AUTOMATIC GRID GENERATION FOR ACCURATE NAVIER-STOKES SIMULATIONS

Paulus R. Lahur*, Atsushi Hashimoto, Keiichi Murakami**, Takashi Aoyama****
***Research Center of Computational Mechanics, Inc. (RCCM), Japan**
**** Japan Aerospace Exploration Agency (JAXA)**

Keywords: *Unstructured, hybrid, hexahedra, prismatic*

Abstract

In this study, we investigated the performance of our automatic grid generation code in Navier-Stokes simulation of a transonic aircraft model. This was done in the form of participation in the 4th Drag Prediction Workshop (DPW4), where NASA Common Research Model (CRM) was used. The grid generation method presented here is based on a hybrid of Cartesian and prismatic grids. The grid generator (known as HexaGrid) requires only a small number of parameters, and it can run automatically. Comparison with the results from tetrahedral-based unstructured grid and multi-block structured grid, which we also submitted to the workshop, suggests that the Cartesian-prismatic hybrid grid method can be very competitive. Drag, as well as other measure of aerodynamic performance, was predicted within reasonable accuracy, even in comparison with structured grid generated manually.

1 Introduction

Grid generation is a critical step in Computational Fluid Dynamics, because it determines not only the accuracy of the result, but also the overall computation time and effort. For Navier-Stokes simulation on a non-trivial geometry, accuracy and computation effort are two competing requirements, where one must be sacrificed at the expense of the other. If computational accuracy is the primary objective, then one usually has to manually generate multi-block structured grid, which takes a long time and requires highly skilled individuals. It is not uncommon for such procedure to take weeks. On the other hand, if saving in computation effort and fast turn-around cycle is the primary objective, then automatic grid generation method is used. The grid is usually either tetrahedral-based or hexahedral-based unstructured grid. The computational accuracy is generally lower than that of the grid generated manually by expert. Thus, a grid generator that can offer the best of both worlds will be a very useful tool indeed.

Motivated by this, Japan Aerospace Exploration Agency (JAXA) and Research Center of Computational Mechanics, Inc. (RCCM) have been developing a grid generator that is both automatic and suitable for accurate Navier-Stokes flow computation. The resulting code is called "HexaGrid," which generates mainly hexahedral grid cells [1, 2]. The grid is a hybrid between Cartesian grid and prismatic grid, where the Cartesian grid fills the far-field region and the prismatic grid fills the region around solid surface to resolve boundary layer flow.

Cartesian grid was chosen due to its speed in filling computational domain, because it does not follow body surface (non-body-fitted) [1-8].

Due to the way the prismatic grid is generated, the grid generator tolerates small defects in body surface, such as small gaps and overlapping elements, which is not uncommon for geometry represented in STL (Stereo Lithography) format [1, 2, 8].

One of the computation targets presented here is aerodynamic force prediction of a benchmark aircraft model known as NASA's common research model (NASA CRM) [9], first

introduced in the 4th Drag Prediction Workshop (DPW4) [10].

A flow solver known as TAS (Tohoku university Aerodynamic Simulation) code is used in this study [11]. The code solves Full Navier-Stokes equations on unstructured grid which may consist of tetrahedral, hexahedral and pyramid cells. It uses cell-vertex finite volume method. For comparison purpose, the computation results from other types of grid are used: multi-block structured grid (generated manually using Gridgen and solved using UPACS, JAXA's in-house flow solver [19]) and a tetrahedral-based unstructured grid (generated using a code called Mixed-Element Grid Generator in 3 Dimensions, MEGG3D [18], and solved using TAS).

2 Grid Generation Method

We use the grid generation software, HexaGrid, in this study, which is a commercial software jointly developed by Japan Aerospace Exploration Agency (JAXA) and Research Center of Computational Mechanics, Inc. (RCCM). It is capable of automatically generating hexahedra grid around solid surface discretized in triangles, in STL format. Virtually all CAD software can output surface geometry in STL format. The procedures of the grid generation are outlined below.

2.1 Cartesian Grid Generation

Cartesian grid is generated by means of successive local refinement. This step starts with one cell that covers the whole computational domain, whose size is set by user. In three-dimensional space, each refinement divides a cell isotropically into eight child cells of equal size and shape. At the beginning, the grid is locally refined until the size of cells intersecting solid surface is smaller than a maximum grid size set by user. Then the grid is further refined until the size of cells intersecting the solid surface with large curvature reaches a minimum grid size. In addition, we can also control grid size anywhere using "Refinement Box" with HexaGrid GUI. For simplicity, we did not use the Refinement Box in this study.

