JOURNAL OF COMPUTATIONAL PHYSICS 53, 319–330 (1984)

A Domain Decomposition and Overlapping Method for the Generation of Three-Dimensional Boundary-Fitted Coordinate Systems

KAZUYOSHI MIKI

AND

TOSHIYUKI TAKAGI

Energy Research Laboratory, Hitachi Ltd.
1168 Moriyama-cho, Hitachi-shi, Ibaraki, 316 Japan

Received April 26, 1983; revised July 15, 1983

A new method is presented for numerically generating boundary-fitted coordinate systems for arbitrarily shaped three-dimensional regions. In the method, the three-dimensional region of interest is decomposed into several hexahedrons, each of which has two grid surfaces overlapped with each of the neighboring hexahedrons. Based on this method, a new computer program GRID-3D has been developed, which allows the generation of an unlimitted number of different types of coordinate systems. Application to a variety of geometries confirms that GRID-3D is a convenient and efficient tool in the generation of boundary-fitted coordinate systems even for a considerably complicated configuration consisting of many different components.

1. INTRODUCTION

In the numerical solution of partial differential equations, the discretization of field requires the following considerations with regard to the arrangement of its grid points:

(a) accurate representation of boundary shapes,

(b) grid concentration in regions of expected large gradients of the physical solution,

(c) smooth coverage of the entire field, and

(d) specification of the relations between grid points, such as computational sequence and relative positions.

In many engineering problems, regions in the immediate vicinities of boundary surfaces usually exhibit large gradients of the solution, and are dominant in determining the character of the entire solution; examples include a viscous boundary

0021-9991/84 $3.00

581/53/2-8

layer in fluid dynamics and stress concentration in solid mechanics. Therefore, the solution accuracy, in particular, depends on the treatment of boundary conditions for the above first and second considerations.

The finite difference method uses one of the Cartesian cylindrical and spherical coordinate systems in such a way that the coordinate lines are coincident with the boundaries. But this approach needs some interpolation between the grid points for complicated boundary shapes with strong curvature or slope discontinuities, which may introduce significant numerical errors. The finite element method, on the other hand, has geometrical advantages for matching complicated boundaries, but it requires a lot of experience and time for constructing the grid points over the field while satisfying the above third and fourth considerations.

One of the most effective ways to overcome these problems is the use of boundary-fitted coordinate systems [1]. The technique for such coordinate systems is based on an automated numerical generation of a curvilinear coordinate system having a coordinate line coincident with each boundary of the physical region of interest. It also has capabilities to exercise control over the spacings of the coordinate lines in the entire field, and to simplify the finite difference expression by transforming three-dimensional geometries in a physical space to a rectangular shape in the transformed space.

This paper presents a new method of numerically generating boundary-fitted coordinate systems for arbitrarily shaped three-dimensional regions. The method uses the techniques of decomposing the physical region into several hexahedrons and of overlapping neighboring hexahedrons with two grid surfaces. Based on this method, a new computer program, called GRID-3D, has been devloped. Its capabilities and applications are also presented here.

2. Description of the Method

The generation of boundary-fitted coordinate systems can be accomplished by solving numerically the elliptic partial differential equations with the Dirichlet boundary conditions [2]. But when dealing with geometrically complicated three-dimensional domains, the generation procedure becomes more complex and has less flexibility in adapting to a broad variety of geometries.

One approach to a complicated three-dimensional domain is to divide it into a number of geometrically simple subdomains, and to patch together coordinate systems generated separately for each subdomain [3]. In this approach, the coordinate lines can remain continuous across the surface of juncture between two adjacent subdomains. However, some first derivatives of the coordinate lines will be discontinuous at the boundary without any provision for control of the intersection angles there, which may introduce significant numerical error or decrease the convergence speed in solving physical partial differential equations.

In the present method, a complicated physical domain is divided into several hexahedrons, each of which has six curved or plane surfaces. Each subdomain grid is generated independently, and then is joined with the others to form a composite grid for the original domain. To ensure the composite grid remains both continuous and smooth across the boundaries, an overlap of two grid surfaces is adopted between any two adjacent hexahedrons.

