INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **31**: 97–111 (1999)

APPLICATIONS OF UNSTRUCTURED HYBRID GRID METHOD TO HIGH-REYNOLDS NUMBER VISCOUS FLOWS

KAZUHIRO NAKAHASHI[a,*], DMITRI SHAROV[b], SHINTARO KANO[a] AND MASATOSHI KODERA[a]

[a] *Department of Aeronautics and Space Engineering, Tohoku University, Aoba-yama 01, Sendai 980-8579, Japan*
[b] *Research Center for Computational Science, Fujitsu Limited, Chiba 261, Japan*

SUMMARY

In this paper, an unstructured hybrid grid method is discussed for its capability to compute three-dimensional compressible viscous flows of complex geometry. A hybrid of prismatic and tetrahedral grids is used to accurately resolve the wall boundary layers for high-Reynolds number viscous flows. The Navier–Stokes equations for compressible flows are solved by a finite volume, cell–vertex scheme. The LU-SGS implicit time integration method is used to reduce the computational time for very fine grids in boundary layer regions. Two kinds of one-equation turbulence models are evaluated here for their accuracy. The method is applied to computations of transonic flows around the ONERA M5 airplane and ONERA M6 wing, and supersonic shock/boundary layer interacting flows inside a scramjet inlet to validate the accuracy and efficiency of the method. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: unstructured grid; Navier–Stokes equations; LU-SGS scheme; one-equation turbulence model; transonic flow; supersonic flow; scramjet

1. INTRODUCTION

Computational fluid dynamics (CFD) has become an indispensable tool for the aerodynamic design and analysis of airplanes and engines. However, engineering applications of the conventional CFD based on structured and block structured grids have been limited to relatively simple components because of the difficulty and cost in generating grids about three-dimensional complex geometries. Recently, tetrahedral unstructured grids are becoming more popular for complex geometry problems because of their several advantages over the conventional structured grids. The tetrahedral grid is flexible enough to treat complex geometries and has an adaptive refinement/unrefinement capability, thus saving the total number of grid points. Owing to the recent developments of automatic surface and volume grid generations, tetrahedral unstructured grids are widely used to solve inviscid flow and low Reynolds number viscous flow problems.

However, the application of the tetrahedral unstructured grid method to high-Reynolds number viscous flows has several critical issues. An accurate resolution of thin boundary layers developed along the wall surface requires very fine grids, which causes a stiffness problem of

---

[*] Correspondence to: Department of Aeronautics and Space Engineering, Tohoku University, Aoba-yama 01, Sendai 980-8579, Japan.

CCC 0271–2091/99/170097–15$17.50

the flow solver. Moreover, the generation of such anisotropic stretched grids without obtuse grid elements near the wall is another critical issue of the unstructured grid methods. One promising approach is to use a hybrid grid [1,2], which is comprised of structured or semi-structured grids for viscous regions and tetrahedral unstructured grids for the rest of computational domain. Although the concept of the hybrid grid is not very new and several hybrid methods appear in the literature, their verification in accuracy, efficiency and capability to treat real engineering problems seems to be relatively unexplored. The paper is aimed at contributing to the verification issue.

In this paper, applications of the unstructured hybrid grid method to compute compressible viscous flows about complex geometries are discussed. The computational region is discretized by a hybrid grid for accurate and efficient computations, as well as the easiness of the grid generation for complex configurations. The hybrid grid for three-dimensional problems consists of prismatic and tetrahedral grids. The prismatic semi-structured grid is generated around viscous boundary surfaces and covers viscous regions, while the tetrahedral grid covers the rest of computational domain. The solution algorithm to compute compressible Navier–Stokes equations is based on an upwind, finite volume method. The stiffness problem due to very fine grids for high-Reynolds number flows is overcome by the LU-SGS implicit time integration method [3].

The present approach has been applied to several external and internal flow problems for the validation of the method. For transonic flow problems, the method is applied to flows around the ONERA M5 airplane and the ONERA M6 wing. These are well-known test cases for validating the accuracy and efficiency of the code. The method is also applied to supersonic shock/boundary layer interacting flow fields, where the accuracy of the viscous term computation including turbulence models becomes important. First, the accuracy and efficiency of the method are discussed for a supersonic flow of double fins on a flat plate as a test case for turbulence models. Then the unstructured hybrid grid is applied to a realistic problem of a scramjet engine in order to validate its capability for three-dimensional high-Reynolds number flow problems.

2. SOLUTION ALGORITHM

2.1. Finite volume discretization

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

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

where $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, e]^T$ is the vector of conservative variables; ρ is the density; u, v, w are the velocity components in the x-, y-, z-directions; and e is the total energy. The vectors $\mathbf{F}(\mathbf{Q})$and $\mathbf{G}(\mathbf{Q})$ represent the inviscid and viscous flux vectors respectively 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 whose boundary is shown in Figure 1. The boundary surface is defined by the tetrahedra centroid C, face centroids B and D, and mid-point of edge A. A similar approach is used to construct dual cells for prisms and pyramids using their centroids, centroids of their faces and mid-points of the edges.

With this control volume, Equation (1) can be written in an algebraic form as follows:

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

where ΔS_{ij} is a segment area of the control volume boundary associated with 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.

