Thin Body Treatment on Unstructured, Cartesian Grid

by
Paulus R. LAHUR, School of Engineering, Nagoya University
Yoshiaki NAKAMURA, Dept. of Aerospace Eng., Nagoya University

ABSTRACT

Cartesian grid does not follow the contour of body, as in commonly used, body-fitted schemes. This grid makes grid generation easy and reduces generation time cycle when handling complex body geometry such as complete aircraft configuration. These are the main advantages over body-fitted schemes. However, in a simple Cartesian grid generation, when the body dimension is less than cell size (thin body), the separate fluid regions in the cell are mistakenly considered as one fluid region (multiple-flow-region problem), which produces erroneous result. In this paper a new cell cutting procedure is proposed, in which the cell is automatically split into separate sub-cells. In sharp edge of body similar treatment is also performed to eliminate truncation of the edge in a simple cell cutting procedure. The results show that the cell splitting algorithm proposed here is effective in preventing the errors while keeping the number of cells reasonably low.

1. INTRODUCTION

Grid generation plays a crucial part in computational fluid dynamics as it has direct influence on the capability to accurately simulate physical phenomena of a flow. This is a major consideration in selecting grid. However, other factors also influence grid selection; one of the biggest is cost efficiency, which demands that the flow simulation process be not only accurate, but also fast and cheap to implement and utilize. As industries such as aircraft maker become more and more competitive, CFD is expected to play a more involved role in the design process to reduce cost and development cycle time, which creates a strong demand to incorporate CFD into current design cycle. This puts more stringent requirements in conventional CFD discipline, namely : fast turn-around time and ease of use [1].

There are three major approaches in grid generation. The first one is structured, body-fitted method such as C-grid. Although it yields the most accurate solution, generating grid around complex body is very difficult and requires a lot of man-hour and expertise.

The second grid generation : unstructured, body fitted grid such as tetrahedral grid, offers a remedy to this problem. It can handle complicated body shape, but to obtain a grid with acceptable quality a lot of man-hour and expertise is still required.

The third type of grid is Cartesian grid. The grid is non-body-fitted, so that it can intersect the body surface. No need to conform to the body shape means it can handle complicated body automatically and in a short period of time. It has been shown that this method can treat very complicated body such as a full body of aircraft with very little user intervention and short generation time. Codes using this method have been employed by NASA (SPLITFLOW) [2] and Boeing (TRANAIR) [3], which find their applications in preliminary design stage.

Due to its early stage in development, many weaknesses still exist in Cartesian grid generation, such as thin body problem. Currently only a few studies have been made to treat these problems. Thus the objective of the present research is to identify the pros and cons of Cartesian grid generation and propose some improvements.

The first problem is referred to as multiple-flow-region problem, or loosely known as thin body problem. This situation arises when the thickness of body is less than cell dimension, as in the region near supersonic wing tip. Although more than one flow regions can exist inside the cell containing the thin body, conventional Cartesian grid generation discerns only one fluid region. This will produce erroneous result. The simplest remedy is to repeatedly refine the particular cell until the problem vanishes. However this will increase the number of very small cells, which is inefficient use of computational resources. Here a new cell-cutting method is proposed, which splits the problem cell.

The second problem arises when an infinitely sharp edge of body such as the leading edge of supersonic wing is present. In a simple Cartesian grid generation, the edge is truncated. As a result, it behaves as if it were blunt, and hence it is called false-blunt-body problem. Repeated cell refinement can also reduce the problem at the expense of computational resources. A cell splitting procedure similar to the case of thin body is proposed to eliminate this problem.

As a study case, a thin double wedge was employed. The results are evaluated by comparing with theoretical values.

2. METHODS

2.1. INPUT

The current grid generation code accepts body surface data as a collection of triangular panels. The parameters to be set by user in this grid generation are the size of computational domain and the maximum level of grid refinement.

2.2. GRID REFINEMENT

First, a uniform and coarse grid is generated in the computational domain according to a simple Cartesian coordinate. The grid is then repeatedly refined based on the distance to body surface, so that the cells on body surface become the finest, and those in far field the coarsest (Fig.1).

As a measure of cell size, the ratio of the double wedge chord, C to the smallest cell side length, Lmin is used : C/Lmin.