Figure 1 illustrates the procedue for generating the grid, which consists of the following steps:

Step 1. Calculate the coordinates of the inner grid points for a hexahedron using the predetermined grid coordinates on the six surfaces of the hexahedron.

Step 2. Transfer the calculated coordinates of the overlapped grid points to the outer surfaces of the neighboring hexahedrons.

Step 3. Using the transferred coordinates, calculate the coordinates of inner grid points for the neighboring hexahedron.

Step 4. This calculation is repeated for all hexahedrons. Unless the convergence requirement is met for all grid points, return to Step 1 with new grid coordinates on the surfaces of hexahedrons.

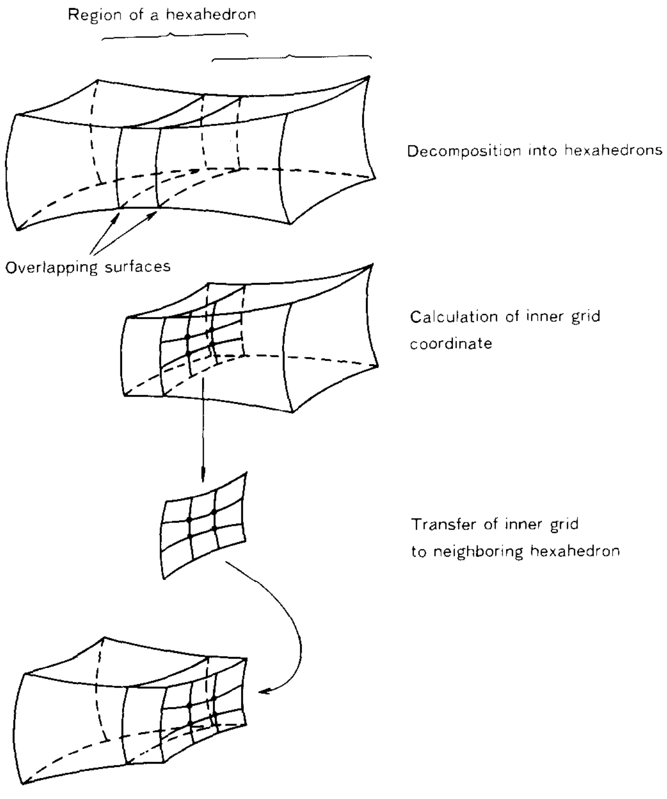

FIG. 1. Procedure for generating the grid using domain decomposition and overlapping.

The inner grid coordinates for each hexahedron are calculated by solving the elliptic system of quasilinear equations [4],

$$\alpha_{11}X_{\xi\xi} + \alpha_{22}X_{\eta\eta} + \alpha_{33}X_{\zeta\zeta} + 2\alpha_{12}X_{\xi\eta} + 2\alpha_{13}X_{\xi\zeta} + 2\alpha_{23}X_{\eta\zeta} + J^2(PX_\xi + QX_\eta + RX_\zeta) = 0, \tag{1a}$$

$$\alpha_{11}Y_{\xi\xi} + \alpha_{22}Y_{\eta\eta} + \alpha_{33}Y_{\zeta\zeta} + 2\alpha_{12}Y_{\xi\eta} + 2\alpha_{13}Y_{\xi\zeta} + 2\alpha_{23}Y_{\eta\zeta} + J^2(PY_\xi + QY_\eta + RY_\zeta) = 0, \tag{1b}$$

$$\alpha_{11}Z_{\xi\xi} + \alpha_{22}Z_{\eta\eta} + \alpha_{33}Z_{\zeta\zeta} + 2\alpha_{12}Z_{\xi\eta} + 2\alpha_{13}Z_{\xi\zeta} + 2\alpha_{23}Z_{\eta\zeta} + J^2(PZ_\xi + QZ_\eta + RZ_\zeta) = 0, \tag{1c}$$

where $[X, Y, Z]$ and $[\xi, \eta, \zeta]$ are coordinates in the physical and transformed spaces, respectively, and P, Q, and R are functions for controlling grid spacing, and

$$\alpha_{jk} = \sum_{m=1}^{3} \beta_{mj}\beta_{mk}, \tag{2}$$

where β_{mj} is the cofactor of the (m, j) element in the matrix,

$$M = \begin{pmatrix} X_\xi & X_\eta & X_\zeta \\ Y_\xi & Y_\eta & Y_\zeta \\ Z_\xi & Z_\eta & Z_\zeta \end{pmatrix}. \tag{3}$$