2.2. Convective flux computation

The numerical flux $\mathbf{h}$ is computed using an approximate Riemann solver of Harten–Lax–van Leer–Einfeldt–Wada [4]. If we evaluate $\mathbf{Q}_{ij}^{\pm}$ in $\mathbf{h}$ at both end points of each edge, we get the first-order scheme. The second-order spatial accuracy is realized by a linear reconstruction of the primitive gas dynamic variables $\mathbf{q} = [\rho, u, v, w, p]^T$ inside the control volume using the following equation:

$$\mathbf{q}(\mathbf{r}) = \mathbf{q}_i + \psi_i \nabla \mathbf{q}_i \cdot (\mathbf{r} - \mathbf{r}_i) \quad (0 \le \psi \le 1), \tag{3}$$

where $\mathbf{r}$ is a vector pointing to point (x, y, z), and i is the node number. The gradients associated with the control volume centroids are volume-averaged gradients computed by the surrounding grid cells

$$\nabla \mathbf{q}_i = \frac{\sum_{e(i)} \nabla \mathbf{q}_e V_e}{\sum_{e(i)} V_e}, \tag{4}$$

where index e denotes the identity of the grid cell (tetrahedron, prism or pyramid) sharing node i. Unlike tetrahedra, the gradients for prismatic and pyramidal cells are not unique, hence prisms and pyramids are divided into tetrahedra to compute the gradients.

Venkatakrishnan's limiter [5] is used because of its superior convergence properties,

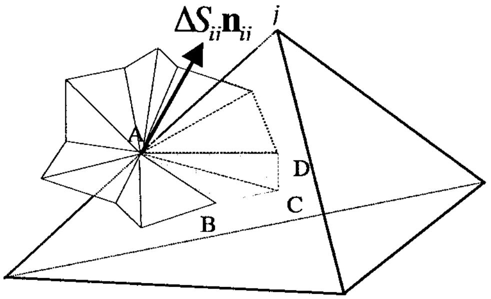


Figure 1. Control volume.

$$\psi_i = \min_{j(i)} \begin{cases} \dfrac{\Delta_{\max}^2 + \varepsilon^2 + 2\Delta_-\Delta_{\max}}{\Delta_{\max}^2 + 2\Delta_-^2 + \Delta_{\max}\Delta_- + \varepsilon^2}, & \text{if } \Delta_- > 0 \\ \dfrac{\Delta_{\min}^2 + \varepsilon^2 + 2\Delta_-\Delta_{\min}}{\Delta_{\min}^2 + 2\Delta_-^2 + \Delta_{\min}\Delta_- + \varepsilon^2}, & \text{if } \Delta_- < 0 \end{cases}, \tag{5}$$

where

$$\begin{aligned} &\Delta_- = \mathbf{q}_{ij} - \mathbf{q}_i = \nabla\mathbf{q}_i \cdot (\mathbf{r}_{ij} - \mathbf{r}_i), \\ &\Delta_{\max} = \max_{j(i)}(\mathbf{q}_j - \mathbf{q}_i), \qquad \Delta_{\min} = \min_{j(i)}(\mathbf{q}_j - \mathbf{q}_i), \\ &\varepsilon^2 = (K\overline{\Delta l})^2, \end{aligned} \tag{6}$$

where $\overline{\Delta l}$ is an average cell length and K is a constant whose value is selected between 0.1 and 0.3.

2.3. Viscous flux computation

To compute viscous stress and heat flux terms in $\mathbf{G}(\mathbf{Q})$, it is necessary to evaluate spatial derivatives of the primitive variables u, v, w at each control volume face. In this paper, the spatial derivatives are evaluated directly at the edges. The gradient of a scalar ϕ is computed for each edge as follows:

$$\nabla\phi_E = \frac{\sum_{e(E)} \nabla\phi_e V_e}{\sum_{e(E)} V_e}, \tag{7}$$

where the index E denotes edge; the summation is performed over all elements (tetrahedra, prisms or pyramids) sharing the edge E.

The turbulent kinetic viscosity is evaluated by two one-equation turbulence models, the Goldberg–Ramakrishnan model (G-R) [6] and the Spalart–Allmaras model (S-A) [7]. These turbulence models do not need a searching procedure along normal lines-to-wall as required in the Baldwin–Lomax algebraic model, so that they are suitable for unstructured grid methods to treat complex geometries.

In the G-R model, turbulent kinetic viscosity v_{t} is given by

$$v_{\mathrm{t}} = C_\mu f_\mu \cdot (vR_{\mathrm{T}}). \tag{8}$$

The transport variable vR_{T} is computed by the following equation:

$$\frac{\mathrm{D}(vR_{\mathrm{T}})}{\mathrm{D}t} = (C_{\varepsilon 2} f_2 - C_{\varepsilon 1})(vR_{\mathrm{T}}P)^{1/2} + \left(v + \frac{v_{\mathrm{t}}}{\sigma_\varepsilon}\right)\nabla^2(vR_{\mathrm{T}}) - \nabla v_t \cdot \frac{\nabla(vR_{\mathrm{T}})}{\sigma_\varepsilon}. \tag{9}$$

For the S-A model, turbulent kinetic viscosity v_{t} is given as

$$v_{\mathrm{t}} = \tilde{v} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{v}}{v}. \tag{10}$$

The transport variable $\tilde{v}$ in the above equation is given by solving the following equation:

$$\frac{\mathrm{D}\tilde{v}}{\mathrm{D}t} = c_{b1}\tilde{S}\tilde{v} + \frac{1}{\sigma}\nabla\cdot[(v + \tilde{v})\nabla\tilde{v}] + \frac{c_{b2}}{\sigma}(\nabla\tilde{v})^2 - c_{w1} f_w \left(\frac{\tilde{v}}{d}\right)^2, \tag{11}$$

where d is the shortest distance of the field point to the wall. The transition terms in the original S-A model was neglected here because of simplicity. The initial value of the transport variable of $20.0/C_\mu$ is used in the G-R model, and $0.1/C_\mu$ in the S-A model. The turbulence equation, Equations (9) or (11), is integrated at each gas dynamic step by the same LU-SGS method.

