Chapter 9

Grid Generation – Structured Grids

9.1 Introductory Remarks

In order to numerically solve the governing partial differential equations (PDEs) of fluid mechanics, approximations to the partial differentials are introduced. These approximations convert the partial derivatives to finite difference expressions, which are used to rewrite the PDEs as algebraic equations. The approximate algebraic equations, referred to as finite difference equations (FDEs), are subsequently solved at discrete points within the domain of interest. Therefore, a set of grid points within the domain, as well as the boundaries of the domain, must be specified.

Typically, the computational domain is selected to be rectangular in shape where the interior grid points are distributed along grid lines. Therefore, the grid points can be identified easily with reference to the appropriate grid lines. This type grid is known as the *structured grid* and is the focus of this chapter. A second category of grid system may be constructed where the grid points cannot be associated with orderly defined grid lines. This type grid system is known as the *unstructured grid* and will be introduced in Chapter 15. In the remainder of this chapter, *structured grid generation* is implied wherever we refer to grid generation.

The generation of a grid, with uniform spacing, is a simple exercise within a rectangular physical domain. Grid points may be specified as coincident with the boundaries of the physical domain, thus making specification of boundary conditions considerably less complex. In the previous chapters, this restriction on the physical domain was enforced, i.e., all applications were limited to rectangular-type domains.

Unfortunately, the majority of the physical domains of interest are nonrectangular. Therefore, imposing a rectangular computational domain on such a physical domain will require some sort of interpolation for the implementation of the boundary conditions. Since the boundary conditions have a dominant influence on the solution of the equation, such an interpolation causes inaccuracies at the place of greatest sensitivity. In addition, unequal grid spacing near the boundaries creates further complications with the FDEs since approximations with nonequal stepsizes must be used. This form of the FDE changes from node to node, creating cumbersome programming details.

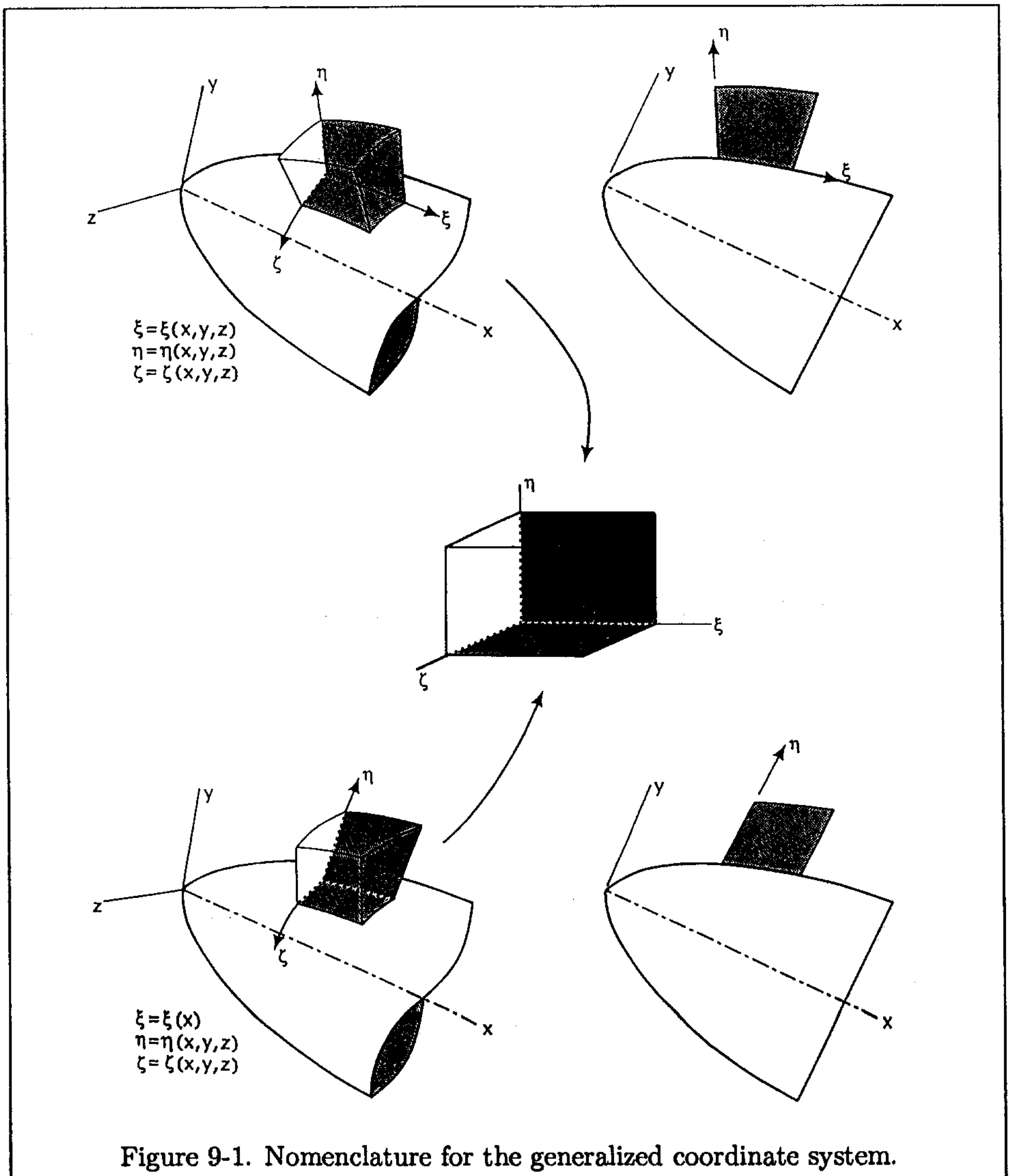


Figure 9-1. Nomenclature for the generalized coordinate system.

To overcome these difficulties, a transformation from physical space to computational space is introduced. This transformation is accomplished by specifying a generalized coordinate system which will map the nonrectangular grid system in the physical space to a rectangular uniform grid spacing in the computational space. The generalized coordinate system may be expressed in many ways. Two examples are illustrated in Figure 9-1. The first example shows the so-called body-fitted coordinate system where two coordinate lines, ξ and ζ, are aligned on the surface along the streamwise and circumferential directions, and where the third coordinate, η, is normal to the surface. In the second example, the ξ coordinate is aligned along the body axis, ζ is in the circumferential direction and η is normal to the body axis.

For illustrative purposes, two-dimensional (2-D) problems will be considered in detail, with a description of the three-dimensional (3-D) problems to follow. A 2-D domain is illustrated in Figure 9-2.

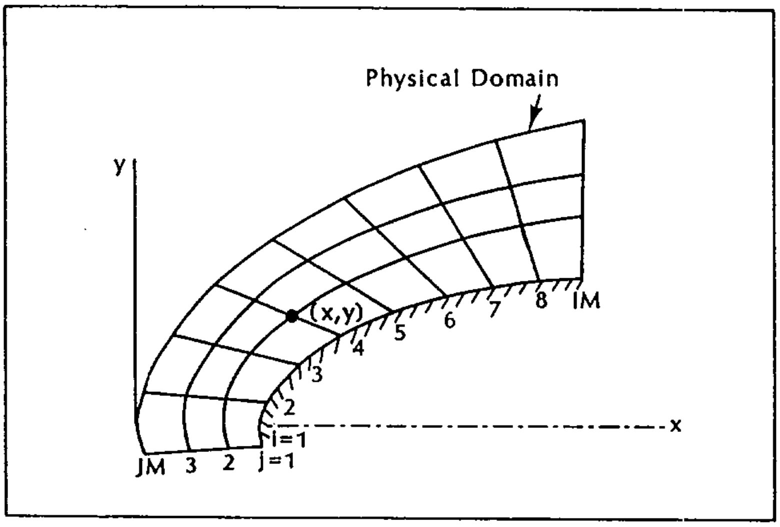


Figure 9-2. A typical 2-D domain for the axisymmetric blunt body problem.

In order to eliminate the difficulty associated with the nonequal stepsizes used in the FDEs, the physical domain is transformed into a rectangular, constant step-size, computational domain. A typical computational domain is shown in Figure 9-3. Note that the computational domain is obtained by deformation of the physical domain, i.e., by twisting and stretching, etc.

The central issue at this point is identifying the location of the grid points in the physical domain. That is, what are the x and y coordinates of a grid point in physical space (Figure 9-2) that correspond to a grid point (i, j) specified in the computational domain (Figure 9-3)? In determining the grid points, a few constraints must be imposed. First, the mapping must be one-to-one; i.e., grid lines of the same family cannot cross each other. Second, from a numerical point of view, a smooth grid point distribution with minimum grid line skewness, orthogonality or near orthogonality, and a concentration of grid points in regions where high flow gradients occur are required. Obviously, all of the desired features may not be met by the use of a particular grid generation technique. Grid generation techniques which emphasize any one or a combination of these features will be presented in the following sections.