And J denotes the Jacobian determinant of the inverse transformation,

$$J = \frac{\partial(X, Y, Z)}{\partial(\xi, \eta, \zeta)} = \det |M|. \tag{4}$$

The boundary conditions are specified on the six plane surfaces of a transformed rectangular body,

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} f_i(\xi_i, \eta, \zeta) \\ g_i(\xi_i, \eta, \zeta) \\ h_i(\xi_i, \eta, \zeta) \end{pmatrix} \qquad [\xi_i, \eta, \zeta] \in \Gamma_i \ (i = 1, 2), \tag{5a}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} f_i(\xi, \eta_i, \zeta) \\ g_i(\xi, \eta_i, \zeta) \\ h_i(\xi, \eta_i, \zeta) \end{pmatrix} \qquad [\xi, \eta_i, \zeta] \partial \Gamma_i \ (i = 3, 4), \tag{5b}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} f_i(\xi, \eta, \zeta_i) \\ g_i(\xi, \eta, \zeta_i) \\ h_i(\xi, \eta, \zeta_i) \end{pmatrix} \qquad [\xi, \eta, \zeta_i] \in \Gamma_i \ (i = 5, 6), \tag{5c}$$

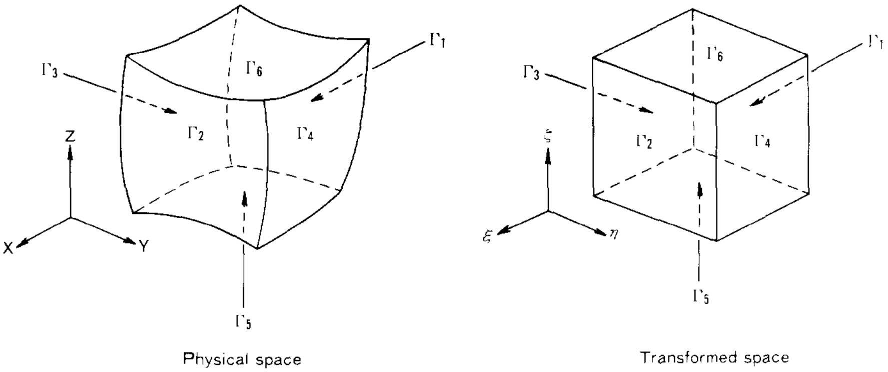

FIG. 2. Boundary surfaces of a hexahedron.

where ξ_i, η_i, and ζ_i are specified constants, and Γ_i $(i = 1, 2, \ldots, 6)$ are the six boundary surfaces as shown in Fig. 2. These boundary conditions are specified in advance on outer surfaces of the original whole domain or determined by the solution of the generating system Eq. (1) for the neighboring hexahedrons.

The functions P, Q, and R in Eq. (1) may be chosen to make the grid concentrate as desired. The following forms of these functions are obtained by modifying the forms incorporated in the TOMCAT program of Thompson *et al.* [5],