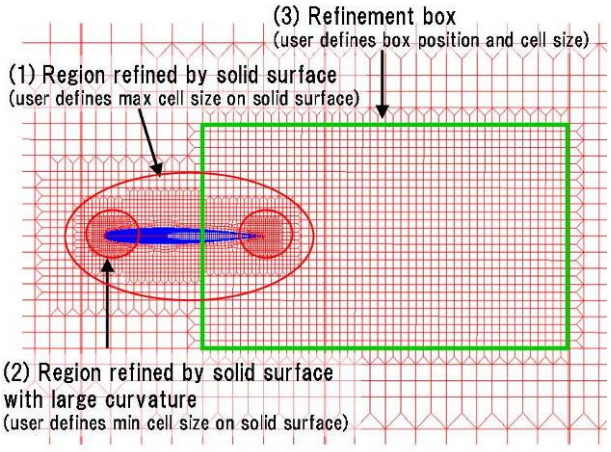

Fig. 1. Cartesian grid refinement

2.2 Removal of Cells near Solid Surface

The purpose of this step is to create sufficient space for prismatic grid around the solid surface as well as to form a base for prismatic grid generation. In Navier-Stokes computation, ideally the boundary layer can fit into this space.

Removal of cells is carried out for those intersecting the solid object and those around the solid surface. The size of Cartesian grid cells removed is in the same order as the total thickness of prismatic grid. Note that, due to the shape of Cartesian grid cells, the resulting grid is non-body-fitted.

2.3 Snapping to Solid Surface

To form a body-fitted grid, the surface of Cartesian grid is then snapped onto the solid surface (Fig. 2). This is done by moving a node on Cartesian grid surface to the closest location on the solid surface. Note that snapping always finds a unique location closest to the present position. This is a very important property, because it means that the method works even when there is a small gap between triangles that form a solid surface, and also when the triangles overlap or intersect each other. This may result in lower cost in preparing the CAD data.

The weakness of the snapping method is that it cannot capture highly concave geometry, because it will always move to the closest location. Thus this method is supplemented with feature capturing method, which forces the Cartesian grid surface to move into the concavity.

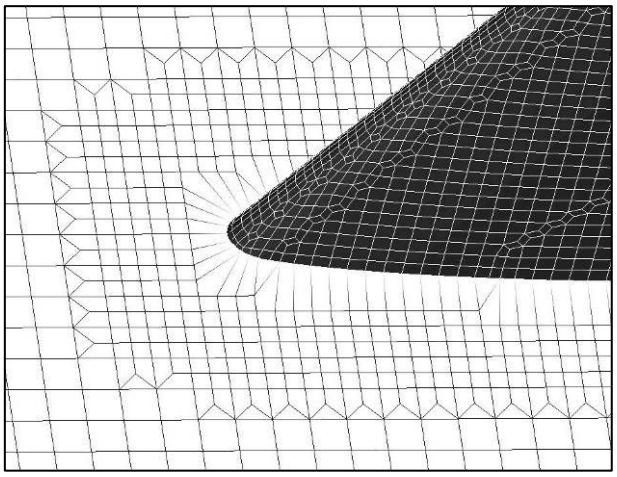

Fig.2. Grid snapping to solid surface

2.4 Prismatic Grid Generation

A number of prismatic grid layers are constructed on the snapped surface, where each layer consists of the same number of cells as shown in Fig. 3. User defines the thickness of the first grid layer, and the expansion factor of thickness. Note that the total thickness of prismatic grid is already determined during removal of Cartesian grid. The number of prismatic grid layers can thus be computed.

This stage is very important for Navier-Stokes computation, because the prismatic grid is responsible for resolving the boundary layer.

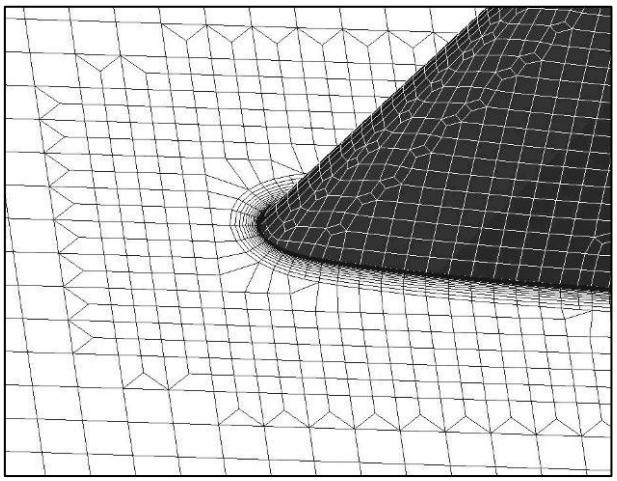

Fig. 3. Prismatic grid generation

2.5 Quality Improvement

Independently, the quality of the Cartesian and prismatic grids is already very good. When they are used together, the only region that needs major improvement is the interface between the two grids, so that they blend smoothly (Fig. 4). A smooth transition of grid cell's size and shape is very important in Navier-Stokes computation. For our study, a Laplacian smoother is found to give a satisfactory result.

In this study, we used grid face flatness and convexity as a measure of grid quality.

