3D-Computational Mesh Generation around a Propell by Elliptic Differential Equation Systen

By

Makoto KOBAYAKAWA* and Ichiro HATANO*

(Received June 16, 1987)

Abstract

The recent rapid progress of large scale super-computers enables us to solve Euler equations and even Navier-Stokes equations numerically. Almost all methods of solution adopt finite difference calculations and, therefore, the generation technique of computational mesh largely affects the stability and convergence of the solutions. In this paper, an analytic method is applied for the generation of a 3D mesh system for Navier-Stokes equations around ATP (Propfan). One of the advantages of this method is that the mesh lines have strong differentiabilities. The differential equation used is the Poisson type, and the right hand side is called the control function because it is able to control the degree of mesh line clustering. Here, the form of the control function was contrived to cluster near the solid surfaces. By this method, several mesh lines are laid in the boundary layer above the blade surfaces.

1. Introduction

Due to the very rapid progress of large scale super-computers, the non-linear partial-differential equations in fluid dynamics, e.g. Euler equations and Navier-Stokes equations, can now be solved numerically. Almost all methods of solutions adopt finite difference calculations in which the flow field is divided into many polyhedrons for three-dimensional cases. Also, the differential equations are transformed into difference equations. Accordingly, the method of division of the flow field is very important, since generated computational mesh largely affects the stability and convergence of the solutions. Moreover, the physical quantities solved change their values depending on whether the mesh is good or bad.

We can find many papers[1-7] about the mesh generation techniques cited in the references. These techniques are divided into two groups as follows:

(i) Algebraic generation method.

(ii) Analytic generation method.

* Department of Aeronautical Engineering.

By method (i), all grid points and coordinate curves can be determined arbitrarily, while the curves are determined by the solutions of elliptic or hyperbolic differential equations in method (ii). Therefore, the location of the coordinate curves can be done easily. Moreover, the normality condition upon the boundaries and the spacing control of the mesh points in the specified region are easily done. Compared to method (i), the analytic method does not so easily control the mesh lines. However, the strong differentiability can be kept along the mesh lines.

In this paper, 3D computational meshes around ATP are generated. The Poisson equation (elliptic equation) is used for this purpose. The right hand side of this equation, called the control function, controls the mesh line locations.

2. Basic Equations

Let us try to map the physical space between two blades of an ATP in the computational space as shown in Fig. 1. The physical space, xyz-space, is surrounded by two blades, nacelle, outer boundary, inflow and outflow boundaries. The computational space, $\xi\eta\zeta$-space, is a rectangular parallelpiped. This is divided into $N_\xi \times N_\eta \times N_\zeta$ small parallelpipedes with each length as $\Delta\xi$, and $\Delta\zeta$. The mapping of this computational space in the boundary fitted mesh lines in the physical space is performed.

The elliptic differential equation is applied for the governing equation, and the mapping is completed by solving a boundary value problem.

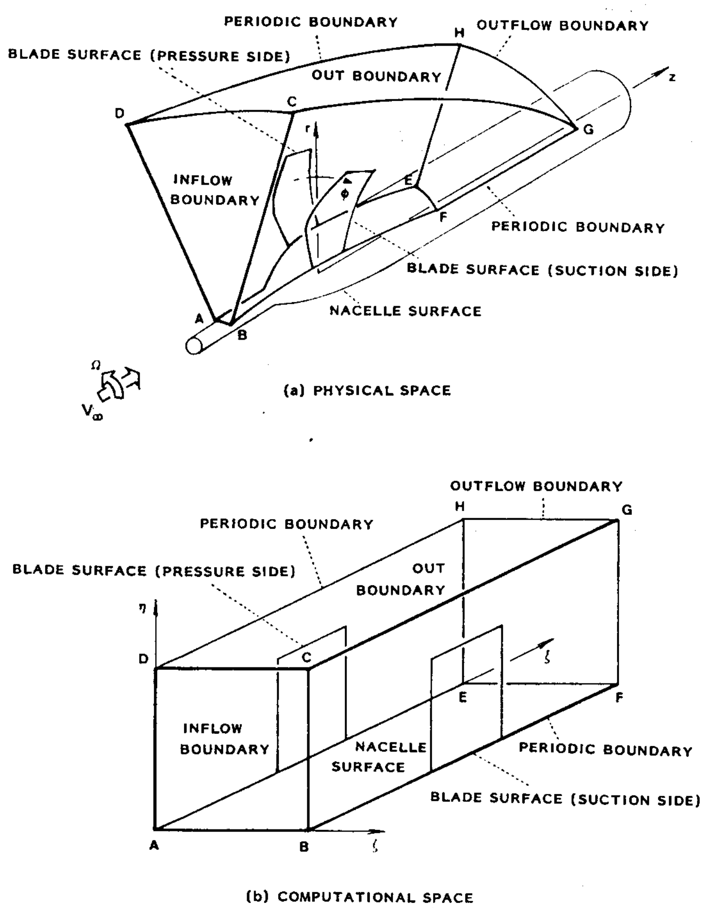
$$\nabla^2 \xi^i = 0\,, \quad \nabla^2 = \sum_{k=1}^{3} \frac{\partial^2}{\partial x_k^2} \tag{1}$$