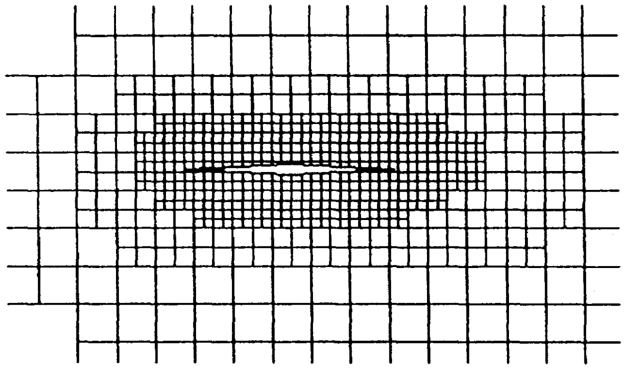

Fig.1 Grid refinement based on distance from body surface

2.3. CELL CUTTING

In a simple cutting scheme, a body is intersected with the faces of a cell, and the flux area of the face, or the part exposed to fluid is calculated. In this method, the procedure can not decide whether a cell contains thin body or not. To overcome this, the cell cutting procedure has been modified as follows :

- Consider each face of a cell.
- Find intersection between edge of face and body surface, which results in pair(s) of points. Each pair of points indicate one body surface (Fig.2).

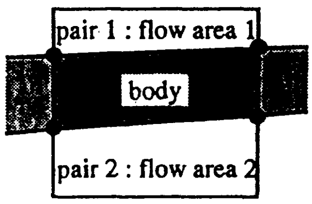

Fig.2 Intersection between cell face and body surface

- Calculate (each) flux face area.
- Repeat the above for all six faces.
- Connect the pair of intersections to form loop(s). Each loop represents one body surface (Fig.3).
- Calculate (each) body surface area and normal vector.
- Calculate (each) fluid volume.

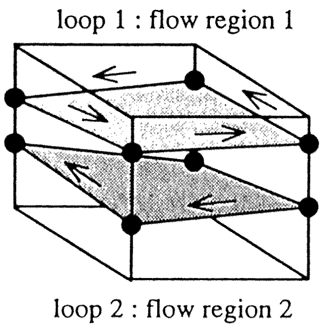

Fig.3 Loop(s) connecting all intersection points.

2.4. SHARP EDGE TREATMENT

Sharp edge is defined as abrupt change in adjacent body normal vectors (more than 90 deg). When a sharp edge is detected in a cell, the cell is split along the edge, using a procedure similar to that of thin body splitting. A splitting panel in front of the edge is added in order to fully split the cell. At the flow computation stage, the sub-cells are then re-merged. See Fig. 4.

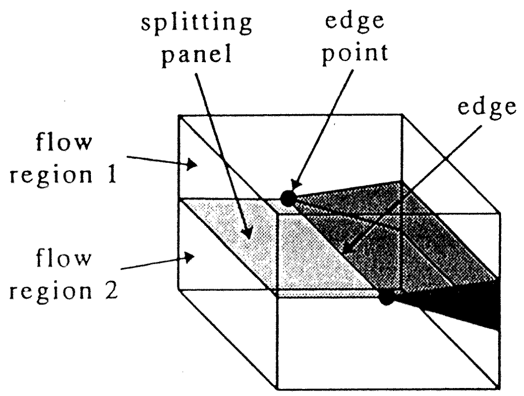

Fig.4 Splitting of a cell containing sharp edge

2.5. FLOW SOLVER

The flow solver is based on finite-volume Euler scheme. The numerical flux is calculated using Hanel's flux vector splitting scheme, which is first order in spatial accuracy. The solution is advanced in time using a three-step Runge-Kutta scheme, and local time-stepping is also employed to accelerate convergence.

2.6. TEST CASE : DOUBLE WEDGE

As a test case, a double wedge with thickness-to-chord ratio of 5% is used. The flow is supersonic throughout the flow field (M_∞ = 3.0) with zero angle of attack. Chord-to-grid ratio (C/Lmin) is 64.0, which corresponds to grid cell with level 4 of refinement. The tests are performed with 2 different types of grid. In the first grid the edge point of the wedge is located at the cell center, whereas in the second one the edge point is off the cell center (see Figs. 5 and 9). Cells with volume less than 30% of a smallest uncut cell are merged with the biggest neighboring cells in order to save time. The CFL number is set at 0.3.

3. RESULTS

The first set of results shows the pressure distribution along the wedge's surface for the case when the edge point of the wedge is at the cell center.