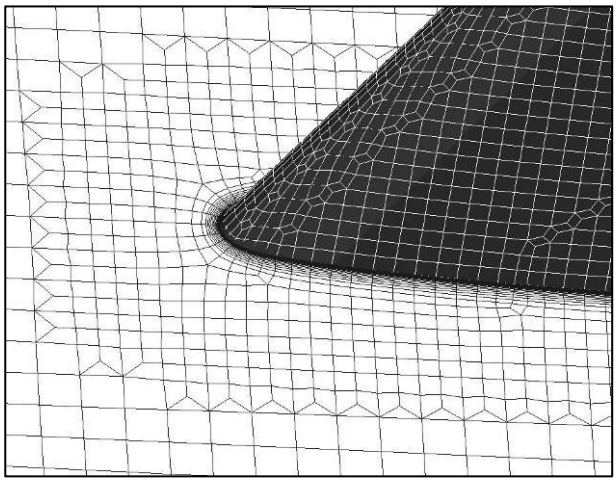

Fig. 4. Grid smoothing

3 Flow Computation Method

A flow solver for unstructured grid known as TAS (Tohoku university Aerodynamic Simulation) code [11], is used in this study. It is a well-validated code and used in a variety of aerospace applications [12]. In TAS, full Navier-Stokes equations are solved on the unstructured grid by a cell-vertex finite volume method. The HLLEW (Harten-Lax-van Leer-Einfeldt-Wada) method is used for the numerical flux computations [13]. The LU-SGS (Lower/Upper Symmetric Gauss-Seidel) implicit method is used for time integration [14]. The second-order spatial accuracy is realized by a linear reconstruction of the primitive variables with Venkatakrishnan's limiter [15] and Unstructured MUSCL-scheme (U-MUSCL) [16]. As for turbulence model, the Spalart-Allmaras model is used [17]. The equations for

the turbulence model are also solved using the second-order scheme. Turbulent transition is not taken into account.

4 Computations

4.1 Model Geometry

The model is NASA Common Research Model (NASA CRM) [9] shown in Fig. 5. It is a wing/body/tail configuration with a wing-body fairing. The supercritical wing is employed for the main wing. The fuselage is representative of a wide-body commercial transport aircraft. Shown here is one of four configurations provided by DPW4, where it has horizontal tail oriented at 0 degree.

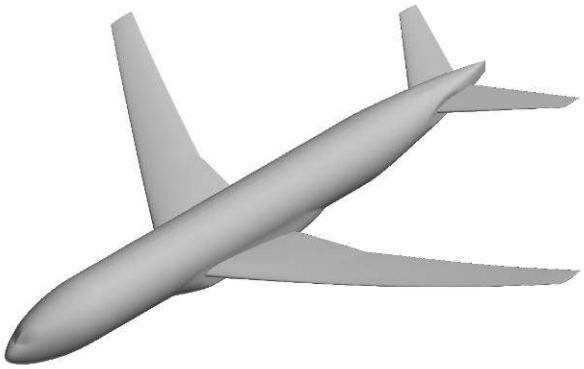

Fig. 5. NASA Common Research Model (CRM)

4.2 Grids

Three types of grid are shown here. The first is Cartesian-prismatic hybrid grid, which is the subject of this study. For comparison, we also used tetrahedral-prismatic hybrid grid, and multi-block structured grid.

4.2.1 Unstructured Cartesian-Prismatic Hybrid Grid Generated with HexaGrid

Three different grids (coarse, medium, and fine grids) are generated with HexaGrid. The medium grid is shown in Fig. 6 and 7. These parameters are the same for all grids: maximum cells size on solid surface of 5 inch, domain size of 100 times reference chord length (Cref), first prism layer thickness of 9.85E-4, and prism layer expansion ratio of 1.25. The difference between the grids is the minimum cell size on solid surface: 5in, 1.25in, and 0.625in for coarse, medium, and fine grids, respectively.

Figure 8 shows the grids of main wing at the cross-section of η=0.50, where η is a ratio of distance between the body axis and the cross-section on the span. The total thickness of prism layer depends on the Cartesian grid size. Hence, the prism layer is relatively thin around the trailing edge for medium and fine grids. Similarly, the number of prism layer decreases with increasing total number of grid (Table 3).

Gridding guidelines are provided by DPW4 [10]. Although we tried to follow them as closely as possible, restrictions of HexaGrid has forced us to deviate slightly.

Number of grid nodes is defined as 3.5 million, 10 million, and 35 million for coarse, medium, and fine grids, respectively. We generated 3.2 million, 11million, and 37 million, respectively, which is quite close (Table 1).

Grid sizing in the direction perpendicular to body surface can be achieved by setting the thickness of the first prism layer and the growth rate of prism layers. Thus these sizes are in compliance with the guideline. However, controlling grid cell size in the direction along body surface is difficult under present method. This is because the Cartesian grid method used here can only generate isotropic grid. Some stretching effect is achieved when snapping the Cartesian grid surface to body surface. For example, when Cartesian grid around leading edge is snapped, the result is elongated grid faces along the body edge. However, there is no direct control of grid face's aspect ratio as in structured grid. Thus, the required minimum of 12 cells across the trailing edge base for the medium mesh had to be neglected here.

