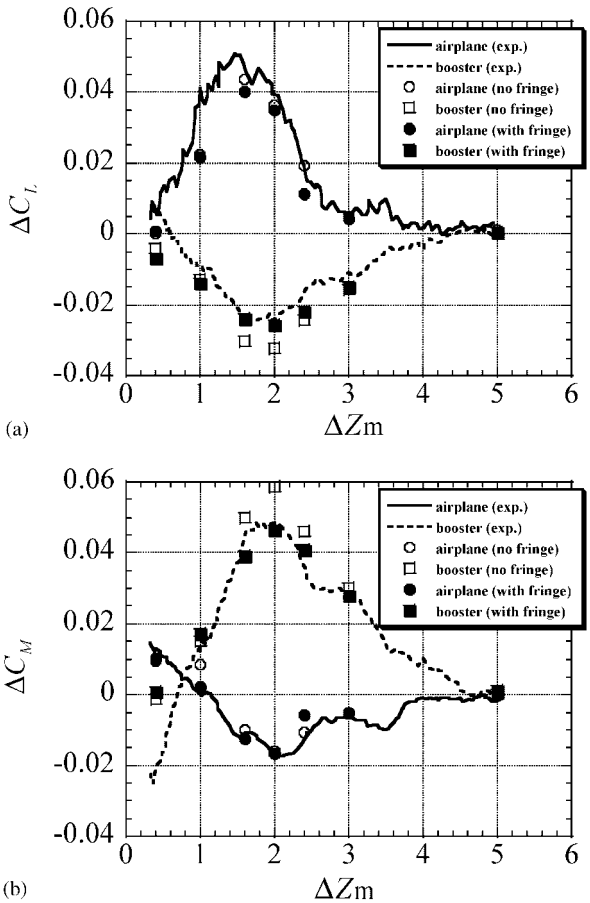
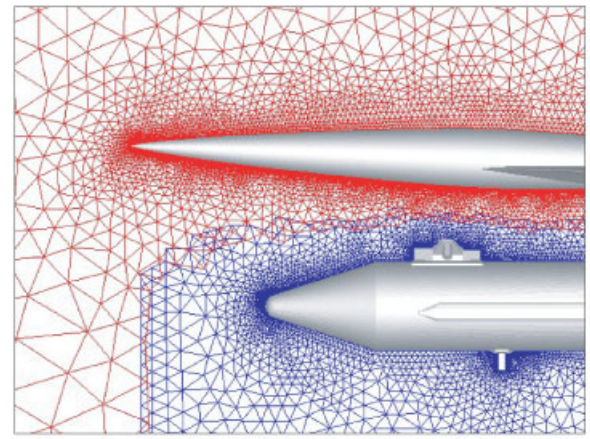


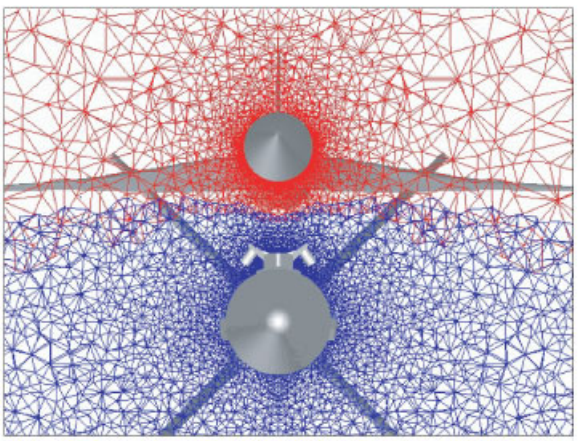
Figure 6. Comparisons of (a) lift and (b) pitching moment coefficients between the experimental and computational results at $M_\infty = 2.5$, angles of attack of the airplane at 2° and the booster at 0°.

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 Plate 2. 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 Figure 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



(a)



(b)

Plate 1. Overset grids on: (a) symmetric plane and (b) a cross-sectional plane.

