CHAPTER SEVENTEEN

Structured Grid Generation

Structured grids are generated in two- or three-dimensional geometries (with plane or curved surfaces). In general, two types of structured grid generation are in use: algebraic methods and partial differential equation (PDE) mapping methods. For more complex geometries, it is preferable to construct multiblocks initially, with refined grids filled in for each of the multiblocks subsequently. Detailed procedures are presented in the following sections.

17.1 ALGEBRAIC METHODS

In algebraic methods, geometric data of the cartesian coordinates in the interior of a domain are generated from the values specified at boundaries through interpolations or specific functions of the curvilinear coordinates. Toward this end, we begin first with the unidirectional interpolations of various functional representations, followed by multidirectional interpolations.

17.1.1 UNIDIRECTIONAL INTERPOLATION

Unidirectional interpolation refers to the functional representation in only one direction. Among the most widely used are Lagrange polynomials, Hermite polynomials, and cubic spline functions. These polynomials, some of which were discussed in Chapter 9, are briefly reviewed below.

(a) Lagrange Polynomials

The Lagrange polynomials, as used in FEM for interpolations of a variable (Section 9.2.2), may be used for grid generation in interpolation between cartesian and curvilinear coordinates (Figure 17.1.1).

$$x = \Phi_N(\xi)x_N, \qquad \Phi_N(\xi_M) = \delta_{NM} \tag{17.1.1}$$

with $\Phi_N(\xi)$ being the Lagrange polynomials

$$\Phi_N = \prod_{M=1,\, M\neq N}^{n} \frac{\xi - \xi_M}{\xi_N - \xi_M}, \qquad \xi = \frac{x}{h} \tag{17.1.2}$$

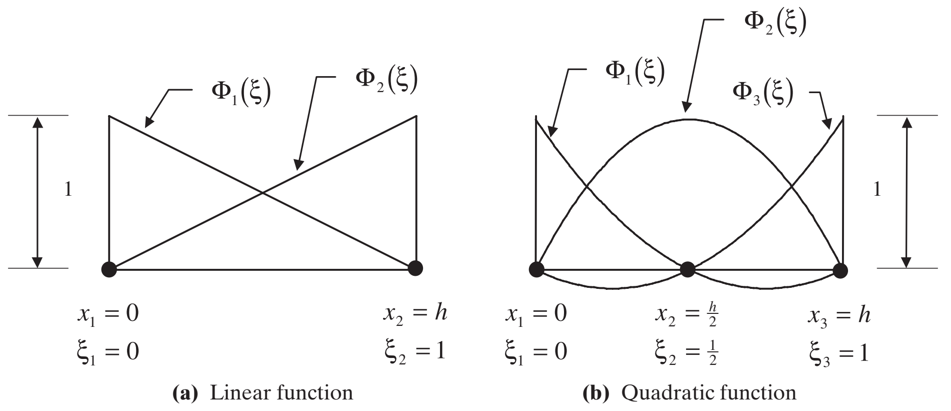


Figure 17.1.1 Lagrange polynomials.

This formula provides:

Linear Interpolation

$$\Phi_1 = 1 - \xi, \qquad \Phi_2 = \xi \tag{17.1.3}$$

Quadratic Interpolation

$$\Phi_1 = 2\left(\xi - \frac{1}{2}\right)(\xi - 1), \qquad \Phi_2 = -4\xi(\xi - 1), \qquad \Phi_3 = 2\xi\left(\xi - \frac{1}{2}\right) \tag{17.1.4}$$

(b) Hermite Polynomials

Hermite polynomials as used in FEM provide functional representation not only of coordinate values, but also of gradients of coordinate values. For example, the functional representation including the zeroth order and first order derivatives of coordinate values are given by cubic functions (Figure 17.1.2),

$$x = H_N^0(\xi)x_N + H_N^1(\xi)\,\theta_N, \qquad \theta_N = \frac{\partial x_N}{\partial \xi} \tag{17.1.5a}$$

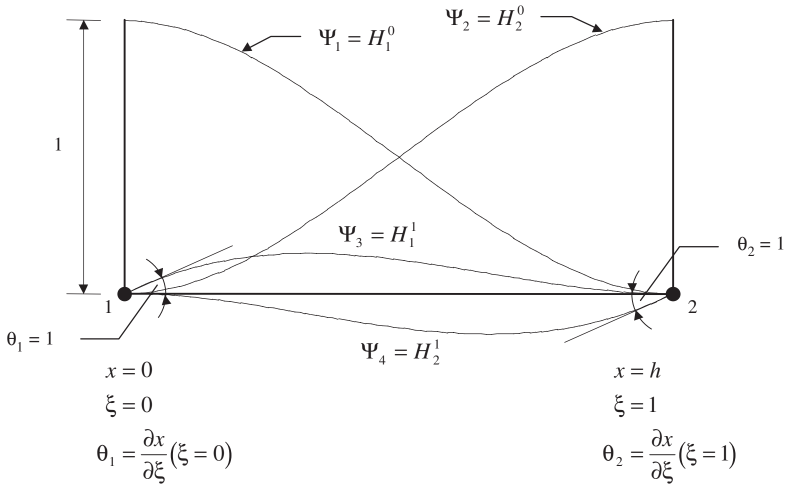


Figure 17.1.2 Hermite polynomial.

or

$$x = H_1^0 x_1 + H_2^0 x_2 + H_1^1\,\theta_1 + H_2^1\,\theta_2 \tag{17.1.5b}$$

with $H_N^0(\xi_M) = \delta_{NM}$, $H_N^1(\xi_M) = \delta_{NM}$
Thus

$$x = \Psi_r Q_r, \quad (r = 1, 2, 3, 4) \tag{17.1.5c}$$

with

$$\Psi_1 = H_1^0(\xi) = 1 - 3\xi^2 + 2\xi^3 \tag{17.1.6a}$$

$$\Psi_2 = H_2^0(\xi) = 3\xi^2 - 2\xi^3 \tag{17.1.6b}$$

$$\Psi_3 = H_1^1(\xi) = \xi - 2\xi^2 + \xi^3 \tag{17.1.6c}$$

$$\Psi_4 = H_2^1(\xi) = \xi^3 - \xi^2 \tag{17.1.6d}$$

These functions match the two boundary values x_1, and x_2 and the first derivatives, $(\partial x/\partial\xi)_1$, and $(\partial x/\partial\xi)_2$ at the two boundaries.

The advantage of specifying $(\partial x/\partial\xi)$ as well as x can be used to make the grid orthogonal at the boundary. This will be useful in multidirectional grid generation.

(c) Cubic Spline Functions

One of the difficulties with conventional polynomial interpolations, particularly if the polynomials are of high order, is the oscillatory character. To remedy this disadvantage, the cubic spline functions can be used to achieve smoother curves.

Consider two arbitrary adjacent points x_i and x_{i+1}. We wish to fit a cubic to these two points and use this cubic as the interpolation function between them.

$$F_i(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \quad (x_i \le x \le x_{i+1}) \tag{17.1.7}$$

Note that two constants in (17.1.7) may be determined by end conditions and two others by the slope (first derivative) and curvature (second derivative). Here the second derivative of a cubic line is a straight line (Figure 17.1.3) so that

$$g''(x) = g''(x_i) + \frac{x - x_i}{x_{i+1} - x_i}[g''(x_{i+1}) - g''(x_i)] \tag{17.1.8}$$

Integrating (17.1.8) twice, we obtain

$$\begin{aligned} g(x) = F_i(x) = {} & \frac{g''(x_i)}{6}\left[\frac{(x_{i+1} - x)^3}{\Delta x_i} - \Delta x_i(x_{i+1} - x)\right] \\ & + \frac{g''(x_{i+1})}{6}\left[\frac{(x - x_i)^3}{\Delta x_i} - \Delta x_i(x - x_i)\right] \\ & + f(x_i)\left(\frac{x_{i+1} - x}{\Delta x_i}\right) + f(x_{i+1})\left(\frac{x - x_i}{\Delta x_i}\right) \end{aligned} \tag{17.1.9}$$

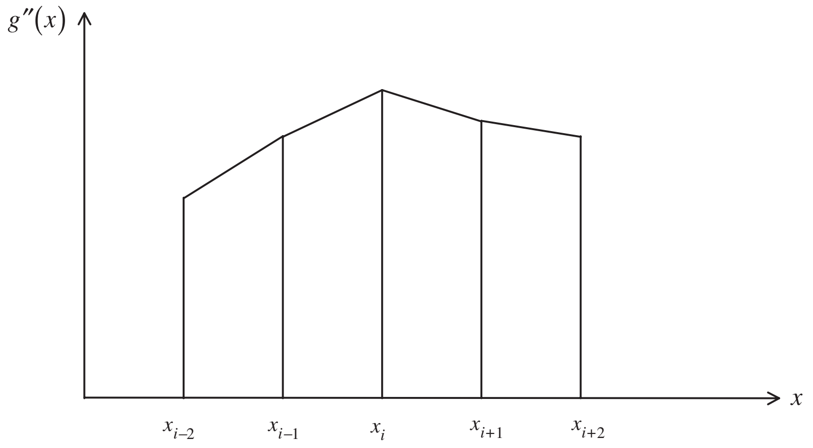


Figure 17.1.3 Cubic spline representation.

with $\Delta x_i = x_{i+1} - x_i, i = 0, 1, \ldots n-1, g(x_i) = f(x_i)$ and $g(x_{i+1}) = f(x_{i+1})$. Since the second derivatives $g''(x_i)$ $(i = 0, 1, \ldots n)$ are still unknown, these must be evaluated as follows:

$$F_i'(x_i) = F_{i-1}'(x_i) \tag{17.1.10a}$$

$$F_i''(x_i) = F_{i-1}''(x_i) \tag{17.1.10b}$$

Evaluation of (17.1.10a) leads to a set of simultaneous linear equations of the form

$$\frac{\Delta x_{i-1}}{\Delta x_i} g''(x_{i-1}) + \frac{2(x_{i+1} - x_{i-1})}{\Delta x_i} g''(x_i) + g''(x_{i+1})$$
$$= 6\left[\frac{f(x_{i+1}) - f(x_i)}{(\Delta x_i)^2} - \frac{f(x_i) - f(x_{i-1})}{(\Delta x_i)(\Delta x_{i-1})}\right] \tag{17.1.11}$$

This represents $n-1$ equations in the $n+1$ unknowns $g''(x_0), g''(x_1), \ldots, g''(x_n)$. The two necessary additional equations are

$$g''(x_0) = 0 \tag{17.1.12a}$$

$$g''(x_n) = 0 \tag{17.1.12b}$$

The resulting $g(x)$ is called a natural cubic spline.

In terms of nondimensional coordinates, (17.1.11) and (17.1.12) are written as

$$(\xi_i - \xi_{i-1})x_{i-1}'' + 2(\xi_{i+1} - \xi_{i-1})x_i'' + (\xi_{i+1} - \xi_i)x_{i+1}'' = 6\left(\frac{x_{i+1} - x_i}{\xi_{i+1} - \xi_i} - \frac{x_i - x_{i-1}}{\xi_i - \xi_{i-1}}\right) \tag{17.1.13}$$

with

$$x_1'' = 0 \tag{17.1.14a}$$

$$x_n'' = 0 \tag{17.1.14b}$$

The solution x'' is substituted into

$$x = \frac{(\xi_{i+1} - \xi)^3}{6\,(\xi_{i+1} - \xi_i)} x_i'' + \frac{(\xi - \xi_i)^3}{6\,(\xi_{i+1} - \xi_i)} x_{i+1}'' + \left[\frac{x_i}{\xi_{i+1} - \xi_i} - \frac{\xi_{i+1} - \xi_i}{6} x_i''\right](\xi_{i+1} - \xi)$$
$$+ \left[\frac{x_{i+1}}{\xi_{i+1} - \xi_i} - \frac{\xi_{i+1} - \xi_i}{6} x_{i+1}''\right](\xi - \xi_i) \qquad (17.1.15)$$

It is seen that (17.1.15) may be written in the form similar to (17.1.1) as a linear combination of interpolation functions and nodal values of the first and second derivatives of x at nodal points i and $i+1$.

Additional interpolation functions useful for surface grid generations are available. These functions will be discussed in Section 17.3.

17.1.2 MULTIDIRECTIONAL INTERPOLATION

There are two multidirectional interpolation methods available: domain vertex methods developed from FEM interpolation functions and transfinite interpolation methods predominantly used in FDM, constructed by means of tensor products of unidirectional functional representation in multidimensions.

17.1.2.1 Domain Vertex Method

Domain vertex methods utilize tensor products of unidirectional interpolation functions for two or three dimensions. Let us consider a two-dimensional domain with physical coordinates (x, y) and transformed computational domain (ξ, η) as shown in Figure 17.1.4a, related by

$$x_i = \hat{\Phi}_N(\xi)\,\hat{\Phi}_M(\eta) x_{iNM}, \quad (i = 1, 2, \;\; N, M = 1, 2) \qquad (17.1.16a)$$

or

$$x_i = \Phi_N(\xi,\, \eta) x_{iN}, \quad (i = 1, 2, \;\; N = 1, 2, 3, 4) \qquad (17.1.16b)$$

where i denotes the physical coordinate directions and N and M represent node numbers in the direction of the coordinate $\hat{\Phi}_N(\xi)$, and $\hat{\Phi}_M(\eta)$ are the unidirectional functions whereas $\Phi_N(\xi,\, \eta)$ indicates the tensor product.

$$\Phi_N(\xi,\, \eta) = \begin{cases} \Phi_1 = (1-\xi)(1-\eta) & = \hat{\Phi}_1(\xi)\,\hat{\Phi}_1(\eta) \\ \Phi_2 = \xi(1-\eta) & = \hat{\Phi}_2(\xi)\,\hat{\Phi}_1(\eta) \\ \Phi_3 = \xi\eta & = \hat{\Phi}_2(\xi)\,\hat{\Phi}_2(\eta) \\ \Phi_4 = (1-\xi)\eta & = \hat{\Phi}_1(\xi)\,\hat{\Phi}_2(\eta) \end{cases} \qquad (17.1.17)$$

which are known as "blending functions."

Similarly for three dimensions (Figure 17.1.4b), we obtain

$$x_i = \hat{\Phi}_N(\xi)\,\hat{\Phi}_M(\eta)\,\hat{\Phi}_P(\zeta) x_{iNMP}, \quad (i = 1, 2, 3, \;\; N, M, P = 1, 2) \qquad (17.1.18a)$$

or

$$x_i = \Phi_N(\xi,\eta,\zeta) x_{iN}, \quad (i = 1, 2, 3, \;\; N = 1, \ldots, 8) \qquad (17.1.18b)$$

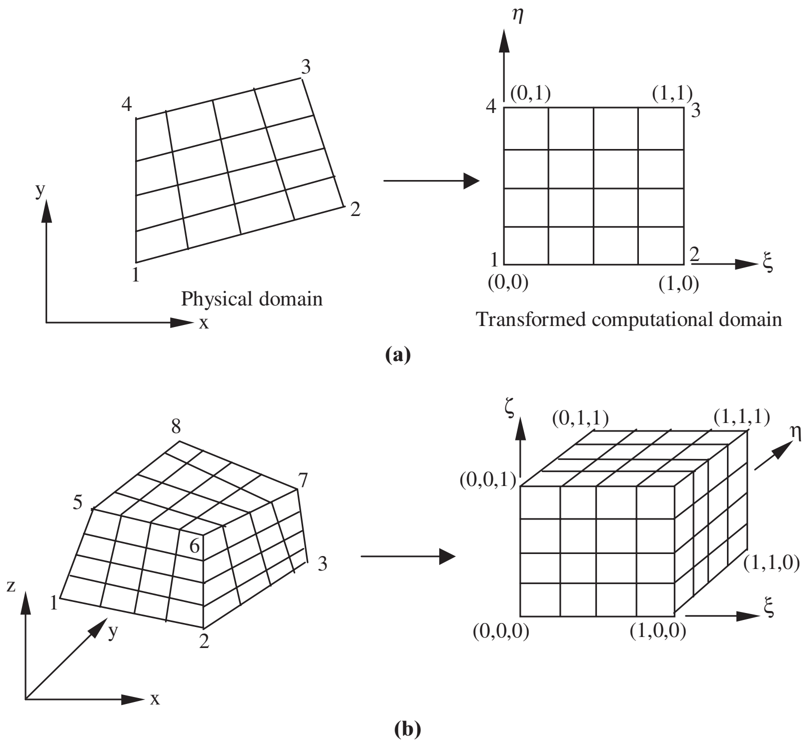


Figure 17.1.4 Multidimensional interpolation, all interior lines (as many as desired) are generated from (17.1.16) with the corner node coordinates and the interior values of ξ and η. (**a**) Two-dimensional domain. (**b**) Three-dimensional domain.