where $\xi^1=\xi$, $\xi^2=\eta$ and $\xi^3=\zeta$
Eq. (1) is a Laplace equation. However, the following Poisson equation is more useful for this purpose.

$$\nabla^2 \xi^i = P^i,\ i = 1, 2, 3 \tag{2}$$

where P^i is called the control function, and the coordinate curves are able to be controlled by this value. For negative P^i, a constant ξ^i surface in the physical space is attracted to the smaller value of x_i, where $x_1=x$, $x_2=y$ and $x_3=z$ and vice versa. In a word, the mesh lines can be dense or coarse due to the negative or positive values of the right hand side of Eq. (2).

Changing the dependent and independent variables with each other, Eq. (2) is rewritten as follows:



(a) PHYSICAL SPACE

(b) COMPUTATIONAL SPACE

Fig. 1. Space around ATP.

$$\sum_{i=1}^{3}\left(\sum_{j=1}^{3} g_{ij}\frac{\partial^2 x_k}{\partial \xi^i \partial \xi^j}+P^i\frac{\partial x_k}{\partial \xi^i}\right)=0 \tag{3}$$

where

$$g^{ij}=\frac{1}{g}\left(g_{km}\,g_{ln}-g_{kn}\,g_{lm}\right)$$

$$g_{ij}=\underset{\sim}{r}_{\xi^i}\cdot\underset{\sim}{r}_{\xi^j},\ \underset{\sim}{r}=[x\,y\,z]^T,\ \underset{\sim}{r}_{\xi^i}=\left[\frac{\partial x}{\partial \xi^i}\ \frac{\partial y}{\partial \xi^i}\ \frac{\partial z}{\partial \xi^i}\right]^T$$

$$g=[\underset{\sim}{r}_{\xi^1}\cdot(\underset{\sim}{r}_{\xi^2}\times\underset{\sim}{r}_{\xi^3})]^2$$

Moreover, put

$$P^i = g^{ii}\,\mathcal{P}^i \tag{4}$$

Eq. (3) is written as follows:

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\left(\frac{\partial^2 x_k}{\partial\xi^i\,\partial\xi^j}+\delta_{ij}\,\mathcal{P}^i\,\frac{\partial x_k}{\partial\xi^i}\right)=0 \tag{5}$$

3. Method of Solution

As described in the previous section, the desired mesh is obtained by solving the boundary value problem of Eq. (5). It is needless to say that the method of solution for Eq. (5) is numerical. Then, the space differentiation is expressed by the finite difference. The central difference with a second accuracy is used herein, i.e.

$$\begin{aligned}
\frac{\partial x}{\partial\xi} &= \frac{x(i+1,j,k)-x(i-1,j,k)}{2\Delta\xi}\\
\frac{\partial^2 x}{\partial^2\xi} &= \frac{x(i+1,j,k)-2x(i,j,k)+x(i-1,j,k)}{\Delta\xi^2}\\
\frac{\partial^2 x}{\partial\xi\partial\eta} &= \frac{x(i+1,j+1,k)-x(i-1,j+1,k)-x(i+1,j-1,k)+x(i-1,j-1,k)}{4\Delta\xi\Delta\eta}
\end{aligned}$$

$$\text{etc.} \tag{6}$$

Substituting these expressions into Eq. (5), $x(i,j,k)$ is solved as follows:

$$\begin{aligned}
x(i,j,k) &= \frac{1}{2C_w}\,(d_1+d_2+d_3+d_4+d_5+d_6\\
&\quad +g^{11}\,x_\xi\,\mathcal{P}+g^{22}\,x_\eta\,\mathcal{Q}+g^{33}\,x_\zeta\,\mathcal{R})
\end{aligned} \tag{7}$$

where

$$\left.\begin{aligned}
d_1 &= \frac{g^{11}}{\Delta\xi^2}\,\{x(i+1,j,k)-x(i-1,j,k)\}\\
d_2 &= \frac{g^{22}}{\Delta\eta^2}\,\{x(i,j+1,k)+x(i,j-1,k)\}\\
d_3 &= \frac{g^{33}}{\Delta\zeta^2}\,\{x(i,j,k+1)+x(i,j,k-1)\}\\
d_4 &= d_{12}(g^{12}+g^{21})\\
d_5 &= d_{13}(g^{13}+g^{31})\\
d_6 &= d_{23}(g^{23}+g^{32})
\end{aligned}\right\} \tag{7a}$$

and

$$\left.\begin{aligned}
d_{12} &= \frac{\partial^2 x}{\partial\xi\partial\eta} = \frac{1}{4\Delta\xi\Delta\eta}\\
&\times\{x(i+1,j+1,k)-x(i+1,j-1,k)-x(i-1,j+1,k)+x(i-1,j-1,k)\}\\
d_{13} &= \frac{\partial^2 x}{\partial\xi\partial\zeta} = \frac{1}{4\Delta\xi\Delta\zeta}\\
&\times\{x(i+1,j,k+1)-x(i+1,j,k-1)-x(i-1,j,k+1)+x(i-1,j,k-1)\}\\
d_{23} &= \frac{\partial^2 x}{\partial\eta\partial\zeta} = \frac{1}{4\Delta\eta\Delta\zeta}\\
&\times\{x(i,j+1,k+1)-x(i,j+1,k-1)-x(i,j-1,k+1)+x(i,j-1,k-1)\}\\
C_w &= \frac{g^{11}}{\Delta\xi^2}+\frac{g^{22}}{\Delta\eta^2}+\frac{g^{33}}{\Delta\zeta^2}
\end{aligned}\right\} \tag{7b}$$

In order to obtain the converged $x(i, j, k)$, we used the iteration method with relaxation, i.e. $x^*(i, j, k)$ is calculated by using the n-th value $x(i, j, k)$ by Eq. (7). Thus, Δx is obtained by

$$\Delta x = x^*(i, j, k)-x(i, j, k) \tag{8}$$

Then, the $(n+1)$-th value of x is obtained as follows:

$$x^{n+1}(i, j, k) = x^n(i, j, k)+\omega\Delta x \tag{9}$$

where ω is the relaxation coefficient which makes the convergence speed faster and the solution to be stable. The above process should be continued until Δx becomes smaller than the preassigned value.

4. Control Function

As described above, mesh lines can be controlled by the control function P^i (or $\mathcal{P}$, $\mathcal{Q}$, $\mathcal{R}$). The purpose of this paper is to generate the mesh system which is applied for the Navier-Stokes computational code around ATP. Therefore, the control function is used for the mesh line clustering near the solid surface where the boundary layer is developing. Since the solid surfaces locate along the ζ-constant plane, we can put

$$\mathcal{P} = \mathcal{Q} = 0 \tag{10}$$

Then, $\mathcal{R}$ is given by

$$\mathcal{R} = -fa[\exp\{-b(\zeta-\zeta_1)\}-\exp\{-b(\zeta_N-\zeta)\}] \tag{11}$$

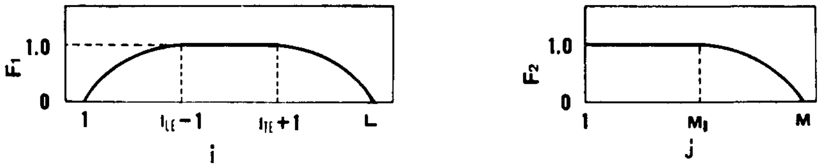


Fig. 2. Control functions

where

$$\left.\begin{aligned} a &= -\frac{(\zeta-\zeta_1)(\zeta-\zeta_N)}{(\zeta_N-\zeta_1)^2}\,a' \\ b &= \frac{1}{\zeta_N-\zeta_1}\,b' \\ f &= f_1\cdot f_2 \end{aligned}\right\} \tag{12}$$

In the above equation, f controls the mesh line location along the ξ and η directions by second order curves, i.e. f_1 and f_2 determine the density along the ξ and η directions respectively, as shown in Fig. 2. Moreover, a' and b' are determined by trial and error.