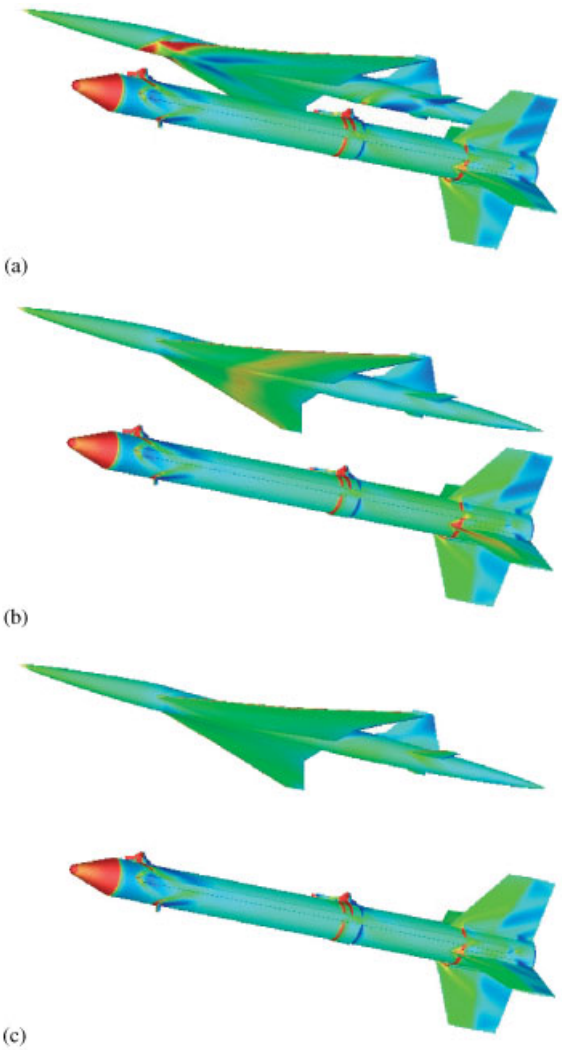


Plate 2. Computed pressure contours of supersonic airplane/rocket booster separation at $M_\infty = 2.5$, angles of attack of the airplane at 2° and the booster at 0° and (a) $\Delta Z = 0.4$ m, (b) $\Delta Z = 2.4$ m, (c) $\Delta Z = 5.0$ m.

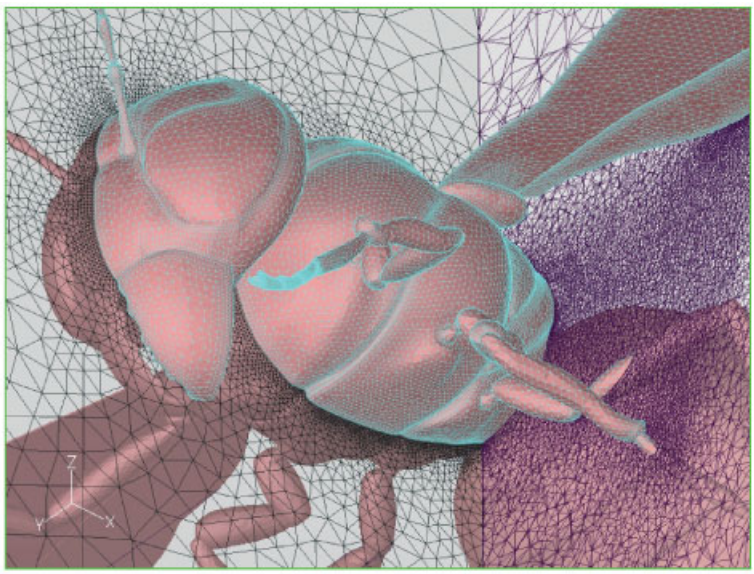


Plate 3. Cut view of tetrahedral grid (741 728 nodes, 4 183 735 cells).

