PIGG - Program for Interactive Grid Generation

John S. Seo* , Moon Sang Kim*, and Yong Hyun Yoon**

대화형 격자구성 코드(PIGG) 개발

서 상 필* 김 문 상* 윤 용 현**

요 약

본 논문은 전산 유체 역학에서 필요한 3차원 다구획 격자구성 프로그램을 개발한 것이다. 유동장에 격자를 구성하는 것은 유동해석 방정식을 푸는 것 보다 일반적으로 시간이 더 소요되므로 격자구성 시간을 단축하기 위해서 사용자 편의를 도모하는 컴퓨터 화면과의 대화형 코드를 개발하였다. PIGG라고 부르는 본 격자구성 프로그램은 형상 모델링, 표면격자 구성, 유동장격자 생성, 그리고 격자의 재구성에 이르는 일련의 과정이 대화형/화면게시 형태로 수행된다. 본 PIGG를 이용하여 전통적인 전투기 형상 격자를 구성한 결과 많은 시간을 단축할 수 있었다.

Key Words: Grid Generation Code(격자 구성 코드), Interactive Grid Generation(대화형 격자구성), Multiblock(다구획), Surface Grid(표면 격자), Field Grid(유동장 격자)

1. Introduction

Advances in computing speed and the availability of larger core memories have allowed more challenging computational mechanics problems to be attempted. Compu tational Fluid Dynamics(CFD) methods rec ently simulate flows about complex geomet ries with more complex physics. A typical CFD application may be divided into three steps: grid generation (pre-processing), flow calculation (processing), and solution visuali zation(post-processing). Each step is usually performed independently. However, there are iterative procedures between each steps.

The grid generation process is extremely time consuming for complex geometries. Advances in graphics workstations have provided the hardware capability for the execution of sophisticated, user-friendly and interactive grid generation software. This development has dramatically reduced the man-hours involved during the grid generation process for many engineering applications. GRIDGEN[1], GENIE++[2], EAGLEView[3] and NGP[4] are examples of such developments.

The purpose of this work is to develop a grid generation code associated with general fighter aircraft flow fields in a graphic interface environment. This paper describes such an approach. The PIGG(Program for Interactive Grid Generation) code is a general three-dimensional algebraic/elliptic grid gene ration system based on multiblock structured grids. The code provides a user-friendly graphic interface for an existing grid genera tion program. This tool builds upon a grid generation program which was delivered by the contractors, KAIST (Korea Advanced Institute of Science and Technology) and KA FA(Korea Air Force Academy), to The Agency for Defence Development(ADD) and adds a graphic interface program. PIGG is designed to be very user-oriented with efficien t and easily recognizable input.

* Agency for Defence Development
** Korea Air Force Academy

"Pictures speak a thousand words." Graphic display provides user with the most informa tion in the least amount of time. Any errors are quickly detected in real-time rather than waiting until the grid generation session is complete. Therefore, PIGG attempts to make the grid generation process simple thus reducing the overall CFD solution turn-around time. Basically, the code consists of two main modules. The first module allows generation of the bounding surface grid with the input files which contain the geometry data and grid generation control parameters. The second module generates the complete three-dimensional grid based on the bounding surface grid. An interactive program based on the technique has been written, and its usage is described herein.

2. Grid Generation Systems

Methodologies for structured grid generation can be classified into direct methods, such as algebraic methods, or indirect methods, such as solving a set of partial differential equati ons(PDE). Elliptic systems are the most common PDE approach for generating structured grids. In this work, both algebraic and elliptic grid generation systems are available to the user for generating a grid. The algebraic system is used to generate the initial grid, and the elliptic system is used to refine the initial grid for better grid quality.

The technique for grid generation is a methodology for establishing the mathematical expression relating the computational domain to a physical domain. The methodology separates the boundary-grid definition from the interior-grid definition. The boundary grid is first defined, and then the interior grid is defined. Field point distribution is controlled by functions evaluated automatically by the code.

2.1 Algebraic Grid System

Algebraic grid generation system is the most important part for the interactive multiblock grid generation systems for arbitrary configurations. Interior grid points $\vec{P}(\xi, \eta, \zeta)$ of a computational domain can be constructed by interpolation from the boundaries. Such coordinates generation procedures are refered to as algebraic generat ion systems. These are many interpolation /approximation methods, some of them perform the interpolation of given points such as the Lagrange interpolation and the transfinite interpolation method, some of them consider the control points to use Bezier, B-spline, nonuniform rational B-splines(NURBS). Basical ly, the control point method has been a design tool which allows designers to develop the shape by specifying control points. By using the inverse process, the control points which would produce a given surface are found[5].