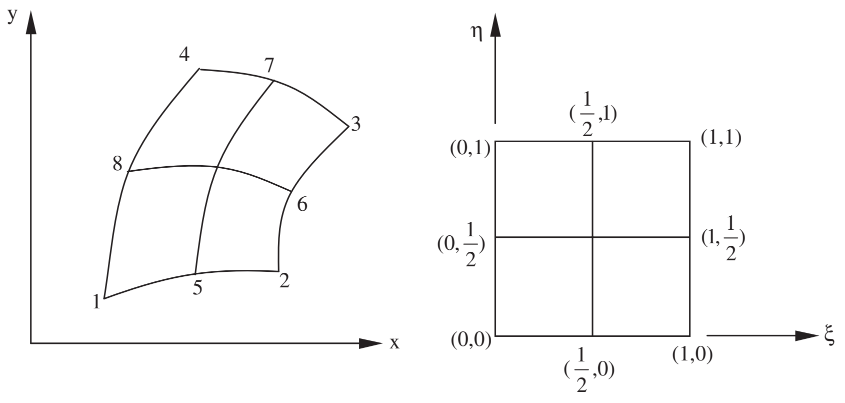
with

$$\Phi_N(\xi,\eta,\zeta) = \begin{cases} \Phi_1 = (1-\xi)(1-\eta)(1-\zeta) \\ \Phi_2 = \xi(1-\eta)(1-\zeta) \\ \Phi_3 = \xi\eta(1-\zeta) \\ \Phi_4 = (1-\xi)\eta(1-\zeta) \\ \Phi_5 = (1-\xi)(1-\eta)\zeta \\ \Phi_6 = \xi(1-\eta)\zeta \\ \Phi_7 = \xi\eta\zeta \\ \Phi_8 = (1-\xi)\eta\zeta \end{cases} \tag{17.1.19}$$

Extensions of the above processes can be made to accommodate higher order interpolations by providing interior nodes along each side (see Figure 17.1.5 for quadratic mapping). Furthermore, triangular elements and tetrahedral elements can also be constructed, following the FEM geometries discussed in Chapter 9.

Example 17.1.1 Trapezoidal Geometry

Given: Four points $A(0,0)$, $B(L,0)$, $C(L,H_2)$, and $D(0, H_1)$. Generate a mesh corresponding to ξ,η at 0.2 apart. Assume $L = 20$, $H_1 = 5$, $H_2 = 10$.



$$\Phi_1 = (1-\xi)(1-\eta)(1-2\xi-2\eta) \qquad \Phi_3 = \xi\eta(3-2\xi-2\eta)$$
$$\Phi_2 = \xi(1-\eta)(2\xi-2\eta-1) \qquad \Phi_4 = (1-\xi)\eta(-2\xi+2\eta-1)$$
$$\Phi_5 = 4\xi(1-\xi)(1-\eta) \qquad \Phi_7 = 4\xi(1-\xi)\eta$$
$$\Phi_6 = 4\xi\eta(1-\eta) \qquad \Phi_8 = 4(1-\xi)\eta(1-\eta)$$

Figure 17.1.5 Quadratic interpolation by inserting any values of ξ and η, interior coordinates are generated from the above functions (as many as desired).

Solution:

$$\begin{aligned}
x &= (1-\xi)(1-\eta)x_1 + \xi(1-\eta)x_2 + \xi\eta x_3 + (1-\xi)\eta x_4 \\
&= [\xi(1-\eta) + \xi\eta]\, L \\
&= 20\,\xi
\end{aligned}$$

with $x_1 = 0, \quad x_2 = L, \quad x_3 = L, \quad x_4 = 0, \quad L = 20$

$$\begin{aligned}
y &= (1-\xi)(1-\eta)y_1 + \xi(1-\eta)y_2 + \xi\eta y_3 + (1-\xi)\eta y_4 \\
&= \xi\eta H_2 + (1-\xi)\eta H_1 \\
&= 10\,\xi\eta + 5(1-\xi)\eta
\end{aligned}$$

with $y_1 = 0, \quad y_2 = 0, \quad y_3 = H_2 = 10, \quad y_4 = H_1 = 5$

The grid points or lines x, y can now be generated, and the results are shown in Figure E17.1.1.

Example 17.1.2

Consider a quarter circular disk as shown in Figure E17.1.2a. Using quadratic Lagrange polynomials develop a program to generate a 7 × 16 mesh:

Solution: The quadratic Lagrange interpolation functions (9 node) are given by

$$\Phi_N(\xi,\eta) = \prod_{M=1,\ N\neq M}^{n} \frac{\xi-\xi_M}{\xi_N-\xi_M}\,\frac{\eta-\eta_M}{\eta_N-\eta_M}$$

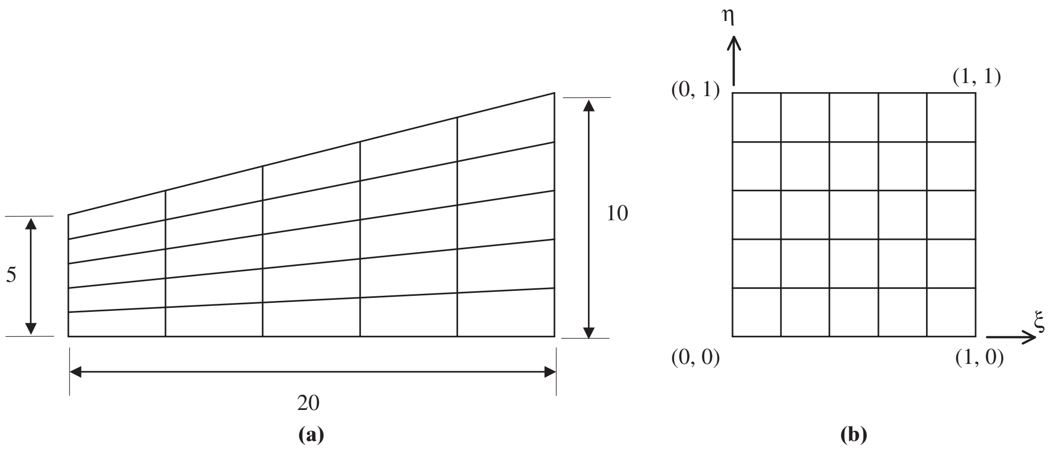


Figure E17.1.1 Physical (trapezoidal) and transformed geometries.

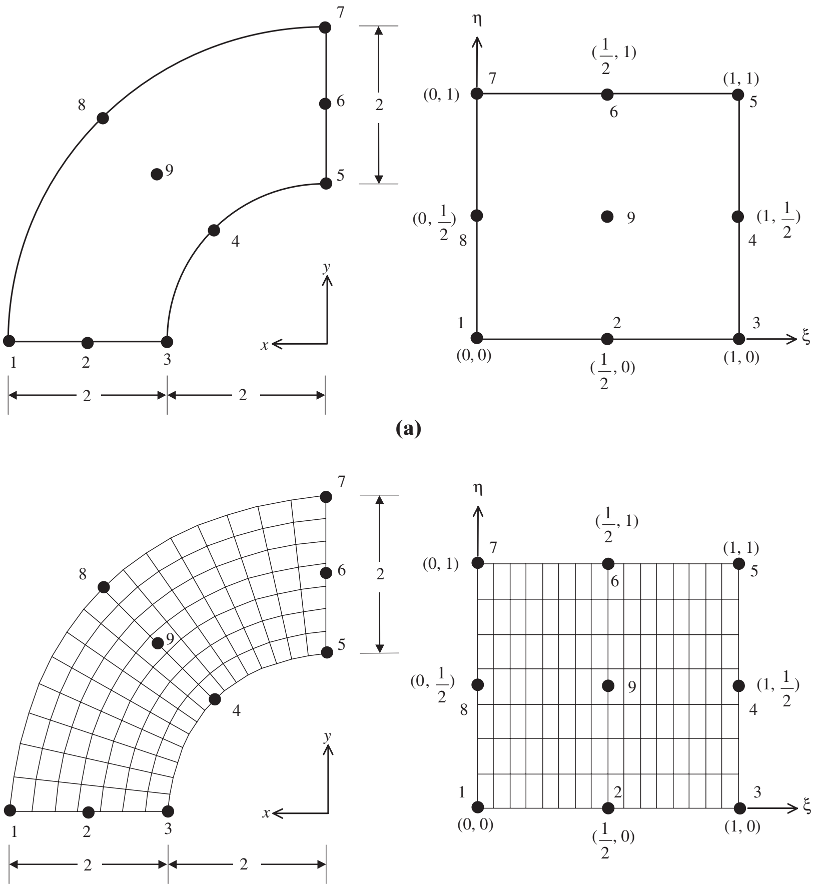


Figure E17.1.2 Quadratic Lagrange polynomials. **(a)** Quarter circle disk. **(b)** Mesh generated for a quarter circular disk using quadratic Lagrange polynomials.

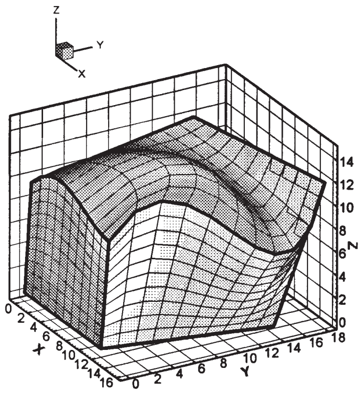
$$\Phi_N(\xi,\eta) = \begin{cases} \Phi_1 = 4\left(\xi - \frac{1}{2}\right)(\xi - 1)\left(\eta - \frac{1}{2}\right)(\eta - 1) \\ \Phi_2 = -8\xi(\xi - 1)\left(\eta - \frac{1}{2}\right)(\eta - 1) \\ \Phi_3 = 4\xi\left(\xi - \frac{1}{2}\right)\left(\eta - \frac{1}{2}\right)(\eta - 1) \\ \Phi_4 = -8\xi\left(\xi - \frac{1}{2}\right)\eta(\eta - 1) \\ \Phi_5 = 4\xi\left(\xi - \frac{1}{2}\right)\eta\left(\eta - \frac{1}{2}\right) \\ \Phi_6 = -8\xi(\xi - 1)\eta\left(\eta - \frac{1}{2}\right) \\ \Phi_7 = 4\left(\xi - \frac{1}{2}\right)(\xi - 1)\eta\left(\eta - \frac{1}{2}\right) \\ \Phi_8 = -8\left(\xi - \frac{1}{2}\right)(\xi - 1)\eta(\eta - 1) \\ \Phi_9 = 16\xi(\xi - 1)\eta(\eta - 1) \end{cases}$$

$$x_i(\xi,\eta) = \Phi_N(\xi,\ \eta)x_{iN}$$

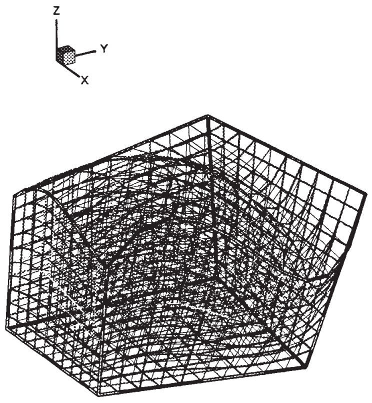
The results are shown in Figure E17.1.2.

Example 17.1.3

Consider a three-dimensional geometry as shown in Figure E17.1.3. Develop a computer program to generate an 11 × 11 × 11 mesh (1331 points) using the domain vertex method. First derive the Lagrange polynomial interpolation functions with the data (x, y, z) as given in Table E17.1.3. Use a cubic between nodes 5–8 and 8–11.



(a)



(b)

Figure E17.1.3 Three-dimensional grid, domain vertex method. **(a)** Surface nodal geometry. **(b)** Interior nodes.

Table E17.1.3 Interpolation Function Data

Point	(x, y, z)	Point	(x, y, z)	Point	(x, y, z)	Point	(x, y, z)
1	(0, 0, 0)	2	(12, 0, 0)	3	(15, 14, 0)	4	(0, 16, 0)
5	(1, 0, 10)	6	(3, 1, 11)	7	(6, 2, 9)	8	(8, 3, 7)
9	(12, 9, 10)	10	(14, 14, 6)	11	(16, 18, 12)	12	(4, 13, 14)

Physical domain coordinates are given:

$$a = \text{Length between 5 and 8} = \sqrt{6} + \sqrt{14} + 3$$
$$b = \text{Length between 8 and 11} = \sqrt{61} + \sqrt{45} + \sqrt{56}$$
$$\Phi_N(\xi,\eta,\zeta) = \prod_{M=1,\ N\neq M}^{n} \frac{\xi-\xi_M}{\xi_N-\xi_M}\,\frac{\eta-\eta_M}{\eta_N-\eta_M}\,\frac{\zeta-\zeta_M}{\zeta_N-\zeta_M}$$

with $n = \hat{n} + 1$, $\hat{n}$ being the total number of inside edge nodes in each direction (ξ, η, ζ).

$$\Phi_1 = \frac{(\xi-\xi_2)}{(\xi_1-\xi_2)}\frac{(\eta-\eta_2)}{(\eta_1-\eta_2)}\frac{(\zeta-\zeta_2)}{(\zeta_1-\zeta_2)} = \frac{(\xi-1)}{(0-1)}\frac{(\eta-1)}{(0-1)}\frac{(\zeta-1)}{(0-1)}$$
$$= -(\xi-1)(\eta-1)(\zeta-1)$$

$$\Phi_2 = \xi(\eta-1)(\zeta-1) \qquad \Phi_3 = -\xi\,\eta(\zeta-1) \qquad \Phi_4 = (\xi-1)\,\eta(\zeta-1)$$

$$\Phi_5 = \frac{a^2}{\sqrt{6}(\sqrt{6}+\sqrt{14})}\left(\xi-\frac{\sqrt{6}}{a}\right)\left(\xi-\frac{\sqrt{6}+\sqrt{14}}{a}\right)(\xi-1)(\eta-1)\zeta$$

$$\Phi_6 = \frac{a^3}{\sqrt{6}\sqrt{14}(\sqrt{6}-a)}\xi\left(\xi-\frac{\sqrt{6}+\sqrt{14}}{a}\right)(\xi-1)(\eta-1)\zeta$$

$$\Phi_7 = \frac{-a^3}{(\sqrt{6}+\sqrt{14})\sqrt{14}(\sqrt{6}+\sqrt{14}-a)}\xi\left(\xi-\frac{\sqrt{6}}{a}\right)(\xi-1)(\eta-1)\zeta$$

$$\Phi_8 = \frac{\xi(\xi-1)(\eta-1)\zeta\left(\xi-\frac{\sqrt{6}}{a}\right)}{\left(1-\frac{\sqrt{6}}{a}\right)}\frac{\left(\xi-\frac{\sqrt{6}+\sqrt{14}}{a}\right)}{\left(1-\frac{\sqrt{6}+\sqrt{14}}{a}\right)}\frac{\left(\eta-\frac{\sqrt{61}}{b}\right)}{\left(\frac{\sqrt{61}}{b}\right)}$$
$$\times\frac{\left(\eta-\frac{\sqrt{61}+\sqrt{45}}{b}\right)}{\left(\frac{\sqrt{61}+\sqrt{45}}{b}\right)}$$

$$\Phi_9 = \frac{\xi\eta\zeta}{\left(\frac{\sqrt{61}}{b}\right)} \frac{\left(\eta - \frac{\sqrt{61}+\sqrt{45}}{b}\right)}{\left(\frac{\sqrt{61}}{b} - \frac{\sqrt{61}+\sqrt{45}}{b}\right)} \frac{(\eta-1)}{\left(\frac{\sqrt{61}}{b} - 1\right)}$$

$$\Phi_{10} = \frac{\xi\eta\zeta}{\left(\frac{\sqrt{61}+\sqrt{45}}{b}\right)} \frac{\left(\eta - \frac{\sqrt{61}}{b}\right)}{\left(\frac{\sqrt{61}+\sqrt{45}}{b} - \frac{\sqrt{61}}{b}\right)} \frac{(\eta-1)}{\left(\frac{\sqrt{61}+\sqrt{45}}{b} - 1\right)}$$

$$\Phi_{11} = \xi\eta\zeta \frac{\left(\eta - \frac{\sqrt{61}}{b}\right)\left(\eta - \frac{\sqrt{61}+\sqrt{45}}{b}\right)}{\left(1 - \frac{\sqrt{61}}{b}\right)\left(1 - \frac{\sqrt{61}+\sqrt{45}}{b}\right)} \qquad \Phi_{12} = -(\xi - 1)\eta\zeta$$

Example 17.1.4

Clustering of boundary layers at the wall or interior domain may be achieved using exponential relations between the physical domain and transformed domain (Figure E17.1.4).