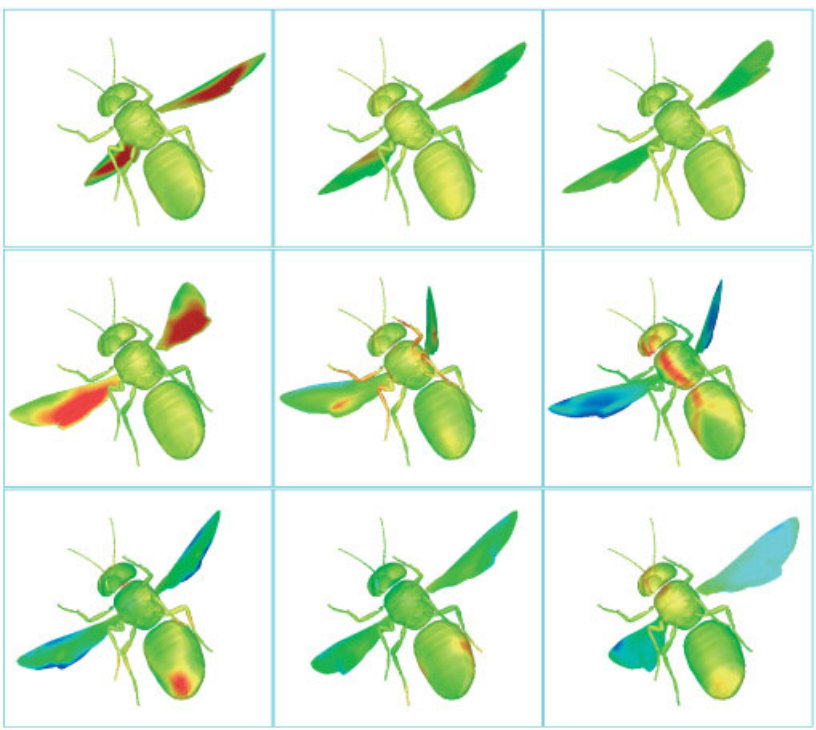


Plate 4. Computed pressure distributions on a lower surface of a flapping hornet.

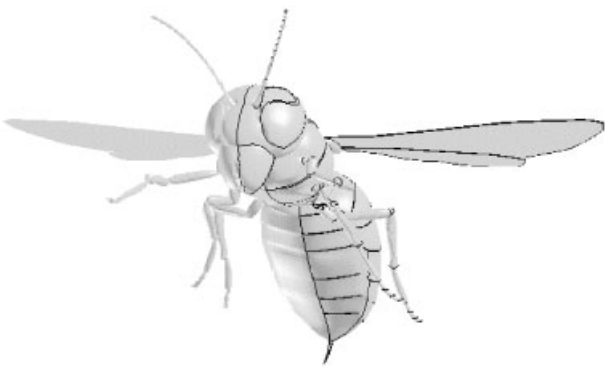


Figure 7. Hornet model and ridges (blacklines).

effect of those small components on the aerodynamic coefficients is negligible. In the present overset unstructured grid approach, to include the small components into the computation is straightforward.

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 Figure 6, 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.

5.2. Flow around a hornet with flapping wings

The method was applied to a hornet model which is shown in Figure 7. The CAD data was given by a DXF file which was converted to the STL file. After the cleanup of the data, the surface grid generation method was applied to the revised surface. In the reconstruction of geometric features, several ridges were added particularly at the legs because the defined surface was relatively rough. Surface triangulation was then performed as shown in Figure 8. The generated surface mesh has 132 626 triangular faces. The required CPU time on Pentium III (600 MHz) PC is about 3 min for the surface grid generation process in this case. The total required time for the surface mesh was about 1 day. Most of these works was to eliminate interfered faces that were contained in original CAD data given by DXF files. If the model is given by a CAD-defined solid model, this modification becomes trivial.

Plate 3 shows cut views of the volume grid which has 4 183 735 tetrahedra in it.

A hornet flies at a speed of about 6 m/s with Reynolds number of 4200 based on the forward flight speed and the chord length of the wing. The flapping frequency is about 100 Hz [17]. In this paper, however, the flow was computed using the compressible Navier–Stokes flow solver instead of the incompressible one. This is mainly because of saving the CPU times for unsteady flow computations.

The computations were performed at a Mach number of 0.12 with the Reynolds number of 4200. With this low Mach number, the compressibility effect is negligible. Reduced frequency defined as $k = \omega^*(c/2U)$ was adjusted to the 100 Hz with 6 m/s flight speed; namely a ratio between the flow velocity and the flapping wing tip velocity was adjusted.