TABLE I

Coefficients Used in Control Functions P, Q, and R

Attracted Planes	Planes, Line, or Point Attracting Neighboring Planes
$a_{l_1} \neq 0$ for $\xi =$ constant plane	$a_{l_1} \neq 0, a_{l_2} = 0, a_{l_3} = 0$ for $\xi = \xi_l$ plane
	$c_{l_1} \neq 0, c_{l_2} = 0, c_{l_3} = 0$
$a_{l_2} \neq 0$ for $\eta =$ constant plane	$a_{l_1} = 0, a_{l_2} \neq 0, a_{l_3} = 0$ for $\eta = \eta_l$ plane
	$c_{l_1} = 0, c_{l_2} \neq 0, c_{l_3} = 0$
$a_{l_3} \neq 0$ for $\zeta =$ constant plane	$a_{l_1} = 0, a_{l_2} = 0, a_{l_3} \neq 0$ for $\zeta = \zeta_l$ plane
	$c_{l_1} = 0, c_{l_2} = 0, c_{l_3} \neq 0$
	$a_{l_1} = 0, a_{l_2} \neq 0, a_{l_3} \neq 0$ for $(\eta, \zeta) = (\eta_l, \zeta_l)$ line
	$c_{l_1} = 0, c_{l_2} \neq 0, c_{l_3} \neq 0$
	$a_{l_1} = 0, a_{l_2} = 0, a_{l_3} \neq 0$ for $(\xi, \zeta) = (\xi_l, \zeta_l)$ line
	$c_{l_1} = 0, c_{l_2} = 0, c_{l_3} \neq 0$
	$a_{l_1} \neq 0, a_{l_2} \neq 0, a_{l_3} = 0$ for $(\xi, \eta) = (\xi_l, \eta_l)$ line
	$c_{l_1} \neq 0, c_{l_2} \neq 0, c_{l_3} = 0$
	$a_{l_1} \neq 0, a_{l_2} \neq 0, a_{l_3} \neq 0$ for $(\xi, \eta, \zeta) = (\xi_l, \eta_l, \zeta_l)$ point
	$c_{l_1} \neq 0, c_{l_2} \neq 0, c_{l_3} \neq 0$

$$P(\xi, \eta, \zeta) = -\sum_{l=1}^{n} a_{l_1} \cdot \operatorname{sgn}(\xi - \xi_l) \cdot \exp(-b_{l_1} \cdot T_l), \tag{6a}$$

$$Q(\xi, \eta, \zeta) = -\sum_{l=1}^{n} a_{l_2} \cdot \operatorname{sgn}(\eta - \eta_l) \cdot \exp(-b_{l_2} \cdot T_l), \tag{6b}$$

$$R(\xi, \eta, \zeta) = -\sum_{l=1}^{n} a_{l_3} \cdot \operatorname{sgn}(\zeta - \zeta_l) \cdot \exp(-b_{l_3} \cdot T_l), \tag{6c}$$

where

$$T_l = \sqrt{c_{l_1}(\xi - \xi_l)^2 + c_{l_2}(\eta - \eta_l)^2 + c_{l_3}(\zeta - \zeta_l)^2}. \tag{7}$$

The sets of the coefficients a_{lm} and c_{lm}, as shown in Table I, attract the ξ, η, and/or $\zeta =$ constant planes to the specified plane, coordinate line, or grid point. The range of the attraction effect is determined by the decay factor b_{lm}. The effect of the controlling functions are shown in the example of Section 3B.

3. Computer Program and Applications

A. *Description of* GRID-3D *Program*

A new computer program GRID-3D has been developed on the basis of the method described in Section 2. Figure 3 illustrates its calculation flow. Main input data provided to the program are geometrical data, the total number of grid points in ξ, η, and ζ directions of each hexahedron, and values of parameters needed for the overlapping between hexahedrons.

The physical grid coordinates must also be given on the outer surfaces of the whole domain under consideration. GRID-3D has two functions which simplify this procedure. One is the use of a geometrical data package, which calculates the physical coordinates of the outer surfaces employing user-supplied geometrical data. The following geometries can be treated by the package: sphere, hemisphere, ellipsoid, cone, rectangular, cylinder, shell, and torus.

The second function in GRID-3D is the synthesis of these geometries by rotation, translation, and scaling. Rotations of a geometry around the X, Y, and Z axes, for instance, is accomplished on the basis of

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = L \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} \qquad (i = 1, 2, 3), \tag{8}$$