We need to emphasize here that, due to the ease of use of HexaGrid, even CFD beginners with little experience in grid generation can generate grids easily and the time needed for training is short.

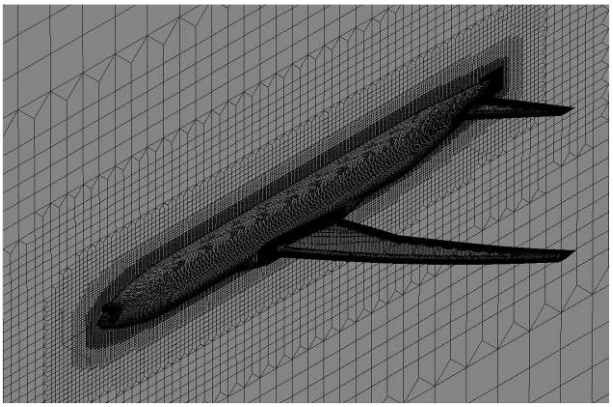

Fig. 6. Cartesian grid around the model

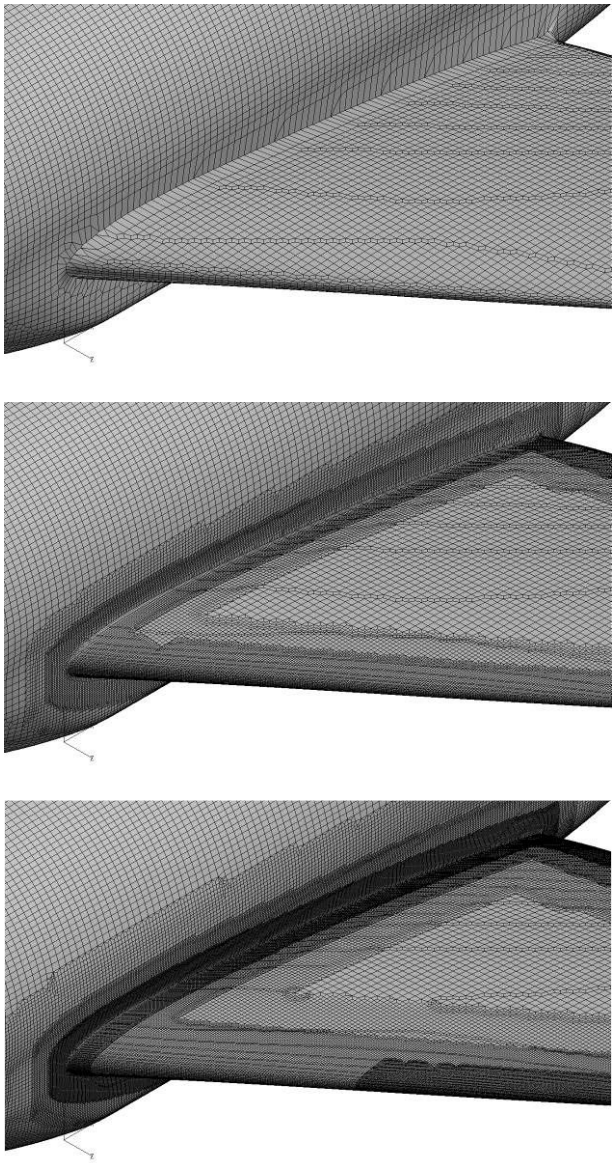

Fig. 7. Top to bottom: coarse, medium and fine grids on fuselage and wing junction

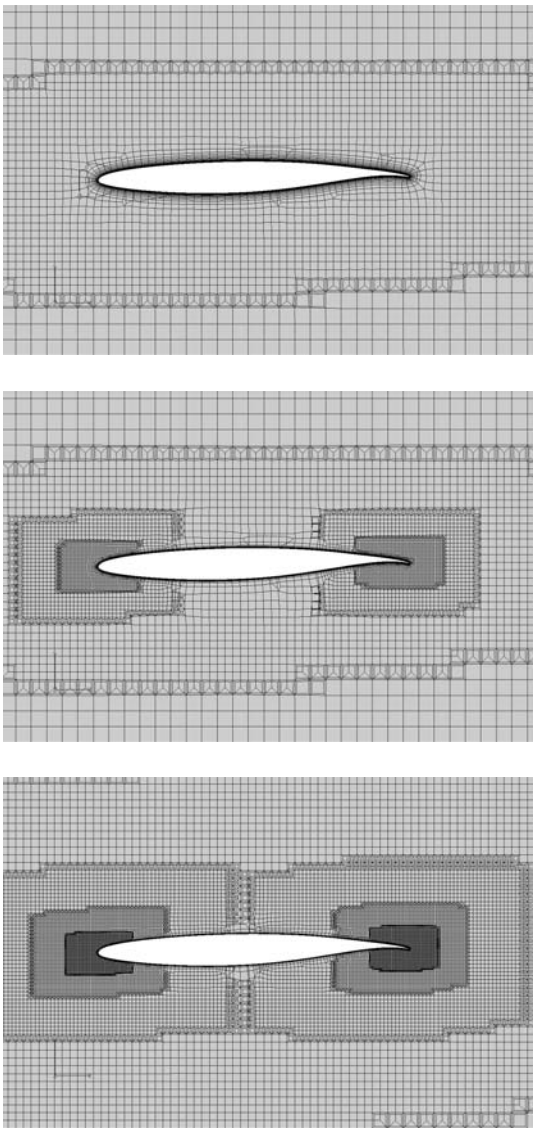

Fig. 8. Top to bottom: coarse, medium and fine grids on main wing's cross section