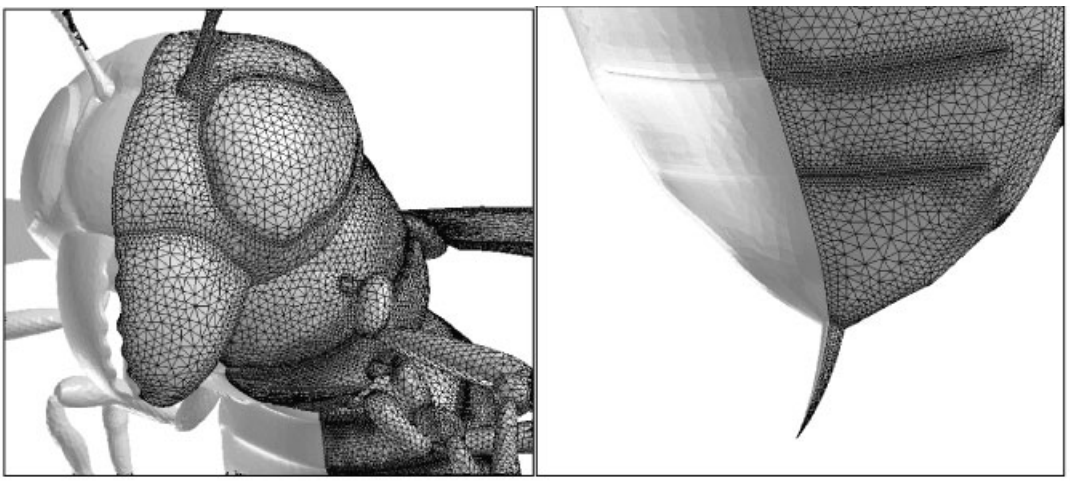


Figure 8. Surface grid around a hornet.

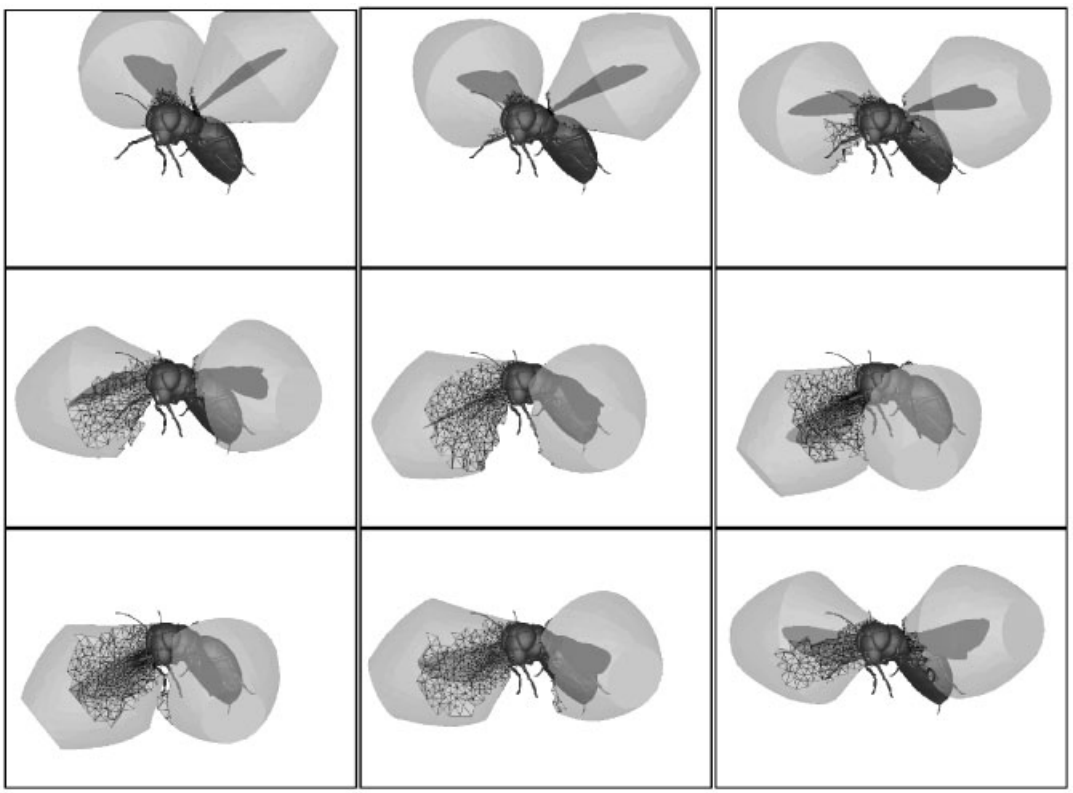


Figure 9. Flapping wing motions and the intergrid boundaries of wing grids.