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}, \tag{9a}$$

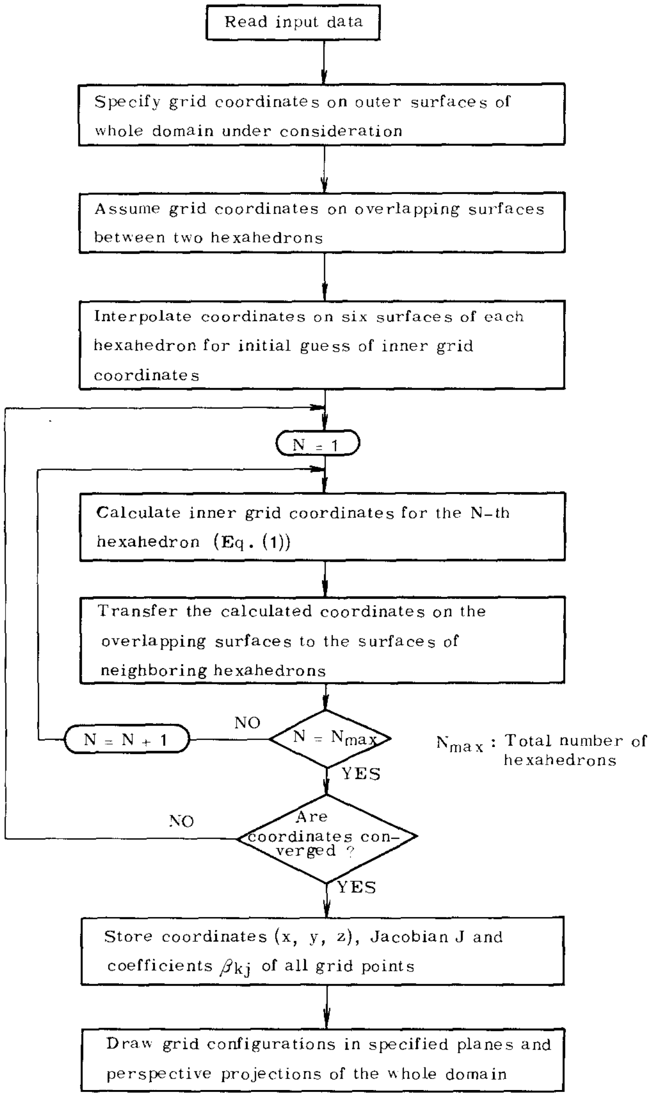

FIG. 3. Calculational steps in GRID-3D.

$$L_2 = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}, \tag{9b}$$

$$L_3 = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{9c}$$

where $[X_0, Y_0, Z_0]$ and $[X, Y, Z]$ are the grid coordinates of the original and the rotated geometries, respectively, and L_1, L_2, and L_3 are rotation matrices around the X, Y, and Z axes by θ radians, respectively.

With the specified grid coordinates on the outer surfaces of the whole domain and the assumed grid coordinates on overlapping surfaces between hexahedrons, the program calculates inner grid coordinates for each hexahedron by solving Eq. (1) using the SOR technique. The calculated coordinates $[X, Y, Z]$, Jacobian J, and coefficients β_{mj} of all grid points are stored in forms to be used in the solution of a partial differential equation arising from a physical problem.

The program can draw the grid configurations in specified planes and the perspective projections of the whole domain. This graphic display capability makes it possible to choose the best grid arrangement for a particular problem.

B. *Numerical Results and Discussion*

When a three-dimensional physical domain is treated by several hexahedrons, the curved smooth surface of the domain must often be assigned to more than one surface of a hexahedron. As a typical example of this point, Fig. 4 shows the grid configuration for a sphere, where the sphere is treated by single hexahedron and the surface of the sphere is divided into six curved surfaces of the same shape. In this and succeeding figures, the perspective projection of a whole domain and the grid configuration in a specified ξ, η, or $\zeta = $ constant plane are shown. In the program, variables ξ, η, and ζ are assumed to take integer values as

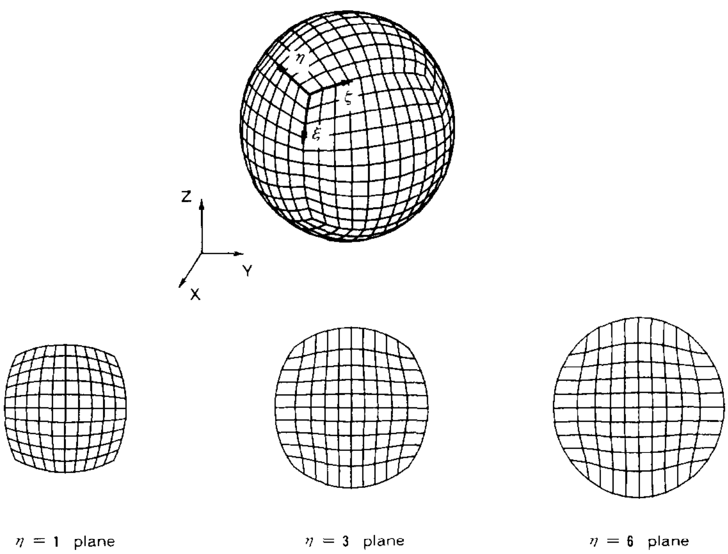

FIG. 4. Perspective projection of coordinate system for a sphere in a physical space.