In order to determine the degree of clustering of the mesh lines, we must estimate the thickness of the boundary layer. Since we put only a thin-layer assumption upon the blade surfaces for the solution of the Navier-Stokes equation, several mesh lines may be laid in the boundary layer. Since the thickness of the turbulent boundary layer is proportional to $\mathrm{Re}^{-1/2}$, $\delta \simeq 10^{-3}$ in the case of $\mathrm{Re} \simeq 10^6$. Accordingly, 10^{-4} is enough for the duration of the two neighboring mesh lines.

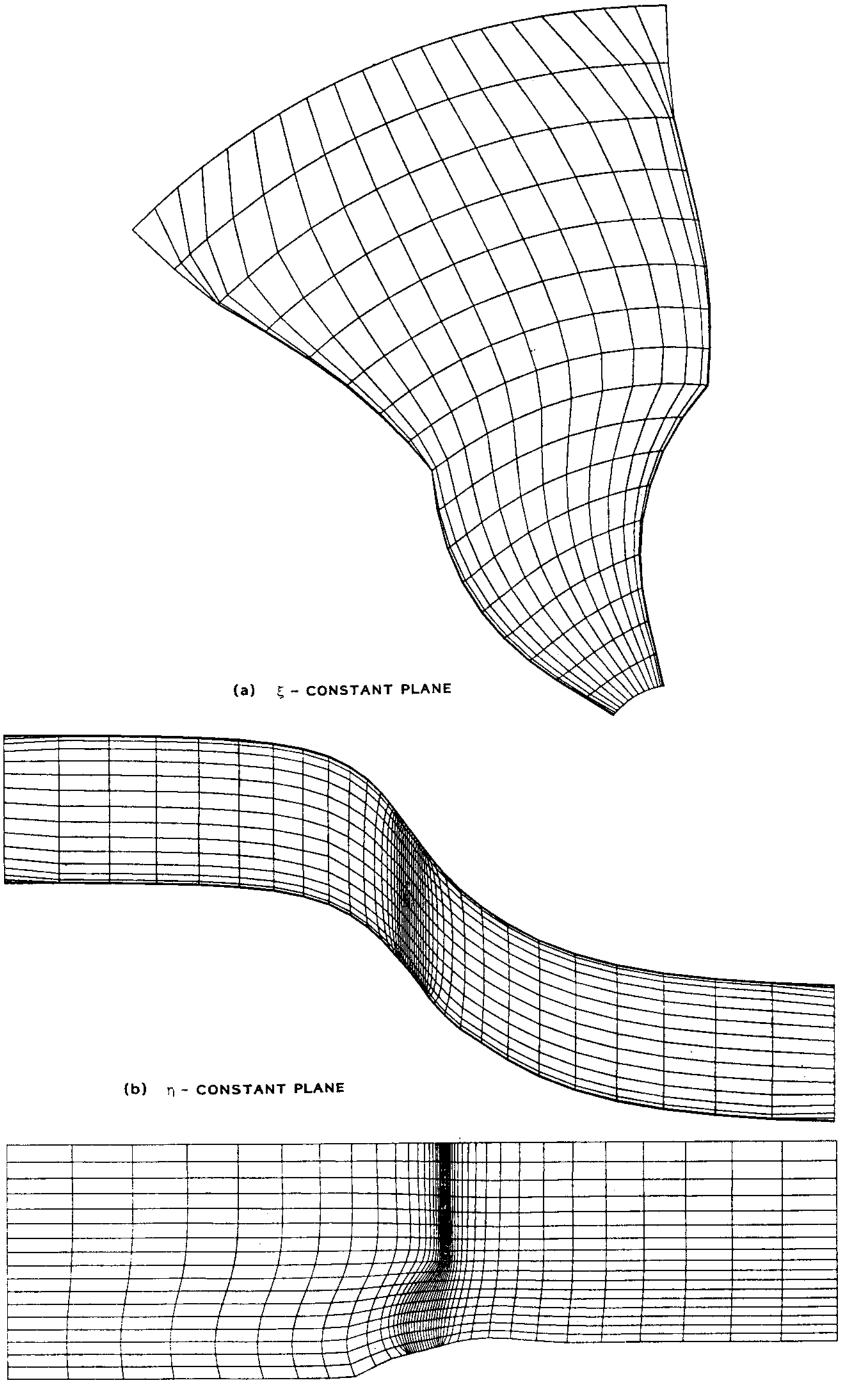
5. Mesh System for Rotating Blades

Applying the above mentioned method to the space between two neighboring blades, we try to generate a computational mesh system around a propeller. By this mesh system, we aim to solve Navier-Stokes equations where a thin boundary layer is assumed on the blade surfaces, while the flow slips on the nacelle surface, i.e. the viscous terms are neglected. Then, the governing equations are Euler ones on the nacelle surface.

The numbers of the mesh lines are 45, 18 and 21 along the x, y and z-directions respectively, and we put a' and b' in Eq. (12) as

$$a' = 1.5 \times 10^3$$
$$b' = 1.25 \times 10$$

The initial mesh system was generated appropriately by an algebraic method.



(a) ξ - CONSTANT PLANE

(b) η - CONSTANT PLANE

(c) ζ - CONSTANT PLANE

Fig. 3. Mesh system around SR-3 ATP.

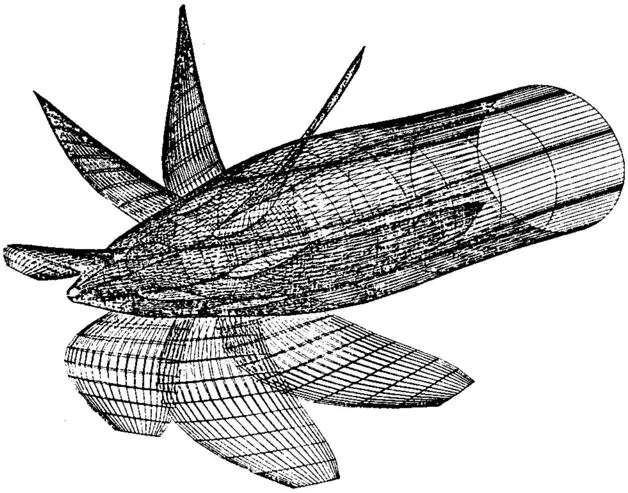


Fig. 4. Meshes on the surfaces of SR-3 ATP.

Using this system, we started the iteration. An example of mesh after 20 iterations is illustrated in Fig. 3. Fig. 4 shows mesh lines on the solid surfaces. The clustering of mesh lines near the blade surfaces is depicted in these figures.