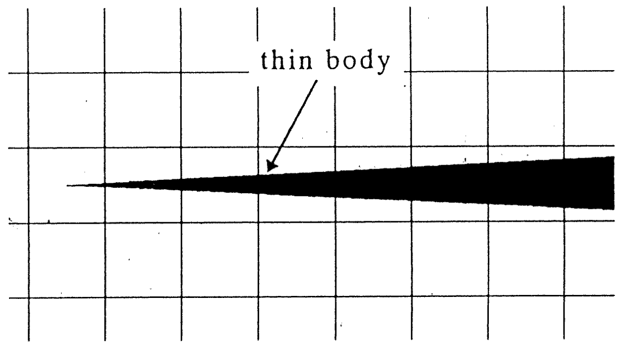

Fig.5 Grid around double wedge with refinement level 4 (C/Lmin = 64), where edge point is at the cell center.

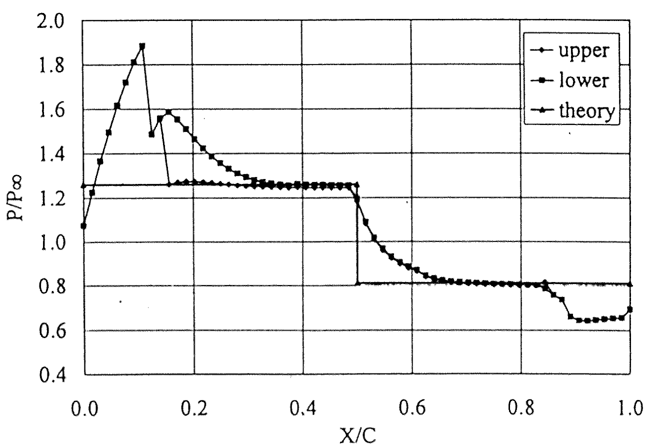

Fig.6 Pressure distribution along double wedge on simple Cartesian grid at C/Lmin=64, where edge point is at the cell center.

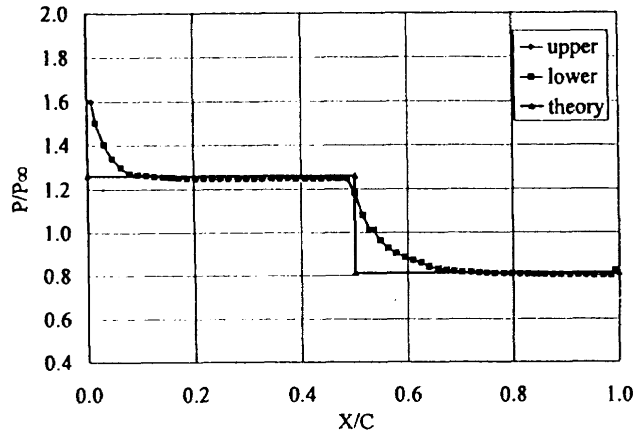

Fig.7 Pressure distribution along double wedge with thin body treatment at C/Lmin=64, where edge point is at the cell center.

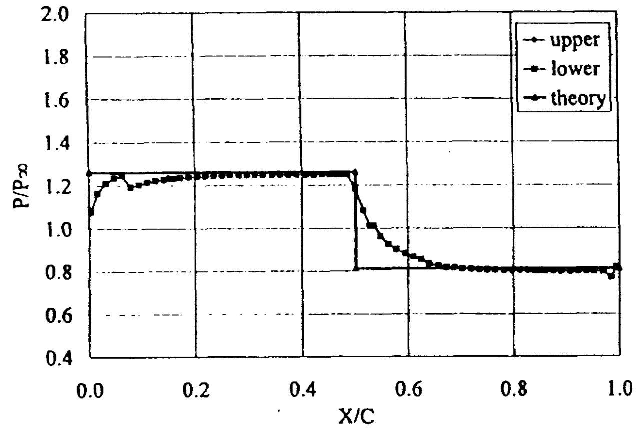

Fig.8 Pressure distribution along double wedge with sharp edge treatment at C/Lmin=64, where edge point is at the cell center.

The second set of results shows the pressure distribution when the edge point is off the cell center.

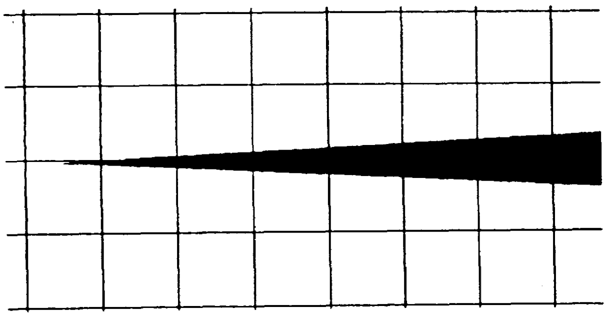

Fig.9 Grid around double wedge with refinement level 4 at C/Lmin = 64, where edge point is off the cell center.