Table 1. Grid generation results from HexaGrid

	Coarse	Medium	Fine
Cartesian grid finest level	13	15	16
No. of prism layers	35	29	26
Total node count	3,213,783	11,055,602	36,601,899
Total cell count	3,644,942	12,654,764	41,630,191
Boundary node count	105,686	295,394	757,593
Boundary face count	106,272	297,697	762,131

4.2.2 Unstructured Tetrahedra-Prismatic Hybrid Grid generated with MEGG3D

A hybrid unstructured grid is generated using an unstructured surface/volume mesh generator, MEGG3D [18]. First, edges (feature lines) of the STL data are set manually with GUI of MEGG3D. The surface grids are triangulated with the direct advancing front method based on the edges and STL data. Then isotropic tetrahedral volume grids are generated using the method of Delaunay tetrahedral meshing. Finally, prismatic layers are added. This grid generator has been well-validated and used for many applications such as drag prediction.

A medium grid generated with MEGG3D is shown in Fig. 9. The grid is well clustered near the leading edge and trailing edge compared with that generated with HexaGrid. However, it is difficult to generate fine mesh across the trailing edge base since the grid is also basically isotropic. The MEGG3D grid in the middle of wing is relatively coarse, whereas HexaGrid is uniformly fine. The grid information is shown in Table 4. The TAS flow solver is also used for this grid.

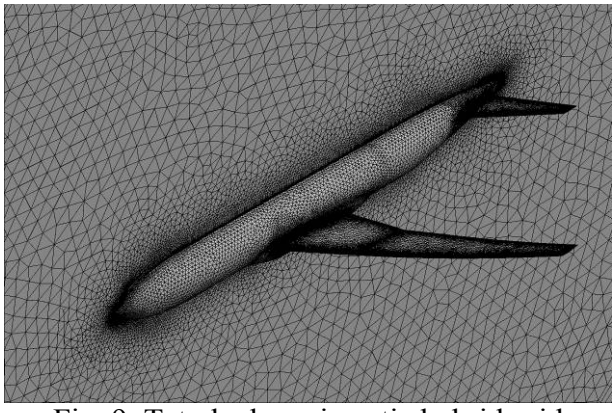

Fig. 9. Tetrahedra-prismatic hybrid grid

4.2.3 Multi-block Structured Grid generated with Gridgen

A multi-block structured grid was also prepared for comparison using a commercial software, Gridgen. A medium grid generated with Gridgen is shown in Fig. 10. The grid is well clustered near the leading edge and trailing edge compared with those generated with HexaGrid and MEGG3D. The grid is fine in the chord-wise direction, whereas it is coarse in the span-wise direction. The grid information is shown in Table 4.

As for a flow solver, UPACS is used for this grid [19]. This flow solver is based on a cell-centered finite volume method. In this study, the second-order scheme of the Roe's flux difference splitting for convection terms is used with the MUSCL extrapolation and the van Albada's limiter. The viscous terms are discretized using a scheme based on 2nd-order central difference. Time integration is carried out using the MFGS implicit method. The Spalart-Allmaras model17 is used for a turbulence model.

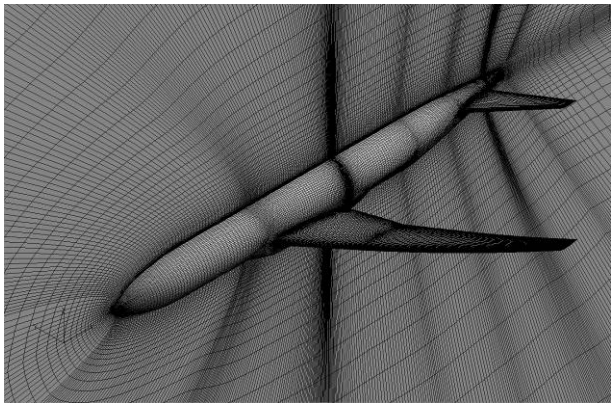

Fig. 10. Multi-block structured grid

Table 2. Grid generation results from MEGG3D and Gridgen

	MEGG3D	Gridgen
Zones	1	283
No. of prism layers	34	40
Node count	13,490,678	9,903,509
Cell count	36,250,881	9,006,808
Boundary node count	369,541	294,408
Boundary face count	798,336	275,608

5 Results

In general, results from the three types of grid discussed above compare very well with the average band of results of all participants in the workshop [20].