The most common algebraic interpolation method in grid generation is the transfinite interpolation(TFI). A general three-dimensional TFI can be expressed in projector form as

$$\begin{aligned} & F_{\xi^1}(\vec{p}) \oplus F_{\xi^2}(\vec{p}) \oplus F_{\xi^3}(\vec{p}) = \\ & F_{\xi^1}(\vec{p}) + F_{\xi^2}(\vec{p}) + F_{\xi^3}(\vec{p}) \\ & -F_{\xi^1}[\ F_{\xi^2}(\vec{p})]\ -F_{\xi^2}[\ F_{\xi^3}(\vec{p})] \\ & -F_{\xi^1}[\ F_{\xi^3}(\vec{p})]\ +F_{\xi^1}[\ F_{\xi^2}[\ F_{\xi^3}(\vec{p})]\ ] \quad (1) \end{aligned}$$

where $F_{\xi^i}(\vec{p})$ is an one-dimensional interpola tion function in the ξ^i-direction. Generating grids with an algebraic method is faster than the other methods, such as PDE methods or variational methods. However, its main disadvantage is the non-smooth result. That is, the discontinuity on the boundaries will propagates into the field. Therefore, PIGG also provides the elliptic grid generation system to smooth the grid.

2.2 Elliptic Grid System

The Poisson-type elliptic grid generation system developed by Thompson et al.[6], is coupled and nonlinear. Therefore, it provides smooth grid lines around an arbitrary shape, but requires longer iteration time and specification of data at boundaries of the domain of interest. To compensate for this point, Kwon's method[7] which is much simpler than Thompson's is introduced. The equations derived by Kwon are linear and uncoupled, so they are much easier to solve. The above two methods are described.

The elliptic grid generation method first introduced by Thompson and Warsi is frequently applied in practice[8]. The transformed Poisson equations in three dimensional coordinates are used, wherein x, y, z are found from given ξ, η, ζ:

$$g^{11}(\vec{r}_{\xi\xi}+P\vec{r}_{\xi})+g^{22}(\vec{r}_{\eta\eta}+Q\vec{r}_{\eta})+g^{33}(\vec{r}_{\zeta\zeta}+R\vec{r}_{\zeta})+2g^{12}\vec{r}_{\xi\eta}+2g^{23}\vec{r}_{\xi\zeta}+2g^{12}\vec{r}_{\eta\zeta}=0 \quad (2)$$

where the forcing functions, P, Q and R are constructed as sums of decaying exponential functions.

Another elliptic grid generation method developed by Kwon is used to generate volume grids. Basically, these equations are transformed by introducing streching functions to the final computational domain so that the generated grid can be clustered in the desired regions. These are Laplace equations which are linear and uncoupled, so they are much easier to solve. The following grid generating equations were derived by Kwon, J.H[9].

$$Ax_{xx}+Bx_x+Cx_{yy}+Dx_y+Ex_{zz}+Fx_z=0$$
$$Ay_{xx}+By_x+Cy_{yy}+Dy_y+Ey_{zz}+Fy_z=0$$
$$Az_{xx}+Bz_x+Cz_{yy}+Dz_y+Ez_{zz}+Fz_z=0 \quad (3)$$

where $A=\dfrac{1}{\left(\frac{df}{dX}\right)^2}$, $B=-\dfrac{\left(\frac{d^2f}{dX^2}\right)}{\left(\frac{df}{dX}\right)^3}$,

$$C=\frac{1}{\left(\frac{dg}{dY}\right)^2}, D=-\frac{\left(\frac{d^2g}{dY^2}\right)}{\left(\frac{dg}{dY}\right)^3}, E=\frac{1}{\left(\frac{dh}{dZ}\right)^2}$$

$$F=-\frac{\left(\frac{d^2h}{dZ^2}\right)}{\left(\frac{dh}{dZ}\right)^3}, x_x=\frac{f}{g'}y_y, y_x=-\frac{g'}{f}x_y$$

etc. He combined a Dirichlet condition and a Neumann condition derived from the transformed Cauchy Riemann conditions. The primes denote ordinary differentials with respect to X or Y. If a Dirichlet condition is used on a boundary for the x-equation, then a Neumann condition is used on that boundary for the y-equation and vice versa. Comparing to Thompson's, Kwon's method has the disadvantage which is no guarantee for non-crossing of grid lines of the same family.