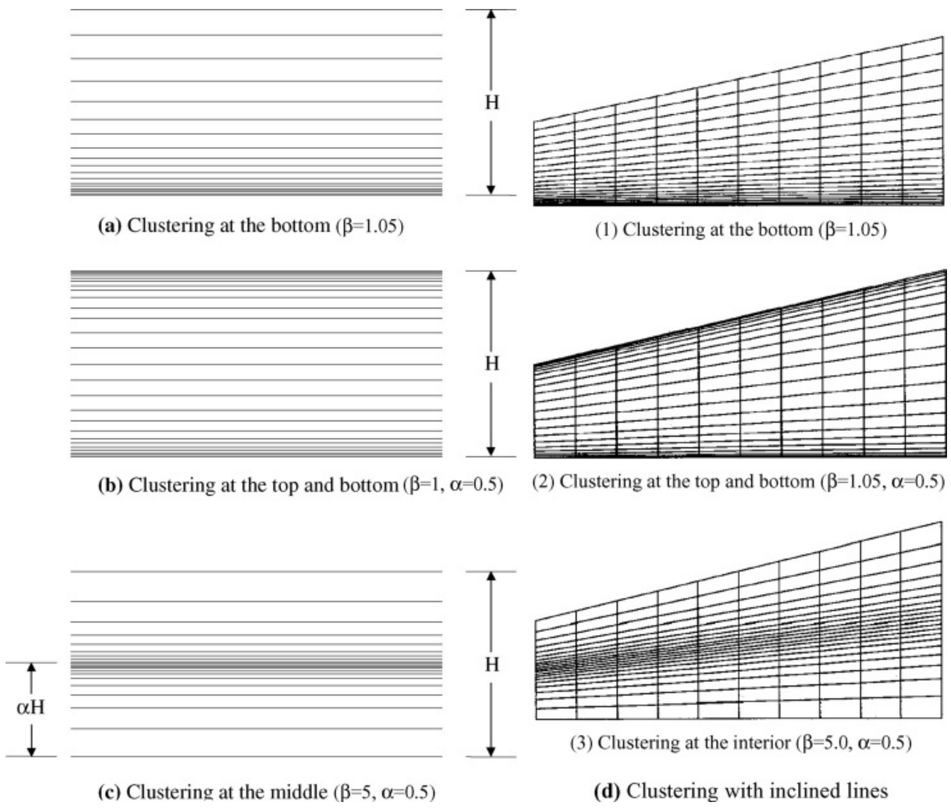

H
(a) Clustering at the bottom (β=1.05)
(1) Clustering at the bottom (β=1.05)
H
(b) Clustering at the top and bottom (β=1, α=0.5)
(2) Clustering at the top and bottom (β=1.05, α=0.5)
αH
H
(3) Clustering at the interior (β=5.0, α=0.5)
(c) Clustering at the middle (β=5, α=0.5)
(d) Clustering with inclined lines

Figure E17.1.4 Clustering of mesh lines.

(a) Clustering at the Bottom Wall

$$x = \xi$$

$$y = H \frac{(\beta+1) - (\beta-1)\left(\dfrac{\beta+1}{\beta-1}\right)^{1-\eta}}{\left(\dfrac{\beta+1}{\beta-1}\right)^{1-\eta} + 1}$$

with $1 < \beta < \infty$

(b) Clustering at Top and Bottom Walls

$$x = \xi$$

$$y = H \frac{(2\alpha+\beta)\left(\dfrac{\beta+1}{\beta-1}\right)^{\frac{\eta-\alpha}{1-\alpha}} + 2\alpha - \beta}{(2\alpha+1)\left[\left(\dfrac{\beta+1}{\beta-1}\right)^{\frac{\eta-\alpha}{1-\alpha}} + 1\right]}$$

with $0 < \alpha, \beta < \infty$

(c) Clustering at Interior Domain

$$x = \xi$$

$$y = \alpha H \left\{ 1 + \frac{\sinh \beta(\eta - A)}{\sinh(\beta A)} \right\}$$

with $0 < \beta < \infty, \quad 0 < \alpha < 1, \qquad A = \dfrac{1}{2\beta} \ln \dfrac{1 + (e^{\beta} - 1)\alpha}{1 + (e^{-\beta} - 1)\alpha}$

Example 17.1.5

Grid generation over a conical body. Consider a conical body with a typical circular cross section of radius R and a physical domain with semi-major and semi-minor axes as shown in Figure E17.1.5.

Grid points y and z are given by

$$y(k,1) = -R\cos\theta$$
$$z(k,1) = R\sin\theta$$

Clustering in the vicinity of the body for the viscous boundary layer can be achieved by

$$y(k,j) = y(k,1) - c(k,j)\cos\theta(k)$$
$$z(k,j) = z(k,1) + c(k,j)\sin\theta(k)$$

where

$$c(k,j) = \delta \left\{ 1 - \frac{\beta\left[\left(\dfrac{\beta+1}{\beta-1}\right)^{\eta} - 1\right]}{\left(\dfrac{\beta+1}{\beta-1}\right)^{\eta} + 1} \right\}$$

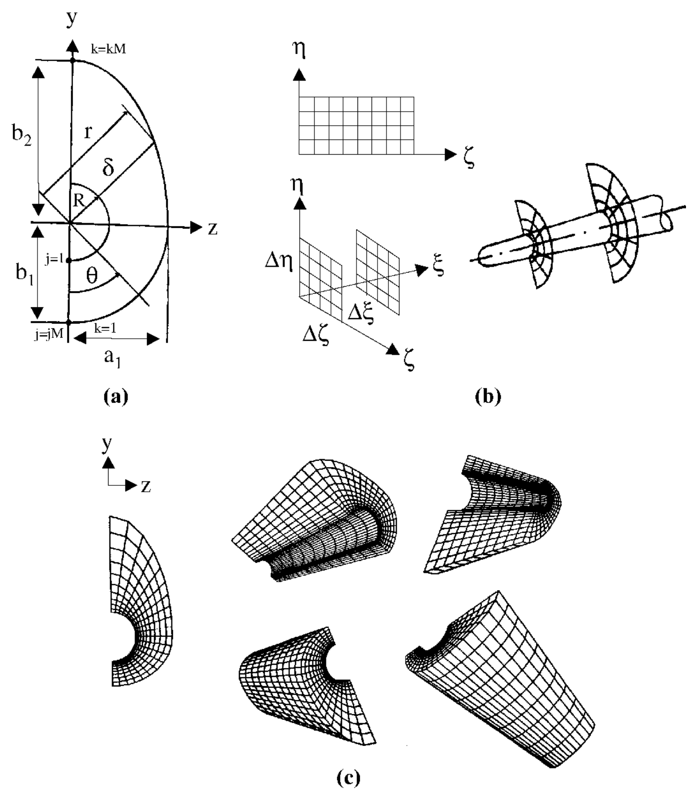


Figure E17.1.5 Algebraic grid generation of conical body. (**a**) Given data. (**b**) Transformed cross sections. (**c**) Finalized mesh.

with

$$\delta(k) = r(k) - R(k), \qquad r = \left[\left(\frac{\sin\theta}{a}\right)^2 + \left(\frac{\cos\theta}{b}\right)^2\right]^{-1/2}$$

The grid generated in Figure E17.1.5(c) will then be repeated in the (x, ξ) direction for the entire three-dimensional domain. Here, $R = 1$, $a_1 = 2.5$, $b_2 = 4$ were chosen in Figure E17.1.5(c).

17.1.2.2 Transfinite Interpolation Methods (TFI)

An alternative approach to the domain vertex methods is to use the unidirectional interpolation functions introduced in Section 17.1.1 and form tensor products in two or three directions as in the domain vertex methods, but with all sides of the boundaries interpolated and matched as well as the corner nodes. To this end, Lagrange polynomials, Hermite polynomials, or spline functions may be used. A transformed computational domain mapped into various arbitrary physical domains is shown in Figure 17.1.6.

Consider a region $\overline{\Omega}$, $[0, 1] \times [0, 1]$ and postulate the existence of a function **F** (vector valued) which maps $\overline{\Omega}$ into Ω such that **F**: F $\rightarrow \overline{\Gamma}$. Our objective is to construct a univalent (one-to-one) function **U:** $\overline{\Omega} \rightarrow \Omega$ which matches **F** on the boundary

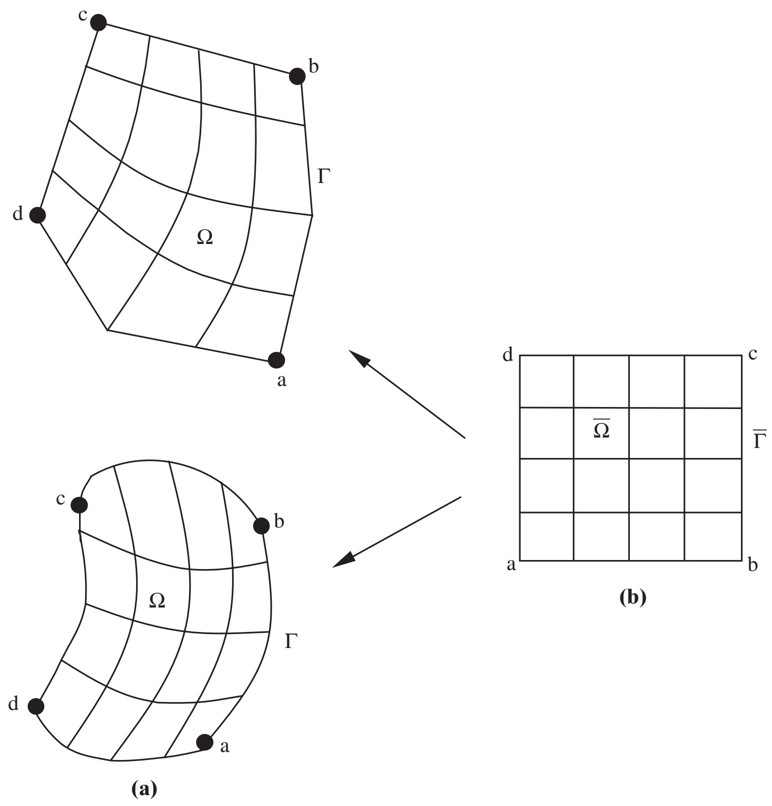


Figure 17.1.6 Physical domain and transformed computational domain for transfinite interpolation. (**a**) Physical domain. (**b**) Transformed computational domain.

of $\overline{\Gamma}$, that is,

$$\mathbf{U}(0,\eta)=\mathbf{F}(0,\eta),\quad \mathbf{U}(\xi,0)=\mathbf{F}(\xi,0) \tag{17.1.20a}$$

$$\mathbf{U}(1,\eta)=\mathbf{F}(1,\eta),\quad \mathbf{U}(\xi,1)=\mathbf{F}(\xi,1) \tag{17.1.20b}$$

A function **U** which interpolates to **F** at a finite set of points is defined as the transfinite interpolant of **F**. The isoparametric interpolation scheme is a special case of the transfinite interpolation schemes.

Consider now a linear operator known as a projector $\wp$, such that $\mathbf{U}\rightarrow\wp[\mathbf{F}]$ is a univalent map of $\overline{\Omega}\rightarrow\Omega$ satisfying the desired interpolatory properties [Gordon and Hall, 1973].

$$\wp(\xi)\,[\mathbf{F}(\eta)]=\Phi_1(\xi)\mathbf{F}_1(0,\eta)+\Phi_2(\xi)\mathbf{F}_2(1,\eta) \tag{17.1.21a}$$

$$\wp(\eta)[\mathbf{F}(\xi)]=\Phi_1(\eta)\mathbf{F}_1(\xi,0)+\Phi_2(\eta)\mathbf{F}_2(\xi,1) \tag{17.1.21b}$$

Then the tensor product projection

$$\wp(\xi)\;\wp(\eta)\,[\mathbf{F}]=\Phi_N(\xi)\Phi_M(\eta)\mathbf{F}_{NM} \tag{17.1.22}$$

interpolates to **F** at four corners of $[0,1]\times[0,1]$. Here $\mathbf{F}_{NM}$ matches the function at the four corners, but it may not match the function on all the boundaries as illustrated in Figure 17.1.7. Similar effects occur on all other boundaries. These discrepancies can be

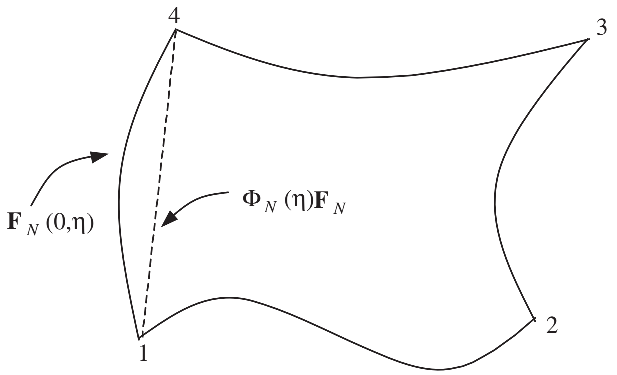


Figure 17.1.7 $\mathbf{F}_{NM}$ match the function at the four corners but not on all boundaries.

removed by subtracting from the sum of (17.1.21a,b) a function formed by interpolating the discrepancies (17.1.22), which represents the Boolean sum projection [Coons, 1967].

$$\mathbf{U} = [\wp(\xi) \oplus \wp(\eta)][\mathbf{F}] = \wp(\xi)\,\mathbf{F}(\eta) + \wp(\eta)\,\mathbf{F}(\xi) - \wp(\xi)\ \wp(\eta)\,[\mathbf{F}] \tag{17.1.23}$$

where the symbol $\oplus$ implies the tensor product and $\mathbf{F}(\eta)$ and $\mathbf{F}(\xi)$ are the parameterization of the sides of the domain and $[\mathbf{F}]$ represents the corresponding vertices. This matches the function not only at the corners but also at all boundaries. Here $\mathbf{U}$ is a transfinite interpolant to $\mathbf{F}$. The functions $\Phi_N(\xi)$ and $\Phi_M(\eta)$ given in (17.1.21) and (17.1.22) are referred to as blending functions. The most commonly used blending functions are of the Lagrange polynomial type

$$\begin{aligned}\mathbf{U} = {} & (1-\xi)\mathbf{F}(0,\eta) + \xi\mathbf{F}(1,\eta) + (1-\eta)\mathbf{F}(\xi,\ 0) + \eta\mathbf{F}(\xi,\ 1) \\ & - [(1-\xi)(1-\eta)\mathbf{F}(0,\ 0) + (1-\xi)\,\eta\mathbf{F}(0,\ 1) + \xi\,(1-\eta)\mathbf{F}(1,\ 0) + \xi\eta\mathbf{F}(1,\ 1)]\end{aligned} \tag{17.1.24}$$

For a quadratic variation of boundaries, the blending function $\wp(\xi)$ and $\wp(\eta)$ can simply be replaced by the quadratic Lagrange polynomials.

The following rules are applied in choosing the transfinite interpolation functions:

(1) Pick four points on Γ and identify these as being the images of the four corners of $\overline{\Gamma}$.
(2) These four points separate $\overline{\Gamma}$ into four curve segments which we identify as being the graphs of the four vector valued functions $\mathbf{F}(0,\eta)$, $\mathbf{F}(1,\eta)$, $\mathbf{F}(\xi,0)$, and $\mathbf{F}(\xi,1)$, that is, the four segments of the boundary of Ω are defined to be the images of the four sides of $\overline{\Omega}$.
(3) Use the formulas of $\mathbf{F}(0,\eta)$, $\mathbf{F}(1,\eta)$, $\mathbf{F}(\xi,0)$, $\mathbf{F}(\xi,1)$ in (17.1.24) to define a bilinearly blended transfinite function $\mathbf{U}(\xi,\eta)$, and recall that $\mathbf{U} = \mathbf{F}$ for points (ξ,η) on the perimeter of $\overline{\Omega}$; that is, $\mathbf{U}$ maps the boundary of $\overline{\Omega}$ onto the boundary of Ω.
(4) Test to see if the univalency criteria are satisfied, that is, Jacobian is nonsingular.
(5) Higher order transfinite interpolation functions should be used if necessary (irregular boundaries).

Grid generation for three-dimensional geometries using transfinite interpolation functions was studied by Coons [1967] and extended by Cook [1974]. The procedure includes the surface nodal point mesh generator and volume nodal point mesh generator.

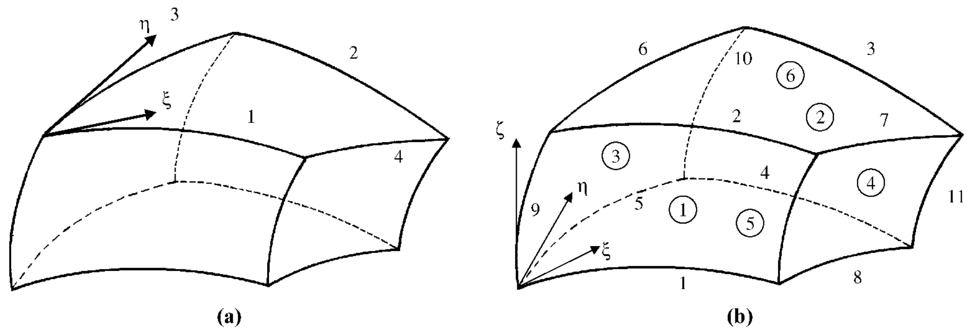


Figure 17.1.8 Surface and volume point mesh generator. (**a**) Surface nodal point mesh generator. (**b**) Volume nodal point mesh generator.

The transfinite interpolation formulas for three-dimensional problems are of the form

$$
\begin{aligned}
\mathbf{U} &= [\wp(\xi) \oplus \wp(\eta) \oplus \wp(\zeta)]\,[\mathbf{F}] \\
&= \wp(\xi)\,[\mathbf{F}(\eta,\zeta)] + \wp(\eta)\,[\mathbf{F}(\xi,\zeta)] + \wp(\zeta)\,[\mathbf{F}(\xi,\eta)] \\
&\quad - [\wp(\xi)\wp(\eta)[\mathbf{F}] + \wp(\eta)\wp(\zeta)\,[\mathbf{F}] + \wp(\zeta)\wp(\xi)[\mathbf{F}] + \wp(\xi)\;\wp(\eta)\wp(\zeta)[\mathbf{F}]]
\end{aligned} \tag{17.1.25}
$$

Consider the coordinate system as shown in Figure 17.1.8 in which the following relations can be established.