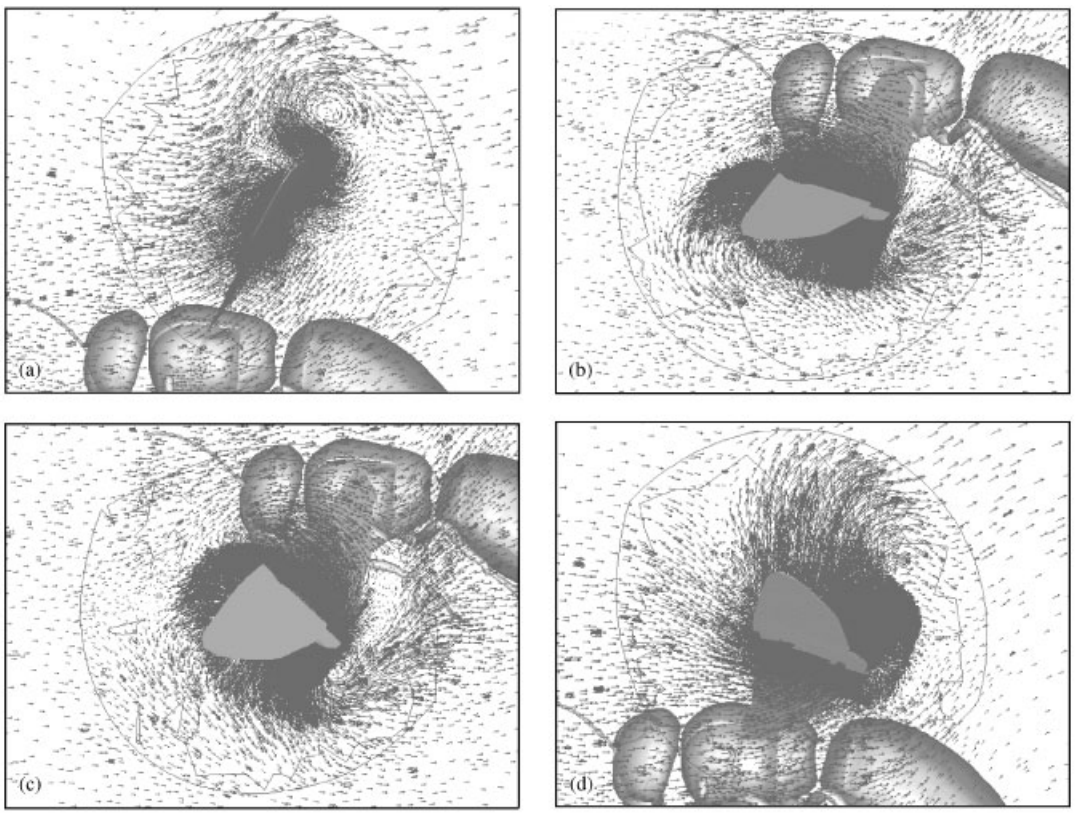


Figure 10. Computed velocity vectors near the wing at 70% semi-span location: (a) beginning of downstroke, (b) beginning of rotation at the end of downstroke motion, (c) beginning of upstroke, (d) beginning of rotation at the end of upstroke motion.

The wing motion must be specified in translational and rotational angles. However, a precise data of the wing motion is not available for a hornet in forward flight. In this paper, the wing motion was specified based on the experimental set-up of Dickinson [18]. Figure 9 shows the flapping motion of the wing with the outer boundaries of the wing grids after the hole cutting. The flapping plane is declined in 30° [17].

Figures 10 show the velocity vectors on a plane at 70% span location of the wing. Solid lines in the figures are the intergrid boundary lines of two overlapping grids. Figure 10(a) is the velocity vectors at the beginning of the downstroke where the wing turns over rapidly from the positive angle of attack to negative. This turn over creates a vortex near the trailing edge. A similar feature can be seen in Figure 10(c) where the wing starts the upstroke. This vortex shedding by the rotation of the wing at each end of up and down strokes may effectively generate lift and thrust as pointed out by Dickinson *et al*. [18]. Figure 10(b) is the flow vectors at the end of the downstroke and the wing is about to turn from negative angle of attack to positive. At this stage, a strong down wash can be observed due to the combined effect of the down stroke speed and the positive angular velocity.

Plate 4 shows the computed pressure distributions of a hornet with flapping wing. Both at the beginning and the end of the down stroke, relatively higher pressure on the lower surface of the wing can be observed. This is also due to the rapid rotation of the wing at each end of up and down strokes. It is also observed that relatively higher pressure region appears on the chest of the hornet when the wing finishes the down stroke and is about to move upward. It is considered that the wake of the down stroke and the angular motion of the wing cause such a higher pressure region on the lower surface of the body.