3. Design of the Graphical User Interface

Graphical user interface design is one of the major issues in the development of contemporary grid code. An interactive program, however, is more complex than a noninteractive program because the user and the program must communicate through questions, responses, and graphical displays. Also, fault tolerances must be coded so that interactive input errors will not cause the program to abort.

PIGG consists of a little over 10,000 lines in C and it is hardwired for the Silicon Graphic IRIS workstations due mainly to its dependence on the Graphics Library(GL)[10], and hence is executable only on IRIS workstations. All user inputs are handled through the mouse/menu combination with text-interface.

2.2 Elliptic Grid System

The Poisson-type elliptic grid generation system developed by Thompson et al.[6], is coupled and nonlinear. Therefore, it provides smooth grid lines around an arbitrary shape, but requires longer iteration time and specification of data at boundaries of the domain of interest. To compensate for this point, Kwon's method[7] which is much simpler than Thompson's is introduced. The equations derived by Kwon are linear and uncoupled, so they are much easier to solve. The above two methods are described.

The elliptic grid generation method first introduced by Thompson and Warsi is frequently applied in practice[8]. The trans formed Poisson equations in three dimension al coordinates are used, wherein x, y, z are found from given ξ, η, ζ:

$$g^{11}(\vec{r}_{\xi\xi}+P\vec{r}_{\xi})+g^{22}(\vec{r}_{\eta\eta}+Q\vec{r}_{\eta})+g^{33}(\vec{r}_{\zeta\zeta}+R\vec{r}_{\zeta})+2g^{12}\vec{r}_{\xi\eta}+2g^{23}\vec{r}_{\xi\zeta}+2g^{12}\vec{r}_{\eta\zeta}=0 \quad (2)$$

where the forcing functions, P, Q and R are constructed as sums of decaying exponential functions.

Another elliptic grid generation method developed by Kwon is used to generate volume grids. Basically, these equations are transformed by introducing strecting functions to the final computational domain so that the generated grid can be clustered in the desired regions. These are Laplace equations which are linear and uncoupled, so they are much easier to solve. The following grid generating equations were derived by Kwon, J.H[9].

$$Ax_{xx}+Bx_x+Cx_{yy}+Dx_y+Ex_{zz}+Fx_z=0$$
$$Ay_{xx}+By_x+Cy_{yy}+Dy_y+Ey_{zz}+Fy_z=0$$
$$Az_{xx}+Bz_x+Cz_{yy}+Dz_y+Ez_{zz}+Fz_z=0 \quad (3)$$

where $A=\dfrac{1}{\left(\dfrac{df}{dX}\right)^2}$, $B=-\dfrac{\left(\dfrac{d^2f}{dX^2}\right)}{\left(\dfrac{df}{dX}\right)^3}$,

$C=\dfrac{1}{\left(\dfrac{dg}{dY}\right)^2}$, $D=-\dfrac{\left(\dfrac{d^2g}{dY^2}\right)}{\left(\dfrac{dg}{dY}\right)^3}$, $E=\dfrac{1}{\left(\dfrac{dh}{dZ}\right)^2}$

$F=-\dfrac{\left(\dfrac{d^2h}{dZ^2}\right)}{\left(\dfrac{dh}{dZ}\right)^3}$, $x_x=\dfrac{f}{g'}y_y$, $y_x=-\dfrac{g'}{f}x_y$

etc. He combined a Dirichlet condition and a Neumann condition derived from the transfor med Cauchy Riemann conditions. The primes denote ordinary differentials with respect to X or Y. If a Dirichlet condition is used on a boundary for the x-equation, then a Neumann condition is used on that boundary for the y-equation and vice versa. Comparing to Thompson's, Kwon's method has the disadvantage which is no guarantee for non-crossing of grid lines of the same family.

3. Design of the Graphical User Interface

Graphical user interface design is one of the major issues in the development of contemporary grid code. An interactive progr am, however, is more complex than a noninter active program because the user and the program must communicate through questions, responses, and graphical displays. Also, fault tolerances must be coded so that interactive input errors will not cause the program to abort.

