Old Dominion University Research Foundation

LANGLEY
GRANT
IN-02-CR

261666
29P.

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
COLLEGE OF ENGINEERING AND TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

APPLICATION OF LAGRANGIAN BLENDING FUNCTIONS FOR GRID GENERATION AROUND AIRPLANE GEOMETRIES

By

J. S. Abolhassani, Visiting Assistant Professor

I. Sadrehaghighi, Graduate Research Assistant

and

S. N. Tiwari, Principal Investigator

Final Report
For the period ended September 30, 1989

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665

Under
Master Contract NAS1-18584
Task Authorization No. 32
Dr. Robert E. Smith Jr., Technical Monitor
ACD-Computer Applications Branch

February 1990

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
COLLEGE OF ENGINEERING AND TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

APPLICATION OF LAGRANGIAN BLENDING FUNCTIONS FOR GRID GENERATION AROUND AIRPLANE GEOMETRIES

By

J. S. Abolhassani, Visiting Assistant Professor

I. Sadrehaghighi, Graduate Research Assistant

and

S. N. Tiwari, Principal Investigator

Final Report
For the period ended September 30, 1989

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665

Under
Master Contract NAS1-18584
Task Authorization No. 32
Dr. Robert E. Smith Jr., Technical Monitor
ACD-Computer Applications Branch

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508-0369

February 1990

FOREWORD

This is the final report on the research project "Grid Generation and Inviscid Flow Computation About an F-16 Airplane."

The period of performance on this research was October 1, 1988 through September 30, 1989. This work was supported by the U.S. Air Force through NASA Langley Research Center, NASA Contract NAS1-18584, Task Authorization No. 32. The contract was monitored by Dr. Robert E. Smith Jr. of the Analysis and Computation Division (Computer Application Branch), NASA Langley Research Center, Mail Stop 125.

APPLICATIONS OF LAGRANGIAN BLENDING FUNCTIONS FOR GRID GENERATION AROUND AIRPLANE GEOMETRIES *

Jamshid S. Abolhassani†
Ideen Sadrehaghighi‡
Surendra N. Tiwari§

Abstract

A simple procedure has been developed and applied for the grid generation around an airplane geometry. This approach is based on a transfinite interpolation with *Lagrangian* interpolation for the blending functions. A monotonic rational quadratic spline interpolation has been employed for the grid distributions.

Introduction

In order to study the flow-field around any aerodynamic configuration, a system of nonlinear partial differential equations must be solved over a highly complex geometry. Regardless of the computational approach, the domain of interest should be discretized into a set of points (for the finite difference methods) or a set of elements (for the finite element methods). This step is commonly referred to as *grid generation.*

Selection of the grid *topology* is the first step in the generation of *structured* grid. In this step, orientation of the computational coordinates must be selected relative

*This paper is declared a work of the U. S. Government and is not subjected to copyright protection in the United States

†Visiting Assistant Professor, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23529-0247, Member, AIAA.

‡Graduate Research Assistant, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23529-0247, Student Member, AIAA.

§Eminent Professor, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23529-0247, Member, AIAA.

to the physical coordinates. For complex geometries, one may have to select different computational coordinate systems for different regions of the physical domain. In this case, one physical domain is mapped into several computational subdomains, and each subdomain is referred to as a block. Therefore, it is possible to have a boundary-fitted coordinate system for a highly complex configuration. The objective of this study is to present a simple procedure for generating a relatively orthogonal grid for a generic airplane geometry.

Topology of an Airplane

In order to establish a grid topology for any geometry, it is essential to examine each component separately [1]. A typical airplane geometry has two important components: the fuselage and the wing. A fuselage has a circular like cross-section which suggests a natural O-type grid. This topology produces a nearly orthogonal grid with one line of polar singularity at the nose. In the streamwise direction, it is possible to have either a C- or an H-type grid. If the fuselage has a small slope near the nose, then it is better to use an H-type grid in the streamwise direction. The other choice is to have a C-type grid in the streamwise direction which is good for the fuselage with a blunt-nose. If the nose is sharp, C-type topology may generate a slightly skewed grid near the nose. In short, the use of an O-H or an O-C topology will result in a nearly orthogonal grid with one line of polar singularity at the nose.

In general, a wing possesses its own natural coordinate system which may not be compatible with the fuselage's coordinate system. However, it is possible to generate an H-, O- or a C- type grid in the streamwise direction, and a C- or an H-type in the crosswise direction. It is conceivable to generate a single-block grid about these two components, but this grid will be skewed for any practical applications. In order to maintain a minimum of C^0 continuity at the interfaces, it is essential to select a compatible topology for the wing and the fuselage. A *dual-block* grid possesses much less skewness than a single-block grid. The dual-block grid consists of two large blocks, one covering the top part of the physical domain, and the other block covering the bottom part of the physical domain. The dual-block topology is a direct result of using an H-type grid for the wing. For higher continuity (C^1 and above), an oscillatory transfinite interpolation can be used to generate the interior grid. Then, it is possible to ensure the orthogonality at the interface as well.

If the wing-tip has a finite area, then the topology of the grid needs to be changed accordingly. This change may create additional blocks which is not desirable. Furthermore, addition of vertical and horizontal tail surfaces may also change the topology of the grid. These changes depend on the geometry of the tails and the required grid topology. If the leading and the trailing edges of the tails are sharp, then it is a good idea to use an H-type grid around them. This will not change the dual-block topology. However, there would be one singular line emanating from the leading and the trailing edges of each tail surface. These singular lines are located on the boundaries of the computational domain which will not have a dramatic effect on the flow code. It is also possible to have an O- or a C-type grid around the tails, this choice will result in creation of additional blocks.

Once the grid topology has been selected, then the grid on the boundary surfaces can be generated. This step depends on how a surface is defined. A surface can be defined either by a set of analytical functions or by a set of cross-sections. The former requires no interpolation and the latter requires some sort of bi-directional interpolation. In this study, the fuselage surface is given by analytical functions and the wing and tails by their cross-sections. More details of the geometry is given in appendix A.

Discretization of a Curved Line

Before generating the surface grid, one needs to compute the grid-point distribution along the entire or part of the boundary edges. This distribution must be monotonic in the parameter space, and it can be computed by an analytical function or by a numerical approximation. Analytical functions are generally limited to simple curves. However, a complex curve can be decomposed into several sections, and analytical functions can be used for each section [2]. The advantages of analytical functions are their simplicity and the guaranteed monotonicity.

Similarly, a numerical approximation can be used to compute the grid-point distribution on a curve. This approach is widely used and care must be taken to insure monotonicity and high accuracy. For example, the natural cubic spline is C^2 continuous, and it can generate smooth grid-point distribution. If the data has a high second derivative, the result may not be monotonic. The cubic spline can be modified to control monotonicity [3] and still be C^2 continuous. However, this scheme may not reproduce the initial data. This method has been applied for two-boundary grid generation with much success [2]. The other option is to use a lower order polynomial (e. g. C^1) with guaranteed monotonicity. For instance, a Monotonic Rational Quadratic Spline interpolation (MRQS) is always monotonic [4] and smooth. Figure 1 shows results from a cubic spline and a monotonic rational quadratic spline. It can be seen clearly that the MRQS can generate a monotonic grid distribution unlike a cubic spline. The MRQS scheme is an explicit scheme and does not require any matrix inversion. The disadvantage of MRQS scheme is its accuracy (C^1). The algorithm for this method can be found in [4].