Boundary 1:	$\eta = 0,$	$x = f_1(\xi),$	$y = g_1(\xi),$	$z = h_1(\xi)$
Boundary 2:	$\eta = 1,$	$x = f_2(\xi),$	$y = g_2(\xi),$	$z = h_2(\xi)$
Boundary 3:	$\xi = 0,$	$x = f_3(\eta),$	$y = g_3(\eta),$	$z = h_3(\eta)$
Boundary 4:	$\xi = 1,$	$x = f_4(\eta),$	$y = g_4(\eta),$	$z = h_4(\eta)$

These definitions lead to the surface nodal point coordinates (Figure 17.1.8a):

$$
\begin{aligned}
x(\xi,\eta) &= (1-\eta)\,f_1(\xi) + \eta f_2(\xi) + (1-\xi)\,f_3(\eta) + \xi\; f_4(\eta) \\
&\quad - x\,(0,0)\,(1-\xi)(1-\eta) - x\,(1,0)\,\xi(1-\eta) \\
&\quad - x\,(0,1)\,(1-\xi)\eta - x\,(1,1)\,\xi\eta
\end{aligned} \tag{17.1.26}
$$

Similarly for $y(\xi,\eta)$ and $z(\xi,\eta)$.

For the volume nodal point mesh generator, we utilize the ξ, η, ζ coordinates normalized as follows (Figure 17.1.8b):

Boundary edge 1:	$\eta = 0,$	$\zeta = 0,$	$x = f_1(\xi),$	$y = g_1(\xi),$	$z = h_1(\xi)$
Boundary edge 2:	$\eta = 0,$	$\zeta = 1,$	$x = f_2(\xi),$	$y = g_2(\xi),$	$z = h_2(\xi)$
$\vdots$					
Boundary edge 12:	$\eta = 0,$	$\xi = 1,$	$x = f_{12}(\zeta),$	$y = g_{12}(\zeta),$	$z = h_{12}(\zeta)$

With these boundary edge functions, the linearly blended interpolation functions are

$$
\begin{aligned}
x(\xi,\eta,\zeta) &= (1-\eta)(1-\zeta)\,f_1(\xi) + (1-\eta)\zeta f_2(\xi) + \eta\zeta f_3(\xi) + \eta(1-\zeta)\,f_4(\xi) \\
&\quad + (1-\xi)(1-\zeta)\,f_5(\eta) + (1-\xi)\zeta\; f_6(\eta) + \xi\,\zeta\; f_7(\eta) + \xi\,(1-\zeta)\,f_8(\eta) \\
&\quad + (1-\xi)(1-\eta)\;f_9(\zeta) + (1-\xi)\eta\,f_{10}(\zeta) + \xi\,\eta\,f_{11}(\zeta) \\
&\quad + \xi\,(1-\eta)\;f_{12}(\zeta) + c\,(\xi,\eta,\zeta)
\end{aligned} \tag{17.1.27}
$$

where

$$\begin{aligned} c(\xi,\eta,\zeta) = -3[&(1-\xi)(1-\eta)(1-\zeta)\,x(0,0,0) + (1-\xi)(1-\eta)\zeta\,x(0,0,1) \\ &+(1-\xi)\eta(1-\zeta)\,x(0,1,0) + (1-\xi)\eta\zeta\,x(0,1,1) \\ &+\xi(1-\eta)(1-\zeta)\,x(1,0,0) + \xi(1-\eta)\zeta\,x(1,0,1) \\ &+\xi\eta(1-\zeta)\,x(1,1,0) + \xi\eta\zeta\,x(1,1,1)] \end{aligned} \tag{17.1.28}$$

Similarly for $y(\xi,\eta,\zeta)$ and $z(\xi,\eta,\zeta)$.

It is desirable to write (17.1.27) in terms of boundary surfaces:

Boundary surface	1 :	$\eta = 0$,	$x = \overline{f}_1(\xi,\zeta)$,	$y = \overline{g}_1(\xi,\zeta)$,	$z = \overline{h}_1(\xi,\zeta)$
	2 :	$\eta = 1$,	$x = \overline{f}_2(\xi,\zeta)$,	$y = \overline{g}_2(\xi,\zeta)$,	$z = \overline{h}_2(\xi,\zeta)$
	3 :	$\xi = 0$,	$x = \overline{f}_3(\eta,\zeta)$,	$y = \overline{g}_3(\eta,\zeta)$,	$z = \overline{h}_3(\eta,\zeta)$
	4 :	$\xi = 1$,	$x = \overline{f}_4(\eta,\zeta)$,	$y = \overline{g}_4(\eta,\zeta)$,	$z = \overline{h}_4(\eta,\zeta)$
	5 :	$\zeta = 0$,	$x = \overline{f}_5(\xi,\eta)$,	$y = \overline{g}_5(\xi,\eta)$,	$z = \overline{h}_5(\xi,\eta)$
	6 :	$\zeta = 1$,	$x = \overline{f}_6(\xi,\eta)$,	$y = \overline{g}_6(\xi,\eta)$,	$z = \overline{h}_6(\xi,\eta)$

Thus, the boundary surface functions may be written in terms of boundary edge functions:

$$\begin{aligned} x(\xi,\eta,\zeta) = \frac{1}{2}\{&(1-\eta)\overline{f}_1(\xi,\zeta) + \eta\overline{f}_2(\xi,\zeta) + (1-\xi)\overline{f}_3(\eta,\zeta) \\ &+\xi\,\overline{f}_4(\eta,\zeta) + (1-\zeta)\overline{f}_5(\xi,\eta) + \zeta\,\overline{f}_6(\xi,\eta) + 2c(\xi,\eta,\zeta)\} \end{aligned} \tag{17.1.29}$$

where

$$\begin{aligned} \overline{f}_1(\xi,\zeta) = &(1-\zeta)\,f_1(\xi) + \zeta\,f_2(\xi) + (1-\xi)\,f_9(\zeta) + \xi\,f_{12}(\zeta) \\ &-(1-\xi)(1-\zeta)\,x(0,0,0) - (1-\xi)\zeta\,x(0,0,1) \\ &-\xi(1-\zeta)\,x(1,0,0) - \xi\zeta x(1,0,1) \end{aligned} \tag{17.1.30}$$

etc.

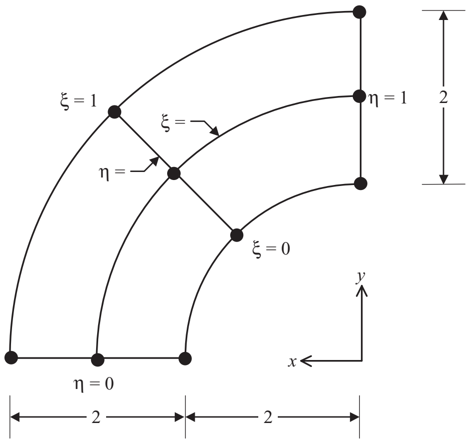
With these coordinate transformation equations, the interior nodal point may be calculated if the interior nodal point can be described in terms of the ξ,η,ζ coordinate system and if the boundary surface functions are known [Cook, 1974].

Example 17.1.6

Repeat Example 17.1.2 using the transfinite interpolation functions (Figure E17.1.6).

The quadratic blending functions are

$$\Phi_N(\xi) = \begin{cases} 2\left(\xi - \frac{1}{2}\right)(\xi - 1) \\ -4\xi(\xi - 1) \\ 2\xi\left(\xi - \frac{1}{2}\right) \end{cases}, \quad \Phi_N(\eta) = \begin{cases} 2\left(\eta - \frac{1}{2}\right)(\eta - 1) \\ -4\eta(\eta - 1) \\ 2\eta\left(\eta - \frac{1}{2}\right) \end{cases}$$



$$x(\xi,\eta) = -(2+2\xi)\cos\frac{\pi}{2}\eta$$

$$y(\xi,\eta) = (2+2\xi)\sin\frac{\pi}{2}\eta$$

$$\mathbf{F}(\xi,\eta) = \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix}$$

Figure E17.1.6 Quarter-circle disk with TIF method.

with the projections

$$\wp_\xi[\mathbf{F}] = \sum_{N=1}^{3} \Phi_N(\xi)\,\mathbf{F}(\xi_N,\eta)$$

$$\wp_\eta[\mathbf{F}] = \sum_{N=1}^{3} \Phi_N(\eta)\,\mathbf{F}(\xi,\eta_N)$$

and the product projections

$$\wp_\xi\,\wp_\eta[\mathbf{F}] = \sum_{N=1}^{3}\sum_{M=1}^{3} \Phi_N(\xi)\,\Phi_M(\eta)\,\mathbf{F}(\xi_N,\eta_M)$$

Thus, the transfinite interpolation functions are

$$\begin{aligned}\mathbf{U}(\xi,\eta) &= \wp_\xi \oplus \wp_\eta[\mathbf{F}] = \wp_\xi[\mathbf{F}] + \wp_\eta[\mathbf{F}] - \wp_\xi\,\wp_\eta[\mathbf{F}] \\ &= \sum_{N=1}^{3} \Phi_N(\xi)\,\mathbf{F}(\xi_N,\eta) + \sum_{M=1}^{3} \Phi_M(\eta)\,\mathbf{F}(\xi,\eta_M) \\ &\quad - \sum_{N=1}^{3}\sum_{M=1}^{3} \Phi_N(\xi)\,\Phi_M(\eta)\,\mathbf{F}(\xi_N,\eta_M)\end{aligned}$$

Thus,

$$\begin{aligned}\begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} &= \sum_{N=1}^{3} \Phi_N(\xi) \begin{bmatrix} x(\xi_N,\eta) \\ y(\xi_N,\eta) \end{bmatrix} + \sum_{M=1}^{3} \Phi_M(\eta) \begin{bmatrix} x(\xi,\eta_M) \\ y(\xi,\eta_M) \end{bmatrix} \\ &\quad - \sum_{N=1}^{3}\sum_{M=1}^{3} \Phi_N(\xi)\,\Phi_M(\eta) \begin{bmatrix} x(\xi_N,\eta_M) \\ y(\xi_N,\eta_M) \end{bmatrix}\end{aligned}$$

The primitive function $\mathbf{F}(\xi, \eta)$ is

$$\mathbf{F}(\xi,\eta) = \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} = \begin{bmatrix} -(2+2\xi)\cos\dfrac{\pi}{2}\eta \\ (2+2\xi)\sin\dfrac{\pi}{2}\eta \end{bmatrix}$$

Thus,

$$\begin{aligned} \mathbf{U}(\xi,\eta) = \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} &= \sum_{N=1}^{3} \Phi_N(\xi) \begin{bmatrix} -(2+2\xi_N)\cos\dfrac{\pi}{2}\eta \\ (2+2\xi_N)\sin\dfrac{\pi}{2}\eta \end{bmatrix} \\ &+ \sum_{M=1}^{3} \Phi_M(\eta) \begin{bmatrix} -(2+2\xi)\cos\dfrac{\pi}{2}\eta_M \\ (2+2\xi)\sin\dfrac{\pi}{2}\eta_M \end{bmatrix} \\ &- \sum_{N=1}^{3}\sum_{M=1}^{3} \Phi_N(\xi)\,\Phi_M(\eta) \begin{bmatrix} -(2+2\xi_N)\cos\dfrac{\pi}{2}\eta_M \\ (2+2\xi_N)\sin\dfrac{\pi}{2}\eta_M \end{bmatrix} \end{aligned}$$

The results are identical to those for the domain vertex method in Example 17.1.2.

Additional discussions on algebraic methods will be presented for surface grid generation in Section 17.3.2. Although algebraic methods are convenient if the geometry can be represented by simple analytical expressions, severe limitations would occur when the computational domain is complicated and suitable functional representation of the geometry is unavailable.

17.2 PDE MAPPING METHODS

Grid generation can be achieved by solving partial differential equations with the dependent and independent variables being the physical domain coordinates and transformed computational domain coordinates, respectively. These PDEs may be of elliptic, hyperbolic, or parabolic form. In general, PDE mapping methods are more complicated than algebraic methods, but provide a smoother grid generation [Thompson, Warsi, and Mastin, 1985].

In the following sections, we shall discuss the basic concepts of elliptic, hyperbolic, and parabolic grid generators, including their advantages and disadvantages.

17.2.1 ELLIPTIC GRID GENERATOR

17.2.1.1 Derivation of Governing Equations

Let us consider a simply connected physical domain and transformed computational domain as shown in Figure 17.2.1. The basic idea stems from the fact that the grid generation in two dimensions is analogous to the solution of Laplace equations for stream function (ψ) and velocity potential function (ϕ).

$$\nabla^2\psi = 0 \tag{17.2.1a}$$

$$\nabla^2\phi = 0 \tag{17.2.1b}$$

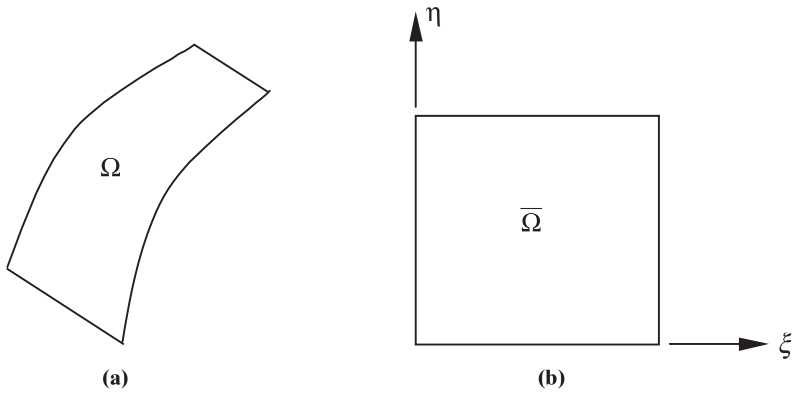


Figure 17.2.1 Simply connected domain. (**a**) Physical domain. (**b**) Transformed computational domain.

Solutions of these equations are shown in Figure 17.2.2. Thus, we may now consider $\psi \rightarrow x$ and $\phi \rightarrow y$ so that we intend to solve

$$\nabla^2 x = \frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^2 x}{\partial \eta^2} = 0 \tag{17.2.2a}$$

$$\nabla^2 y = \frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 y}{\partial \eta^2} = 0 \tag{17.2.2b}$$

Toward this end, we employ the curvilinear coordinates for the equation

$$\nabla^2 \mathbf{r} = 0 \tag{17.2.3}$$

where $\mathbf{r} = (x, y, z)^T$ and ∇ is the curvilinear differential operator defined as

$$\nabla = \mathbf{g}^i \frac{\partial}{\partial \xi_i} \quad (i = 1, 2, 3) \tag{17.2.4}$$

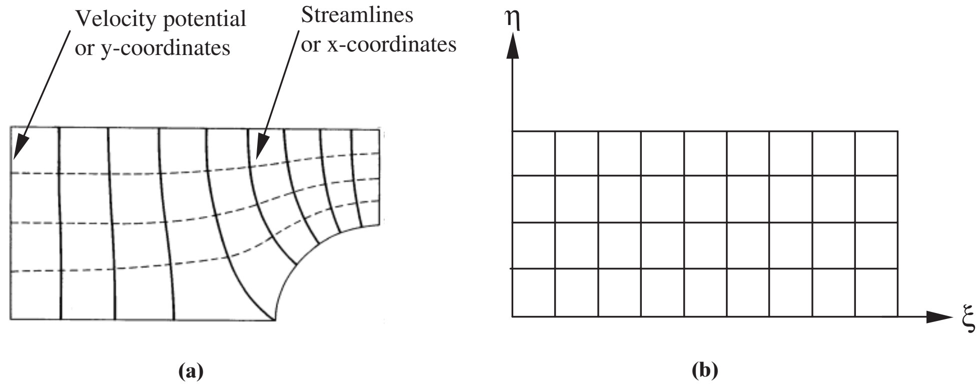


Figure 17.2.2 Analogy of streamlines and velocity potential lines to the generation of x- and y-coordinate grid lines. (**a**) Physical domain. (**b**) Computational domain.

with $\mathbf{g}^i$ being the contravariant tangent vector,

$$\mathbf{g}^i = \frac{\partial \xi_i}{\partial x_m}\mathbf{i}_m \tag{17.2.5}$$

Here x_m refers to the cartesian spatial coordinates and ξ_i denotes the curvilinear coordinates.

Using the standard tensor analysis, we obtain [Chung, 1988]

$$\begin{aligned}\nabla^2\mathbf{r} &= \left(\mathbf{g}^j \frac{\partial}{\partial \xi_j}\cdot \mathbf{g}^i \frac{\partial}{\partial \xi_i}\right)\mathbf{r}\\ &= \mathbf{g}^j\cdot\mathbf{g}^i_{,j}\mathbf{r}_{,i} + \mathbf{g}^j\cdot\mathbf{g}^i\mathbf{r}_{,ij}\\ &= \mathbf{g}^j\cdot(g^{ik}\mathbf{g}_k)_{,j}\mathbf{r}_{,i} + g^{ij}\mathbf{r}_{,ij} = 0\\ &= g^{ij}_{,i}\mathbf{r}_{,j} + g^{ij}\Gamma^k_{ki}\mathbf{r}_{,j} + g^{ij}\mathbf{r}_{,ij}\end{aligned} \tag{17.2.6a}$$

or

$$\nabla^2\mathbf{r} = \frac{1}{\sqrt{g}}\left(\sqrt{g}g^{ij}\mathbf{r}_{,j}\right)_{,i} = 0 \tag{17.2.6b}$$

where the comma denotes partial derivatives with respect to the curvilinear coordinates, Γ^r_{st} represents the Christoffel symbol of the second kind, g^{ij} is the contravariant metric tensor, and g is the determinant of the covariant metric tensor g_{ij}.