PIGG consists of a little over 10,000 lines in C and it is hardwired for the Silicon Graphic IRIS workstations due mainly to its dependence on the Graphics Library(GL)[10], and hence is executable only on IRIS workstations. All user inputs are handled through the mouse/menu combination with text-interface.

4. Applications and Results

The graphic editor permits the choice of the basic graphic primitive(point, curve, or surface) to work with, and other general functions such as the display window. The first step in grid generation is choosing a topology. After deciding on the grid topology, the next step is the computation of a suitable surface grid which defines the aircraft geometry. A full wing-body-tail geometry of aircraft has been used to test the capability of PIGG. Originally, the surface grid was performed using commercial CAD system, which provided cross sections for each component. New surface grid of a fighter aircraft was generated by using PIGG as shown in Figure 1.

The design of the block structure is most important step. The major considerations involved in the design of a successful block structure are the description of the geometry of the domain boundaries to a required accuracy, and the limitations of the code to be employed in the analysis. For the present application, an H-type grid was required for analysis. The requirements included the capturing of all the sharp edges of the aircraft around the leading and trailing edges of the wings, engine inlet, etc. Working with these requirements the baseline block structure shown in Figure 2.

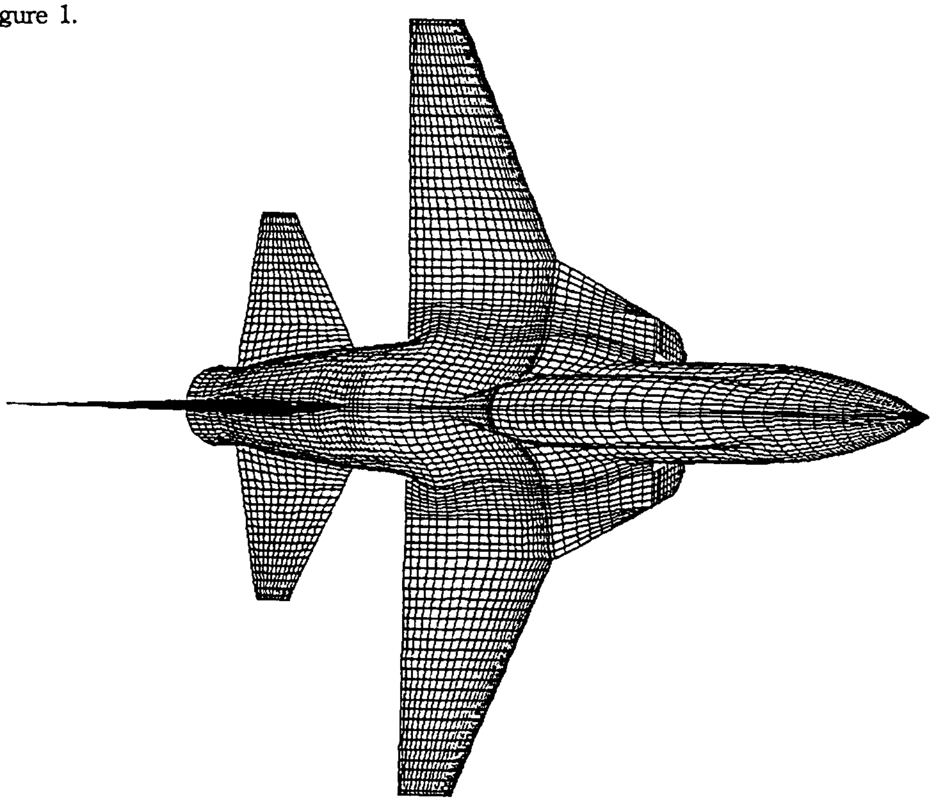

Figure 1 View of a fighter aircraft surface grid

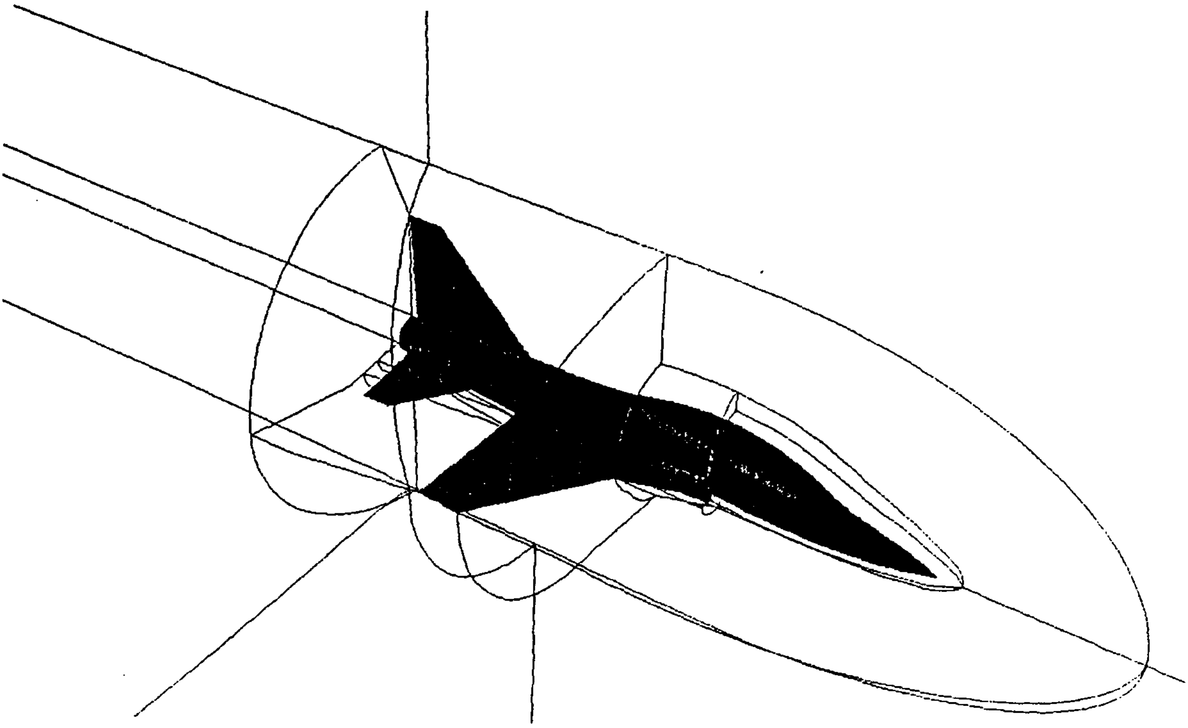

Figure 2 Design of the block structure for the model fighter aircraft

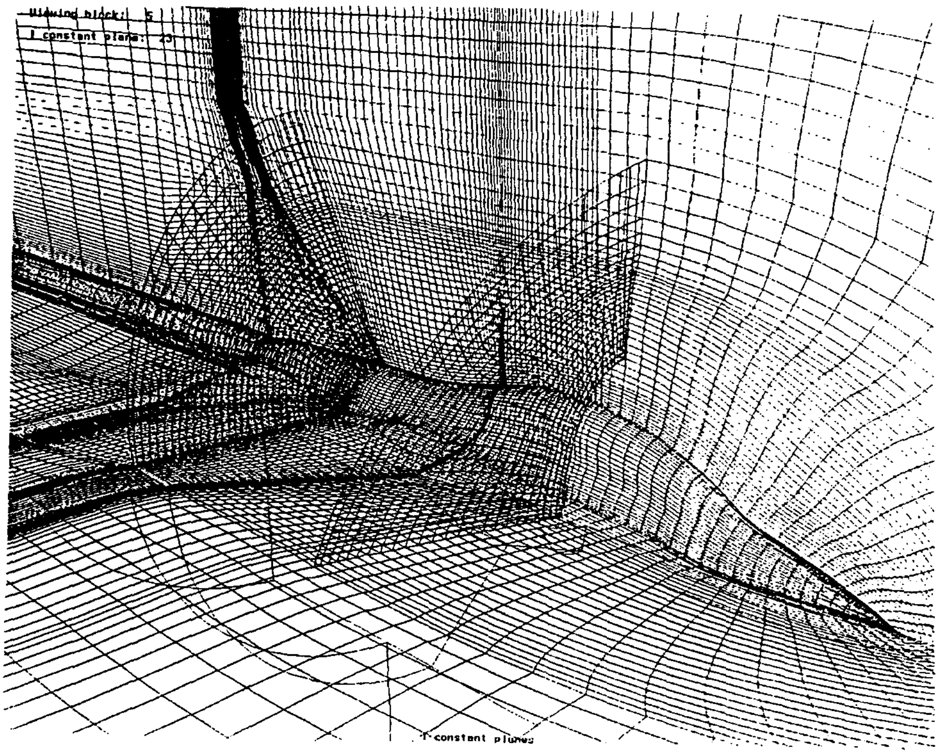

Figure 3 Fine grid details for fighter aircraft.