Surface Grid Generation

Once the grid is generated on the boundary edges, the surface grid can be generated by analytical functions or by an appropriate bi-directional interpolation. Surfaces are either a physical surface (solid) or a nonphysical surface (fictitious). Figure 2 shows the grid on the physical surfaces of a typical airplane geometry. The next step is to generate grid points on the remaining surface portions (nonphysical). For example, the symmetry surface in x-y plane (Fig. 3) is surrounded by a number of lines. This region can be divided into a number of subregions as shown in Fig. 3. In this case, it is divided into five subregions. This subdivision is arbitrary; however, it is a good idea to subdivide along a computational coordinate direction (e. g. constant ξ).

As it was mentioned before, each subregion may be defined by two or more grid lines. For each subregion, grid points can be generated by either an algebraic or a differential method. An extensive discussion of both methods can be found in [5]. Most algebraic methods are either based on a *transfinite* [6,8] or a *multi-surface* method [7]. In this investigation, a transfinite interpolation has been used with the application of a Lagrangian interpolation for the blending functions. A significant extension of the original formulation by Gordon and Hall [6] has made it possible to generate grids for highly complex geometries with a high degree of local control [8].

If the coordinates of some part of a surface $[\vec{F}(\xi,\eta)]$ with their derivatives are known, then it is possible to generate the interior grid by a transfinite interpolation. In general, a two-step transfinite interpolation (or multi-variant interpolation) for curved surfaces can be expressed as

$$\vec{F}(\xi,\eta) = \{x(\xi,\eta), \quad y(\xi,\eta), \quad z(\xi,\eta)\}^T, \tag{1a}$$

$$\vec{F}_1(\xi,\eta) = \sum_{\ell=1}^{L}\sum_{n=0}^{P} \alpha_\ell^{(n)}(\xi)\vec{A}_\ell^{(n)}(\eta), \tag{1b}$$

$$\vec{F}(\xi,\eta) = \vec{F}_1(\xi,\eta) + \sum_{\ell=1}^{M}\sum_{n=0}^{Q} \beta_\ell^{(n)}(\eta)\left[\vec{B}_\ell^{(n)}(\xi) - \frac{\partial^n \vec{F}_1}{\partial\eta^n}(\xi,\eta_\ell)\right], \tag{1c}$$

where the $\vec{A}_\ell^n$ and $\vec{B}_\ell^n$ are the known coordinate lines on the surface with their derivatives,

$$\frac{\partial^n \vec{F}}{\partial\xi^n}(\xi_\ell,\eta) = \vec{A}_\ell^n(\eta), \tag{1d}$$

$$\ell = 1,2,\ldots,L, \quad n = 0,1,\ldots,P,$$

$$\frac{\partial^n \vec{F}}{\partial\eta^n}(\xi,\eta_\ell) = \vec{B}_\ell^n(\xi), \tag{1e}$$

$$\ell = 1,2,\ldots,M, \quad n = 0,1,\ldots,Q,$$

and $\alpha_\ell^{(n)}(\xi)$ and $\beta_\ell^{(n)}(\eta)$ are the univariant blending functions. These functions are subjected to the following Cardinal conditions

$$\frac{\partial^m \alpha_\ell^{(n)}(\xi_i)}{\partial\xi^m} = \delta_{\ell,i}\delta_{n,m}, \tag{2a}$$

$$\frac{\partial^m \beta_\ell^{(n)}(\eta_i)}{\partial\eta^m} = \delta_{\ell,i}\delta_{n,m}. \tag{2b}$$

These conditions allow the input boundaries to be reproduced.

Selection of the blending function depends on the number of specified boundaries. If only two boundaries are defined in one computational direction, then the Lagrangian interpolation would convert to a simple linear interpolation,

$$\alpha_1(\xi) = \frac{(\xi_2 - \xi)}{(\xi_2 - \xi_1)}, \qquad \alpha_2(\xi) = \frac{(\xi - \xi_1)}{(\xi_2 - \xi_1)}. \tag{3}$$

This works if the boundaries do not contain sharp discontinuities. Otherwise, these discontinuities will propagate into the interior regions. One way to alleviate this problem is to construct a blending function that has a very small value away from the boundaries. For example, the following blending functions have these criteria [8].

$$\alpha_1(\xi) = \frac{e^{K\frac{(\xi_2-\xi)}{(\xi_2-\xi_1)}} - 1}{e^K - 1}, \tag{4a}$$

$$\alpha_2(\xi) = \frac{e^{K\frac{(\xi-\xi_1)}{(\xi_2-\xi_1)}} - 1}{e^K - 1} \tag{4b}$$

where K is a negative number greater than one. The larger the K the less the discontinuity will propagate. A similar blending function can be constructed for the η direction.

The other choice for the blending function is the *Lagrangian* interpolation which satisfies the Cardinal conditions. For example, if the lines in the ξ-direction are given at $\xi_1, \xi_2, \ldots, \xi_n$, then the blending function $\alpha_\ell(\xi)$ can be defined as,

$$\alpha_\ell(\xi) = \prod_{\substack{j=1 \\ j\neq\ell}}^{n} \frac{(\xi - \xi_j)}{(\xi_\ell - \xi_j)}. \tag{5}$$

For n=2, this equation will reduce to Eq. (3). For a surface which is defined by several lines, one can use the general definition in Eq. (5).

In Fig. 3, lines AC, AM, DE, FH, EH, IJ, KL, JL, and MP are known. In Region I, the interior grid can be generated by using two lines at DE, and one line at AB and AD, and the normal derivative at AB. Line BE is computed as a part of the solution. Regions II and III can be generated in the same way. The grid in Region IV can be generated by using two lines at the interface (GJLN), two lines at GH, and one line at NO. Then, the HO interface is computed as a part of the solution. By using two lines at the interface (GJLN), the grid lines are C^1 continuous across the interface. Lastly, the grid for Region V can be generated by using two lines at the interface (BEHO), and lines OP and BC. Therefore, it is possible to generate grids that are C^1 continuous at the interfaces without specifying the interfaces or their derivatives.

A general purpose subroutine is written for Eqs. (1) and (5) which can be found in Appendix B. Some results of this procedure are shown in Figs. 3 and 4, and a nonphysical surface can be generated in a similar fashion. The outer boundary and the outflow boundary are shown in Figs. 5 and 6, respectively. The solid lines show the grid on the solid surfaces, and the dotted lines show the grid on the nonphysical boundaries.

Interior Generation Procedure

In general, decomposition of the physical domain produces several blocks. Each block is usually defined by six sides, and each side can be defined by either a surface, plane, line, or a point. If one side of a block collapses to a line or a point, then there would be a singularity in the block. In some instances, a block may have been defined by less than six surfaces. Once the surfaces are defined, the interior grid can be computed by any standard grid generation technique. In this study, a transfinite interpolation has been used to generate the interior grid points.