Equation (17.2.6) may be recast in the form

$$\nabla^2\mathbf{r} = g^{ij}\mathbf{r}_{,ij} + P^j\mathbf{r}_{,j} = 0 \tag{17.2.7}$$

where P^j is known as the control function

$$\begin{aligned}P^j &= g^{ij}_{,i} + g^{ij}\Gamma^k_{ki} = g^{ij}_{,i} + \frac{1}{g}\frac{\partial g}{\partial g_{ij}}\Gamma^k_{ki}\\ &= \frac{\partial}{\partial \xi_i}\left(\frac{\partial \xi_i}{\partial x_m}\frac{\partial \xi_j}{\partial x_m}\right) + \frac{\partial \xi_i}{\partial x_m}\frac{\partial \xi_j}{\partial x_m}\frac{\partial^2 x_p}{\partial \xi_i\partial \xi_k}\frac{\partial \xi_k}{\partial x_p}\end{aligned} \tag{17.2.8}$$

with $g^{ij}_{,i} = 0$.

Physically, the derivative of the contravariant metric tensor and the product of covariant metric tensor and the Christoffel symbol of the second kind represent the deformation process between the physical domain and the transformed computational domain.

In particular, P^j represents control functions capable of inducing two lines or two points to be pulled (attraction, tension) or pushed away (repelled, compression) as effected by the first derivatives and to be bent or twisted as dictated by second derivatives. This behavior is analogous to the differential equations corresponding to normal and shear strains and flexural (bending and torsion) strains in elasticity.

Notice that in this process of "deformation" or geometric transformation, the Laplace equation (17.2.3) has been changed into a Poisson equation (17.2.7).

For three dimensions, (17.2.7) is expanded as

$$\begin{aligned}&g^{11}\mathbf{r}_{,11} + g^{22}\mathbf{r}_{,22} + g^{33}\mathbf{r}_{,33} + 2g^{12}\mathbf{r}_{,12} + 2g^{23}\mathbf{r}_{,23} + 2g^{31}\mathbf{r}_{,31}\\ &\quad + P^1\mathbf{r}_{,1} + P^2\mathbf{r}_{,2} + P^3\mathbf{r}_{,3} = 0\end{aligned} \tag{17.2.9}$$

with

$$g^{ij} = \frac{1}{g}\frac{\partial g}{\partial g_{ij}} = \frac{1}{g}\frac{\partial}{\partial g_{ij}}\begin{vmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{vmatrix}$$

$$g^{11} = \frac{1}{g}(g_{22}g_{33} - g_{23}g_{32}), \qquad g^{22} = \frac{1}{g}(g_{33}g_{11} - g_{31}g_{13}), \qquad g^{33} = \frac{1}{g}(g_{11}g_{22} - g_{12}g_{21})$$

$$g^{12} = \frac{1}{g}(g_{23}g_{31} - g_{21}g_{33}), \qquad g^{13} = \frac{1}{g}(g_{32}g_{21} - g_{31}g_{22}), \qquad g^{23} = \frac{1}{g}(g_{31}g_{12} - g_{32}g_{11})$$

Similarly for two dimensions,

$$g^{11}\mathbf{r}_{,11} + g^{22}\mathbf{r}_{,22} + 2g^{12}\mathbf{r}_{,12} + P^1\mathbf{r}_{,1} + P^2\mathbf{r}_{,2} = 0 \tag{17.2.10a}$$

or

$$\frac{1}{g}(g_{22}\mathbf{r}_{,11} + g_{11}\mathbf{r}_{,22} - 2g_{12}\mathbf{r}_{,12}) + P^1\mathbf{r}_{,1} + P^2\mathbf{r}_{,2} = 0 \tag{17.2.10b}$$

with

$$g = |g_{ij}| = \begin{vmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{vmatrix} = J^2$$

$$g_{11} = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2, \qquad g_{22} = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2, \qquad g_{12} = \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}$$

where the Jacobian J is given by

$$J = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{vmatrix}$$

Note that the contravariant component P^i is the same as the physical component P_i since the control function is a scalar to be prescribed.

Finally, we obtain from (17.2.10b) two equations, using the notation $x_\xi = \partial x/\partial \xi$, etc:

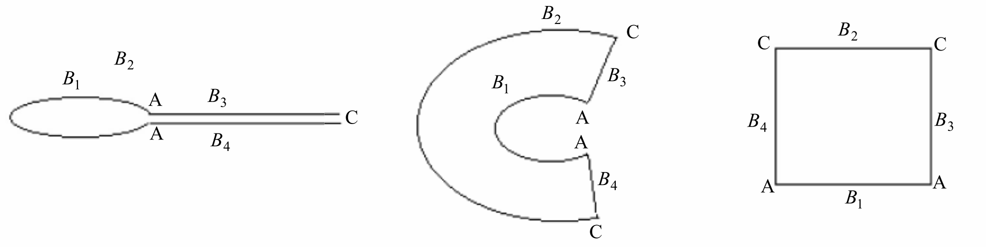
$$(x_\eta^2 + y_\eta^2)x_{\xi\xi} + (x_\xi^2 + y_\xi^2)x_{\eta\eta} - 2(x_\xi x_\eta + y_\xi y_\eta)x_{\xi\eta} = -J^2(Px_\xi + Qx_\eta) \tag{17.2.11a}$$

and

$$(x_\eta^2 + y_\eta^2)y_{\xi\xi} + (x_\xi^2 + y_\xi^2)y_{\eta\eta} - 2(x_\xi x_\eta + y_\xi y_\eta)y_{\xi\eta} = -J^2(Py_\xi + Qy_\eta) \tag{17.2.11b}$$

with $P_1 = P$ and $P_2 = Q$.

Note that these equations are nonlinear and must be solved iteratively to determine the grid coordinate values (x, y). Geometries for this purpose are assumed to be amenable to one-to-one transformation (mapping) between physical domain and computational domain whether simply connected, doubly connected, or multiply connected. A typical doubly connected domain and a multiply connected domain are shown in Figure 17.2.3. Note that transformed computational domain is obtained by introducing the process of unwrapping of the doubly or multiply connected domain. In this way,

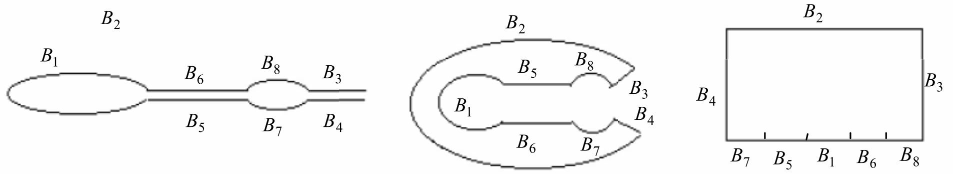


(1) Doubly connected domain and branch cut

(2) Unwrapping of the doubly connected domain

(3) Transformed computational domain

(a)



(1) Physical domain for a multiply connected domain

(2) Unwrapping of a multiply connected domain

(3) Computational domain

(b)

Figure 17.2.3 Doubly and multiply connected domains (O-type). (**a**) Doubly connected domain. (**b**) Multiply connected domain.

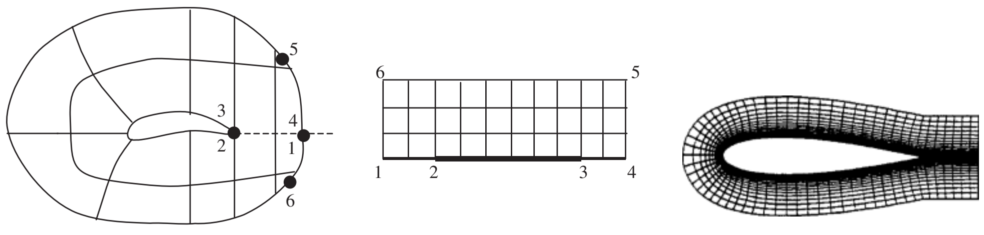


Figure 17.2.4 Doubly and multiply connected domains (C-type). **(a)** Physical domain. **(b)** Transformed computational domain. **(c)** Final mesh.

the arbitrary geometries are made structured into a square (2-D) and a cube (3-D). This type of grid is known as an O-type. Another type is the C-type, which occurs when the exterior boundary is rectangular and grid lines are in the horizontal and vertical directions (Figure 17.2.4).

Prior to the generation of grids by solving the elliptic equations, it is preferable to obtain first preliminary grids using the algebraic method (vertex domain or transfinite interpolations). This information is used as a starter for the elliptic grid generation to obtain smoother grids.

In general, along the airfoil, the grid may become skewed. This is undesirable in computations of flowfields. For a remedy, we seek an orthogonality of grids at the surface [Steger and Sorenson, 1980]. Consider a surface segment ds given by

$$ds = (dx^2 + dy^2)^{\frac{1}{2}} = [(x_\xi d\xi + x_\eta d\eta)^2 + (y_\xi d\xi + y_\eta d\eta)^2]^{\frac{1}{2}} \tag{17.2.12}$$

Here we set $d\xi = 0$ for orthogonality (constant ξ line). Thus, we obtain

$$ds = \left(x_\eta^2 + y_\eta^2\right)^{\frac{1}{2}} d\eta \tag{17.2.13}$$

where

$$x_\eta = s_\eta \frac{(-x_\xi \cos\theta - y_\xi \sin\theta)}{\sqrt{(x_\xi^2 + y_\xi^2)}}$$

$$y_\eta = s_\eta \frac{(-y_\xi \cos\theta + x_\xi \sin\theta)}{\sqrt{(x_\xi^2 + y_\xi^2)}}$$

For $\theta = \pi/2$ (orthogonality) we have

$$x_\eta = \frac{-s_\eta y_\xi}{\sqrt{(x_\xi^2 + y_\xi^2)}}$$

$$y_\eta = \frac{s_\eta x_\xi}{\sqrt{(x_\xi^2 + y_\xi^2)}}$$

$$s_\eta = \left.\frac{\partial s}{\partial \eta}\right|_{\xi=const} = \frac{\Delta s}{\Delta \eta}$$

The finite difference representation for the second derivatives of x and y with respect to η may be written as [Steger and Sorenson, 1980],

$$x_{\eta\eta}(i,\,1) = \frac{-7x_{i,1} + 8x_{i,2} - x_{i,3}}{2\Delta\eta^2} - \frac{3x_\eta(i,\,1)}{\Delta\eta}$$

$$y_{\eta\eta}(i,\,1) = \frac{-7y_{i,1} + 8y_{i,2} - y_{i,3}}{2\Delta\eta^2} - \frac{3y_\eta(i,\,1)}{\Delta\eta}$$

Solutions of elliptic equations (17.2.11) will proceed with central differences for the left-hand side terms (second order derivatives). The first order terms on the right-hand side may be forward-differenced for $P > 0$ and backward-differenced for $Q < 0$.

The control functions, P and Q, are to be used for clustering of grids and are discussed in the following section.

17.2.1.2 Control Functions

In view of the governing equations (17.2.7) or (17.2.11a,b), we may seek to determine the control functions, P and Q, in the form

$$\begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} = \begin{bmatrix} R \\ S \end{bmatrix} \tag{17.2.14}$$

where

$$R = -\frac{1}{J^2}\left[\left(x_\eta^2 + y_\eta^2\right)x_{\xi\xi} + \left(x_\xi^2 + y_\xi^2\right)x_{\eta\eta} - 2(x_\xi x_\eta + y_\xi y_\eta)x_{\xi\eta}\right]$$

$$S = -\frac{1}{J^2}\left[\left(x_\eta^2 + y_\eta^2\right)y_{\xi\xi} + \left(x_\xi^2 + y_\xi^2\right)y_{\eta\eta} - 2(x_\xi x_\eta + y_\xi y_\eta)y_{\xi\eta}\right]$$

Solving for the control functions P and Q,

$$P = \frac{1}{J}(y_\eta R - x_\eta S) \tag{17.2.15a}$$

$$Q = \frac{1}{J}(x_\xi S - y_\xi R) \tag{17.2.15b}$$

The one-dimensional case of (17.2.15) can be shown to be in the form

$$P = -\frac{\partial^2 x}{\partial \xi^2}\bigg/\frac{\partial x}{\partial \xi}$$

which physically corresponds to (17.2.8), representing the deformation process between the physical domain and transformed computational domain, in which the first and second derivatives imply compression or tension and bending or twisting, respectively. Thus, the control functions may be assumed to be of the form

$$P = \hat{P}\left[\alpha(\xi)e^{-\beta(\xi,\,\eta)}\right] \tag{17.2.16a}$$

$$Q = \hat{Q}\left[\alpha(\eta)e^{-\gamma(\xi,\,\eta)}\right] \tag{17.2.16b}$$

Accordingly, we may adopt a form [Thompson et al., 1985]

$$
\begin{aligned}
P(\xi, \eta) = & -\sum_{i=1}^{n} a_i \, |\xi - \xi_i| \exp[-c_i \, |\xi - \xi_i|] \\
& -\sum_{i=1}^{m} b_i \, |\xi - \xi_i| \exp[-d_i \sqrt{(\xi - \xi_i)^2 + (\eta - \eta_i)^2}] \\
Q(\xi, \eta) = & -\sum_{i=1}^{n} a_i \, |\eta - \eta_i| \exp[-c_i \, |\eta - \eta_i|] \\
& -\sum_{i=1}^{m} b_i \, |\eta - \eta_i| \exp[-d_i \sqrt{(\xi - \xi_i)^2 + (\eta - \eta_i)^2}]
\end{aligned}
\tag{17.2.17}
$$

where n and m denote the number of lines of ξ and η of the grid, respectively, with a_i and b_i being the amplification factors, and c_i and d_i being the decay factors.

(1) Amplification factors (a_i, b_i):

$a_i > 0$ lines ξ are attracted to lines, ξ_i

$b_i > 0$ lines ξ are attracted to points (ξ_i, η_i)

Similarly for η coordinates.

(2) Decay factors (c_i, d_i):
These decay factors are to modulate the amplifications from a_i and b_i.

For $a_i < 0$ and $b_i < 0$ the attraction is transformed into a repulsion. Obviously $P = Q = 0$ removes these effects.

In summary, advantages and disadvantages of the elliptic grid generators are as follows:

Advantages

(1) Smooth grid point distribution is achieved. Boundary point discontinuities are smoothed out in the interior domain.
(2) Orthogonality at boundaries can be maintained.

Disadvantages

(1) Computer time is large.
(2) Control functions are often difficult to determine.

Example 17.2.1 Elliptic Grid Generation and Comparison with TFI Method

The results are shown in Figure E17.2.1, with all of them using the 51×31 O-type grid.

17.2.2 HYPERBOLIC GRID GENERATOR

In dealing with an open domain, the hyperbolic grid generator is well suited and efficient. This is because the solution of a hyperbolic differential equation utilizes a marching scheme, which is computationally efficient. There are two methods commonly used to develop a hyperbolic grid generator: one is the cell area (Jacobian) method, and the second is an arc-length method [Steger and Sorenson, 1980].

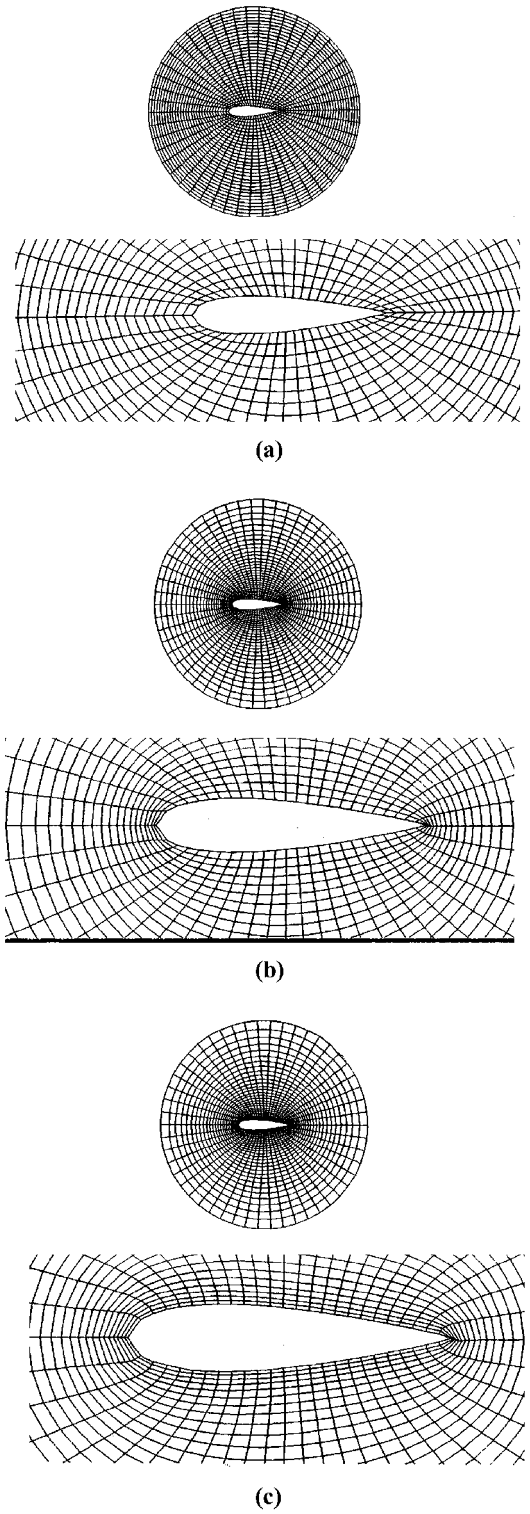


Figure E17.2.1 Elliptic grid generation compared with TFI method. (**a**) Elliptic grid generation without control function. (**b**) Elliptic grid generation with control function. (**c**) Transfinite interpolation approach.