$$\xi = I \qquad (I = 1, 2, ..., I_{max}),$$
$$\eta = J \qquad (J = 1, 2, ..., J_{max}),$$
$$\zeta = K \qquad (K = 1, 2, ..., K_{max}),$$

so that $\eta = 1$ and $\eta = 11$ planes in Fig. 4 correspond to Γ_3 and Γ_4 surfaces of the hexahedron in Fig. 2, respectively. In this example, the corner of the hexahedron corresponds to a point on the smooth surface of the sphere, which results in zero Jacobian at the corner. Therefore, some special treatment is required for solving a physical partial differential equation at the corner of the hexahedron. Figure 4 also demonstrates that a sphere can be treated by a rectangular-type coordinate system, which is more useful than the spherical coordinate system when more grid points are required in the outer region of a sphere without concentrating grid points in the center region.

The following two examples demonstrate how the same type of coordinate system can be generated for different geometries. Figure 5 shows the grid configuration for a 90 degree-bent cylinder. The circumference of the circle is divided into four equal lengths; these arcs correspond to the constant ξ or η lines, while the axial direction corresponds to the ζ-direction. Note that $\zeta = 1$ and $\zeta = K_{max}$ planes, which correspond to Γ_5 and Γ_6 planes, respectively, in Fig. 2, have the same grid configuration as the inner $\zeta = 2$ and $\zeta = K_{max} - 1$ planes have. GRID-3D uses a special treatment for $\zeta = 1$ and $\zeta = K_{max}$ planes, that is, although grid coordinates on

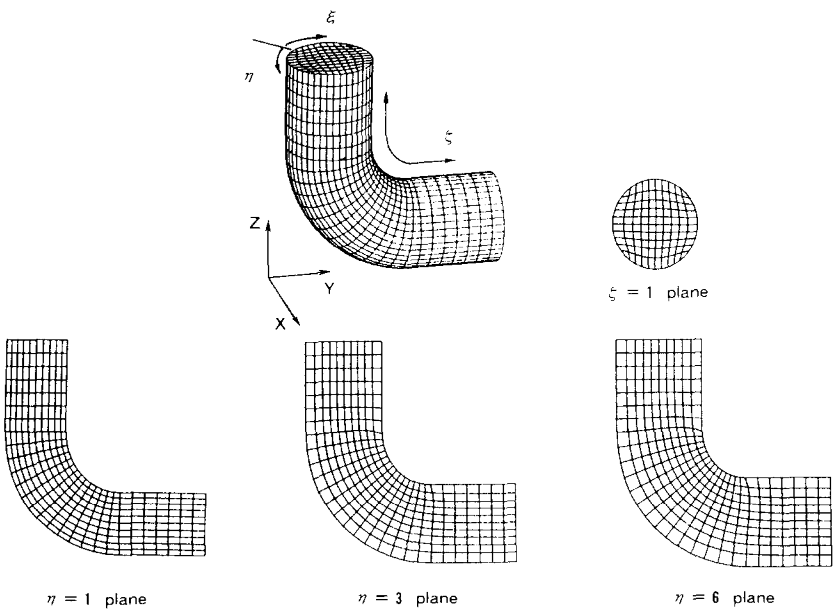

FIG. 5. Perspective projection of cordinate system for a 90°-bent cylinder in a physical space.