Once the boundary surfaces ($\vec{F}(\xi,\eta,\zeta)$) are known, then it is possible to generate the interior grid by a transfinite interpolation. The three-step transfinite interpolation can be expressed for the vector $\vec{F}(\xi,\eta,\zeta)$ as

$$\vec{F}(\xi,\eta,\zeta) = \{x(\xi,\eta,\zeta),\, y(\xi,\eta,\zeta),\, z(\xi,\eta,\zeta)\}^T, \tag{6a}$$

$$\vec{F}_1(\xi,\eta,\zeta) = \sum_{\ell=1}^{L}\sum_{n=0}^{P} \alpha_\ell^{(n)}(\xi)\vec{A}_\ell^{(n)}(\eta,\zeta), \tag{6b}$$

$$\vec{F}_2(\xi,\eta,\zeta) = \vec{F}_1(\xi,\eta,\zeta) + \sum_{\ell=1}^{M}\sum_{n=0}^{Q} \beta_\ell^{(n)}(\eta)\left[\vec{B}_\ell^{(n)}(\xi,\zeta) - \frac{\partial^n \vec{F}_1}{\partial \eta^n}(\xi,\eta_\ell,\zeta)\right], \tag{6c}$$

$$\vec{F}(\xi,\eta,\zeta) = \vec{F}_2(\xi,\eta,\zeta) + \sum_{\ell=1}^{N}\sum_{n=0}^{R} \gamma_\ell^{(n)}(\zeta)\left[\vec{C}_\ell^{(n)}(\xi,\eta) - \frac{\partial^n \vec{F}_2}{\partial \zeta^n}(\xi,\eta,\zeta_\ell)\right], \tag{6d}$$

where the $\vec{A}_\ell^n$, $\vec{B}_\ell^n$ and $\vec{C}_\ell^n$ are the known surface locations and their derivatives.

In this study, the interior grid points are generated based on the definition of six surfaces, and the derivatives at the boundary are not included. Equations (4a-4b) are used for the blending functions in all directions. The method described here has been used to write a computer code which is capable of generating a grid over an airplane with fuselage, wing and tails. Some of these results are shown in Figs. 7-9.

Results and Conclusions

A computer program has been developed to generate a multi-block grid around an airplane geometry. The technique is based on a three-dimensional transfinite interpolation with Lagrangian interpolation function for the blending functions. By using a Lagrangian interpolation, it is possible to enforce continuity across the interfaces without the derivative information. This procedure is proven to be very simple and effective. It is also possible to control the grid spacing by using a monotonic rational quadratic spline interpolation.

Acknowledgements

The authors would like to thank Ms. Joan I. Pitts of NASA Langley (Computer Application Branch) for her technical assistance. Authors were partially supported by a Grant from NASA Langley Research Center (NCCI-68). The authors would like to thank Dr. Joseph Shang of Wright Research and Development (Flight Dynamics Lab) for initiating this investigation.

References

[1] Abolhassani, J. S., Smith, R.E., " *Three-Dimensional Grid Generation About a Submarine,* "**Numerical Grid Generation in Computational Fluid Mechanics**, Edited by S. Sengupta, et. al., Pineridge Press Limited, Swansea, United Kingdom, pp. 505-515, 1988.

[2] Smith, R. E., Wiese, M. R., "*Interactive Algebraic Grid Generation*," NASA Technical Paper TP-2533, March, 1986.

[3] Reinsch, C. H., "*Smoothing by Spline Functions*," **Numerical Mathematics**, Volume 10, No. 3, pp. 177-183, 1967.

[4] Gregory, J. A., Delbourgo, R., "*Piecewise Rational Quadratic Interpolation to Monotone Data*," **IMA Journal of Numerical Analysis**, Volume 2, pp. 123-130, 1982.

[5] Thompson, J. F., Warsi, Z. U. A., Mastin, C. W., **Numerical Grid Generation: Foundations and Applications**, North-Holland, New York, 1985.

[6] Gordon, W. J., Hall, C. A., "*Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation*," **International Journal of Numerical Methods in Engineering**, Volume 7, pp. 461-477, 1973.

[7] Eiseman, P. R., "Grid Generation for Fluid Mechanics Computations," **Annual Review of Fluid Mechanics**, Vol. 17, 1985, pp. 487-522.

[8] Smith, R. E., Eriksson, L. -E., "*Algebraic Grid Generation*," **Computer Methods in Applied Mechanics and Engineering**, Volume 64, pp. 285-300, 1987.

Appendices

A. Solid Surfaces

All solid surfaces are generated by analytical functions. In the present study, solid surfaces are composed of fuselage, wing, horizontal and vertical tail surfaces. The fuselage consists of a cylindrical section and a nose. The nose is either sharp or blunt. A sharp nose is approximated by rotating a parabola about the center line. This parabola can be expressed as,

$$y = ax^2 + bx + c. \tag{A.1}$$

Similarly, a quarter of an ellipse is used for the blunt nose.

$$1 = \left(\frac{x-a}{a}\right)^2 + \left(\frac{y-b}{b}\right)^2 \tag{A.2}$$

In Eqs. (A.1-A.2), constants a, b, and c are selected such that the surface is continuous (at least C^1) at the intersection of the nose and the cylindrical section. In the streamwise direction, the grid points are generated using a monotonic rational spline. In the crosswise direction, the grid points are generated using an exponential function.

Surfaces of the wing and tails are described by several cross-sections using standard four digit NACA airfoil. The wing has a NACA-0010 for a cross-section with closed wing tip. The grid points are concentrated more toward the leading and trailing edges. In the cross-wise direction, grid points are concentrated near the fuselage. The intersection of the wing and fuselage are generated analytically. Grid points for the vertical and horizontal tails are generated in a similar fashion. The cross-sections of horizontal and vertical tails are NACA 0006 and 0005, respectively.

B. Computer Code for a Transfinite Interpolation

This subroutine is based on a transfinite interpolation with a Lagrangian blending function (Eqs. 1 and 5). The following section describes the subroutine arguments.

Definitions:,

F Grid position (x, y, or z)

IL, JL Number of grid points in i and j-directions, respectively.

II(i), JJ(j) This array stores the locations of known grid lines in i- and j-directions, respectively (1 for known grid lines).

IS, IE, JS, JE Starting and ending of region (computational) of interest.

IMAX, JMAX Array dimensions.