17.2.2.1 Cell Area (Jacobian) Method

In this method, we establish orthogonality of grid lines and a Jacobian relation as follows:

(a) Orthogonality of Grid Lines

$$g_{12} = \mathbf{g}_1 \cdot \mathbf{g}_2 = \frac{\partial x_m}{\partial \xi_1}\mathbf{i}_m \cdot \frac{\partial x_n}{\partial \xi_2}\mathbf{i}_n = 0$$

$$(x_\xi \mathbf{i}_1 + y_\xi \mathbf{i}_2) \cdot (x_\eta \mathbf{i}_1 + y_\eta \mathbf{i}_2) = 0$$

or

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{17.2.18}$$

(b) Jacobian Relation

$$x_\xi y_\eta - x_\eta y_\xi = J(\xi, \eta) \tag{17.2.19}$$

Here (17.2.18) and (17.2.19) represent a system of hyperbolic equations. These equations are nonlinear and may be solved using the standard Newton's iterative scheme, with an algebraic grid to estimate the Jacobian.

Initially, we assume that

$$x_\xi y_\eta = x_\xi^{k+1} y_\eta^k + x_\xi^k y_\eta^{k+1} - x_\xi^k y_\eta^k \tag{17.2.20}$$

Dropping $k+1$ for simplicity, the orthogonality and the Jacobian relation may be written, respectively, as

$$x_\xi x_\eta^k + x_\xi^k x_\eta - x_\xi^k x_\eta^k + y_\xi y_\eta^k + y_\xi^k y_\eta - y_\xi^k y_\eta^k = 0 \tag{17.2.21a}$$

and

$$x_\xi y_\eta^k + x_\xi^k y_\eta - x_\xi^k y_\eta^k - x_\eta y_\xi^k - x_\eta^k y_\xi + x_\eta^k y_\xi^k = J \tag{17.2.21b}$$

We note here that

$$x_\xi^k x_\eta^k + y_\xi^k y_\eta^k = 0 \tag{17.2.22a}$$

$$x_\eta^k y_\xi^k - x_\xi^k y_\eta^k = -J^k \tag{17.2.22b}$$

Thus, (17.2.21a,b) can be rewritten as

$$x_\xi x_\eta^k + x_\xi^k x_\eta + y_\xi y_\eta^k + y_\xi^k y_\eta = 0 \tag{17.2.23a}$$

$$x_\xi y_\eta^k + x_\xi^k y_\eta - x_\eta y_\xi^k - x_\eta^k y_\xi = J + J^k \tag{17.2.23b}$$

Let

$$A = \begin{bmatrix} x_\eta^k & y_\eta^k \\ y_\eta^k & -x_\eta^k \end{bmatrix}, \qquad B = \begin{bmatrix} x_\xi^k & y_\xi^k \\ -y_\xi^k & x_\xi^k \end{bmatrix}, \qquad R = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad H = \begin{bmatrix} 0 \\ J + J^k \end{bmatrix}$$

Then

$$AR_\xi + BR_\eta = H \tag{17.2.24a}$$

or

$$CR_\xi + R_\eta = B^{-1}H \tag{17.2.24b}$$

with

$$C = B^{-1}A = \frac{1}{D}\begin{bmatrix} x_\xi^k x_\eta^k - y_\xi^k y_\eta^k & x_\xi^k y_\eta^k + x_\eta^k y_\xi^k \\ x_\xi^k y_\eta^k + x_\eta^k y_\xi^k & -(x_\xi^k x_\eta^k - y_\xi^k y_\eta^k) \end{bmatrix}$$

$$D = (x_\xi^k)^2 + (y_\xi^k)^2$$

Thus, (17.2.24b) becomes hyperbolic if the eigenvalues of C

$$\lambda = \pm\left[\frac{(x_\eta^k)^2 + (y_\eta^k)^2}{D}\right]^{\frac{1}{2}}$$

are real. For real eigenvalues, we must assure that

$$(x_\xi^k)^2 + (y_\xi^k)^2 \neq 0$$

Now the solution of (17.2.24a) can be obtained with the use of central differences for ξ-derivatives and first order backward differences for η-derivatives. This will result in a block diagonal system, marching in the η-direction with an initial distribution of grid points on the surface and boundary lines given. At the boundaries either forward or backward differences may be employed, with the orthogonality conditions enforced. Further details are found in Steger and Sorenson [1980].

17.2.2.2 Arc-Length Method

In this method, the Jacobian equation (17.2.19) is replaced by the relation defining the tangent line

$$\mathbf{g}_i \cdot \mathbf{g}_i = g_{ii} = g_{11} + g_{22} = F(\xi, \eta) \tag{17.2.25a}$$

or

$$\begin{aligned} F(\xi, \eta) &= (x_\xi \mathbf{i}_1 + y_\xi \mathbf{i}_2) \cdot (x_\xi \mathbf{i}_1 + y_\xi \mathbf{i}_2) + (x_\eta \mathbf{i}_1 + y_\eta \mathbf{i}_2) \cdot (x_\eta \mathbf{i}_1 + y_\eta \mathbf{i}_2) \\ &= x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2 \end{aligned} \tag{17.2.25b}$$

This relation may also be obtained by

$$ds^2 = dx^2 + dy^2 \tag{17.2.26a}$$

which represents an arc-length

$$ds^2 = (x_\xi d\xi + x_\eta d\eta)^2 + (y_\xi d\xi + y_\eta d\eta)^2 \tag{17.2.26b}$$

Setting $\Delta\xi = \Delta\eta = 1$, we obtain

$$\Delta s^2 = x_\xi^2 + y_\xi^2 + x_\eta^2 + y_\eta^2 \tag{17.2.27}$$

Equating (17.2.25b) and (17.2.27) leads to

$$F(\xi, \eta) = \Delta s^2 \tag{17.2.28}$$

The arc-length Δs may be specified by the user. For a constant ξ-line, we obtain

$$\Delta s^2 = x_\eta^2 + y_\eta^2 \tag{17.2.29}$$

Linearization and finite difference approximations for (17.2.28) and (17.2.29) can be carried out similarly as in the cell area method.

In summary, it is seen that the hyperbolic grid generation system is less general, although it is much faster than the elliptic generation system. The specification of the cell volume distribution avoids the possible grid line overlapping that otherwise can occur with concave boundaries. Disadvantages include boundary slope discontinuities being propagated into the field, with shocklike solutions possibly resulting in an unsmooth grid generation.

17.2.3 PARABOLIC GRID GENERATOR

The parabolic system provides a compromise between the elliptic and hyperbolic systems:

(a) *Diffusiveness:* Propagation of boundary discontinuities are prevented similarly as in the elliptic system.
(b) *Marching scheme:* Solutions are fast, similar to the hyperbolic systems.

The governing equations are modified from the Poisson equations as [Nakamura, 1991]

$$x_\eta - Ax_{\xi\xi} = S_x \tag{17.2.30a}$$

$$y_\eta - Ay_{\xi\xi} = S_y \tag{17.2.30b}$$

where $A =$ constant and S_x, $S_y =$ source terms.

Here, the source terms act as control functions. Implementations of (17.2.30) are not as convenient as in the case of elliptic and hyperbolic systems, but the solution of a tridiagonal system for (17.2.30a,b) is much faster than the elliptic grid generator. However, orthogonality is not achieved as directly as in the hyperbolic system. Implementation of control functions through the source terms S_x and S_y remains undeveloped.

17.3 SURFACE GRID GENERATION

A surface mesh is a prerequisite for three-dimensional grid generation. Although the surface grid generation is considered a part of the unstructured three-dimensional mesh generation, it is often convenient to obtain the surface grid in a structured configuration using algebraic methods [De Boor, 1972; Bezier, 1986; Farin, 1988] or elliptic PDE methods [Warsi and Koomullil, 1991; Arina and Casella, 1991; Nakamura et al., 1991]. It is possible to combine the algebraic or elliptic PDE approaches in a structured fashion close to the surface with unstructured grids elsewhere away from the surface. Such a scheme is particularly useful in boundary layer flows.

17.3.1 ELLIPTIC PDE METHODS

The elliptic PDE methods for surface grid generation require derivations of governing equations based on the theory of surfaces or differential geometries. A brief review of the theory of differential geometry applicable to surface grid generation is given below [Chung, 1988, p. 229]:

17.3.1.1 Differential Geometry

Consider a reference surface characterized by a curvilinear coordinate system ($\xi^1, \xi^2, \xi^3 = 0$) with an origin located at P by a position vector $\mathbf{r}_o$, as shown in Figure 17.3.1a. Here, the usual practice of writing the curvilinear coordinates in terms of contravariant component ξ^i with indices placed as superscripts will be followed unlike in the previous sections. Let ξ^3 be the distance along the normal to the reference surface ($\xi^3 = 0$) and $\hat{\mathbf{n}}_3 = \mathbf{n}$ be the unit normal vector. An arbitrary point Q on the ξ^3 coordinate is defined by a position vector $\mathbf{r} = x_i\mathbf{i}_i$ where x_i's are the cartesian coordinate ($i = 1, 2, 3$):

$$\mathbf{r} = x_i\mathbf{i}_i = \mathbf{r}_o + \xi^3\hat{\mathbf{a}}_3 = \mathbf{r}_o + \xi^3\mathbf{n} \tag{17.3.1}$$

The tangent base vectors along the curvilinear coordinates $\xi^\alpha(\alpha = 1, 2)$ on the reference surface, often called the middle surface, are represented by the partial derivatives of $\mathbf{r}_o$ with respect to ξ^α:

$$\frac{\partial \mathbf{r}_o}{\partial \xi^\alpha} = \mathbf{r}_{o,\alpha} = \mathbf{a}_\alpha \tag{17.3.2}$$

Here, $\mathbf{a}_\alpha$ is the covariant surface tangent vector. Likewise, the tangent vectors along ξ^α on the arbitrary surface at $\mathbf{r}$ are

$$\frac{\partial \mathbf{r}}{\partial \xi^\alpha} = \mathbf{r}_{,\alpha} = \mathbf{r}_{o,\alpha} + \xi^3\mathbf{n}_{,\alpha} = \mathbf{g}_\alpha \tag{17.3.3}$$

or

$$\mathbf{g}_\alpha = \mathbf{a}_\alpha + \xi^3\mathbf{n}_{,\alpha} \tag{17.3.4a}$$

and

$$\mathbf{g}_3 = \mathbf{a}_3 = \hat{\mathbf{a}}_3 = \mathbf{n} \tag{17.3.4b}$$

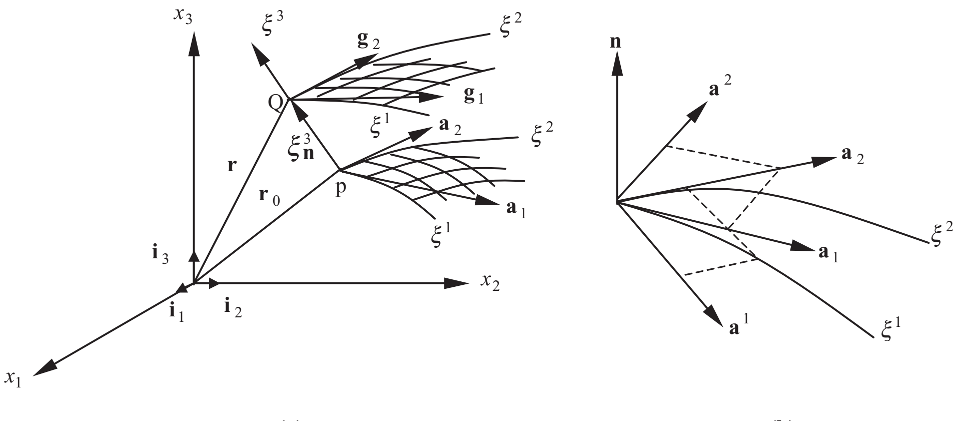


Figure 17.3.1 Surface geometry coordinates. **(a)** Surface geometry. **(b)** Covariant and contravariant components of metric tensors.

The reciprocal base vector or the contravariant component of the tangent vector $\mathbf{a}^i$ has the property (Figure 17.3.1b),

$$\mathbf{a}^i \cdot \mathbf{a}_j = \delta^i_j \tag{17.3.5}$$

and

$$\mathbf{a}^\alpha = a^{\alpha\beta}\mathbf{a}_\beta, \qquad \mathbf{a}_\alpha = a_{\alpha\beta}\mathbf{a}^\beta \tag{17.3.6}$$

in which $a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta$, $a^{\alpha\beta} = \mathbf{a}^\alpha \cdot \mathbf{a}^\beta$ are the covariant and contravariant components of the metric tensor, respectively. Note that $\mathbf{a}^\alpha$ is the contravariant surface tangent vector normal to the ξ^α surface. It also follows that

$$\begin{aligned}
&a^{\alpha\beta} = g^{\alpha\beta}(\xi^1, \xi^2, 0), \qquad a_{\alpha\beta} = g_{\alpha\beta}(\xi^1, \xi^2, 0)\\
&a^{\alpha\beta}a_{\beta\gamma} = \delta^\alpha_\gamma\\
&|a_{\alpha\beta}| = a = g\,(\xi^1, \xi^2, 0)\\
&|a^{\alpha\beta}| = \frac{1}{a}\\
&a^{11} = \frac{a_{22}}{a}, \qquad a^{22} = \frac{a_{11}}{a}, \qquad a^{12} = -\frac{a_{12}}{a}
\end{aligned} \tag{17.3.7}$$

An elemental volume bound by the coordinate surface is given by

$$d\Omega = \mathbf{g}_1 d\xi^1 \times \mathbf{g}_2 d\xi^2 \cdot \mathbf{g}_3 d\xi^3 = \sqrt{g}_{123}\,\mathbf{g}^3 \cdot \mathbf{g}_3 d\xi^1 d\xi^2 d\xi^3 = \sqrt{g} d\xi^1 d\xi^2 d\xi^3 \tag{17.3.8}$$

The curvatures of a surface are defined through scalar products of the base vectors and the derivatives of the base vectors through the Christoffel symbols of the first kind $(\Gamma_{\alpha\beta\gamma})$ and the second kind $\Gamma^\gamma_{\alpha\beta}$:

$$\Gamma_{\alpha\beta\gamma}(\xi^1, \xi^2, 0) = \mathbf{a}_\gamma \cdot \mathbf{a}_{\alpha,\beta} = \Gamma_{\alpha\beta\gamma} \tag{17.3.9a}$$

$$\Gamma^\gamma_{\alpha\beta}(\xi^1, \xi^2, 0) = \mathbf{a}^\gamma \cdot \mathbf{a}_{\alpha,\beta} = -\mathbf{a}_\alpha \cdot \mathbf{a}^\gamma_{,\beta} = \Gamma^\gamma_{\alpha\beta} \tag{17.3.9b}$$

$$\Gamma_{\alpha\beta\gamma} = \frac{1}{2}(a_{\alpha\gamma,\beta} + a_{\beta\gamma,\alpha} - a_{\alpha\beta,\gamma}) \tag{17.3.9c}$$

$$\Gamma_{\alpha\beta\gamma} = a_{\gamma\eta}\,\Gamma^\eta_{\alpha\beta}, \qquad \Gamma^\eta_{\alpha\beta} = a^{\eta\gamma}\,\Gamma_{\alpha\beta\gamma} \tag{17.3.9d}$$

A scalar product of the normal vector $\mathbf{n}$ and the derivatives of the tangent base vectors is known as a curvature tensor:

$$b_{\alpha\beta} = \mathbf{n} \cdot \mathbf{a}_{\alpha,\beta} = -\mathbf{a}_\alpha \cdot \mathbf{n}_{,\beta} = \Gamma_{\alpha\beta3}(\xi^1, \xi^2, 0) = b_{\beta\alpha} \tag{17.3.10a}$$

$$b^\alpha_\beta = \mathbf{n} \cdot \mathbf{a}^\alpha_{,\beta} = -\mathbf{a}^\alpha \cdot \mathbf{n}_{,\beta} = -\Gamma^\alpha_{3\beta}(\xi^1, \xi^2, 0) \tag{17.3.10b}$$