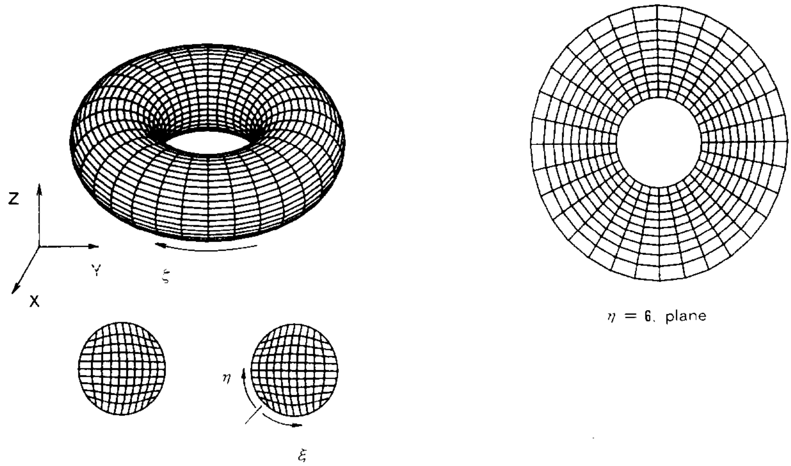

FIG. 6. Perspective projection of coordinate system for a torus in a physical space.

these planes are specified in advance, they are changed in an iteration procedure by transferring the calculated coordinates on the adjacent inner plane to the outer $\zeta = 1$ or $\zeta = K_{\max}$ plane. This smoothness of the grid coordinates along the axial direction is more convenient for some engineering problems, such as a fluid flow problem in an elbow pipe.

The curvilinear coordinate system in Fig. 5 can be used in a straightforward manner for a torus, as indicated in Fig. 6. In this example, the $\zeta = 1$ and $\zeta = \mathrm{K}_{\max}$ planes are joined to the $\zeta = \mathrm{K}_{\max} - 1$ and $\zeta = 2$ planes, respectively, so that the torus

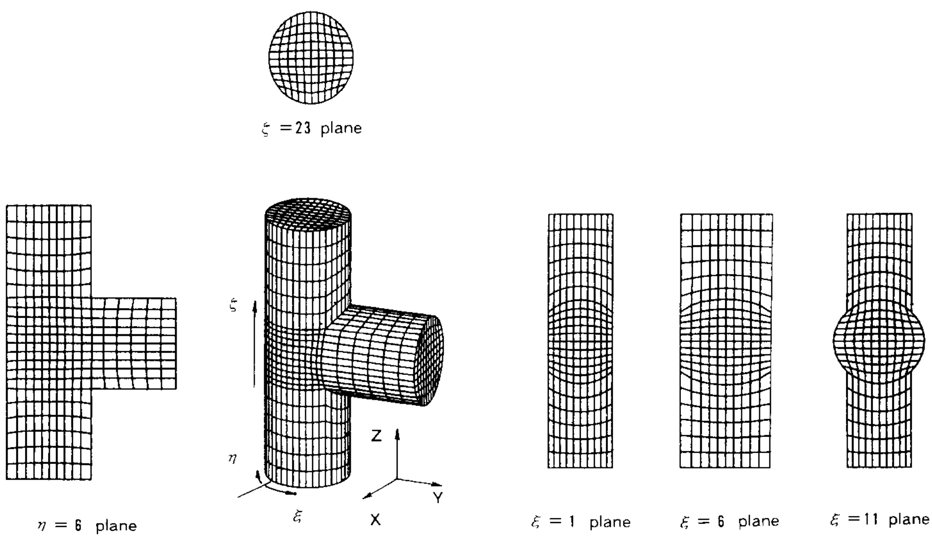

FIG. 7. Perspective projection of coordinate system for a T-shaped cylinder in a physical space.

is treated by a single hexahedron with two overlapping planes. Note that there is smooth coverage of grid points in the entire domain.

The next example shows grid configuration generated by using two hexahedrons overlapping each other. The geometry shown in Fig. 7 consists of two cylinders joined to each other at right angles, each of which is treated by a hexahedron. Although the cylindrical coordinate system is best suited for a cylinder, it is extremely difficult to match the coordinates in the junction region between two cylinders. The rectangular-type coordinate system, as shown in Fig. 7, is much more flexible in matching between the other rectangular-type coordinate system. The coordinate system in the figure shows an accurate representation of the boundary shape and smooth coverage of grid points even in the overlapping region between the two hexahedrons. It should be noted that almost an orthogonal grid configuration can be obtained in $\eta = 6$ symmetry plane.