Example: Consider surface IV in Fig. 3. In this case, five grid lines are known: two lines at GH, two lines at GJLN, and one line at NO. The size of the grid is 95 in the i-direction and 50 in the j-direction. Point G is at (70,15) and point O is at (95,25).

```
IL=95

JL=50

II(i)=0 except i=69, 70, and 95,

JJ(j)=0 except j=14, and 15,

IS=70

IE=95

JS=15

JE=25

IMAX=95
```

```
JMAX=95
CALL TRANS2D(F,IL,JL,II,JJ,IS,IE,JS,JE,
                               IMAX,JMAX)

C
      SUBROUTINE TRANS2D(F,IL,JL,II,JJ,IS,IE,JS,JE,IMAX,JMAX)
      PARAMETER(NMAX=95)
      DIMENSION F(IMAX,JMAX),II(NMAX),JJ(NMAX)
      DIMENSION PH(NMAX,NMAX),SI(NMAX,NMAX),III(NMAX),JJJ(NMAX)
C
      IF(IL.GT.NMAX.OR.JL.GT.NMAX) THEN
        WRITE(*,*)  'THE PARAMETERS ARE SMALL, STOP IN SUB TRANS2D'
          STOP
      END IF
      NM=0
      DO 100 I=1,IL
         IF(II(I).NE.0) THEN
           NM=NM+1
           III(NM)=I
         END IF
 100  CONTINUE
C
      MM=0
      DO 110 J=1,JL
         IF(JJ(J).NE.0) THEN
           MM=MM+1
           JJJ(MM)=J
         END IF
 110  CONTINUE
C
C.....SETUP THE BLENDING FUNCTIONS (LAGRANGIAN INTERPOLATION)
C
      DO 200 N=1,NM
         DO 210 I=1,IL
            PH(N,I)=1.0
            DO 220 N1=1,NM
               IF(N1.EQ.N) GOTO 220
               PH(N,I)=PH(N,I)*FLOAT(I-III(N1))/FLOAT(III(N)-III(N1))
 220        CONTINUE
 210     CONTINUE
 200  CONTINUE
C
      DO 300 M=1,MM
         DO 310 J=1,JL
            SI(M,J)=1.0
            DO 320 M1=1,MM
               IF(M1.EQ.M) GOTO 320
               SI(M,J)=SI(M,J)*FLOAT(J-JJJ(M1))/FLOAT(JJJ(M)-JJJ(M1))
 320        CONTINUE
 310     CONTINUE
 300  CONTINUE
C
C.....COMPUTE THE TRANSFINITE INTERPOLATION
C
      DO 1000 J=JS,JE
```

```
         IF(JJ(J).NE.0) GOTO 1000
         DO 1100 I=IS,IE
            IF(II(I).NE.0) GOTO 1100
C
            F(I,J)=0.
C
            DO 1200 N=1,NM
               F(I,J)=F(I,J)+PH(N,I)*F(III(N),J)
 1200       CONTINUE
C
            DO 1300 M=1,MM
               F(I,J)=F(I,J)+SI(M,J)*F(I,JJJ(M))
 1300       CONTINUE
C
            DO 1400 N=1,NM
               DO 1500 M=1,MM
                  F(I,J)=F(I,J)-PH(N,I)*SI(M,J)*F(III(N),JJJ(M))
 1500          CONTINUE
 1400       CONTINUE
 1100    CONTINUE
 1000 CONTINUE
C
      RETURN
      END
```

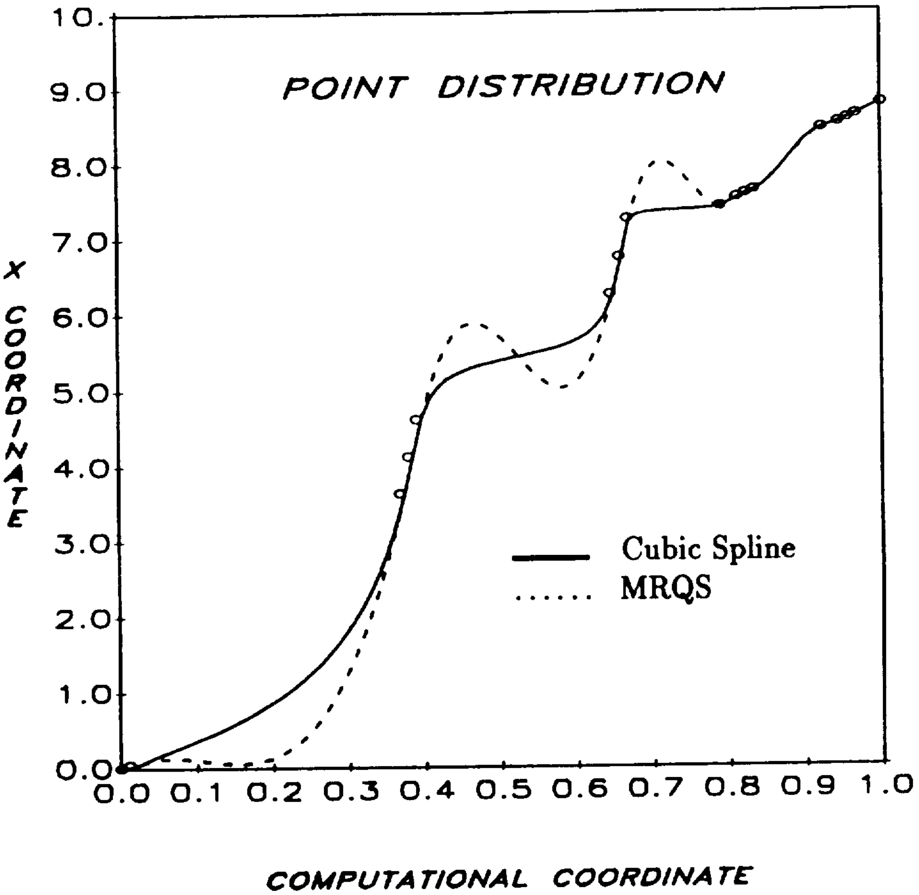

Fig. 1 Point Distribution in the Stream-Wise Direction.