2.4. LU-SGS method

Implicit time integration is required for computing high-Reynolds number flows because of severe CFL conditions due to very fine grids near the wall boundary. Here, the LU-SGS implicit method [3] is applied to integrate Equation (2) in time. With $\Delta\mathbf{Q} = \mathbf{Q}^{n+1} - \mathbf{Q}^n$, an implicit time integration of Equation (2) can be written as

$$\frac{V_i}{\Delta t}\Delta\mathbf{Q}_i = -\sum_{j(i)}\Delta S_{ij}\mathbf{h}_{ij}^{n+1} + \sum_{j(i)}\Delta S_{ij}\mathbf{G}_{ij}^n. \tag{12}$$

By linealizing and splitting the numerical flux vector 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 [8] leads to the following equations:

$$\left(\frac{V_i}{\Delta t}\mathbf{I} + \sum_{j(i)}\Delta S_{ij}\mathbf{A}_i^+\right)\Delta\mathbf{Q}_i + \sum_{j(i)}\Delta S_{ij}\mathbf{A}_j^-\Delta\mathbf{Q}_j = \mathbf{R}_i, \tag{13}$$

where

$$\mathbf{R}_i = -\sum_{j(i)}\Delta S_{ij}\mathbf{h}_{ij}^n + \sum_{j(i)}\Delta S_{ij}\mathbf{G}_{ij}^n. \tag{14}$$

The LU-SGS method on an unstructured grid can be derived by splitting node points $j(i)$ into two groups, $j\in L(i)$ and $j\in U(i)$, for the second summation in left-hand-side of Equation (13).

$$\left(\frac{V_j}{\Delta t}\mathbf{I} + \sum_{j(i)}\Delta S_{ij}\mathbf{A}_i^+\right)\Delta\mathbf{Q}_i + \sum_{j\in L(i)}\Delta S_{ij}\mathbf{A}_j^-\Delta\mathbf{Q}_j + \sum_{j\in U(i)}\Delta S_{ij}\mathbf{A}_j^-\Delta\mathbf{Q}_j = \mathbf{R}_i. \tag{15}$$

This can be solved by the following two sweeps:

$$\text{Forward sweep: } \Delta\mathbf{Q}_i^* = \mathbf{D}^{-1}\left[\mathbf{R}_i - \sum_{j\in L(i)}\Delta S_{ij}\mathbf{A}_j^-\Delta\mathbf{Q}_j^*\right] \tag{16a}$$

$$\text{Backward sweep: } \Delta\mathbf{Q}_i = \Delta\mathbf{Q}_i^* - \mathbf{D}^{-1}\sum_{j\in U(i)}\Delta S_{ij}\mathbf{A}_j^-\Delta\mathbf{Q}_j, \tag{16b}$$

where

$$\mathbf{D} = \left(\frac{V_j}{\Delta t}\mathbf{I} + \sum_{j(i)}\Delta S_{ij}\mathbf{A}_i^*\right). \tag{17}$$

The term $\mathbf{D}$ is diagonalized by using the Jameson–Turkel approximation of the Jacobian [9] as $\mathbf{A}^{\pm} = 0.5(\mathbf{A} \pm \rho_A\mathbf{I})$, where ρ_A is a spectral radius of Jacobian $\mathbf{A}$. For viscous flows, ρ_A is replaced by

$$\rho_A^* = \rho_A + 2\,\frac{\mu + \mu_{\mathrm{T}}}{Re\cdot\rho\cdot h}, \tag{18}$$

where μ and μ_{T} are kinetic and turbulent viscosities, and h is the distance between i and j.

Using the fact [10] that $\Sigma_{j(i)} \Delta S_{ij}\mathbf{A} = 0$, Equation (17) becomes a diagonal matrix

$$\mathbf{D} = \left(\frac{V_j}{\Delta t} + 0.5 \sum_{j(i)} \Delta S_{ij}\rho_A\right)\mathbf{I}. \tag{19}$$

With these simplifications, the final form of the LU-SGS method for the unstructured grid becomes

$$\text{Forward sweep: } \Delta\mathbf{Q}_i^* = \mathbf{D}^{-1}\left[\mathbf{R}_i - 0.5 \sum_{j\in L(i)} \Delta S_{ij}(\Delta\mathbf{h}_j^* - \rho_A \Delta\mathbf{Q}_j^*)\right] \tag{20a}$$

$$\text{Backward sweep: } \Delta\mathbf{Q}_i = \Delta\mathbf{Q}_i^* - 0.5\mathbf{D}^{-1} \sum_{j\in U(i)} \Delta S_{ij}(\Delta\mathbf{h}_j - \rho_A \Delta\mathbf{Q}_j), \tag{20b}$$

where $\Delta\mathbf{h} = \mathbf{h}(\mathbf{Q} + \Delta\mathbf{Q}) - \mathbf{h}(\mathbf{Q})$.

The lower/upper splitting of Equation (15) for the structured grid is realized by using the hyper planes $i + j + k = \text{constant}$. For the unstructured grid, there are no grid lines so that another strategy is necessary to perform the sweeps. In Reference [10] it was proposed to sweep through node numbers from node 1 to N and backwards. Namely, lower nodes $j \in L(i)$ are surrounding nodes whose number is less than i. The upper nodes $j \in U(i)$ are surrounding nodes whose number exceeds the current node number i. Here we employs a grid reordering technique to improve the convergence and the vectorization. Details of this grid reordering are described in Reference [3].

3. GRID GENERATION

3.1. *Surface grid generation*

Surface grids of three-dimensional bodies are generated by the direct surface meshing method [11]. This method applies the advancing front method directly to the body surface in the physical space. It does not use a mapping so that the mesh size can be automatically controlled by adapting to the local surface curvature. Also, without mapping, surface definition for meshing can be more flexible.