Once the block structure was designed and the geometry of the individual blocks defined, the database containing the blocks was updated to include the details of the grid that was to be generated in each block. Each block was picked on the screen individually in an interactive manner, and the number of grid points in each coordinate direction was specified. The grid was then generated for each block and connected between the neighboring blocks. In this case, one interior (engine duct intake) block and 17 exterior blocks consisting of 625,000 grid points were generated in 5 weeks. Additional 2-3 weeks were spent refining the grid quality. The user can inspect and modify only a single block at a time. It was very time-efficient and robust. Figure 3 shows details of that grid at different columns of blocks near the aircraft surface.

5. Conclusions

The development of PIGG lasted approximately 12 months. In writing PIGG, one of the important considerations is that the programmer should have some experience in that field. Only from experience one can gain the knowledge of grid generator requirements. Feedback is an essential part of the software development because large programs are written for multi-users and each user has some good and bad opinions to offer. The main programmers job is to determine which opinions to incorporate into the program.

PIGG, by no means is the all-encom passing grid generation tool. It contains the basic functions to generate algebraic grids for a complex geometry. Further enhancements to the program should attend to the following deficiencies. For one, PIGG lacks the surface projection capability where any grid shapes can be forced to adhere exactly to a database geometry shape. This is important for accurately capturing the desired surface shapes. Many times, simple algebraic grid is insufficient; it propagates undesirable edge discontinuities into the interior of the grids. A widely used elliptic and hyperbolic grid generation functions with user-controlled edge angle and spacing would greatly improve PIGG. Another enhancement could be incorporating some advanced graphics display such as solid surface shading and hidden line removal. This would provide user with clearer display of the grid objects. Also, an additional user interface such as the ability to pick grid objects from the screen would greatly save time for seasoned users while an on-line help can aid the novice users. Finally, a sophisticated boundary conditions checker can improve the CFD analysis process greatly, by automa ting the boundary conditions error corrections.

References

[1] Stainbrenner, J. P. "Enhancements to the GRIDGEN System for Increased User Efficiency and Grid Quality", AIAA-92- 0662, AIAA 30th Aerospace Sciences Meeting, Reno, NV, January 1992.

[2] Soni, B. K., "GENIE: Generation of Computational Geometry-Grids for Internal /External Flow Configurations", P.915, Numerical Grid Generation in Computational Fluid Dynamics, Sengupta, S., Hauser, J., Eiseman, P. R., Thompson, J. F.(Eds.), Proceedings of the second International Conference, Pineridge Press, 1988

[3] Soni, B. K., Thompson, J. F., Stokes, M. L., and Shih, M. H., "GENIE^{++}, EAGLEView and TIGER: General Purpose and Special Purpose Graphically Interactive Grid Systems", AIAA-92-0071. ,AIAA 30th Aerospace Sciences Meeting, Reno, NV., January 1992.

[4]Thompson, J. F., "The National Grid Project", Computer Systems in Engineering, Vol.3, Nos.1-4, pp. 393-399, 1992.
5]Yoon, Y. H. " Enhancements and Extensions of EAGLE Grid Generation System", Ph.D Thesis, Miss. State Univ., 1991
[6]. Thompson, J. F. , Warsi, Z. U. A. , and Mastin. C. W , "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Systems - A Review Journal of Computational Physics, Vol.47, P.1-108, 1982
[7].Kwon, J. H. , "Numerical Grid Generations for Cascades With and Without Solution -Adaption, Ph.D. Thesis, Cornell Univ., 1986
[8].Thompson, J. F. , Warsi, Z. U. A., and Mastin. C. W " Numerical Grid Generation: Foundation and Applications", North-Holland, 1985
[9]Kwon, J. H. and Yoon, Y. H. et al., "Numerical Grid Generation of Three-Dimensional Grids", ATRC-413-9253, 1992
[10]Maitz, K. Oleson, L., and Walsh, K., "Graphic Library Reference Manual, C Edition," Version 3.0, Silicon Graphics Inc., Mountain View, CA, 1989.