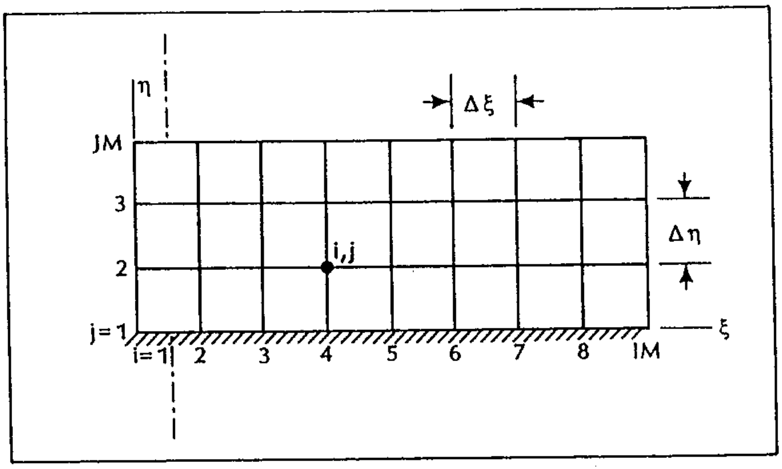


Figure 9-3. Computational domain with constant stepsizes $\Delta\xi$ and $\Delta\eta$.

In general, grid generation techniques may be classified as (1) algebraic methods, (2) partial differential methods, or (3) conformal mappings based on complex variables. In addition, the grid system may be categorized as fixed or adaptive. Obviously, a fixed grid system is generated prior to the solution of the governing equations of fluid motion and remains fixed independent of the solution. On the other hand, an adaptive grid system evolves as a result of the solution of the equations of fluid motion. For example, grid points may move toward regions of high gradients such as in the neighborhood of a shock wave.

Conformal mappings based on complex variables are limited to 2-D problems and require a reasonable knowledge of complex variables. In addition, the determination of the mapping function is sometimes a difficult task. Therefore, this method will not be discussed here. In the following sections, grid generation techniques based on algebraic and PDE methods are presented.

9.2 Transformation of the Governing Partial Differential Equations

The equations of fluid motion include the continuity, momentum and energy equations. For a single phase continuum flow, the transformation of this system of equations will be presented in Chapter 11. In this section, a simple 2-D problem is proposed to familiarize the reader with the processes involved in the transformation of a PDE and the complexity of the resultant equation. It should be mentioned that the form and type of the transformed equation remains the same as the original PDE; i.e., if the original equation is parabolic, then the transformed equation is also parabolic. A mathematical proof is given in Reference [9-1]. Now, define the following relations between the physical and computational spaces:

$$\xi = \xi(x,y) \tag{9-1}$$

$$\eta = \eta(x,y) \tag{9-2}$$

The chain rule for partial differentiation yields the following expression:

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} \tag{9-3}$$

The partial derivatives will be denoted using the subscripts notation, i.e., $\frac{\partial \xi}{\partial x} = \xi_x$. Hence,

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \tag{9-4}$$

and similarly,

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} \tag{9-5}$$

Now consider a model PDE, such as

$$\frac{\partial u}{\partial x} + a\frac{\partial u}{\partial y} = 0 \tag{9-6}$$

This equation may be transformed from physical space to computational space using Equations (9-4) and (9-5). As a result,

$$\xi_x \frac{\partial u}{\partial \xi} + \eta_x \frac{\partial u}{\partial \eta} + a\left(\xi_y \frac{\partial u}{\partial \xi} + \eta_y \frac{\partial u}{\partial \eta}\right) = 0$$

which may be rearranged as

$$(\xi_x + a\xi_y)\frac{\partial u}{\partial \xi} + (\eta_x + a\eta_y)\frac{\partial u}{\partial \eta} = 0 \tag{9-7}$$

This equation is the one which will be solved in the computational domain. Also note that the transformation derivatives ξ_x, ξ_y, η_x, and η_y must be determined from the functional relations (9-1) and (9-2). The determination of the transformation derivatives will be addressed briefly in the next section and in more detail for a 3-D case in Chapter 11. Comparing the original PDE (9-6) and the transformed equation given in (9-7), it is obvious that the transformed equation is more complicated than the original equation. Generally that is always the case. Thus, a trade-off is introduced whereby advantages gained by using the generalized coordinates are somehow counterbalanced by the resultant complexity of the PDE. However, the advantages by far outweigh the complexity of the transformed PDE.

9.3 Metrics and the Jacobian of Transformation

Recall that in Equations (9-4) and (9-5), terms such as ξ_x, ξ_y, η_x, and η_y appear. These transformation derivatives are defined as the metrics of transformation or simply as the metrics. The interpretation of the metrics is obvious considering the following approximation:

$$\xi_x = \frac{\partial \xi}{\partial x} \cong \frac{\Delta \xi}{\Delta x}$$

This expression indicates that the metrics represent the ratio of arc lengths in the computational space to that of the physical space. The computation of the metrics is considered next.

From Equations (9-1) and (9-2) the following differential expressions are obtained

$$d\xi = \xi_x dx + \xi_y dy \tag{9-8}$$

$$d\eta = \eta_x dx + \eta_y dy \tag{9-9}$$

which are written in a compact form as

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \tag{9-10}$$

Reversing the role of independent variables, i.e.,

$$x = x(\xi, \eta)$$

$$y = y(\xi, \eta)$$

The following may be written

$$dx = x_\xi d\xi + x_\eta d\eta \tag{9-11}$$

$$dy = y_\xi d\xi + y_\eta d\eta \tag{9-12}$$

In a compact form they are written as

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{9-13}$$

Comparing Equations (9-10) and (9-13), it can be concluded that

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}^{-1}$$

from which

$$\xi_x = Jy_\eta \tag{9-14}$$

$$\xi_y = -Jx_\eta \tag{9-15}$$

$$\eta_x = -Jy_\xi \tag{9-16}$$

$$\eta_y = Jx_\xi \tag{9-17}$$

where

$$J = \frac{1}{x_\xi y_\eta - y_\xi x_\eta} \tag{9-18}$$

and is defined as the Jacobian of transformation.

The Jacobian, J, is interpreted as the ratio of the areas (volumes in 3-D) in the computational space to that of the physical space.

Note that the actual values of the metrics or the Jacobian could be negative. Obviously, the values depend on the specification of physical and computational grid systems. The computed values of the metrics for various grid systems will be presented shortly. For grid generation methods where analytical expressions for the metrics can be written, they may be analytically evaluated or determined numerically by the use of finite difference expressions. This point is illustrated in the example problem given in Section 9.5.

9.4 Grid Generation Techniques

Before proceeding with the investigation of various grid generation techniques, the objectives will be summarized. A grid system with the following features is desired:

(1) A mapping which guarantees one-to-one correspondence ensuring grid lines of the same family do not cross each other;

(2) Smoothness of the grid point distribution;

(3) Orthogonality or near orthogonality of the grid lines;

(4) Options for grid point clustering.

As mentioned previously, some of the features enumerated, as in (2)–(4) above, may not be achievable with a particular grid generation technique. In the next few sections, various grid generation techniques are introduced.

9.5 Algebraic Grid Generation Techniques

The simplest grid generation technique is the algebraic method. The major advantage of this scheme is the speed with which a grid can be generated. An algebraic equation is used to relate the grid points in the computational domain to those of the physical domain. This objective is met by using an interpolation scheme between the specified boundary grid points to generate the interior grid points. Clearly, many algebraic equations (or interpolation schemes, if preferred) can be introduced for this purpose. To illustrate the procedure, consider the simple physical domain depicted in Figure 9-4.

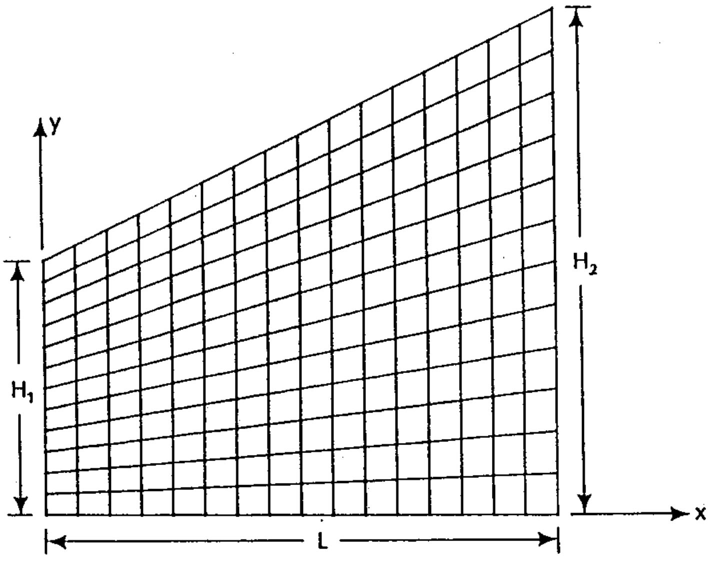


Figure 9-4. The physical space which must be transformed to a uniform rectangular computational space.