3.2. *Volume grid generation*

The hybrid grids comprised of tetrahedra, prisms and pyramids are generated by the method described in Reference [12]. The prismatic semi-structured grid is generated around viscous boundary surfaces and covers viscous regions, while the tetrahedral grid covers the rest of the computational domain. The Delaunay incremental insertion method for tetrahedral grid generation is used. The prismatic grid is structured in directions normal to the boundary faces but the number of prisms generated from one boundary face is variable from face to face. Unlike conventional prismatic grid generators, this technique works well even in regions of cavities and gaps. Pyramidal elements appear between tetrahedral and prisms. This method has shown its robustness for a variety of geometries without user intervention. Only boundary surface triangulation has to be specified.

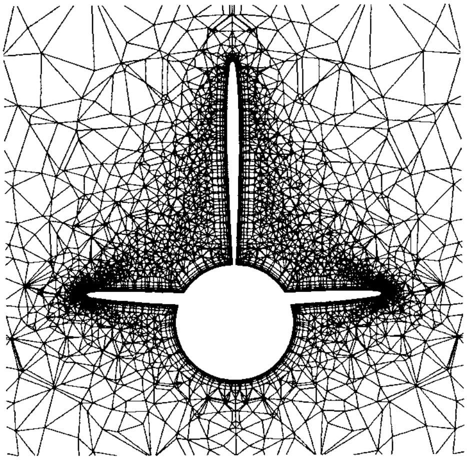


Figure 2. Cut view of hybrid grid near the tail of ONERA M5 airplane.

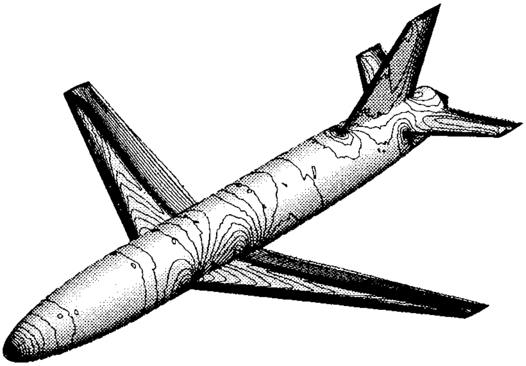


Figure 3. Computed pressure contours of ONERA M5.

4. COMPUTED RESULTS

4.1. Transonic viscous flow past ONERA M5

Viscous flow around the ONERA M5 configuration was computed by the present method. The hybrid grid of this geometry contains 367579 nodes, 514291 tetrahedra, 536601 prisms and 1020 pyramids. A cut view of the grid near the tail of the airplane is shown in Figure 2. The flow is computed with free-stream conditions of $M_\infty = 0.84$, angle of attack $\alpha = -1°$, $Re = 10^6$. The pressure distribution along the airplane surface using the one-equation turbulence model of Goldberg–Ramakrishnan is shown in Figure 3. Comparison of the computed pressure coefficients with the experimental result [13] showed good agreement.

The convergence histories of the Navier–Stokes as well as the Euler computations are shown in Figure 4. The CFL number used for these computations was 10^5. Converged solutions were obtained in about 2000 iterations. The convergence for the NS computation stalled as compared with the Euler computation. This was mainly due to the grid irregularity near the interface between the prismatic grid and the tetrahedral grid. The implicit part of the method takes 70% of the time required for explicit flux computations. The method is approximately ten times faster compared with the explicit method with local time stepping for the NS computation. Extra storage required for the implicit method is approximately 5% of the total memory requirements.

4.2. Transonic viscous flow past ONERA M6

The ONERA M6 is a wing model frequently used to validate the Navier–Stokes codes. The computed result of this configuration is very sensitive to the grid density, so that a very fine grid was used for the present computation. The hybrid grid of this geometry contains 830476 nodes, 815687 tetrahedra, 1340338 prisms and 7363 pyramids. The minimum spacing of the prismatic grid normal to the wall is 2×10^{-5}. The flow is computed with free-stream conditions of $M_\infty = 0.84$, angle of attack $\alpha = 3.06°$, $Re = 1.46 \times 10^7$. The turbulent boundary layer was assumed on the entire wing surface and the Goldberg–Ramakrishnan turbulence model was used.

The computed pressure coefficients for several spanwise locations are shown in Figure 5 together with the inviscid result and the experimental result [14]. The inviscid result was obtained with the same surface grid but a fully tetrahedral grid. Cut views of the grids and the computed pressure contours at 65% spanwise location are shown in Figure 6. As shown in Figures 5 and 6, the computed result with the hybrid grid gives better resolution even with the same surface grid. The prismatic grid retains the surface grid density to some extent from the surface. Another important fact is that the use of the prismatic grid in the boundary layer region significantly saves the total number of grid cells as compared with the fully tetrahedral grid for the same spatial resolution. If we generate a tetrahedral viscous grid by dividing the prismatic cells of the present hybrid grid, the total number of cells will become three times as many as the present grid. Therefore, a hybrid of prismatic and tetrahedral grids has a big advantage over the fully tetrahedral grid in memory requirement and computational time for high-Reynolds number viscous flow problems.

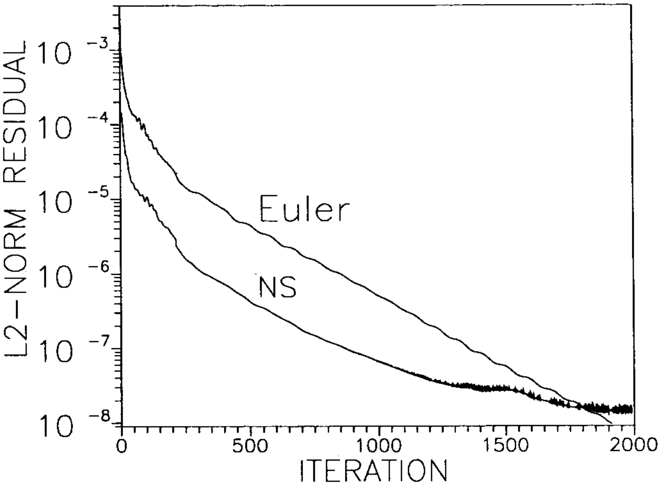


Figure 4. Convergence history of ONERA M5 computations.