Note that $\mathbf{n} \cdot \mathbf{n}_{,\alpha} = 0$, $\Gamma_{3\alpha3} = \Gamma^3_{3\alpha} = 0$. Combining (17.3.9) and (17.3.10), we obtain

$$\mathbf{a}_{\alpha,\beta} = \Gamma^\gamma_{\alpha\beta}\mathbf{a}_\gamma + b_{\alpha\beta}\mathbf{n} = \Gamma_{\alpha\beta\gamma}\mathbf{a}^\gamma + b_{\alpha\beta}\mathbf{n} \tag{17.3.11a}$$

$$\mathbf{a}^\alpha_{,\beta} = -\Gamma^\alpha_{\beta\gamma}\mathbf{a}^\gamma + b^\alpha_\beta\mathbf{n} \tag{17.3.11b}$$

$$\mathbf{n}_{,\beta} = -b_{\alpha\beta}\mathbf{a}^\alpha = -b^\alpha_\beta\mathbf{a}_\alpha \tag{17.3.11c}$$

In view of (17.3.4) and (17.3.11), it follows that

$$\mathbf{g}_\alpha = \mathbf{a}_\alpha - \xi^3 b^\beta_\alpha \mathbf{a}_\beta = \mathbf{a}_\alpha - \xi^3 b_{\alpha\beta}\mathbf{a}^\beta \tag{17.3.12}$$

$$g_{\alpha\beta} = a_{\alpha\beta} - 2\xi^3 b_{\alpha\beta} + (\xi^3)^2 b_{\alpha\gamma} b^\gamma_\beta \tag{17.3.13a}$$

$$g_{\alpha 3} = 0, \qquad g_{33} = 1 \tag{17.3.13b}$$

The changes in the position vector and the normal vector are given by

$$d\,\mathbf{r}_o = \mathbf{r}_{o,\alpha} d\,\xi^\alpha = \mathbf{a}_\alpha d\,\xi^\alpha \tag{17.3.14a}$$

$$d\,\mathbf{n} = \mathbf{n}_{,\alpha} d\,\xi^\alpha = -b_{\alpha\beta}\mathbf{a}^\beta d\,\xi^\alpha \tag{17.3.14b}$$

The scalar products of (17.3.14) are

$$d\,\mathbf{r}_o \cdot d\,\mathbf{r}_o = ds_o^2 = \mathbf{r}_{o,\alpha} d\,\xi^\alpha \cdot \mathbf{r}_{o,\beta} d\,\xi^\beta = a_{\alpha\beta} d\,\xi^\alpha d\,\xi^\beta \tag{17.3.15a}$$

$$d\,\mathbf{r}_o \cdot d\,\mathbf{n} = \mathbf{a}_\alpha d\,\xi^\alpha \cdot \mathbf{n}_{,\beta} d\,\xi^\beta = -b_{\alpha\beta} d\,\xi^\alpha d\,\xi^\beta \tag{17.3.15b}$$

$$\begin{aligned} d\,\mathbf{n} \cdot d\,\mathbf{n} &= \mathbf{n}_{,\alpha} d\,\xi^\alpha \cdot \mathbf{n}_{,\beta} d\,\xi^\beta = \left(-b^\gamma_\alpha \mathbf{a}_\gamma\right) \cdot \left(-b^\mu_\beta \mathbf{a}_\mu\right) d\,\xi^\alpha d\,\xi^\beta \\ &= b^\gamma_\alpha b^\mu_\beta a_{\gamma\mu} d\,\xi^\alpha d\,\xi^\beta = b^\gamma_\alpha b_{\beta\gamma} d\,\xi^\alpha d\,\xi^\beta = c_{\alpha\beta} d\,\xi^\alpha d\,\xi^\beta \end{aligned} \tag{17.3.15c}$$

Here, $a_{\alpha\beta}$, $b_{\alpha\beta}$, and $c_{\alpha\beta}$ are called the first, second, and third fundamental tensors, respectively.

It can be shown that the second order covariant derivative of any covariant component of a first order tensor is of the form

$$\begin{aligned} A_{i\,|\,jk} &= \left(A_{i\,|\,j}\right)_{,k} - \Gamma^r_{ik} A_{r\,|\,j} - \Gamma^r_{jk} A_{i\,|\,r} \\ &= \left(A_{i,j} - \Gamma^r_{ij} A_r\right)_{,k} - \Gamma^r_{ik}\left(A_{r,j} - \Gamma^s_{rj} A_s\right) - \Gamma^r_{jk}\left(A_{i,r} - \Gamma^s_{ir} A_s\right) \\ &= A_{i,jk} - \left(\Gamma^r_{ij}\right)_{,k} A_r - \Gamma^r_{ij} A_{r,k} - \Gamma^r_{ik} A_{r,j} + \Gamma^r_{ik}\Gamma^s_{rj} A_s - \Gamma^r_{jk} A_{i,r} + \Gamma^r_{jk}\Gamma^s_{ir} A_s \end{aligned} \tag{17.3.16a}$$

Similarly,

$$A_{i\,|\,kj} = A_{i,kj} - \left(\Gamma^r_{ik}\right)_{,j} A_r - \Gamma^r_{ik} A_{r,j} - \Gamma^r_{ij} A_{r,k} + \Gamma^r_{ij}\Gamma^s_{rk} A_s - \Gamma^r_{kj} A_{i,r} + \Gamma^r_{kj}\Gamma^t_{ir} A_t \tag{17.3.16b}$$

Subtracting (17.3.16b) from (17.3.16a) yields

$$\begin{aligned} A_{i\,|\,jk} - A_{i\,|\,kj} &= \Gamma^r_{ik}\Gamma^s_{rj} A_s - \left(\Gamma^r_{ij}\right)_{,k} A_r - \Gamma^r_{ij}\Gamma^s_{rk} A_s + \left(\Gamma^r_{ik}\right)_{,j} A_r \\ &= \left[\left(\Gamma^r_{ik}\right)_{,j} - \left(\Gamma^r_{ij}\right)_{,k} + \Gamma^s_{ik}\Gamma^r_{sj} - \Gamma^s_{ij}\Gamma^r_{sk}\right] A_r \\ &= R^r_{ijk} A_r \end{aligned} \tag{17.3.17}$$

where R^r_{ijk} is a mixed tensor of order four, known as the Riemann-Christoffel tensor of the second kind. Since the left-hand side of (17.3.17) is zero, it follows that

$$R^r_{ijk} = 0 \tag{17.3.18}$$

The associated tensor

$$R_{ijkl} = g_{ir} R^r_{jkl} \tag{17.3.19}$$

is the Riemann-Christoffel tensor of the first kind, which may be written in the form

$$R_{ijkl} = \frac{1}{2}(g_{il,jk} + g_{jk,il} - g_{ik,jl} - g_{jl,ik}) + g^{mn}(\Gamma_{jkm}\Gamma_{iln} - \Gamma_{jlm}\Gamma_{ikn}) \tag{17.3.20}$$

$$R_{ijkl} = -R_{jikl} = -R_{ijlk} = R_{klij} \tag{17.3.21}$$

which implies that R_{ijkl} is skew-symmetric in ij and kl. We also note that there are six different components of R_{ijkl}, namely,

$$R_{3131}, \qquad R_{3232}, \qquad R_{1212}, \qquad R_{3132}, \qquad R_{3212}, \qquad R_{3112}$$

The Riemann-Christoffel tensors for the reference surface with $\xi^3 = 0$ are often of the form

$$R^{\lambda}_{\alpha\beta\gamma} = \Gamma^{\lambda}_{\alpha\gamma,\beta} - \Gamma^{\lambda}_{\alpha\beta,\gamma} + \Gamma^{\mu}_{\alpha\rho}\Gamma^{\lambda}_{\mu\beta} - \Gamma^{\mu}_{\alpha\beta}\Gamma^{\lambda}_{\mu\gamma} + \Gamma^{3}_{\alpha\gamma}\Gamma^{\lambda}_{3\beta} - \Gamma^{3}_{\alpha\beta}\Gamma^{\lambda}_{3\gamma} \tag{17.3.22}$$

or

$$\begin{aligned}
R^{\lambda}_{\alpha\beta\gamma} &= \overline{R}^{\lambda}_{\alpha\beta\gamma} + \Gamma^{3}_{\alpha\gamma}\Gamma^{\lambda}_{3\beta} - \Gamma^{3}_{\alpha\beta}\Gamma^{\lambda}_{3\gamma} = 0 \\
R^{3}_{\alpha\beta\gamma} &= \Gamma^{3}_{\alpha\gamma,\beta} - \Gamma^{3}_{\alpha\beta,\gamma} + \Gamma^{m}_{\alpha,\gamma}\Gamma^{3}_{m\beta} - \Gamma^{m}_{\alpha\beta}\Gamma^{3}_{m\gamma} = 0 \\
\overline{R}^{\lambda}_{\alpha\beta\gamma} &= \Gamma^{3}_{\alpha\beta}\Gamma^{\lambda}_{3\gamma} - \Gamma^{3}_{\alpha\gamma}\Gamma^{\lambda}_{3\beta} = b_{\alpha\beta}\left(-b^{\lambda}_{\gamma}\right) - b_{\alpha\gamma}\left(-b^{\lambda}_{\beta}\right) \\
\overline{R}_{\lambda\alpha\beta\gamma} &= a_{\lambda\mu}\overline{R}^{\mu}_{\alpha\beta\gamma} = b_{\alpha\gamma}b_{\lambda\beta} - b_{\alpha\beta}b_{\lambda\gamma}
\end{aligned}$$

From the symmetry of $\Gamma^{\gamma}_{\alpha\beta}$ and $b_{\alpha\beta}$, we obtain

$$\overline{R}_{\alpha\gamma\beta\gamma} = \overline{R}_{\alpha\beta\gamma\gamma} \quad (\alpha, \gamma \text{ are not summed})$$

and

$$\overline{R}_{1212} = \overline{R}_{2121} = -\overline{R}_{2112} = -\overline{R}_{1221}$$

Hence, every nonzero component of $R_{\alpha\beta\gamma\delta}$ is equal to $\overline{R}_{1212}$ or to $-\overline{R}_{1212}$, and it follows that

$$\overline{R}_{1212} = |b_{\alpha\beta}| = b_{11}b_{22} - b_{12}^2 \tag{17.3.23}$$

We introduce an invariant K, called the Gaussian curvature:

$$K = \frac{\overline{R}_{1212}}{a} = \frac{|b_{\alpha\beta}|}{|a_{\alpha\beta}|} = \left|b^{\alpha}_{\beta}\right| = b^1_1 b^2_2 - b^1_2 b^2_1 \tag{17.3.24}$$

Another important invariant, H, called the mean curvature of the surface, is of the form

$$H = \frac{1}{2}a^{\alpha\beta}b_{\alpha\beta} = \frac{1}{2}b^{\alpha}_{\alpha} = \frac{1}{2}\left(b^1_1 + b^2_2\right) \tag{17.3.25}$$

Since $R^3_{\alpha\beta\gamma}$ also vanishes, we obtain from (17.3.22) that

$$\begin{aligned}
R^{3}_{\alpha\beta\gamma} &= \Gamma^{3}_{\alpha\gamma,\beta} - \Gamma^{3}_{\alpha\beta\gamma} + \Gamma^{m}_{\alpha\gamma}\Gamma^{3}_{m\beta} + \Gamma^{m}_{\alpha\beta}\Gamma^{3}_{m\gamma} \\
&= b_{\alpha\gamma,\beta} - b_{\alpha\beta,\gamma} + \Gamma^{m}_{\alpha\gamma}b_{mb} - \Gamma^{m}_{\alpha\beta}b_{m\gamma}
\end{aligned}$$

Defining $b_{\alpha\gamma|\beta} = b_{\alpha\gamma,\beta} - b_{\alpha\lambda}\Gamma^{\lambda}_{\gamma\beta} - b_{\lambda\gamma}\Gamma^{\lambda}_{\alpha\beta}$, we have

$$b_{\alpha\gamma|\beta} = b_{\alpha\beta|\gamma} \tag{17.3.26}$$

which represents either of two equations, namely,

$$b_{11|2} = b_{12|1} \quad \text{or} \quad b_{21|2} = b_{22|1} \tag{17.3.27}$$

These equations, (17.3.27), are called the Codazzi equations of the surface and are useful in establishing compatibility of deformations.

17.3.1.2 Surface Grid Generation

Returning to (17.3.11a), we write the derivative of the surface tangent base vector as

$$\mathbf{a}_{\alpha,\beta} = \mathbf{r}_{o,\alpha\beta} = \Gamma^{\gamma}_{\alpha\beta}\mathbf{r}_{o,\gamma} + b_{\alpha\beta}\mathbf{n} \tag{17.3.28}$$

where $\mathbf{r}_o$, the position vector to the surface, implies the cartesian coordinate values of the surface grid. Multiplying (17.3.28) by $a^{\alpha\beta}$, we obtain

$$\begin{aligned} a^{\alpha\beta}\mathbf{r}_{o,\alpha\beta} &= a^{\alpha\beta}\Gamma^{\gamma}_{\alpha\beta}\mathbf{r}_{o,\gamma} + a^{\alpha\beta}b_{\alpha\beta}\mathbf{n} \\ &= P^{\gamma}\mathbf{r}_{o,\gamma} + a^{\alpha\beta}b_{\alpha\beta}\mathbf{n} \end{aligned} \tag{17.3.29}$$

where

$$P^{\gamma} = a^{\alpha\beta}\Gamma^{\gamma}_{\alpha\beta} \tag{17.3.30}$$

is the control function. Note also that

$$g^{\alpha\beta}b_{\alpha\beta}|_{surface} = a^{\alpha\beta}b_{\alpha\beta} = b^{\alpha}_{\alpha} \tag{17.3.31}$$

This is known as the principal curvature, which is twice the mean curvature (17.3.26).

It is seen that if the surface is degenerated into a plane, then $b^{\alpha}_{\alpha} = 0$ (zero mean curvature), and (17.3.30) becomes identical to that of a two-dimensional plane geometry as given in (17.2.7).

The governing equation for the surface grid generation takes the form

$$a^{11}\mathbf{r}_{o,11} + a^{22}\mathbf{r}_{o,22} + 2a^{12}\mathbf{r}_{o,12} = P^{1}\mathbf{r}_{o,1} + P^{2}\mathbf{r}_{o,2} + \left(b^{1}_{1} + b^{2}_{2}\right)\mathbf{n} \tag{17.3.32a}$$

or

$$\frac{1}{a}(a_{22}\mathbf{r}_{o,11} + a_{11}\mathbf{r}_{o,22} - 2a_{12}\mathbf{r}_{o,12}) - P_{1}\mathbf{r}_{o,1} - P_{2}\mathbf{r}_{o,2} = \left(b^{1}_{1} + b^{2}_{2}\right)\mathbf{n} \tag{17.3.32b}$$

with

$$\mathbf{r}_{o,11} = \begin{bmatrix} x_{\xi\xi} \\ y_{\xi\xi} \\ z_{\xi\xi} \end{bmatrix}, \qquad \mathbf{r}_{o,22} = \begin{bmatrix} x_{\eta\eta} \\ y_{\eta\eta} \\ z_{\eta\eta} \end{bmatrix}, \qquad \mathbf{r}_{o,12} = \begin{bmatrix} x_{\xi\eta} \\ y_{\xi\eta} \\ z_{\xi\eta} \end{bmatrix}, \qquad \sqrt{a} = \begin{vmatrix} x_{\xi} & x_{\eta} & 0 \\ y_{\xi} & y_{\eta} & 0 \\ z_{\xi} & z_{\eta} & 1 \end{vmatrix}$$

$$a_{11} = x_{\xi}^2 + y_{\xi}^2 + z_{\xi}^2$$
$$a_{22} = x_{\eta}^2 + y_{\eta}^2 + z_{\eta}^2$$
$$a_{12} = x_{\xi}x_{\eta} + y_{\xi}y_{\eta} + z_{\xi}z_{\eta}$$

Principal curvatures are given by

$$a^{\alpha\beta}b_{\alpha\beta} = b^{\alpha}_{\alpha} = \mathbf{n}\cdot\mathbf{a}^{\alpha}_{,\alpha} = -\mathbf{a}^{\alpha}\cdot\mathbf{n}_{,\alpha} = -\Gamma^{\alpha}_{\alpha 3}(\xi^1,\ \xi^2,\ 0) = -a^{\alpha\gamma}\Gamma_{\alpha 3\gamma}$$
$$= -\left(a^{11}\Gamma_{131} + a^{12}\Gamma_{132} + a^{21}\Gamma_{231} + a^{22}\Gamma_{232}\right) \tag{17.3.33}$$

with

$$a^{11} = \frac{a_{22}}{a},\qquad a^{22} = \frac{a_{11}}{a},\qquad a^{12} = -\frac{a_{12}}{a}$$
$$\Gamma_{\alpha 3\beta} = \frac{\partial^2 x_m}{\partial\xi^\alpha\partial\xi^3}\frac{\partial x_m}{\partial\xi^\beta} = \frac{\partial^2 x_1}{\partial\xi^\alpha\partial\xi^3}\frac{\partial x_1}{\partial\xi^\beta} + \frac{\partial^2 x_2}{\partial\xi^\alpha\partial\xi^3}\frac{\partial x_2}{\partial\xi^\beta} + \frac{\partial^2 x_3}{\partial\xi^\alpha\partial\xi^3}\frac{\partial x_3}{\partial\xi^\beta} \tag{17.3.34}$$

Example 17.3.1

Consider surface coordinates (x, y, z) given as

$$z = f(x, y),\quad \text{e.g.,}\quad z = h\sin\frac{\pi x}{A}\sin\frac{\pi y}{B}$$

(a) Surface Area

$$dA = \sqrt{1 + z_x^2 + z_y^2}\,dx\,dy$$

(b) Surface Unit Normal Vector

$$\mathbf{n} = n_i\,\mathbf{i}_i,\qquad \mathbf{n} = \frac{\mathbf{a}_1\times\mathbf{a}_2}{\sqrt{a}}$$

$$n_1 = \frac{-z_x}{\sqrt{1 + z_x^2 + z_y^2}},\qquad n_2 = \frac{-z_y}{\sqrt{1 + z_x^2 + z_y^2}},\qquad n_3 = \frac{1}{\sqrt{1 + z_x^2 + z_y^2}}$$

(c) Surface Length Element

$$ds = \sqrt{\left(1 + z_x^2\right)dx^2 + 2z_x z_y dx dy + \left(1 + z_y^2\right)dy^2}$$

(d) Principal Curvatures

$$b^{\alpha}_{\alpha} = \frac{\left(1 + z_y^2\right)z_{xx} - 2z_x z_y z_{xy} + \left(1 + z_x^2\right)z_{yy}}{\left(1 + z_x^2 + z_y^2\right)^{\frac{3}{2}}}$$

Example 17.3.2

Prolate ellipsoid defined by

$$x = a\cos\zeta,\qquad y = b\sin\zeta\cos\zeta,\qquad z = b\sin\zeta\sin\zeta$$

From (17.3.33) and (17.3.34) we obtain the curvature tensor as

$$b^{\alpha}_{\alpha} = \frac{-a[a^2\sin^2\zeta + b^2(1 + \cos^2\zeta)]}{b\,(a^2\sin^2\zeta + b^2\cos^2\zeta)^{\frac{3}{2}}}$$

The governing equations (17.3.32) may be solved using finite differences or finite elements. Control functions can be selected similarly as discussed in Section 17.2. These functions are set before the solution algorithm begins, either directly through input or by calculation from the boundary point distributions.