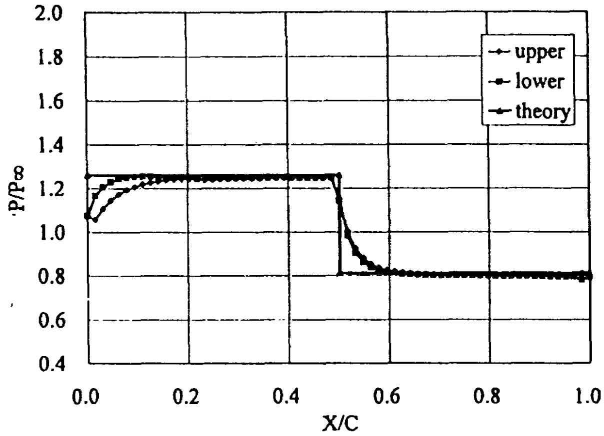

Fig.10 Pressure - distribution along double wedge on simple Cartesian grid at C/Lmin=64, where edge point is off the cell center.

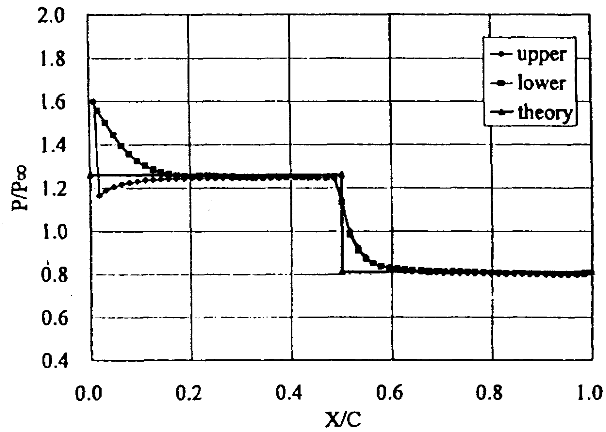

Fig.11 Pressure distribution along double wedge with thin body treatment at C/Lmin=64, where edge point is off the cell center.

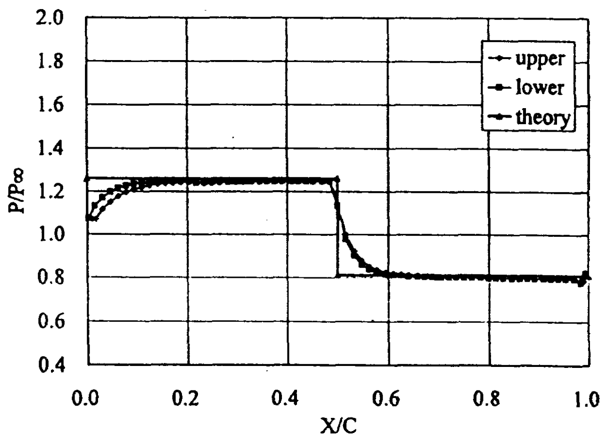

Fig.12 Pressure distribution along double wedge with sharp edge treatment at C/Lmin=64, where edge point is off the cell center.

The aerodynamics coefficients C_L and C_D for the improved Cartesian grid are presented in tables 1 and 2, which are compared with theoretical values.

C_L	position of edge point	
treatment	center	off-center
thin body	3.2E-5	3.8E-3
sharp edge	2.2E-5	3.9E-4

Table 1 C_L for double wedge with refinement level 4. Theoretical value is 0.0.

C_D (error)	position of edge point	
treatment	center	off-center
thin body	3.4E-3 (5%)	3.5E-3 (1%)
sharp edge	3.1E-3 (13%)	3.2E-3 (9%)

Table 2 C_D for double wedge with refinement level 4. Theoretical value is 3.6E-3.

The grids used in the calculation have approximately 17,000 cells. It takes 30 seconds of CPU time to generate the simple Cartesian grid and one minute for the grid with the cell splitting method on a NEC EWS 4800/360MP workstation. The flow solver advances 700 computational steps in 20 minutes of CPU time to reduce the maximum numerical residual by 5 orders of magnitude. On average, the present code takes 0.1 ms per cell per computational step.