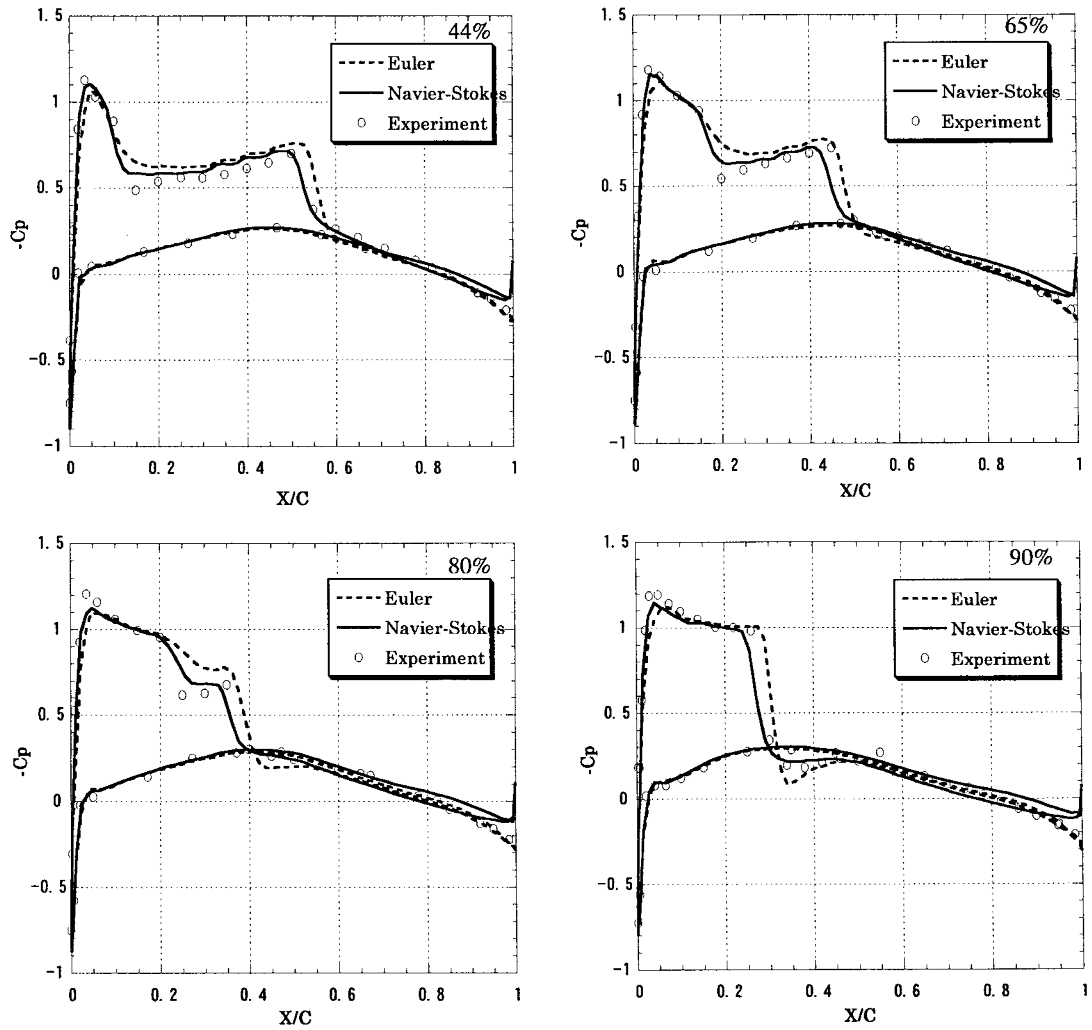


Figure 5. Comparison of pressure coefficients of ONERA M6 wing.

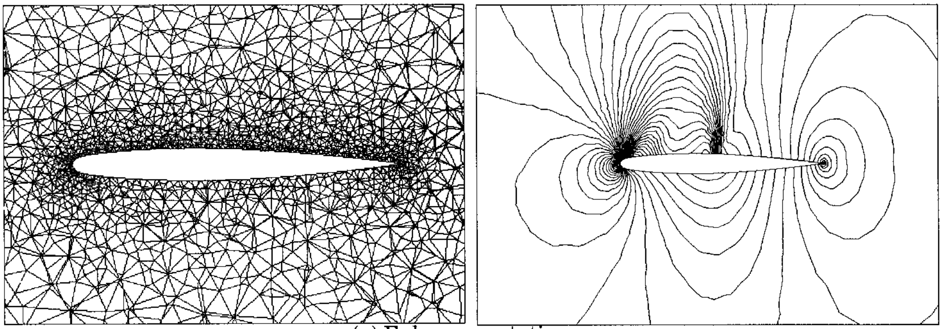
4.3. *Shock/boundary layer interactions between double fines*

The accuracy of the present method with turbulence models for a shock/boundary layer interaction flow field was investigated for a supersonic flow between double fins on a flat plate, as shown in Figure 7. Flow conditions are $M_\infty = 2.95$ and $Re = 2.5 \times 10^5$ based on a thickness of the inflow boundary layer, and the adiabatic wall is assumed. The boundary layer properties are specified at the inflow boundary position, whose distance is $5\delta_\infty$ from the fin leading edge. The boundary layer properties at the inlet boundary were obtained by solving a flat plate flow field by the present method beforehand.