5.1 Comparison of Cp and Cf distribution

Here, we compare the medium grid results at CL=0.500, which are the same results used in

the grid convergence study. The Cp distribution over the whole surface is shown in Fig. 11. The Cp and Cf distributions at η=0.50 of the main wing are shown in Figs. 12 and 13. Generally, HexaGrid+TAS results agree well with the other results except for the cross-section near wing tip, though the grid generated with HexaGrid is coarser near the leading edge than those generated with MEGG3D and Gridgen. The shock wave in the middle of wing is well captured in the case of HexaGrid, where the grid generated with HexaGrid is isotropic and relatively fine compared to the other. The friction coefficient of HexaGrid becomes less than the others behind the shock wave location, since the boundary layer is affected by the larger pressure gradient. The difference of shock strength and the related friction may cause the difference of friction drag.

The solutions in the middle of wing are all different. The grid of HexaGrid is isotropic and uniformly fine, and its Cp distribution seems to be the best among the three results. However, the suction peak at the leading edge is not well resolved in the case of HexaGrid due to the insufficient grid resolution. The grid of MEGG3D is fine near the leading and trailing edges, but it is coarse in the middle of wing. As a result, the Cp distribution is smeared due to the coarse grid. The grid of UPACS is also fine near the leading and trailing edges, but it is anisotropic and relatively coarse in the span-wise direction. Therefore, the pressure distribution is smeared in the span-wise direction. The merit of using uniformly fine grid is pointed out from the comparison. It seems to be better and more efficient to use a little more uniform grid, decreasing number of grid near the leading and tailing edges, than that defined by the gridding guidelines.

The Cp distributions of the three grids are almost same at the tail wing. However, the friction of HexaGrid is less than the others probably due to the coarse grid at the leading edge.

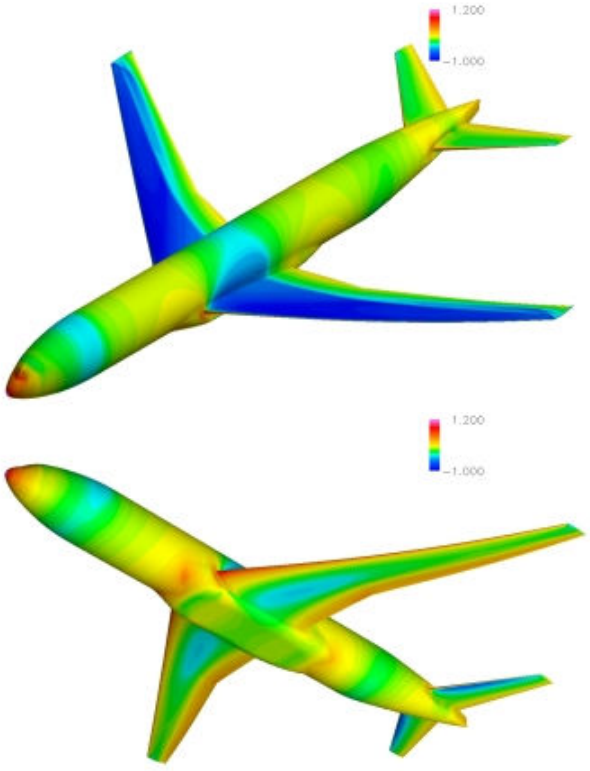

Fig. 11. Cp contours (HexaGrid+TAS on medium grid at CL=0.500)

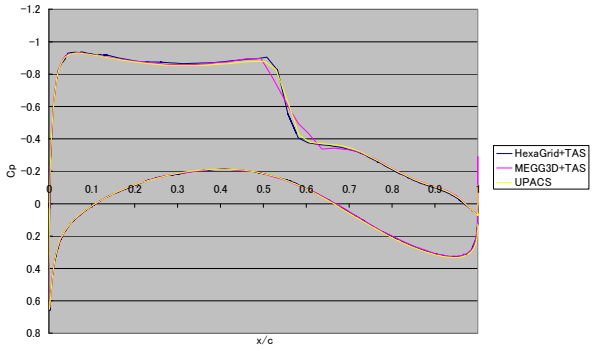

Fig.12 Cp distribution on the main wing at the cross-section of η=0.50

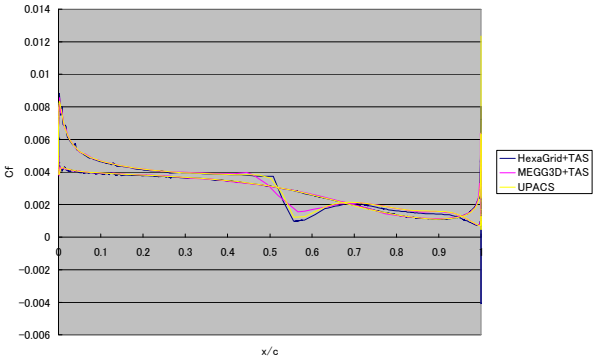

Fig.13 Cf distribution on the main wing at the cross-section of η=0.50

5.2 CL-α, CL-CD polar curve

Figure 14 shows the CL-α graph, where the three results are compared. The result of HexaGrid agrees well with those of MEGG3D and UPACS except for the attack angle of 4□ since the flow separation become large at the attack angle. The separation line is largely different among the three grids. The result of HexaGrid shows the largest separation and its flow pattern is close to UPACS result.