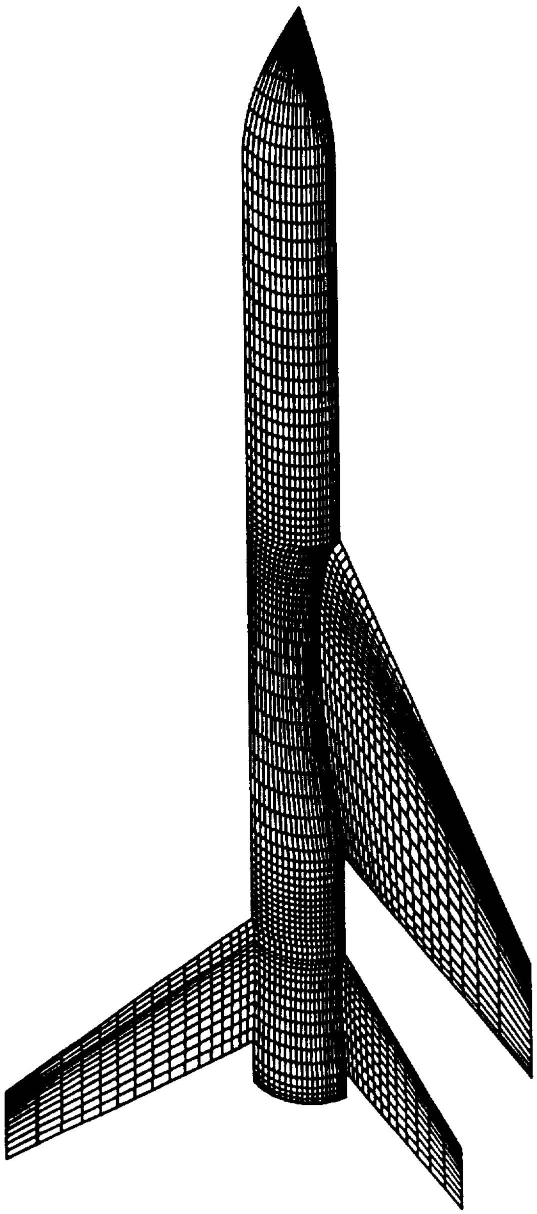

Fig. 2 Grid on the Solid Surfaces

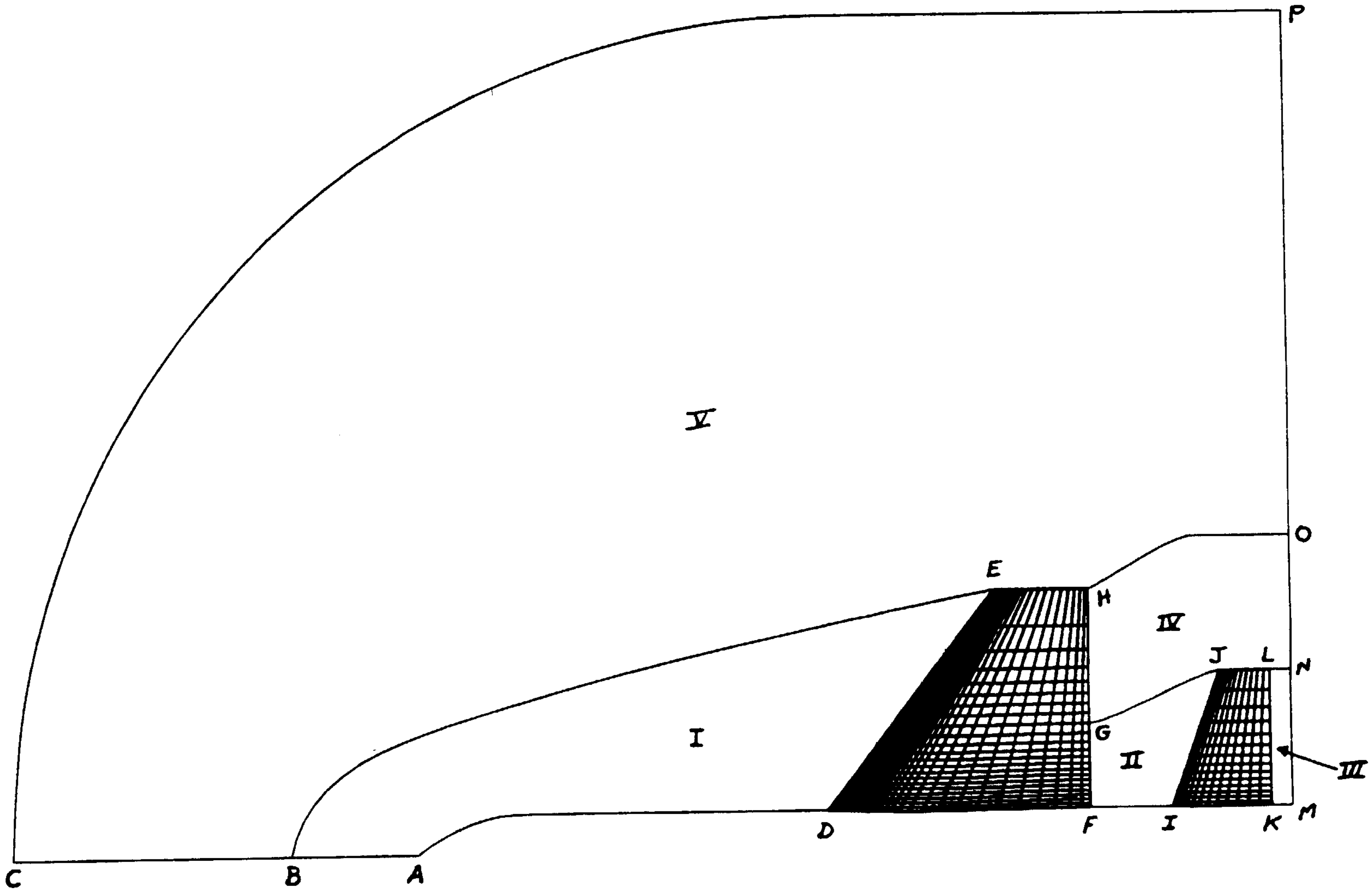

Fig. 3 Symmetry Surface (xy Plane)