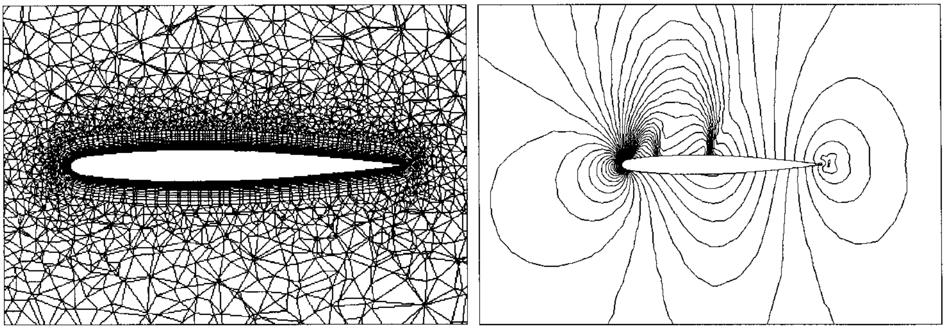
The computed density contours at several flow sections using the Goldberg–Ramakrishnan turbulence model are shown in Figure 8. It is clearly depicted that the shock waves generated at the sharp edge of the fin interact with the boundary layer on the flat plate. The pressure distributions on the flat plate are shown in Figure 9, where experimental and computed results

[15] using the conventional structured grids are also shown for comparison. The present results, with both the Goldberg–Ramakrishnan and the Spalart–Allmaras turbulence models, show better agreements with the experimental results.

The number of total grid points is about 200000 and the necessary memory is 34 M words. The computation is converged after iterating about 3000 times. The computational cost is about 6 h for 1 CPU on NEC SX4.



(a) Euler computation



(b) Navier-Stokes computation

Figure 6. Cut views of grids and pressure contours at 65% span location.

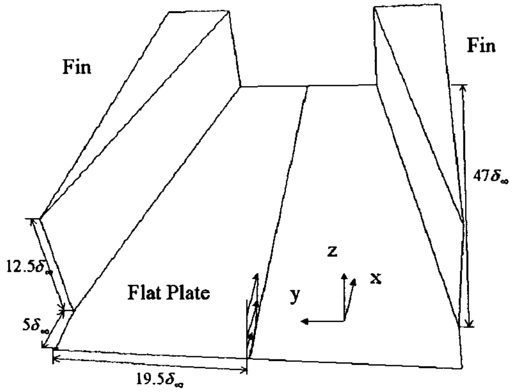


Figure 7. Double fin flow field.

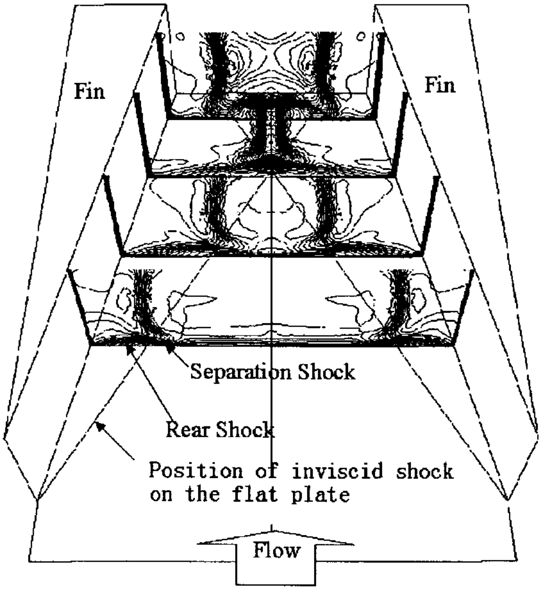


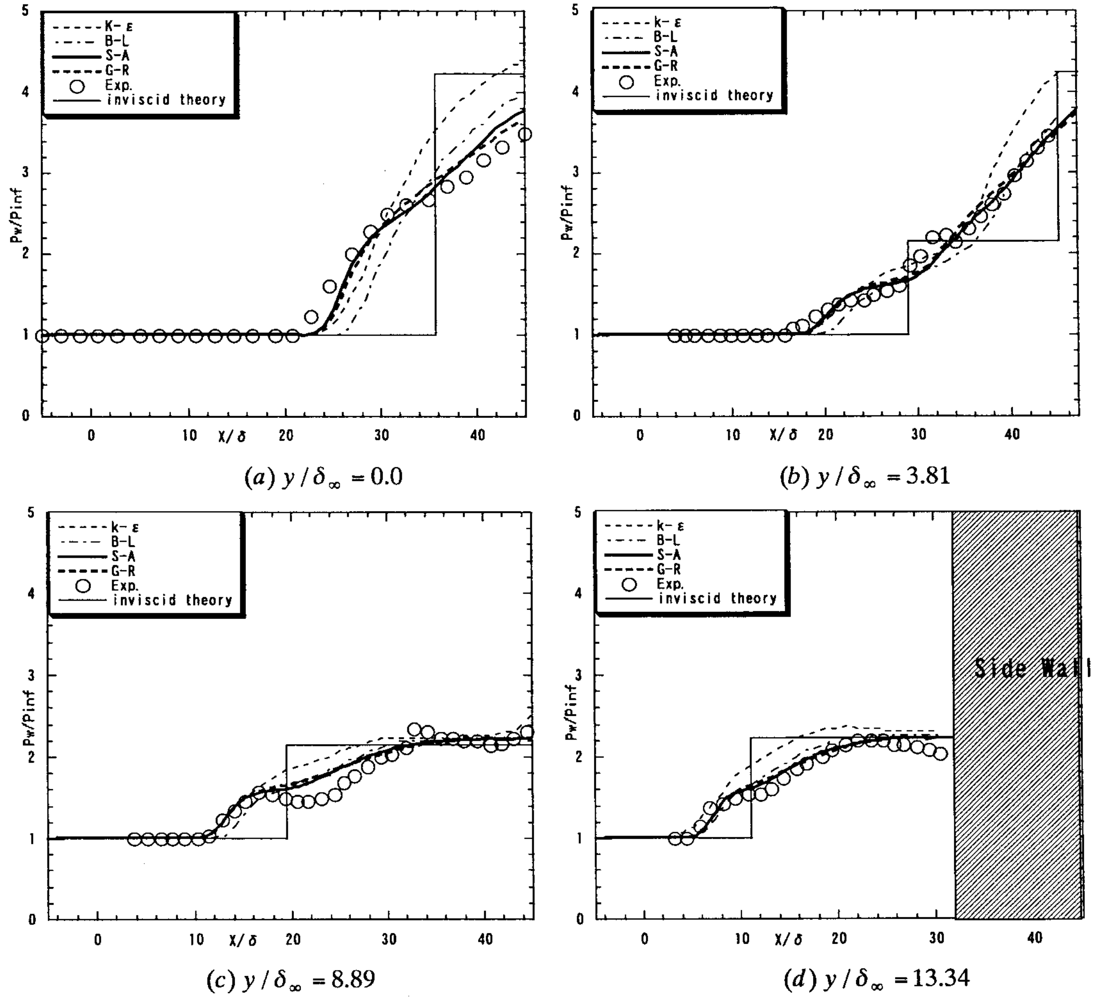
Figure 8. Density contours of double fin flow field.