6. Conclusion

In order to solve Navier-Stokes equations numerically around ATP, it is necessary to generate a computational mesh system suitable for the calculations. In this paper, an analytic method in which the elliptic differential equation is solved is applied for the generation of the 3D mesh system. One of the advantages of this method is that mesh lines have strong differentiabilities. The differential equation used is the Poisson type, and the right hand side is called the control function because it is able to control the degree of mesh line clustering. Here, the form of the control function was contrived to cluster near the solid surfaces. By this method, several mesh lines are laid in the boundary layer above the blade surfaces.

References

1) J. F. Middlecoff and P. D. Thomas; AIAA Journal, 18, 652 (1980).
2) C. W. Mastin; "Numerical Grid Generation", North Holland (1982).
3) C. F. Shieh; "Numerical Grid Generation", North Holland (1982).
4) J. F. Thompson; "Numerical Grid Generation", North Holland (1982).
5) L. E. Ericksson; AIAA Journal, 20, 1313 (1982).
6) J. F. Thompson; AIAA Journal, 22, 1505 (1984).
7) A. Shmilovich and D. A. Caughey; Journal of Aircraft, 22, 467 (1985).

ERRATA

The following correction should be made in the paper title on p.265 of VOLUME XLIX PART 4, OCTOBER 1987.

3D-Computational Mesh Generation around a Propell by Elliptic Differential Equation Systen

↓

3D-Computational Mesh Generation Around a Propeller by Elliptic Differential Equation System