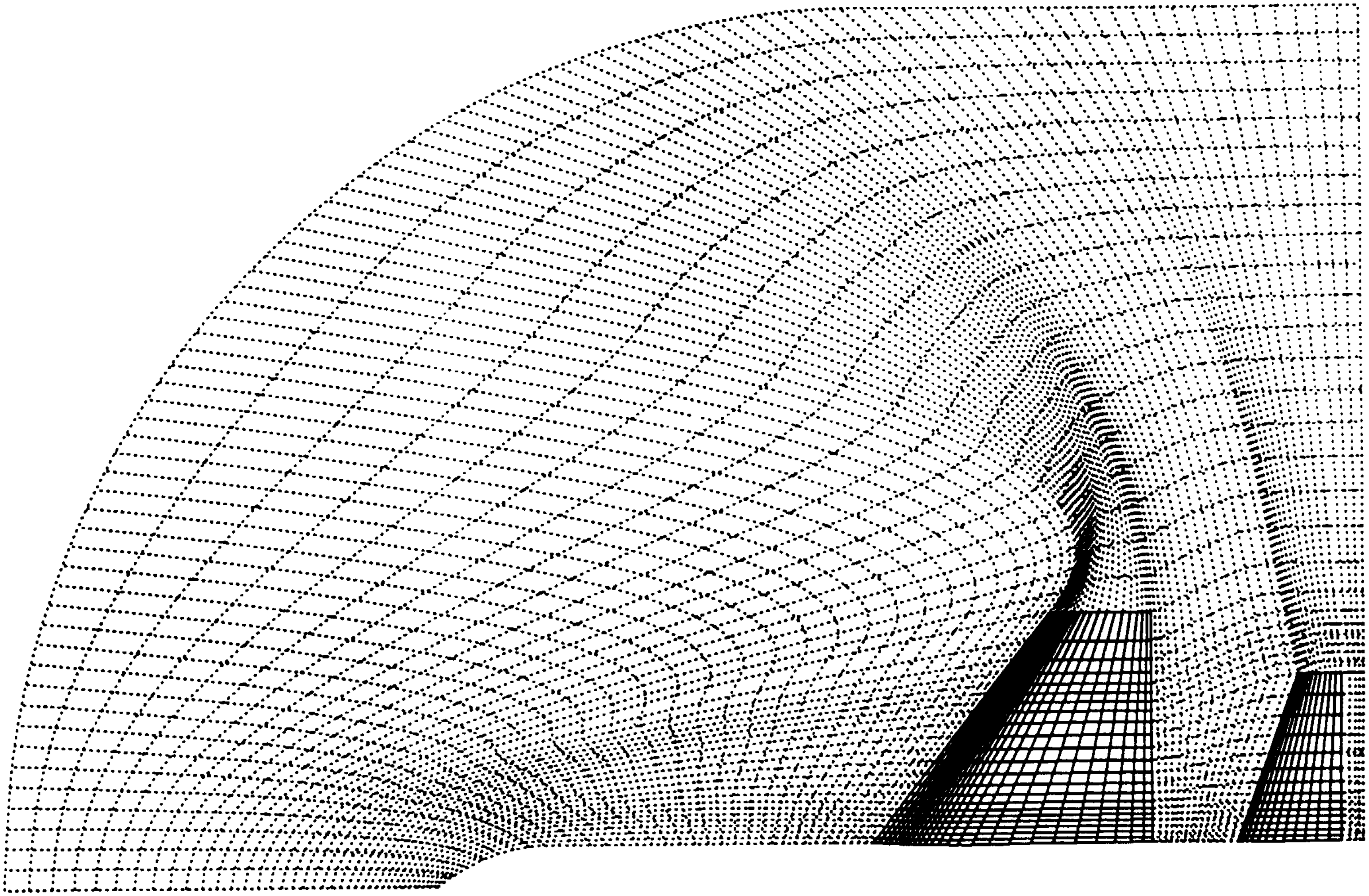

Fig. 4 Grid on the Symmetry Surface (xy Plane)

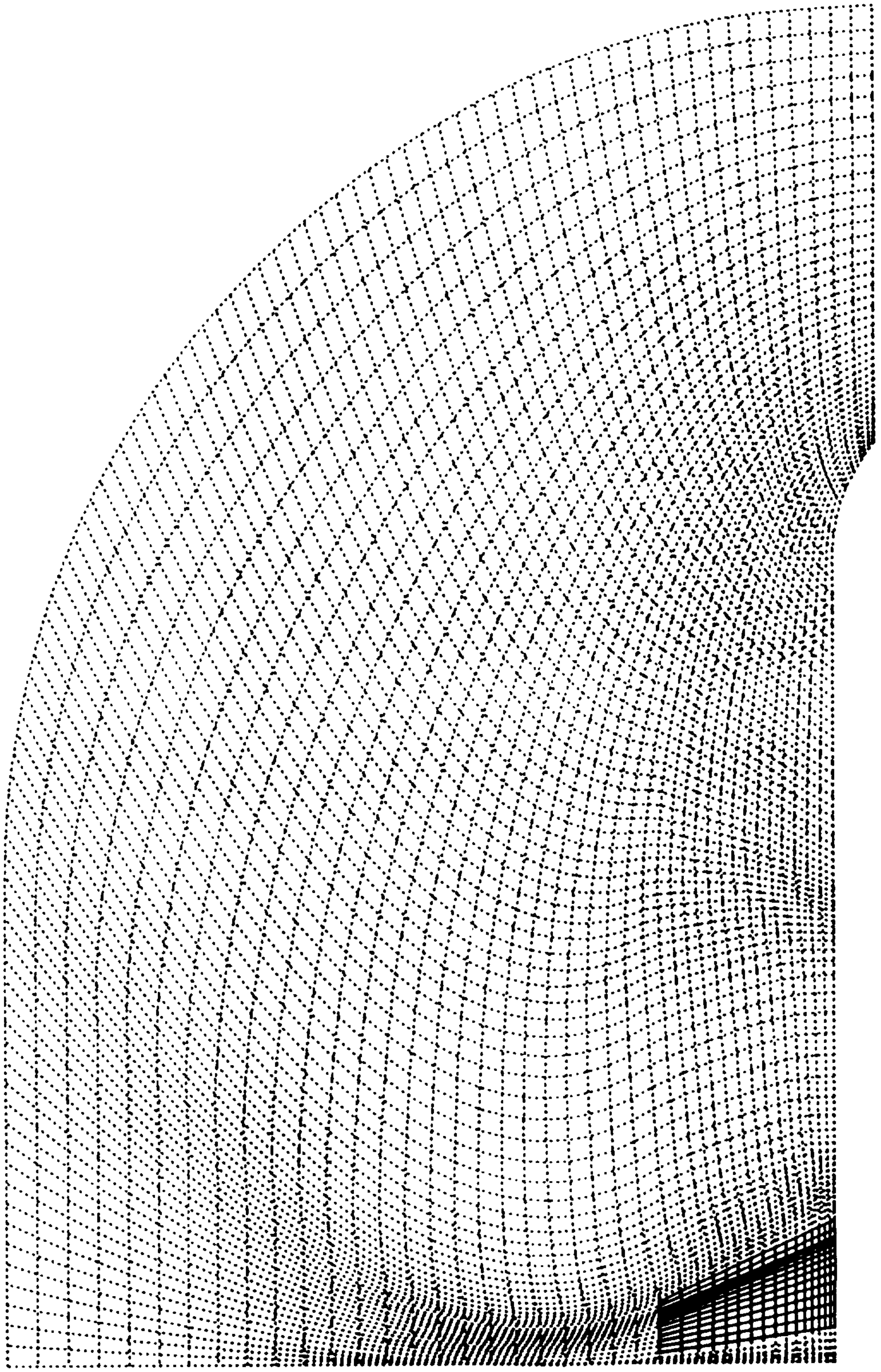

Fig. 5 Grid on the Symmetry Surface (xz Plane)