4.4. Internal flows of SCRAM jet engines

As a practical application, an internal flow of a scramjet (supersonic combustion ram jet) engine was computed by the present method. The experimental research [16] had been conducted at the National Aerospace Laboratory of Japan, Kakuda Research Center. The model engine is schematically shown in Figure 10, and the hybrid grid for this model is shown in Figure 11. Backward-facing steps are used for fuel injection between inlet and isolator. A short strut is integrated on the top wall of the throat section to attain intensive combustion. The flow was computed for a cold flow condition with inflow Mach number of 5.4 and Reynolds number of 1.515×10^6. The reference length L is the height of the inlet entrance and the inflow boundary layer thickness is $0.23L$, which was taken from the experiment. The boundary layer properties were specified at the inflow boundary position and those properties were obtained by the same way with the previous section. The wall temperature was kept constant at 300 K. The Goldberg–Ramakrishnan turbulence model was used for the computation.

The computed pressure contours is shown in Figure 12 and a comparison of the pressure distributions on the wall with the experiment is shown in Figure 13. Even with the complex feature of the flow field due to the shock wave/boundary layer interactions, the result shows good agreement with the experimental data. Figure 14 shows oil flow patterns on the side wall in the both cases of the experiment and the computation. The computed oil flow depicts the essential feature of the experimental oil flow.

Although the configuration of the scramjet engine can be treated by zonal structured grid methods, the singular lines and points appeared in the structured grids often cause a difficulty in convergence for high Mach number computations. In our experience for both structured and unstructured grid methods for this configuration, the total CPU time required for the present hybrid grid computation was much smaller than that of the structured grid method.



(*a*) $y/\delta_\infty = 0.0$ (*b*) $y/\delta_\infty = 3.81$

(*c*) $y/\delta_\infty = 8.89$ (*d*) $y/\delta_\infty = 13.34$

Figure 9. Pressure distributions on the flat plate.

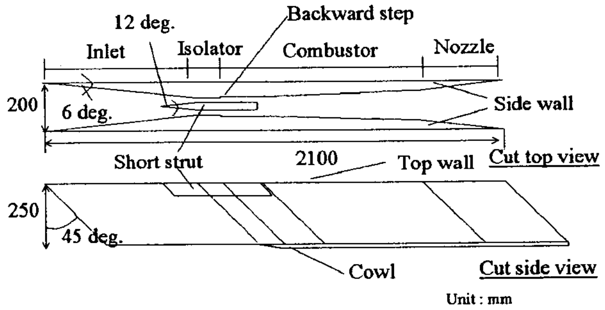


Figure 10. Schematic figure of scramjet.

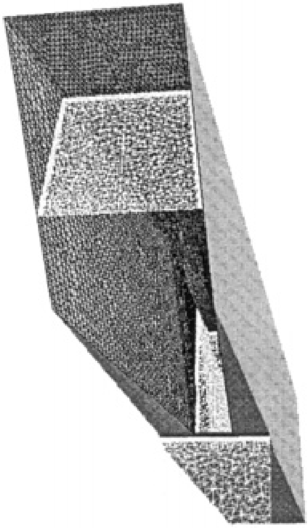


Figure 11. Hybrid grid of scramjet inlet.

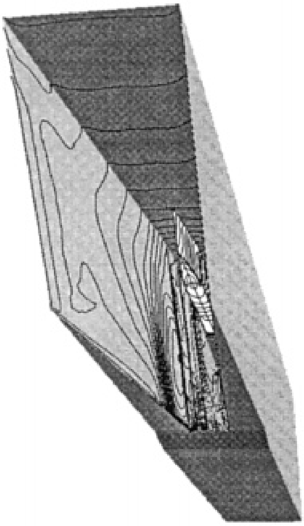
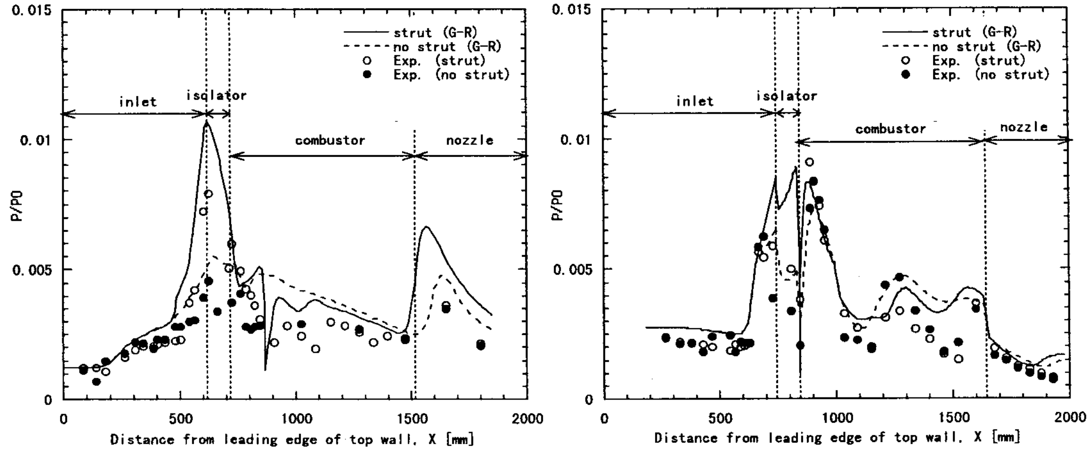


Figure 12. Pressure contours of scramjet inlet.