6. CONCLUSION

Overset unstructured grid method was successfully applied to simulate flowfields around separating airplane/rocket booster and a realistic model of a hornet with flapping wing in forward flight. Detailed surface meshing was achieved by the direct advancing front method coupled with geometrical feature extraction. Pre-treatment of the Stereolithography (STL) data is also an important key technology for these kinds of complex configurations. Overset unstructured grid approach allows the treatment of large movements of bodies without any difficulty.

The computed results of the flying hornet show qualitatively reasonable flow features. At each end of up and down strokes, vortexes are shed by the twisting of the wing that may effectively generate the lift and thrust. However, lift and thrust variations in time were not good at this stage. The movement of the wing was taken from an experimental set-up for a fly in hovering. To obtain reasonable lift and thrust with flapping motion, the specification of the flapping movement of the wing must be improved. Moreover, wings of insects are usually very thin and elastic. The deformation of the wing must be another important factor to effectively produce the lift and thrust for insects. The overset unstructured grid will be a practical approach even for such a flow simulation if it is coupled with a dynamic mesh to treat the wing deformation.

REFERENCES

1. Ito Y, Nakahashi K. Direct surface triangulation using stereolithography data. *AIAA Journal* 2002; **40**: 490–496.
2. Ito Y, Nakahashi K. Surface triangulation for polygonal models based on CAD data. *International Journal for Numerical Methods in Fluids* 2002; **39**:75–96.
3. Nakahashi K, Sharov D. Direct surface triangulation using the advancing front method. *AIAA Paper* 95-1686-CP 1995; 442–451.
4. Löhner R. Regridding surface triangulation. *Journal of Computational Physics* 1996; **126**:1–10.
5. Vinokur M. On one-dimensional stretching functions for finite-difference calculations. *Journal of Computational Physics* 1983; **50**:215–234.
6. Sharov D, Nakahashi K. A boundary recovery algorithm for delaunay tetrahedral meshing. *5th International Conference on Numerical Grid Generation in Computational Field Simulations*, 1996; 229–238.
7. Steger JL, Dougherty FC, Benek JA. A chimera grid scheme. *ASME Mini-Symposium on Advances in Grid Generation*, 1982.
8. Nakahashi K, Togashi F, Sharov D. An intergrid-boundary definition method for overset unstructured grid approach. *AIAA Journal* 2000; **38**:2077–2084.
9. Hariharan N, Wang ZJ, Buning P. Application of conservative chimera methodology in finite difference settings. *AIAA Paper* 97-0627, 1997.
10. Obayashi S, Guruswamy GP. Convergence acceleration of an aeroelastic Navier–Stokes solver. *AIAA Paper* 94-2268, 1994.
11. Venkatakrishnan V. On the accuracy of limiters and convergence to steady state solutions. *AIAA Paper* 93-0880, 1993.

12. Jameson A, Turkel E. Implicit schemes and LU decompositions. *Mathematics of Computation* 1981; **37**: 385–397.
13. Sharov D, Nakahashi K. Reordering of hybrid unstructured grids for lower-upper symmetric Gauss–Seidel computations. *AIAA Journal* 1998; **36**:484–486.
14. Nakahashi K, Sharov D, Kano D, Kodera M. Applications of unstructured hybrid grid method to high-Reynolds number viscous flows. *International Journal for Numerical Methods in Fluids* 1999; **31**:97–111.
15. Iwamiya T. NAL SST project and aerodynamic design of experimental aircraft. *Proceedings of the 4th ECCOMAS Computational Fluid Dynamics Conference*, Athens. John Wiley: Chichester, 1998; 580–585.
16. Togashi F, Nakahashi K, Ito Y, Iwamiya T, Shinbo Y. Flow simulation of NAL experimental supersonic airplane/booster separation using overset unstructured grids. *Computers and Fluids* 2001; **30**:673–688.
17. Azuma A. *A Dictionary About the Moving of Creatures*. Asakura Syoten Publishing House: Tokyo, 1997 (in Japanese).
18. Dickinson MH, Lehmann F-O, Sane D. Wing rotation and the aerodynamic basis of insect flight. *Science* 1999; **284**:1954–1960.