Introducing the following algebraic relations will transform this nonrectangular physical domain into a rectangular domain:

$$\xi = x \tag{9-19}$$

$$\eta = \frac{y}{y_t} \tag{9-20}$$

In (9-20), y_t represents the upper boundary which is expressed as

$$y_t = H_1 + \frac{H_2 - H_1}{L} x$$

Thus, it may be written that

$$\xi = x \tag{9-21}$$

$$\eta = \frac{y}{H_1 + \frac{H_2 - H_1}{L} x} \tag{9-22}$$

which can be rearranged as

$$x = \xi \tag{9-23}$$

and

$$y = \left(H_1 + \frac{H_2 - H_1}{L} \xi\right) \eta \tag{9-24}$$

The grid system is generated as follows. The geometry in the physical space is defined. For this particular problem, it is accomplished by specifying values of L, H_1, and H_2. Next, the desired number of grid points defined by IM (the maximum number of grid points in ξ) and JM (the maximum number of grid points in η) is specified. The equal grid spacing in the computational domain is produced as follows:

$$\Delta\xi = \frac{L}{IM - 1} \tag{9-25}$$

$$\Delta\eta = \frac{1.0}{JM - 1} \tag{9-26}$$

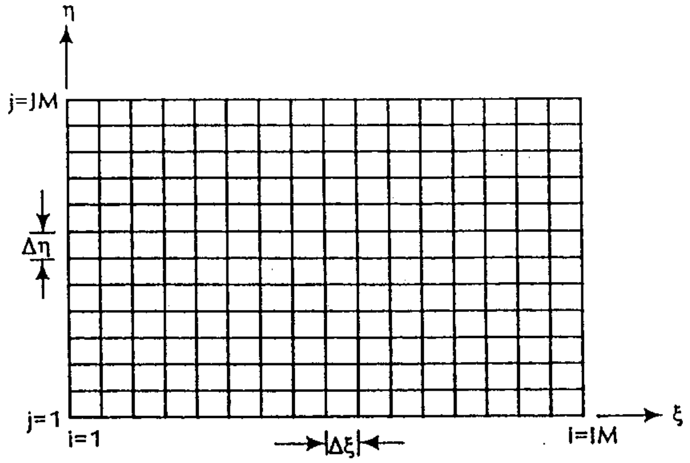


Figure 9-5. The rectangular computational domain with uniform grid spacing.

Note that in Equation (9-20), η has been normalized, i.e., its value varies from zero to one. With the equal step-sizes provided by Equations (9-25) and (9-26), the uniform computational domain is constructed, which is shown in Figure 9-5 for a 17 × 13 grid system. Therefore, the values of ξ and η are known at each grid point within this domain. Now Equations (9-23) and (9-24) are used to identify the corresponding grid points in the physical space. For illustrative purposes, the grid system generated for a physical domain defined by $L = 4$, $H_1 = 2$, and $H_2 = 4$ is shown in Figure 9-4.

As discussed previously, the metrics and the Jacobian of the transformation must be evaluated before any transformed PDEs can be solved. In many instances, when an algebraic model is used the metrics may be calculated analytically. This aspect is obviously an advantage of the algebraic methods, since numerical computation of the metrics will require additional computation time and may introduce some errors into the system of equations of motion that are to be solved. To illustrate this point, the metrics in the proposed example are computed both analytically and numerically. From Equation (9-21), metrics ξ_x and ξ_y can be determined analytically, resulting in

$$\xi_x = 1 \qquad (9\text{-}27)$$

$$\xi_y = 0 \tag{9-28}$$

Similarly, the following is obtained from Equation (9-22)

$$\eta_x = -\frac{(H_2 - H_1)y/L}{[H_1 + (H_2 - H_1)x/L]^2} \quad \text{or}$$

$$\eta_x = -\frac{H_2 - H_1}{L} \frac{\eta}{[H_1 + (H_2 - H_1)\xi/L]} \tag{9-29}$$

and

$$\eta_y = \frac{1}{H_1 + (H_2 - H_1)x/L} \quad \text{or}$$

$$\eta_y = \frac{1}{H_1 + (H_2 - H_1)\xi/L} \tag{9-30}$$

and the Jacobian of transformation is

$$J = \frac{1}{x_\xi y_\eta - y_\xi x_\eta}$$

To compute the metrics numerically, Equations (9-14) through (9-18) are used. Thus, the terms x_ξ, y_ξ, x_η, and y_η are computed initially, from which the Jacobian may be evaluated. These expressions are computed numerically using finite difference approximations. For example, a second-order central difference approximation may be used to compute the transformation derivative x_η for the interior grid points, i.e.,

$$x_\eta = \frac{x_{i,j+1} - x_{i,j-1}}{2\Delta\eta}$$

The transformation derivatives at the boundaries are evaluated with forward or backward second-order approximations. For example, x_η at the $j = 1$ boundary is computed using the forward difference approximation

$$x_\eta = \frac{-3x_{i,1} + 4x_{i,2} - x_{i,3}}{2\Delta\eta}$$

A comparison of the metric η_x evaluated numerically and analytically by Equation (9-29) and the error introduced in the numerical computations is shown in Table (9-1).

j	Analytic	Numerical	ERROR
4	$-.52632E-01$	$-.52632E-01$	$.44409E-15$
8	$-.12281E+00$	$-.12281E+00$	$-.13323E-14$
12	$-.19298E+00$	$-.19298E+00$	$-.26645E-14$

Table 9-1. Comparison of the metric η_x at $i = 4$ evaluated analytically and numerically.

Notice that for this simple grid the errors in the numerical computation of the metrics are extremely small (practically zero!). The distributions of metrics are shown in Figures 9-6a through 9-6d. Each tic mark on the vertical axis represents a numerical change of 0.25 in Figures 9-6a and 9-6b, −0.1 in Figure 9-6c, and 0.125 in Figure 9-6d. These figures illustrate the smooth variations of the metrics. Erratic metric distributions, especially those with some sort of discontinuity, will certainly invite disaster! It is strongly recommended that the metric distributions be investigated prior to solving the governing PDEs of fluid motion.

The simple algebraic model just investigated does not include an option for clustering. Next, some algebraic expressions which employ clustering techniques are presented.

For flow problems where large gradients are concentrated in a specific region, additional resolution of the flow properties is essential. As an example, flow in the vicinity of a solid surface in a viscous fluid possesses large flow gradients. Accurate computation of flow gradients in this region will require many grid points within the domain. Rather than using a nearly uniform grid distribution in the physical domain, grid points may be clustered in the regions of high flow gradients, which reduces the total number of grid points and thus increases efficiency. Some examples of such algebraic expressions with clustering options are provided below. As a first example, consider the transformation given by