5. CONCLUSIONS

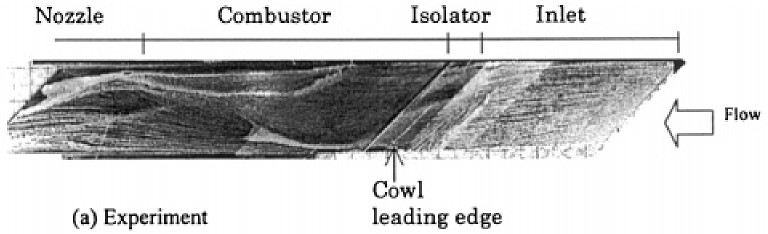
Capabilities of the unstructured hybrid grid method have been estimated by its application to several high Reynolds number viscous flow problems. The method is based on a finite volume, cell–vertex scheme and LU-SGS implicit method. Computations of transonic flows around airplane and wing, as well as supersonic flows inside a scramjet engine have validated the capability of the method in predicting the high-Reynolds number flows accurately and efficiently. Although the unstructured grid method has an inherent problem of memory overhead, the solution quality and the computational efficiency for the Navier–Stokes



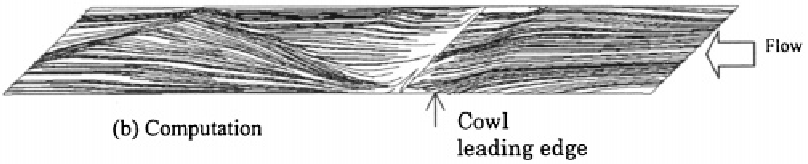
(a) Top wall center line

(b) Side wall center line

Figure 13. Pressure distributions on the wall.



(a) Experiment



(b) Computation

Figure 14. Experimental and computed oil flow patterns on the side wall.

equations became the same level of those by the conventional structured grid method. With the grid flexibility and the easy refinement capability, the advantage of the unstructured grid method is apparent for computations of complex three-dimensional configurations.

REFERENCES

1. K. Nakahashi, 'Prismatic grid method', in H. Hafez and K. Oshima (eds.), *Computational Fluid Dynamics Review 1995*, Wiley, New York, 1995, pp. 87–105.
2. V. Parthasarathy and Y. Kallinderis, 'Adaptive prismatic–terahedral grid refinement and redistribution for viscous flows', *AIAA J.*, **34**, 707–716 (1996).
3. D. Sharov and K. Nakahashi, 'Reordering of hybrid unstructured grids for lower–upper symmetric Gauss–Seidel computations', *AIAA J.*, **36**, 484–486 (1998).
4. S. Obayashi and G.P. Guruswamy, 'Convergence acceleration of an aeroelastic Navier–Stokes solver', *AIAA Paper 94-2268*, 1994.
5. V. Venkatakrishnan, 'On the accuracy of limiters and convergence to steady state solutions', *AIAA Paper 93-0880*, 1993.

6. U.C. Goldberg and S.V. Ramakrishnan, 'A pointwise version of Baldwin–Barth turbulence model', *Comput. Fluid Dyn.*, **1**, 321–338 (1993).
7. P.R. Spallart and S.R. Allmaras, 'A one-equation turbulence model for aerodynamic flows', *AIAA Paper 92-0439*, 1992.
8. A. Jameson and S. Yoon, 'Lower–upper implicit schemes with multiple grids for the Euler equations', *AIAA J.*, **25**, 929–935 (1987).
9. A. Jameson and E. Turkel, 'Implicit schemes and LU decompositions', *Math. Comput.*, **37**, 385–397 (1981).
10. I. Men'shov and Y. Nakamura, 'Implementation of the LU-SGS method for an arbitrary finite volume discretization', *Proc. of Japanese 9th CFD Symposium*, 1995, pp. 123–124.
11. K. Nakahashi and D. Sharov, 'Direct surface triangulation using the advancing front method', *Proc. 12th AIAA CFD Conference*, *AIAA 95-1686-CP*, 1995, pp. 442–451.
12. D. Sharov and K. Nakahashi, 'Hybrid prismatic/tetrahedral grid generation for viscous flow applications', *AIAA J.*, **36**, 157–162 (1998).
13. Proceedings of the 10th NAL Symposium on Aircraft Computational Aerodynamics, CFD Workshop on GK Airfoil and ONERA M5 Geometry, Tokyo, 1992.
14. V. Schmitt and F. Charpin, 'Pressure distribution on the ONERA M6 wing at transonic Mach numbers', *AGARD-AR-138*, **B1**, 1979.
15. N. Narayanswami, D.D. Knight, S.M. Bogdonoff and C.C. Horstman, 'Interaction between crossing oblique shocks and a turbulent boundary layer', *AIAA J.*, **30**, 1945–1952 (1992).
16. T. Kanda, T. Hiraiwa, T. Mitani, S. Tomioka and N. Chinzei, 'Mach 6 testing of a scramjet engine model', *J. Propul. Power*, **13**, 543–551 (1997).