In order to generate grid and solve the flow mentioned above, user does not have to input anything other than the dimension of computational domain, the highest level of grid refinement and flow conditions.

4. DISCUSSION

It is evident in Fig. 6 that in the region where body thickness is less than that of cells a simple Cartesian Grid can produce grossly inaccurate flow solution. Note that the edge point at the cell center, as shown in Fig. 5 is the worst position as far as the thin body problem is concerned. In this position, a great portion of the body have the thin body problem. In contrast, when the position of the edge point is off the cell center (Fig. 9), the solution is much more accurate (Fig. 10), since only a few cells experience the problem. This also indicates that a simple Cartesian Grid generation is sensitive to relative position of the body.

Splitting the cells containing thin body greatly increases the accuracy of the solutions, as can be seen from the comparison between Figs. 6 and 7. This suggests that the simple Cartesian grid generation will have to use a much more refined grid in order to obtain comparable accuracy.

In Figs. 7 and 11 it can be seen that the pressure at the leading edge point jumps beyond the theoretical value as if there were a stagnation point there (false-blunt-body effect). This is due to the truncation of sharp leading edge. The leading edge becomes blunt, and hence the pressure rises up in the stagnation region. The shape of the sharp edge should be preserved in order to prevent this from happening. When the cells that contain a sharp leading edge are split, the pressure jump disappears. See Figs. 8 and 12. Note that the solutions also become less sensitive to relative position of the body.

C_L values in Table 1 can be viewed as a measure of symmetry. As expected, when the edge point is at the cell center, relative position of the body to the grid is symmetrical, which gives symmetrical solution, and hence almost no lift is produced. On the other hand, when the edge is off the cell center, the symmetry is lost. This shows that the non-body-fitted approach is sensitive to relative position of geometry. It is also noted that the sharp edge splitting method reduces the sensitivity up to one order of magnitude.

From Table 2 it can be seen that C_D values of this particular configuration has error up to 13%, which is quite large. This is primarily the effect of inaccuracy of the flow solver algorithm, which is only one order accurate, in expansion region halfway of the body. The excess pressure in this region provides a 'push forward', which reduces the drag. When there is a pressure jump at the leading edge point due to sharp edge truncation, a 'push backward' is produced and compensates for the 'push forward'. This explains the 'higher accuracy' of the cases with no split of cells containing sharp leading edge.

In Fig. 8, a pressure drop is observed in the region where the solution should be smooth. This can be explained as the effect of approximating the solution at body surface. In the current code, flow solution is assumed to be constant everywhere inside a cell (cell-centered method). A much better approximation is expected when the flow solution is assumed to reside at the centroid of a cell, and the values everywhere else are obtained by means of interpolation (or extrapolation in the case of body surface).

An increase in time to generate grid using this new method is observed, although it is still quite fast to generate tens of thousands of cells in the order of minutes.

The simplicity of use is unchanged. As in the simple Cartesian Grid generation, the process is automatic and free from user intervention.

5. CONCLUDING REMARKS

It has been shown that the cell splitting method described above is an effective means to treat thin body with sharp edge such as supersonic wing. Splitting a cell that contains thin body yields a physically more accurate solution using relatively fewer number of cells.

Although the time needed to generate the grid is longer than that of simple Cartesian grid generation, it is still quite fast (in the order of minutes for tens of thousands of cells).

The grid generation method outlined above is automatic and relatively free of user intervention.

Non-body-fitted approach outlined in this paper still has its weakness. Most noticeable is the sensitivity of the grid to relative position of the body and inaccurate approximation of values at body surface. These will be the next targets of this research.

REFERENCES

[1] D. Ives, R. Miller, W. Siddons, K. VanDyke, *Grid Generation - A View from The Trenches*, NASA CP3291, Surface Modeling, Grid Generations, and Related Issues in Computational Fluid Dynamics (CFD) Solutions, May 1995.

[2] D.B. Finley, S.L. Karman Jr., *Euler Technology Assessment for Preliminary Aircraft Design - Compressibility Predictions by Employing The Cartesian Unstructured Grid SPLITFLOW Code*, NASA CR4710, March 1996.

[3] S.S. Samant et al. *TRANAIR : A Computer Code for Transonic Analysis of Arbitrary Configurations*, AIAA Paper 87-0034, January 1987.

[4] A. Ochi, *Numerical Calculation of Flow around SST by Use of Unstructured Grid*, Master Thesis, Dept. of Aerospace Eng., Nagoya University, 1996 (in Japanese).