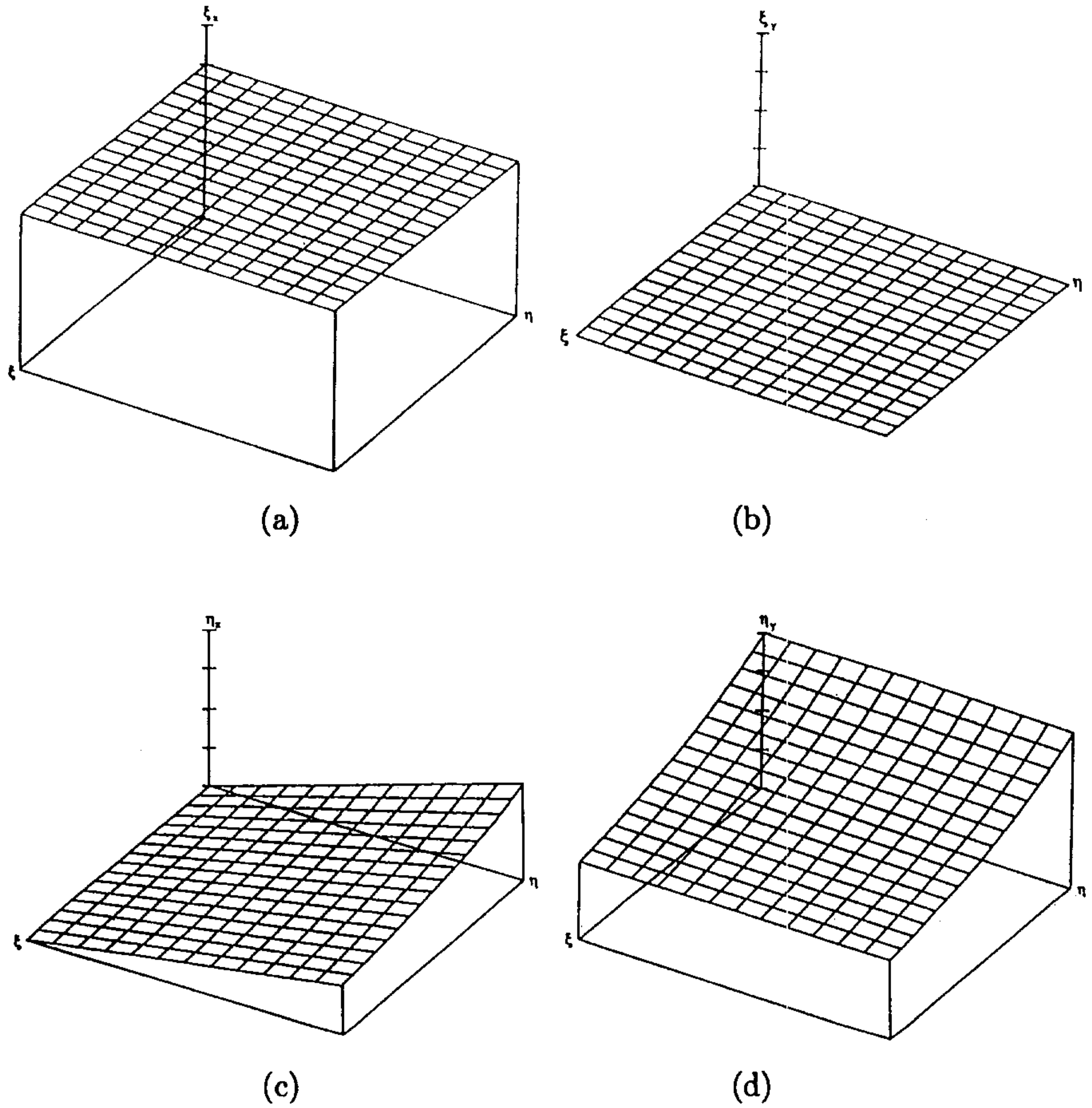
$$\xi = x \tag{9-31}$$

$$\eta = 1 - \frac{ln\{[\beta + 1 - (y/H)]/[\beta - 1 + (y/H)]\}}{ln[(\beta+1)/(\beta-1)]} \tag{9-32}$$

In this equation, β is the clustering parameter within the range of 1 to ∞. As the value of β approaches 1, more grid points are clustered near the surface, where $y = 0$. From Equations (9-31) and (9-32), the inverse of the transformation can be written as

$$x = \xi \tag{9-33}$$

$$y = H\,\frac{(\beta+1) - (\beta-1)\,\{[(\beta+1)/(\beta-1)]^{1-\eta}\}}{[(\beta+1)/(\beta-1)]^{1-\eta} + 1} \tag{9-34}$$



(a) (b)

(c) (d)

Figures 9-6a, b, c, d. Distribution of metrics for the domain shown in Figure 9-4.

The metrics of transformation may be determined analytically from the algebraic relations (9-31) and (9-32) and are given below:

$$\xi_x = 1 \tag{9-35}$$

$$\eta_x = 0 \tag{9-36}$$

$$\xi_y = 0 \tag{9-37}$$

$$\eta_y = \frac{2\beta}{H\,\{\beta^2 - [1-(y/H)]^2\}\,\{ln[(\beta+1)/(\beta-1)]\}} \tag{9-38}$$

The physical and computational domains, using the transformations (9-31) and (9-32), are shown in Figures 9-7a, 9-7b, and 9-7c. The grid system is generated for a 21 × 24 grid, and clustering parameter values of 1.05 and 1.2 are used, respectively. The distributions of metric η_y are shown in Figures 9-8a and 9-8b. Each tic mark on the vertical axis represents a change of 0.5 in Figures 9-8a and 9-8b. Obviously, this type of grid is suitable for boundary-layer type computations, where grid clustering near the surface is required.

For a physical domain enclosed by lower and upper solid surfaces, clustering at both surfaces must be considered. A flow field within a 2-D duct is such an example. The following algebraic equations may be employed for this purpose:

$$\xi = x \tag{9-39}$$

$$\eta = \alpha + (1-\alpha)\,\frac{ln\,\{\{\beta + [(2\alpha+1)y/H] - 2\alpha\}/\{\beta - [(2\alpha+1)y/H] + 2\alpha\}\}}{ln[(\beta+1)/(\beta-1)]} \tag{9-40}$$

where β is the clustering parameter, and α defines where the clustering takes place. When $\alpha = 0$, the clustering is at $y = H$; whereas, when $\alpha = 1/2$, clustering is distributed equally at $y = 0$ and $y = H$. The inverse transformation is given by

$$x = \xi \tag{9-41}$$

$$y = H\,\frac{(2\alpha+\beta)\,[(\beta+1)/(\beta-1)]^{(\eta-\alpha)/(1-\alpha)} + 2\alpha - \beta}{(2\alpha+1)\,\{1 + [(\beta+1)/(\beta-1)]^{(\eta-\alpha)/(1-\alpha)}\}} \tag{9-42}$$

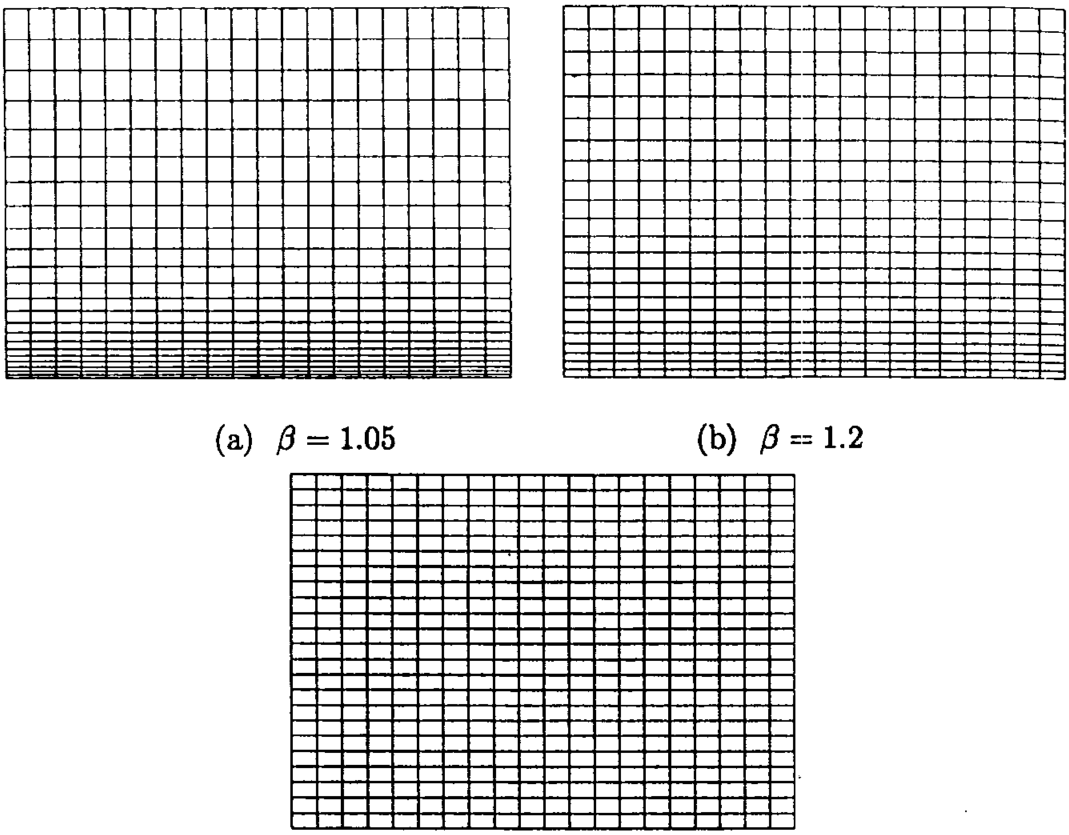
Analytical expressions for the metrics are determined from Equations (9-39) and (9-40) as

$$\xi_x = 1 \tag{9-43}$$

$$\xi_y = 0 \tag{9-44}$$

$$\eta_x = 0 \tag{9-45}$$

$$\eta_y = \frac{2\beta(2\alpha+1)(1-\alpha)}{H\,\{\beta^2 - [(2\alpha+1)y/H - 2\alpha]^2\}\,ln[(\beta+1)/(\beta-1)]} \tag{9-46}$$