Figure 15 shows the CL-CD polar curve. The result of HexaGrid is close to that of UPACS except for the attack angle of 4 deg. The difference is less than 5 counts, which is quite good

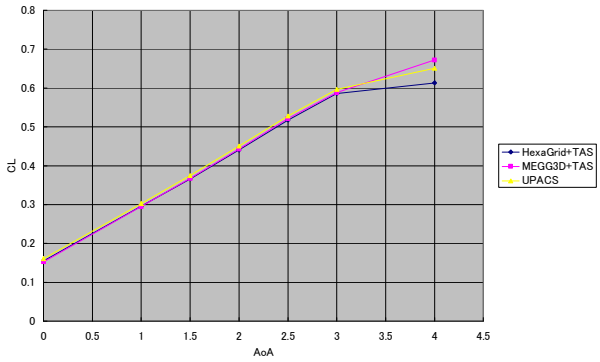

Fig. 14 CL-α

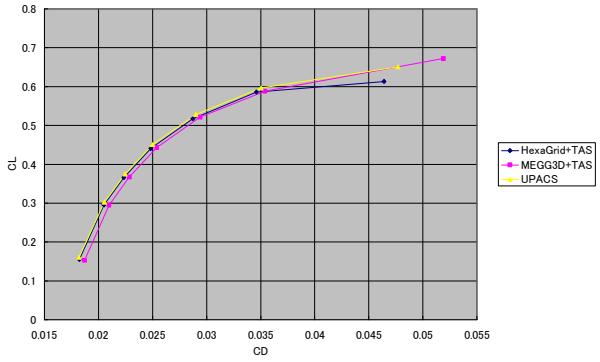

Fig. 15 CL-CD

5.3 Grid convergence

Drag coefficients are computed at CL=0.500 for the coarse, medium, and fine grids. The Free stream Mach number is 0.85 and the Reynolds number based on the reference length, Cref, is 5 millions.

Drag coefficients for the three grids are shown in Fig. 16, where HexaGrid+TAS is compared with MEGG3D+TAS and UPACS. Total drag is a summation of pressure drag and friction drag. The horizontal axis is the grid index, 1/(number of grid nodes)$^{(2/3)}$. The result of HexaGrid shows good grid convergence and the predicted drag coefficients are between the results with MEGG3D+TAS and UPACS (Fig. 16). The drag predicted with the automatic grid generation has almost the same accuracy as that with the manual grid generation. Although the pressure drag of HexaGrid+TAS is close to that of MEGG3D+TAS (Fig. 17), the friction drag behavior of HexaGrid is different from the others (Fig. 18). The friction drags of MEGG3D+TAS and UPACS have good convergence tendency, whereas that of HexaGrid shows different trend with finer grid. One of the reasons is the difference in grid stretching along the solid surface. MEGG3D and UPACS use high stretching factors for grid near the edges, and HexaGrid can not produce grid of similar property.