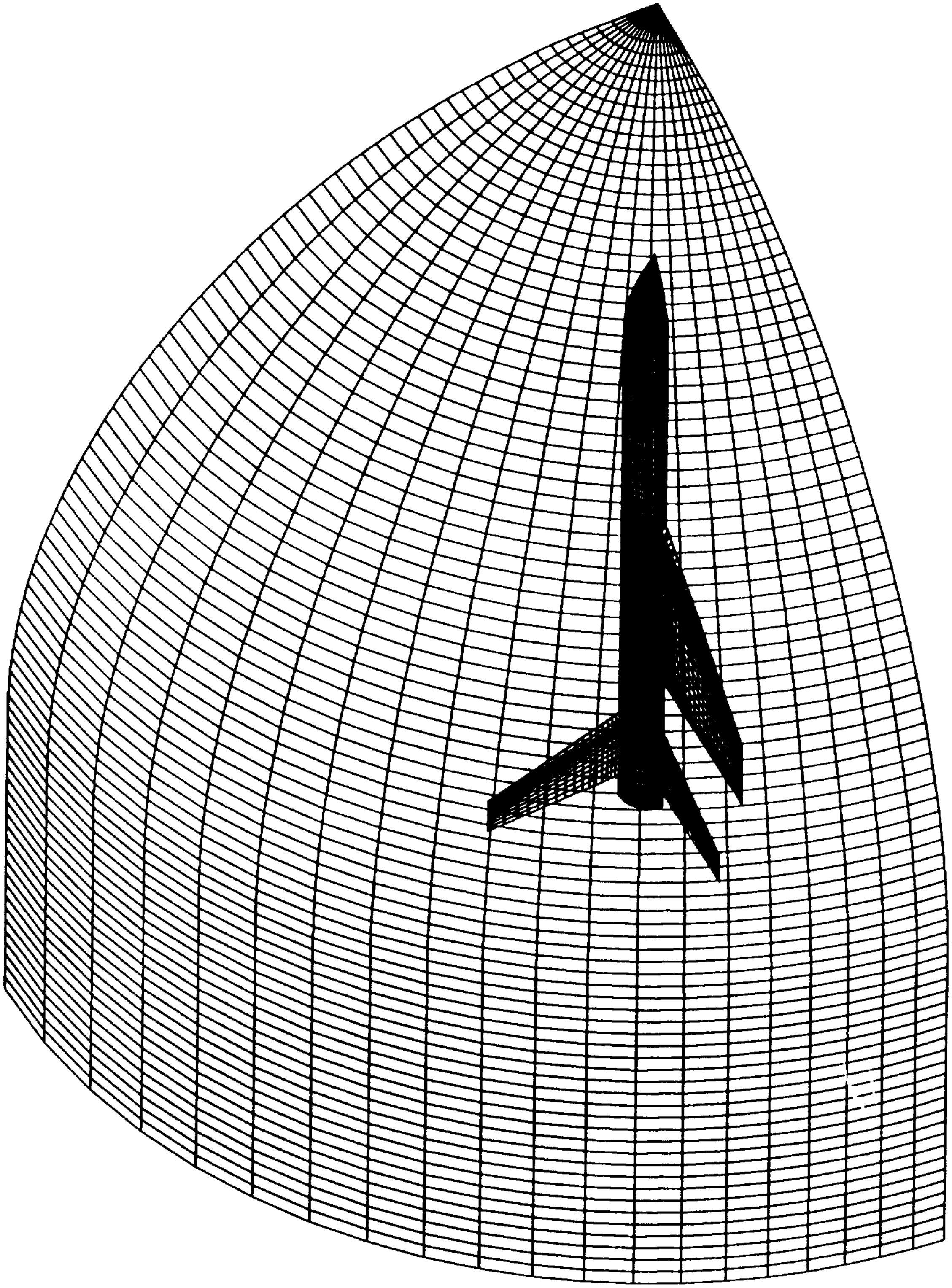

Fig. 6 Grid on the Outer Boundary

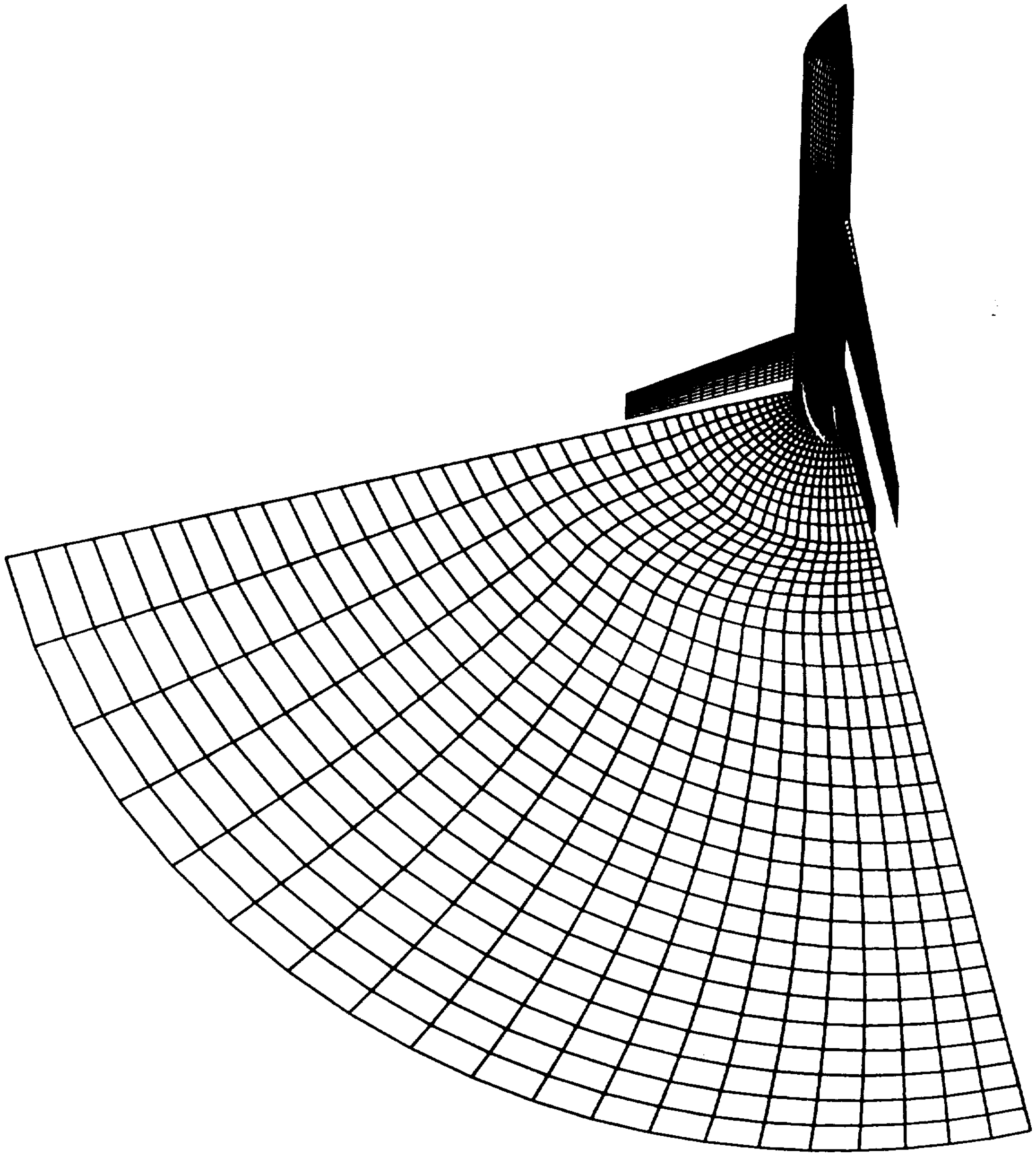

Fig. 7 Grid on the Outflow Boundary