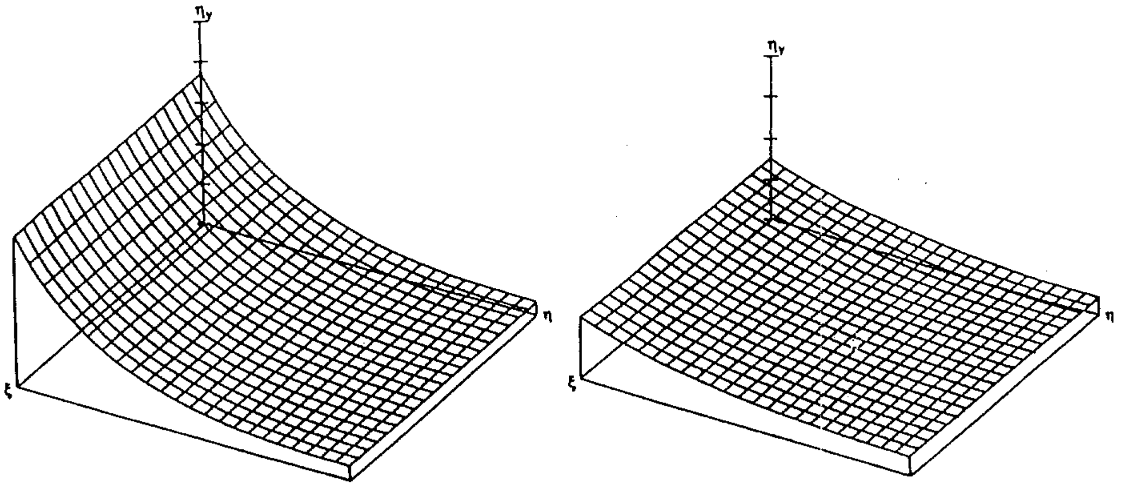


(a) $\beta = 1.05$

(b) $\beta = 1.2$

(c) Computational domain.

Figures 9-7a, b, c. The physical and computational domains generated by transformation functions (9-31) and (9-32).



Figures 9-8a, b. Metric distributions for the domains of Figures 9-7a and 9-7b.

Grid systems generated for a domain with 21 × 24 grid points are shown in Figures 9-9a and 9-9b for $\alpha = 1/2$ and clustering parameters of 1.05 and 1.2, respectively.

The corresponding η_y metric distributions are shown in Figures 9-10a and 9-10b. Each tic mark on the vertical axis represents a change of 0.5 in Figures 9-10a and 9-10b.

For problems where clustering in the interior of the domain is required, the following relations may be utilized

$$\xi = x \tag{9-47}$$

$$\eta = A + \frac{1}{\beta}\sinh^{-1}\left[\left(\frac{y}{D} - 1\right)\sinh(\beta A)\right] \tag{9-48}$$

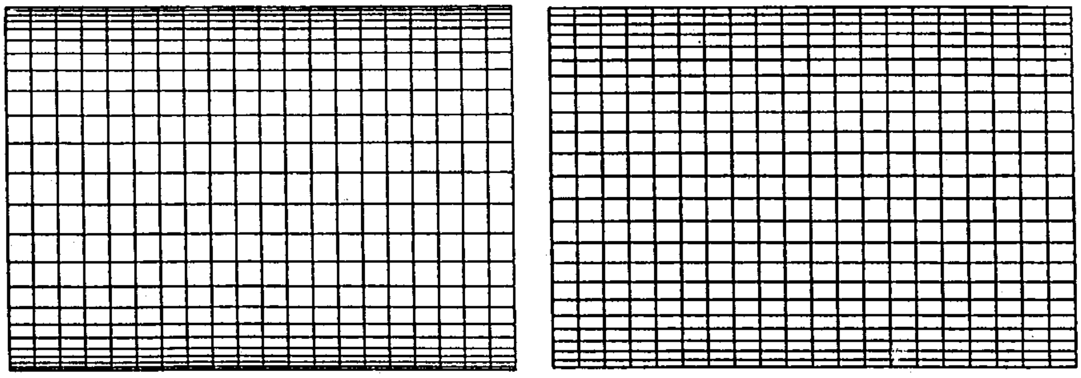
where

$$A = \frac{1}{2\beta}\, ln \left[\frac{1 + (e^{\beta} - 1)(D/H)}{1 + (e^{-\beta} - 1)(D/H)}\right]$$

In Equation (9-48), β is the clustering parameter in the range of $0 < \beta < \infty$, and D is the y coordinate where clustering is desired. The inverse transformation is given by:

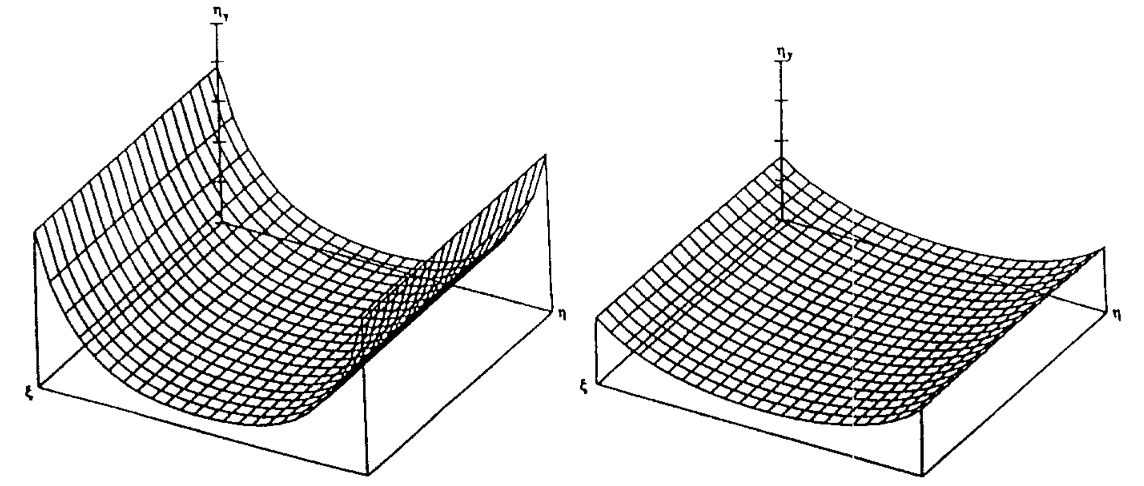
$$x = \xi \tag{9-49}$$

$$y = D\left\{1 + \frac{\sinh[\beta(\eta - A)]}{\sinh(\beta A)}\right\} \tag{9-50}$$



(a) $\beta = 1.05$ (b) $\beta = 1.2$

Figures 9-9a, b. Grid system generated by the transformation function (9-42).



Figures 9-10a, b. Metric distributions for the domains shown in Figure 9-9.

Note that for $\beta = 0$, no clustering is enforced, while a denser clustering of grid points near D is produced for larger values of β. Analytical expressions for the metrics are determined from Equations (9-47) and (9-48) and are given by:

$$\xi_x = 1 \tag{9-51}$$

$$\xi_y = 0 \tag{9-52}$$

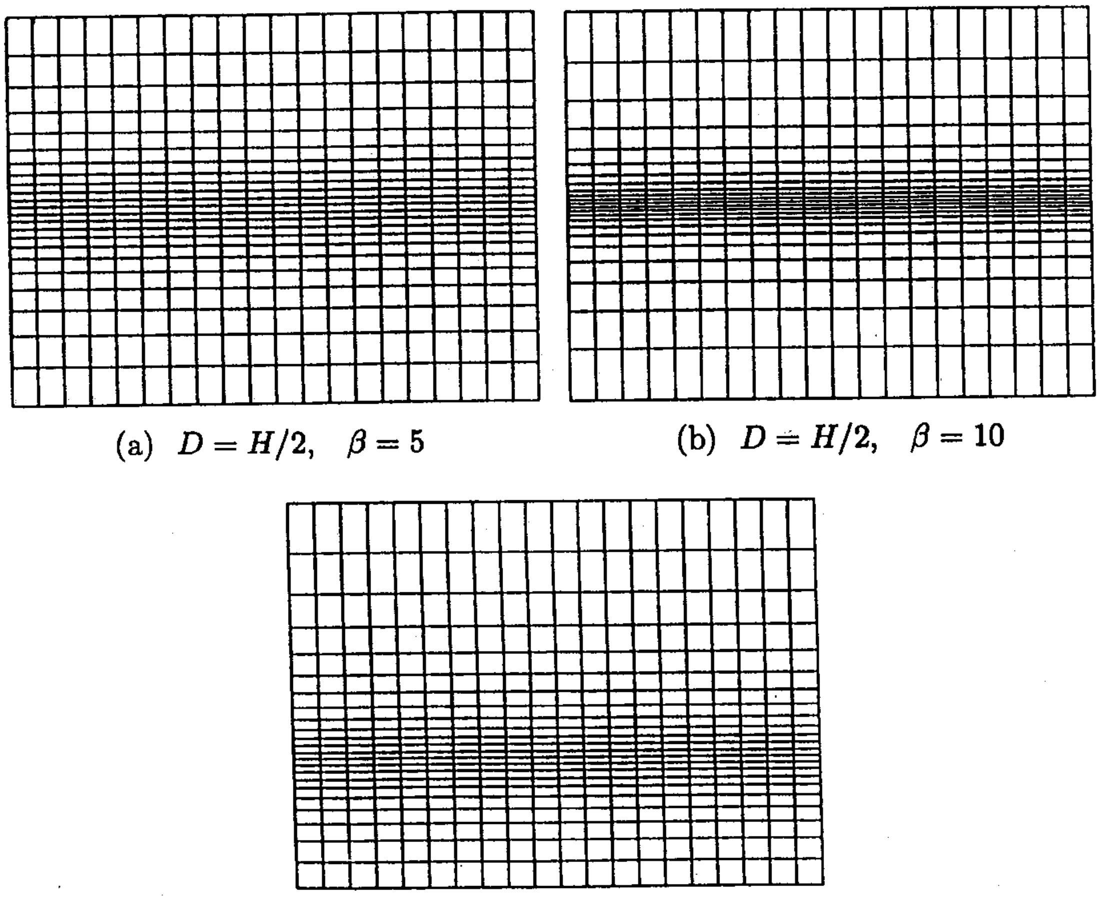
$$\eta_x = 0 \tag{9-53}$$

$$\eta_y = \frac{\sinh(\beta A)}{\beta D\left\{1 + [(y/D) - 1]^2 \sinh^2(\beta A)\right\}^{0.5}} \tag{9-54}$$

The algebraic expressions given by (9-49) and (9-50) are used to generate the grid systems shown in Figures 9-11a through 9-11c. The clustering is specified at $D = H/2$ for the domains shown in Figures 9-11a and 9-11b and at $D = H/4$ for the domain shown in Figure 9-11c. The values of β are 5, 10, and 5, respectively. The distributions of metric η_y are illustrated in Figures 9-12a through 9-12c. Each tic mark on the vertical axis represents a change of 0.5 in Figures 9-12a through 9-12c.

Next, an algebraic expression for generating a grid system around an arbitrary shape is investigated. For simplicity, a conical body shape with a circular cross-section will be assumed. The grid system is determined at cross sections where the relevant PDEs are to be solved. An example would be the solution of parabolized Navier-Stokes equations over a conical configuration in a supersonic flow field. The flow field is depicted in Figure 9-13 where cross-sectional planes are normal to the body axis.

An alternate choice would be the selection of a grid system normal to the body surface. For many applications where the flow is symmetrical, only half of the domain needs to be solved. With that in mind, consider the generation of a grid system at a cross section.



(a) $D = H/2, \quad \beta = 5$

(b) $D = H/2, \quad \beta = 10$

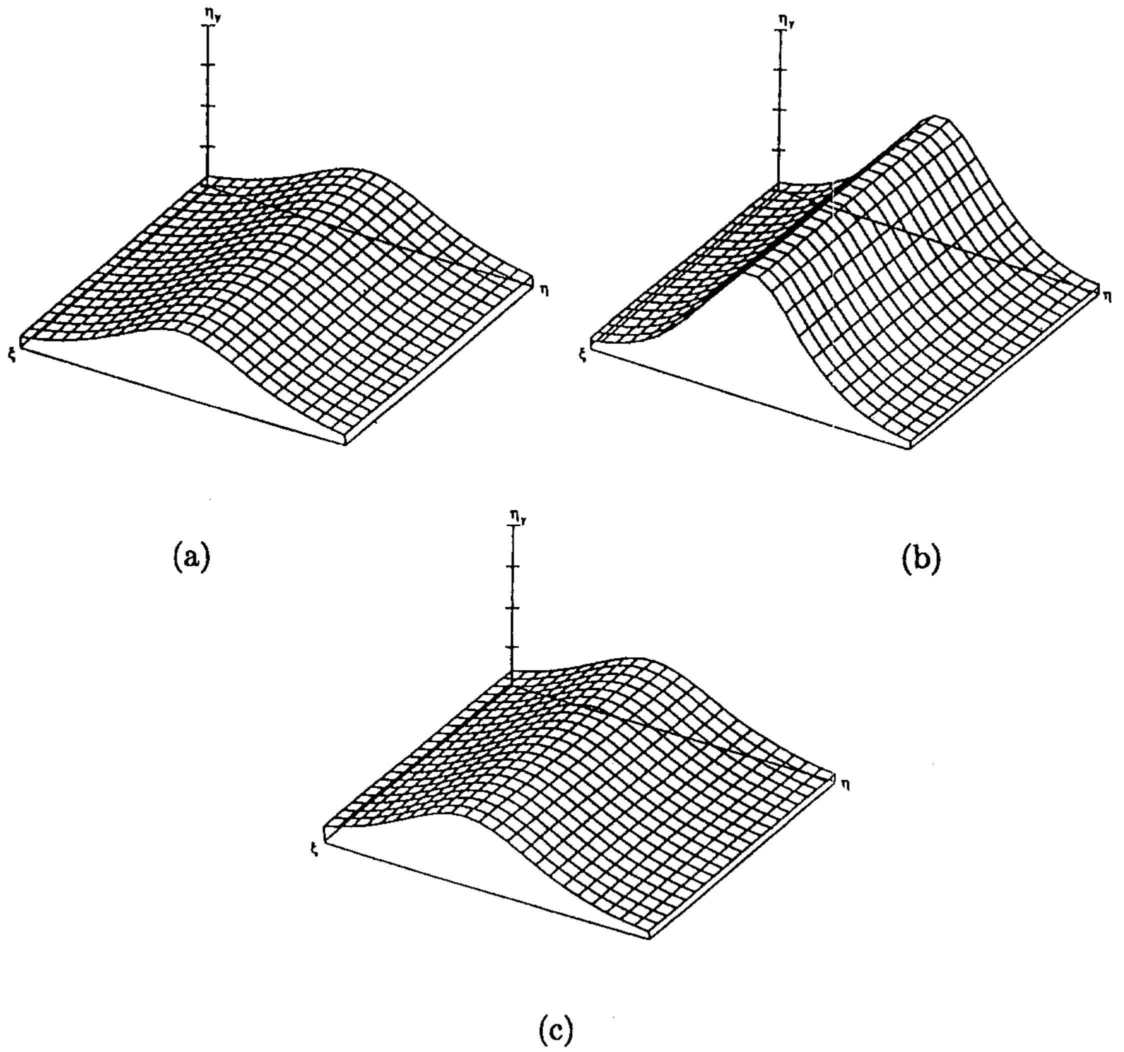
(c) $D = H/4, \quad \beta = 5$

Figures 9-11a, b, c. Grid systems generated by the transformation functions (9-49) and (9-50).

Note that since the physical domain is changing at each streamwise station, a new grid system must be generated at each station. However, no difficulties arise since the grid generating procedure is coded as a subroutine and called at each x-station.

A typical procedure is described next.

First a domain bounded by the body, the plane of symmetry, and the free stream must be defined. The outer boundary in the free stream must be far enough away to include the bow shock generated by the conical configuration. An alternate selection for the outer boundary would be at the bow shock itself. However, the shock location is unknown; therefore, Rankine-Hugoniot relations must be utilized to determine its location. This procedure is known as shock fitting and will be investigated later in Chapter 13.



Figures 9-12a, b, c. Metric distributions for the domains shown in Figure 9-11.

For now, the outer boundary is specified as the free stream. This specification is accomplished by defining two elliptical shapes (or any other geometries you may choose) with semi-major and semi-minor axes denoted by a_1, b_1, and b_2. This specification is illustrated in Figure 9-14. The inner boundary represents the circular cone and will be defined by its radius, R. The number of grid points in the circumferential direction is specified by KM, and for this example they are distributed equally around the body. This distribution is accomplished by defining the incremental angle DELTHET as DELTHET $= \pi/(KM - 1)$, and subsequently computing θ at each k. The nomenclature is shown in Figure 9-15. Note that $k = 1$ is chosen on the windward side of the body and $k = KM$ is on the leeward side. The y and z coordinates of the grid points on the body are easily computed from

$$y(1,k) = -R\cos\theta \tag{9-55}$$

$$z(1,k) = R\sin\theta \tag{9-56}$$

The corresponding grid points on the outer boundary are defined by rays emanating from the origin with the appropriate angular positions.