The last example demonstrates the grid generation capability of the GRID-3D program. Figure 8 shows the grid concentration in a particular region by using the controlling functions P, Q, and R given in Eq. (6). In this example, ξ and $\eta =$ constant planes are attracted to the four outer surfaces ($\xi = 1$, $\xi = 11$, $\eta = 1$, and $\eta = 11$ planes) of a 90 degree-bent cylinder. Increasing the coefficients a_{lm} in Eq. (7), ξ and $\eta =$ constant planes move closer to the boundary. This kind of grid concentration can improve the accuracy of the numerical solution for fluid flow problems in an elbow pipe.

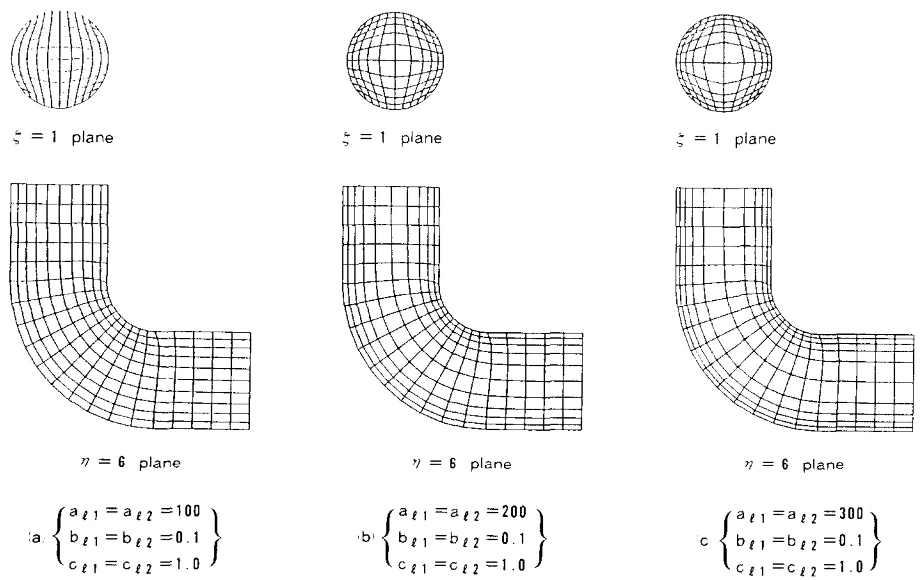

FIG. 8. Grid concentration along outer boundary of the 90°-bent cylinder.

4. Conclusion

A new method has been presented for numerically generating boundary-fitted coordinate systems for arbitrarily shaped three-dimensional regions. The method uses the techniques of decomposing the three-dimensional region into several hexahedrons and of overlapping them with two grid surfaces. An important feature inherent in the method is that the coordinate lines can remain both continuous and smooth throughout the whole domain despite the domain division.

A computer program GRID-3D has been developed on the basis of the method, and was applied to a variety of geometries, including a sphere, a 90 degree-bent cylinder, a torus, and a T-shaped cylinder. These applications confirmed that GRID-3D is a convenient and efficient tool in the generation of boundary-fitted coordinate systems even for a considerably complicated geometry consisting of many different components. Therefore, as a pre-processing aid, GRID-3D will provide significant progress toward improved productivity in the field of computer-aided design (CAD).

Acknowledgments

The authors would like to acknowledge Drs. K. Taniguchi, S. Yamada, S. Kobayashi, and R. Takeda of Energy Research Laboratory, Hitachi, Ltd., for their sustained encouragement. Acknowledgment is also due to Drs. W. T. Sha and B. Chen of Argonne National Laboratory for their valuable discussions.

References

1. J. F. Thompson and Z. U. A. Warsi, *J. Comput. Phys.* **47** (1982), 1.
2. J. F. Thompson, F. C. Thames, and C. W. Mastin, *J. Comput. Phys.* **15** (1974), 299.
3. P. E. Rubbert and K. D. Lee, "Numerical Grid Generation," p. 235, Elsevier, Amsterdam/New York, 1982.
4. C. W. Mastin and J. F. Thompson, *Numer. Math.*, **29** (1978), 387.
5. J. F. Thompson, F. C. Thames, and C. W. Mastin, *J. Comput. Phys.* **24** (1977), 274.