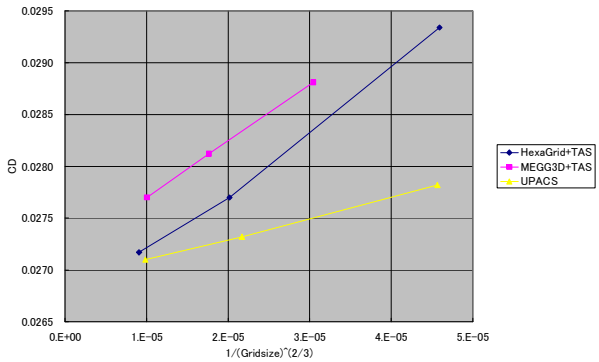

Fig. 16 Grid convergence of total drag

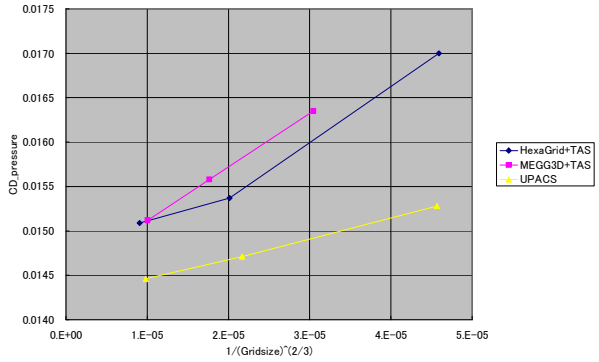

Fig. 17 Grid convergence of pressure drag

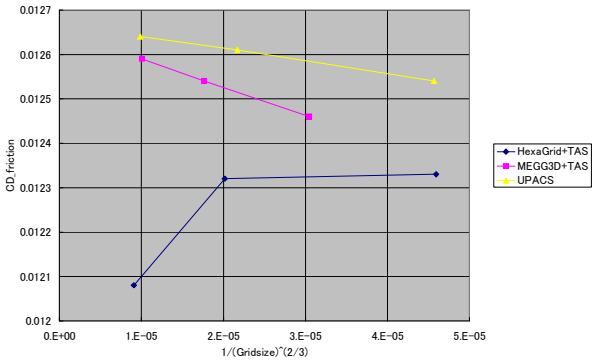

Fig. 18 Grid convergence of friction drag

5 Conclusions

Generally, HexaGrid results are in good agreement with the other results. From the pressure and friction distributions, it is evident that the flow characteristics, such as shock wave on the wing's upper surface, are very well captured. Relatively higher grid resolution and isotropic property of the grid is found to be very helpful in this region. On the other hand, the same property is also responsible for lack of resolution near leading and trailing edges, in comparison to the other methods that support grid face stretching.

The results of HexaGrid are quite close to that of UPACS, which is a structured grid, when there is no or little flow separation. The difference is less than 5 drag counts, which is very good, considering that this is the result of automatic grid generation. These results give us a strong indication that a grid generation that is both automatic and suitable for accurate Navier-Stokes computation can indeed be achieved, at least for flight conditions similar to the case presented here. Work on grid quality is currently underway to improve the results even further.

References

[1] Lahur, P. R., "Automatic Hexahedra Grid Generation Method for Component-based Surface Geometry," AIAA paper 2005-5242, 2005.

[2] Hashimoto, A., Murakami, K., Aoyama, T., Lahur P., "Lift and Drag Prediction Using Automatic Hexahedra Grid Generation Method," AIAA paper 2009-1365.

[3] Aftosmis, M.J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," VKI Lecture Series, 1997-02, 1997.

[4] Deister, F. and Hirschel, E.H., "Adaptive Cartesian/Prism Grid Generation and Solutions for Arbitrary Geometries," AIAA paper 99-0782, 1999.

[5] Leatham, M., Stokes, S., Shaw, J.A., Cooper, J., Appa, J., and Blaylock, T.A., "Automatic Mesh Generation for Rapid-Response Navier-Stokes Calculations," AIAA paper 2000-2247, 2000.

[6] Karman, S.L.Jr., "SPLITFLOW: A 3D Unstructured Cartesian/Prismatic Grid CFD Code for Complex Geometries," AIAA paper 95-0343, 1995.

[7] Tchon, K.F., Hirsch, C., and Schneiders, R., "Octree-based Hexahedral Mesh Generation for Viscous Flow Simulations," AIAA paper 97-1980, 1997.

[8] Wang, Z.J. and Chen, R.F., "Anisotropic Solution-Adaptive Viscous Cartesian Grid Method for Turbulent Flow Simulations," AIAA Journal, Vol. 40, No. 10, 2002, pp. 1969-1978.

[9] Vassberg, J.C., DeHaan, M.A., Rivers, S.M., Wahls, R.A., "Development of a Common Research Model for Applied CFD Validation Studies," AIAA paper 2008-6919.

[10] http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/

[11] Nakahashi, K., Ito, Y., and Togashi, F, "Some challenges of realistic flow simulations by unstructured grid CFD", International Journal for Numerical Methods in Fluids, Vol.43, 2003, pp.769-783.

[12] Murayama, M. and Yamamoto, K., "Comparison Study of Drag Prediction for the 3rd CFD Drag Prediction Workshop by Structured and Unstructured Mesh Method," AIAA paper 2007-258, 2007.

[13] Obayashi, S. and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier-Stokes Solver," AIAA Journal, Vol. 33, No. 6, 1995, pp. 1134-1141.

[14] 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.

[15] Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," Journal of Computational Physics, Vol.118, 1995, pp.120-130.

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

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

[18] Ito, Y., Shih, A. M. and Soni, B. K., "Unstructured Mesh Generation Using MEGG3D - Mixed-Element Grid Generator in Three Dimensions," Proceedings of the International Conference on Numerical Geometry, Grid Generation and Scientific Computing

(NUMGRID2008), Moscow, Russia, June 2008, pp. 5-11.

[19] Takaki, R., Yamamoto, K., Yamane, T., Enomoto, S. and Mukai, J., "The Development of the UPACS CFD Environment," High Performance Computing, Proc. of ISHPC 2003, Springer, pp. 307-319, 2003.

[20] Vassberg JC, Tinoco EN, Mani M, Rider B, Zickuhr T, Levy DW, Brodersenk OP, Eisfeldk B, Crippak S, Wahls RA, Morrison JH, Mavriplis DJ, Murayama M, "Summary of the Fourth AIAA CFD Drag Prediction Workshop," AIAA 2010-4547, 2010

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS2010 proceedings or as individual off-prints from the proceedings.