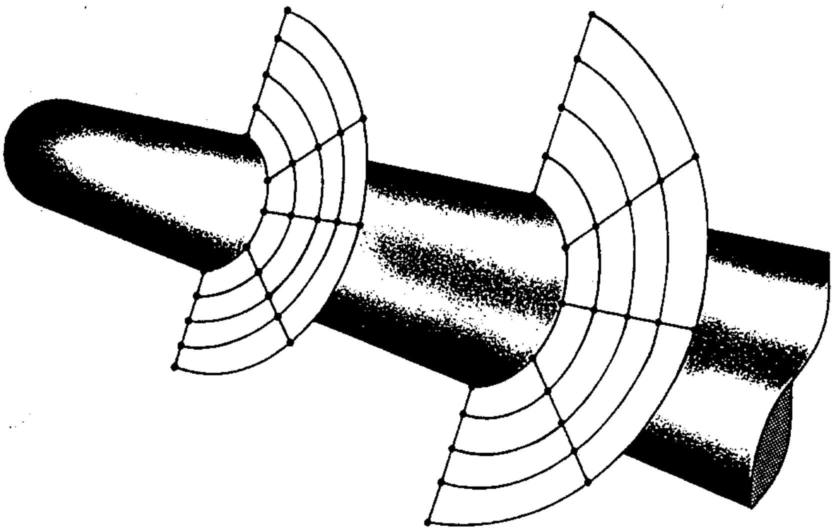


Figure 9-13. The grid system in the physical domain.

For this purpose, the length of the rays are obtained from the equation of an ellipse as

$$r = \frac{1}{\sqrt{\left(\frac{\sin^2\theta}{a^2}\right) + \left(\frac{\cos^2\theta}{b^2}\right)}}$$

where appropriate values of a and b are used. Subsequently, the y and z coordinates of the grid points on the outer boundary are determined by similar expressions given in Equations (9-55) and (9-56).

For viscous flow computations, clustering in the vicinity of the body is desirable. For this purpose the following expression is used:

$$c(k,j) = \delta\left\{1 - \beta\left[\left(\frac{\beta+1}{\beta-1}\right)^{\eta} - 1\right] \Big/ \left[\left(\frac{\beta+1}{\beta-1}\right)^{\eta} + 1\right]\right\} \tag{9-57}$$

In (9-57), δ is the radial distance between the body and the outer boundary, i.e.,

$$\delta(k) = r(k) - R(k) \tag{9-58}$$

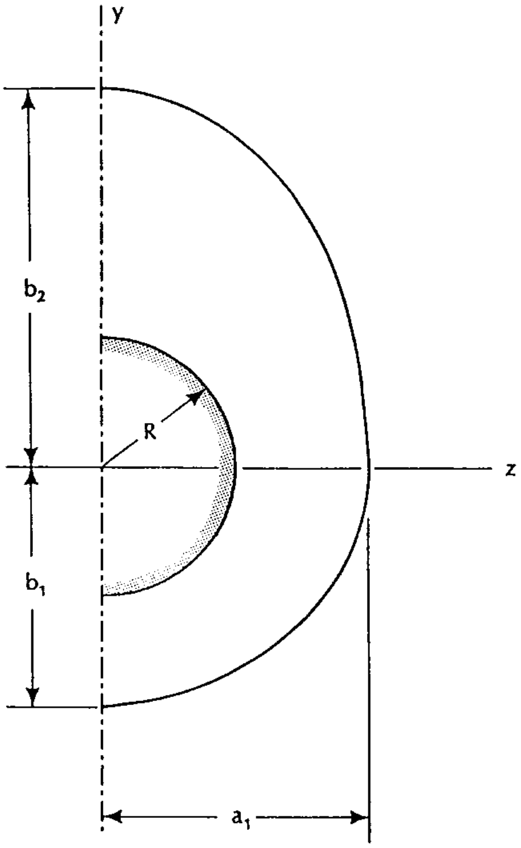


Figure 9-14. Nomenclature required to describe the physical domain.

and β is the clustering parameter. The clustering function given by Equation (9-57) places $\eta = 0.0$ at the outer boundary and $\eta = 1.0$ at the surface. The computational domain is shown in Figure 9-16. A computational domain for the entire problem equivalent to the physical space depicted in Figure 9-13 is shown in Figure 9-17. The y and z coordinates within the physical domain are evaluated from the following equations

$$y(k,j) = y(k,1) - c(k,j)\cos\alpha(k) \tag{9-59}$$

and

$$z(k,j) = z(k,1) + c(k,j)\sin\alpha(k) \tag{9-60}$$

where α is defined as the angle between the normal to the body in the cross-sectional plane and the vertical direction. This relationship is also shown in Figure 9-15.

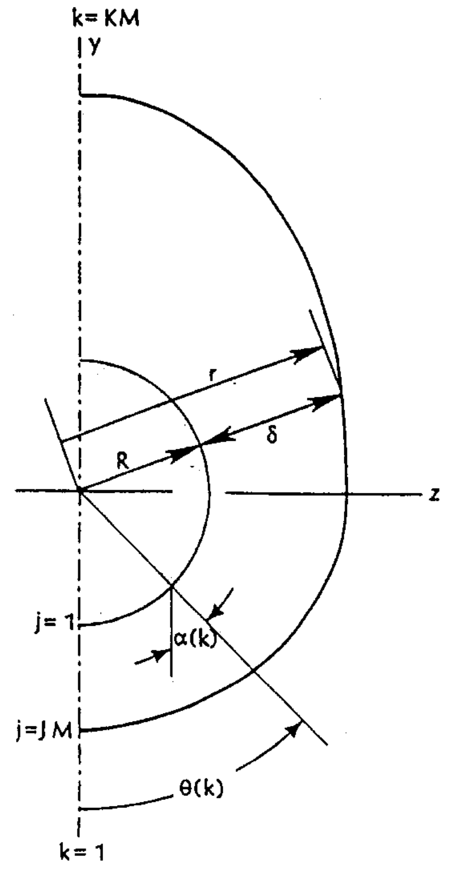


Figure 9-15. Nomenclature required to define the grid system.

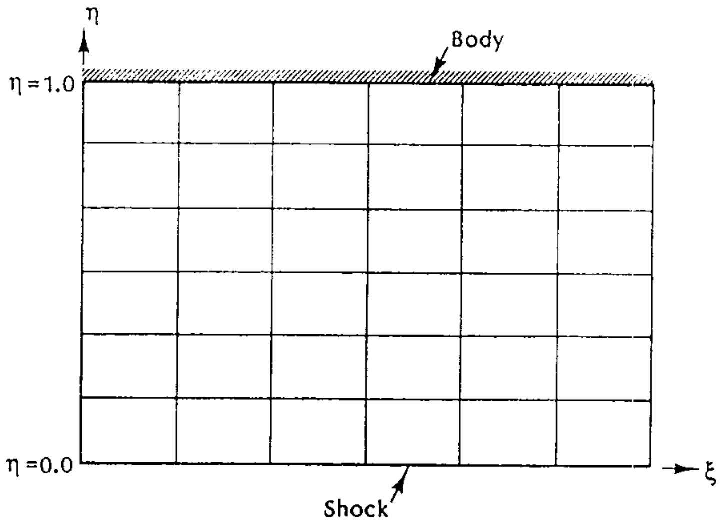


Figure 9-16. Computational grid system at a cross section.

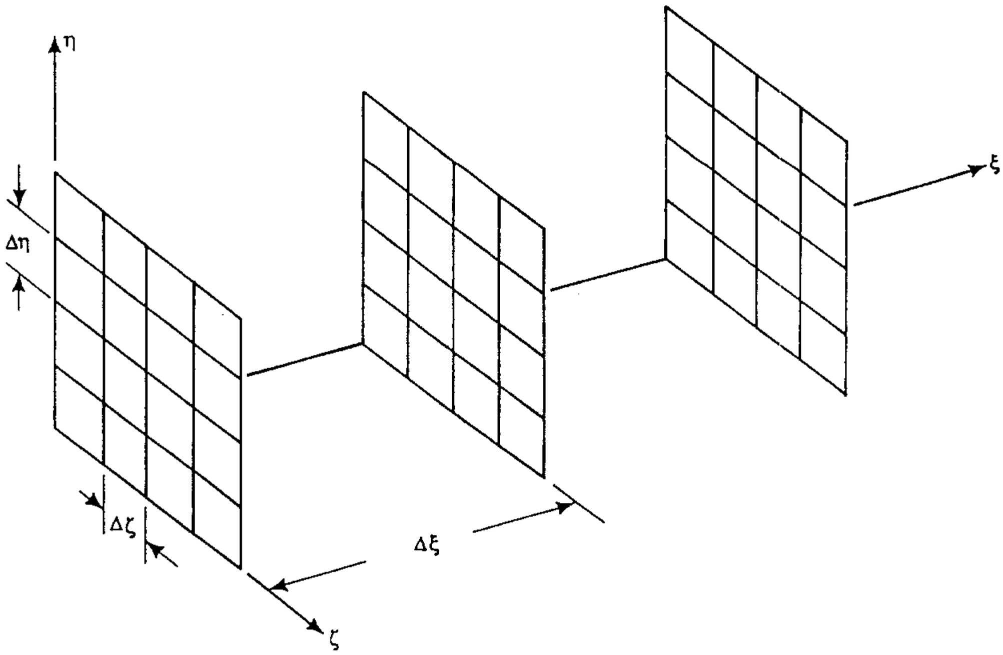


Figure 9-17. Computational domain for the 3-D problem.