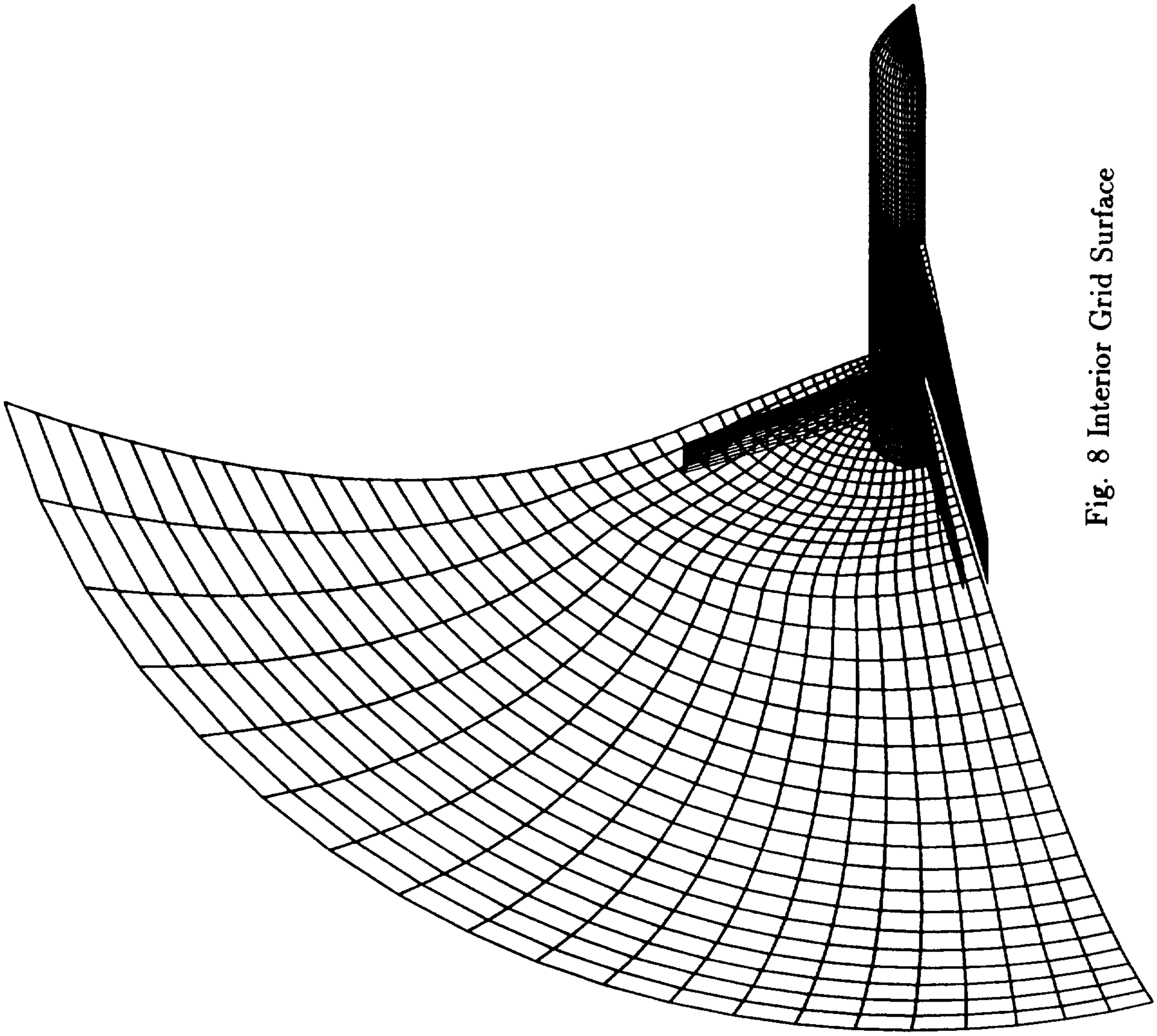

Fig. 8 Interior Grid Surface

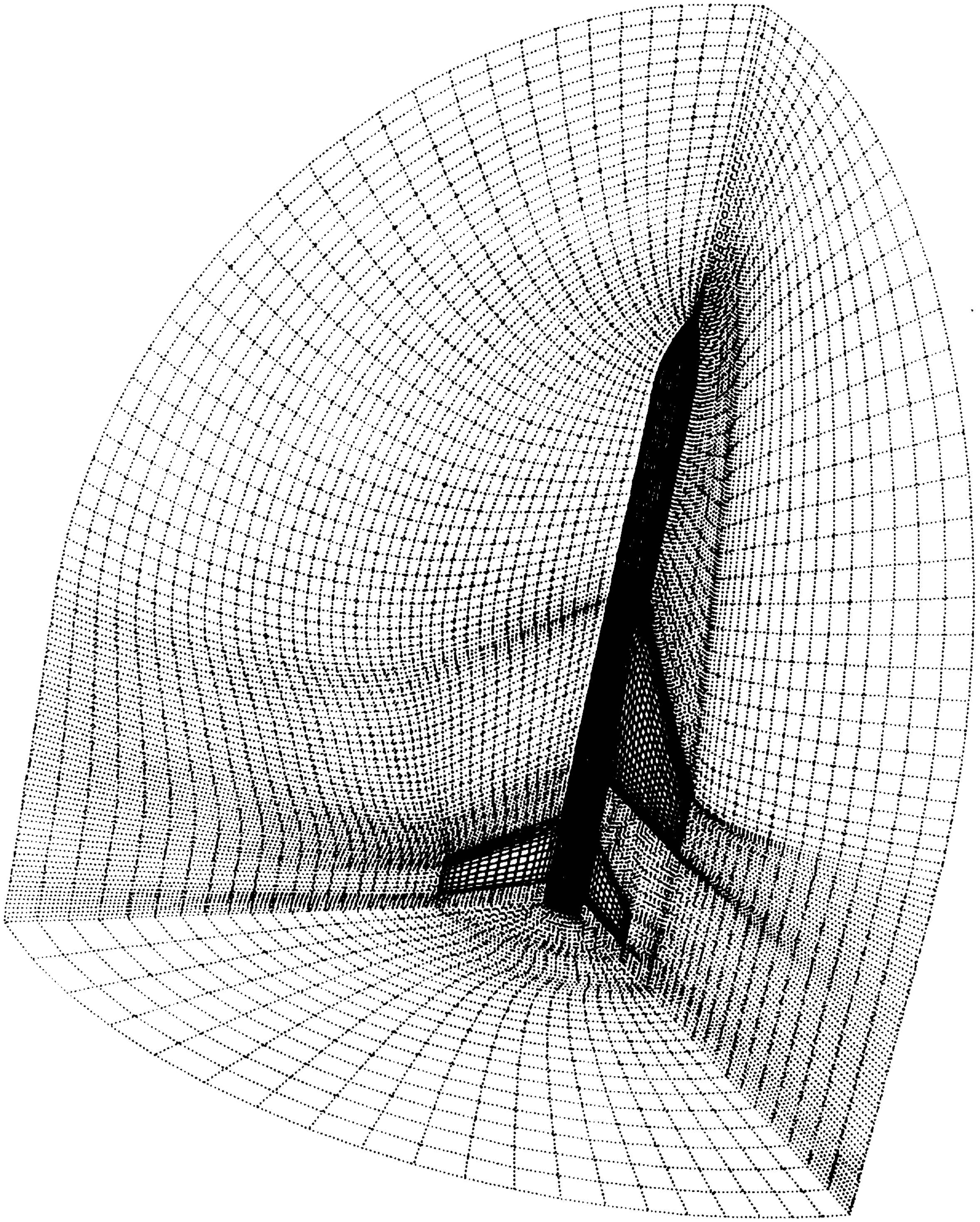

Fig. 9 Grid on the Physical and Nonphysical Surfaces