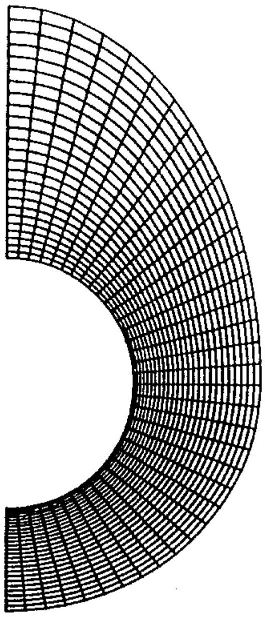
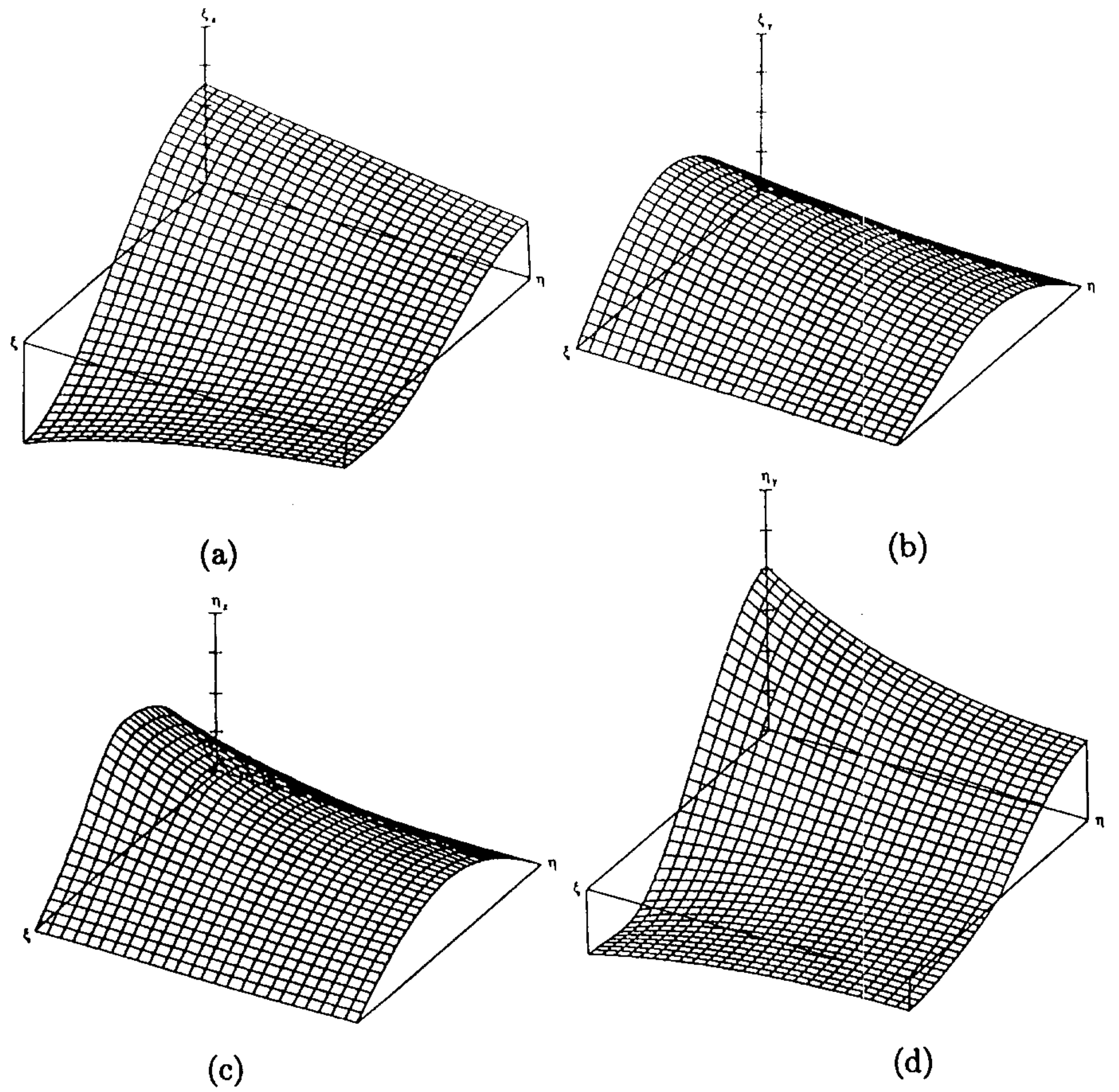


Figure 9-18. Cross-sectional grid system generated for a circular conical configuration.

A grid system generated for a 37 × 26 grid is shown in Figure 9-18 for $\beta = 1.4$. The configuration is defined by $R = 1.0$, $a_1 = 2.0$, $b_1 = 1.8$, and $b_2 = 3.0$. For this problem the metrics are evaluated numerically by using a second-order central difference approximation in the interior of the domain and by second-order forward and backward difference approximations at the boundaries. The metric distributions are illustrated in Figures 9-19a through 9-19d, where the smoothness of the metrics is clearly evident. Note that if the metric distribution had discontinuities, further investigation of the suggested grid system and the solution procedure used to obtain the grid and the metrics would be required.

The algebraic expressions used to generate the grid systems just presented are a few among many appearing in various literature. However, the procedures used to generate grids by algebraic methods are fundamentally similar.

For many applications, algebraic models provide a reasonable grid system with continuous and smooth metric distributions. However, if grid smoothness, skewness, and orthogonality are of concern, grid systems generated by solving PDEs must be used. This option includes elliptic, parabolic, or hyperbolic grid generators, which will be discussed in the next section. For the time being, briefly consider the elliptic grid generators. In this technique, some type of elliptic PDEs is solved to identify the coordinates of the grid points in the physical space. This approach is similar to what was just accomplished with algebraic models, except now a system of PDEs must be solved.



Figures 9-19a, b, c, d. Metric distributions for the domain shown in Figure 9-18.

Recall that iterative procedures were recommended to solve elliptic PDEs. To start such a method, an initial estimate of the grid points must be provided. This initial distribution is easily obtained by use of algebraic methods. Thus, algebraic techniques are used not only to generate grid points for which the equations of fluid motion are solved, but they may also be used as an initial grid point distribution to start an iterative solution procedure for elliptic grid generators. This point will be illustrated in the next section.

Before closing this section, the following conclusions are drawn. The advantages of the algebraic grid generation methods are:

(1) Computationally, they are very fast;

(2) Metrics may be evaluated analytically, thus avoiding numerical errors;

(3) The ability to cluster grid points in different regions can be easily implemented.

The disadvantages are:

(1) Discontinuities at a boundary may propagate into the interior region which could lead to errors due to sudden changes in the metrics;

(2) Control of grid smoothness and skewness is a difficult task.

Some of the disadvantages of the algebraic grid generators are overcome by the use of PDE grid generators which is, of course, accomplished with increased computational time. This procedure will be described in the following section.

9.6 Partial Differential Equation Techniques

A grid generation scheme which is gaining popularity is one in which PDEs are used to create the grid system. In these methods, a system of PDEs is solved for the location of the grid points in the physical space, whereas the computational domain is a rectangular shape with uniform grid spacing. These methods may be categorized as an elliptic, parabolic, or hyperbolic system of PDEs. The elliptic grid generator is the most extensively developed method. It is commonly used for 2-D problems and the procedure has been extended to 3-D problems. Parabolic and hyperbolic grid generators are not as well developed but have some very interesting features. In this section all three methods will be introduced.

The presentation of various schemes will be limited to 2-D problems; however, methods which can be extended to 3-D problems will be identified. By 3-D problems (in grid generation), we refer to situations where all three coordinates are transformed. For some 3-D applications, a coordinate transformation in the streamwise direction may not be required, i.e., $\xi = x$. For such problems, the grid system will be generated only in a 2-D sense at each streamwise location as needed. Thus, the 2-D grid generators are used extensively for 3-D computations. A typical grid system was shown in Figure 9-13.

Due to the advantages previously introduced, the generalized coordinate system will be used in the grid generation techniques which are presented next.

9.7 Elliptic Grid Generators

For domains where all the physical boundaries are specified, elliptic grid generators work very well. A system of elliptic equations in the form of Laplace's equation