17.3.2 ALGEBRAIC METHODS

In algebraic methods, we are not concerned with differential equations, but rather involved in points, curves, elementary surfaces, and the global surface. Earlier works on this subject include Coons [1967], DeBoor [1972], Bezier [1986], Farin [1987, 1988], and George [1991], among others.

17.3.2.1 Points and Curves

Control points which are used in defining some higher order entities (curves and surfaces), points of the curves and surfaces, and the points created by the mesh generator are to be addressed in the algebraic methods.

A point is given either explicitly or is the result of a computation (intersection of two curves). Furthermore, the points given can be present in the surface approximation or else merely serve as supports for information. In this case, they will not exist in this approximation but are used to define the set of points to be created on the surface.

The curves are created from points and relatively complex functions to ensure certain continuity properties (in particular at the junction of two curves). Three types of construction can be established:

(a) The curve is defined by points and passes through them.
(b) The curve is defined by points but does not necessarily pass through them.
(c) The curve is defined by points and additional constraints such as directional derivatives.

We are now confronted with the problem of constructing a piecewise polynomial function of s, of degree n and of class $C^{r_i -1}$ in s_i with $0 \leq r_i \leq n$, such that

$$C(s) = SMQ \tag{17.3.35}$$

where M is the matrix of coefficients of dimension $(n+1) \times (n+1)$, with S and Q being the basis polynomials of the representation (a line vector) and the control (column) vector so that

$$S = [s^n,\ s^{n-1}, \ldots, s, 1] \tag{17.3.36a}$$

$$Q = \left[q_o,\ q_1, \ldots, q_{\frac{n+1}{2}}, \dot{q}_o, \dot{q}_1, \ldots, \dot{q}_{\frac{n+1}{2}}\right] \tag{17.3.36b}$$

To illustrate, we shall examine the Lagrange polynomial, Hermite polynomial, and Bezier curve.

(a) Lagrange Polynomial

The Lagrange polynomials in the context of (17.3.35) are written as

$$C(s) = \sum_{i=0}^{n} \phi_i(s) Q_i \tag{17.3.37}$$

with

$$\phi_i(s) = \prod_{\substack{r=0 \\ r \neq i}}^{n} \frac{s - s_r}{s_i - s_r} \tag{17.3.38}$$

in which $n+1$ specified points are involved and

$$\phi_i(s_j) = \delta_{ij}$$
$$C(s_i) = Q_i$$

With these definitions, the recurrence formula for (17.3.37) becomes

$$C_i^m(s) = \frac{s_{i+m} - s}{s_{i+m} - s_i} C_i^{m-1}(s) + \frac{s - s_i}{s_{i+m} - s_i} C_{i+1}^{m-1}(s) = 0 \tag{17.3.39}$$

with $i = 0, \ldots n - m$, $m = 1, \ldots n$, which is known as the Aitken's algorithm.

Notice that (17.3.35) and (17.3.39) are identical. To see this, let us consider $n = 1$. Then, (17.3.35) becomes

$$C(s) = [s \quad 1] \begin{bmatrix} \dfrac{1}{s_o - s_1} & \dfrac{1}{s_1 - s_0} \\ \dfrac{-s_1}{s_o - s_1} & \dfrac{-s_o}{s_1 - s_0} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = [s \quad 1] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$
$$= (1 - s) Q_1 + s Q_2$$

The same result arises from (17.3.38).

Similarly, for $n = 2$, we obtain

$$C(s) = [s^2 \quad s \quad 1] \begin{bmatrix} \dfrac{1}{(s_o - s_1)(s_o - s_2)} & \dfrac{1}{(s_1 - s_0)(s_1 - s_2)} & \dfrac{1}{(s_2 - s_0)(s_2 - s_1)} \\ \dfrac{-s_1 - s_2}{(s_o - s_1)(s_o - s_2)} & \dfrac{-s_o - s_2}{(s_1 - s_0)(s_1 - s_2)} & \dfrac{-s_o - s_1}{(s_2 - s_0)(s_2 - s_1)} \\ \dfrac{s_1 s_2}{(s_o - s_1)(s_o - s_2)} & \dfrac{s_o s_2}{(s_1 - s_0)(s_1 - s_2)} & \dfrac{s_o s_1}{(s_2 - s_0)(s_2 - s_1)} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

or

$$C(s) = [s^2 \quad s \quad 1] \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$
$$= (2s^2 - 3s + 1) Q_1 + (-4s^2 + 4s) Q_2 + (2s^2 - s) Q_3$$

It is seen that this is the second order Lagrange polynomial representation.

(b) Hermite Polynomial

Proceeding similarly as in the Lagrange polynomial, but with derivatives of Q, we write for $n = 3$,

$$S = [\, s^3 \quad s^2 \quad s \quad 1\,], \qquad Q = [\, q_o \quad q_1 \quad \dot{q}_o \quad \dot{q}_1\,], \qquad M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

representing the cubic Hermite polynomials.

(c) Bezier Curve

An algebraic form of this approximation uses the Bernstein polynomials of the form

$$C(s) = \sum_{i=0}^{n} c_i^n s^i (1-s)^{n-i} Q_i \tag{17.3.40}$$

with

$$c_i^n = \frac{n!}{(n-i)!\,i!} \tag{17.3.41}$$

for which the matrix of coefficient takes the form

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{with} \quad S = [\, s^3 \quad s^2 \quad s \quad 1\,], \quad Q = [\, q_o \quad q_1 \quad q_2 \quad q_3\,] \tag{17.3.42}$$

These polynomials can be shown to be identical to the cubic Hermite polynomials if we consider a third degree polynomial satisfying the following four constraints:

$$C_i(0) = Q_i, \qquad C_i(1) = Q_{i+1}$$
$$\dot{C}_i(0) = \dot{Q}_i, \qquad \dot{C}_i(1) = \dot{Q}_{i+1}$$

To this end, we set

$$C_i(s) = a_i + b_i s + c_i s^2 + d_i s^3$$

and obtain

$$Q_i = a_i$$
$$Q_{i+1} = a_i + b_i + c_i + d_i$$
$$\dot{Q}_i = b_i$$
$$\dot{Q}_{i+1} = b_i + 2c_i + 3d_i$$

This gives

$$S = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_i & Q_{i+1} & \dot{Q}_i & \dot{Q}_{i+1} \end{bmatrix}, \quad M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{17.3.43}$$

Here, $C_i(s) = SMQ$ represents the cubic Hermite polynomial.

Another example is given for the case involving four consecutive points. (Q_{i-1}, Q_i, Q_{i+1}, Q_{i+2}) with a cubic polynomial mapped between [0, 1] and the curve passing through Q_i and Q_{i+1} and its tangent at these points being fixed to the value $\dot{Q}_i = \frac{1}{2}(Q_{i+2} - Q_i)$. These conditions lead to

$$Q_i = a_i$$
$$Q_{i+1} = a_i + b_i + c_i + d_i$$
$$Q_{i+1} - Q_{i-1} = 2b_i$$
$$Q_{i+2} - Q_i = 2b_i + 4c_i + 6d_i$$

and

$$S = [s^3 \quad s^2 \quad s \quad 1], \quad Q = \begin{bmatrix} q_{i-1} & q_i & q_{i+1} & q_{i+2} \end{bmatrix}, \quad M = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \tag{17.3.44}$$

This is known as the Catmull-Rom form.

A general form of (17.3.44), called the cardinal spline basis, is given as

$$M = \begin{bmatrix} -\alpha & 2-\alpha & \alpha-2 & \alpha \\ 2\alpha & \alpha-3 & 3-2\alpha & -\alpha \\ -\alpha & 0 & \alpha & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{17.3.45}$$

where $\alpha = 1$ leads to the Catmull-Rom form.

Similarly, the coefficient matrices for B-spline and Beta spline forms are given as follows:

B-Spline

$$M = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 1 \end{bmatrix} \tag{17.3.46}$$

Beta Spline

$$M = \frac{1}{\Delta}\begin{bmatrix} -2\beta_1^3 & 2(\beta_2 + \beta_1^3 + \beta_1^2 + \beta_1) & -2(\beta_2 + \beta_1^2 + \beta_1 + 1) & 2 \\ 6\beta_1^3 & -3(\beta_2 + 2\beta_1^3 + 2\beta_1^2) & 3(\beta_2 + 2\beta_1^2) & 0 \\ -6\beta_1^3 & 6(\beta_1^3 - \beta_1) & 6\beta_1 & 0 \\ 2\beta_1^3 & \beta_2 + 4(\beta_1^2 + \beta_1) & 2 & 0 \end{bmatrix} \tag{17.3.47}$$

with $\Delta = \beta_2 + 2\beta_1^3 + 4\beta_1^2 + 4\beta_1 + 2$. For $\beta_1 = 1$, $\beta_2 = 0$ the classic B-spline form is found there, β_1 (the bias) and β_2 (the tension) are introduced in B-spline form in order to control the curve by moving it toward the control points.

17.3.2.2 Elementary and Global Surfaces

The different methods to construct a curve can be extended to a surface by using tensor product in two or three directions.

$$C(s,u) = SMQ(u) \tag{17.3.48}$$

with

$$Q(u) = UMQ_{(ij)} \tag{17.3.49}$$

where i denotes the dependence with respect to parameter s and j that with respect to parameter u, U is the equivalent in u to S (i.e., the associated basis polynomial), and $Q_{(ij)}$, is a $(n+1)\times(n+1)$ matrix constructed on control points. Substituting (17.3.49) into (17.3.48) yields

$$\overline{C}(s,u) = SMQ_{(ij)}^T M^T U^T \tag{17.3.50a}$$

or

$$\overline{C}(s,u) = \sum_{i=0}^{n}\sum_{j=0}^{m} b_{ij} s_i u_j \tag{17.3.50b}$$

where b_{ij} depends on the method selected (n and m being arbitrary). In case of the Bezier form, $\overline{C}(s,u)$ can be expressed in terms of the Bernstein polynomials:

$$B_i^n(s) = C_i^n s^i (1-s)^{n-i} \tag{17.3.51}$$

so that

$$\overline{C}(s,u) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(s) B_j^m(s)\, Q_{(ij)} \tag{17.3.52}$$

This represents the surface by Bezier patches leading to quadrilateral elements (Figure 17.3.2a). To produce triangular patches (Figure 17.3.2b), we use the polynomials

$$B_{ijk}^n(r,\, s,\, t) = \frac{n!}{i\,!\,j!\,k!} r^i s^j t^k, \quad i+j+k=n \tag{17.3.53}$$

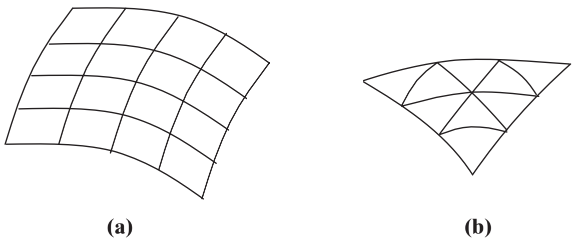


Figure 17.3.2 Quadrilateral and triangular element patches. **(a)** Quadrilateral element. **(b)** Triangular element.

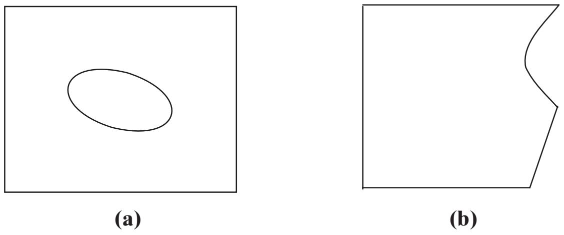


Figure 17.3.3 Basis patches. (**a**) Hole inside. (**b**) Deformed corner.

where r, s, t denote the barycentric coordinates, Q_{ijk}, representing a series of points representing a triangular network acting as control points.

A global surface is defined as a series of elementary surfaces. Let us consider two cases: (a) a patch with a hole (Figure 17.3.3a) and (b) a patch deformed at one of its corners (Figure 17.3.3b) for which the region of interest is the zone remaining or its complementary.

Let us consider the Catmull-Rom form of the third degree. The surface is described by a series of patches which are constructed using the control point s and the representation chosen. Each patch is joined to its neighbors with the properties present in this representation.

The surface is defined by a coarse grid, derived from the control points used in the case of the description for the Catmull-Rom method. Thus, to define a grid on the surface which includes n_1 divisions in one direction (s) and n_2 in the other (u), one has to provide $n_1 + 3$ series of $n_2 + 3$ control points. The end points only serve to define the shape of the surface by its boundary. Each patch is then determined by its four points and by the points of its neighbors (Figure 17.3.4), passing through the four points which define it and sharing a continuous junction with its neighboring patches as the points of the latter are taken into account in its definition.

17.3.2.3 Surface Mesh Generation

Assume that the surface under consideration is known by a series of elementary surfaces of the types described above. The global mesh can be seen as the union of the different elementary surface meshes. In order to obtain a valid mesh, we ensure that any point common to two patches must be defined in the same way in each patch that contains it. This implies that the lines bordering each patch are meshed in the same way in all the patches containing them.

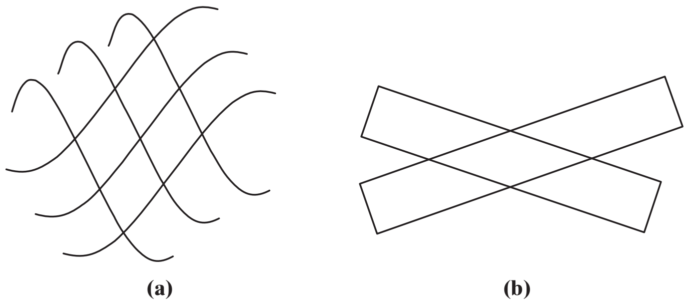


Figure 17.3.4 Surface description by patches. (**a**) Curved line patches. (**b**) Straight line patches.

Toward this end for each boundary line when considering a patch processed previously, we now perform discretization, compatible with the previously meshed lines. The global surface is obtained using, for example, the Catmull-Rom form of the third degree.

Example 17.3.3

Describe in detail the implementation of a Bezier curve for surface grid generation.

(1) Initial Step

A global surface is obtained from the union of elementary surfaces or patches. For the Catmull-Rom method, the surface is defined by a coarse grid of patches derived from user-specified control points. To define a grid on the surface which has n divisions in the s-direction and m divisions in the u-direction requires $(n+2) \times (m+2)$ control points. The extra end points serve to define the shape of the surface at its boundary. Each Bezier patch is then determined from its four points (the vertices of the quadrilateral element) and the points in its corresponding neighbors.

(2) Valid Mesh

In order to obtain a valid mesh, we must ensure that any point which is common to two patches is defined in the same way for each patch that contains it. This implies that the lines bordering each patch are meshed the same way in all patches containing them.

(3) Creation of Mesh

When all the lines forming the boundaries of the patches have been discretized, the mesh of all the patches is created as follows:

(3-1) If the patch is quadrilateral or triangular and if none of its boundary lines contains intermediary points, then it is considered an element of the mesh.

(3-2) If the patch is quadrilateral or triangular and if all of its boundary lines contain a given number of intermediary points compatible with a regular partitioning, then it is meshed by a suitable method. For example, use the Catmull-Rom method as follows:

Step 1

```
do for $i = 0$ and $i = N$
do for $j = 0$ to $M$, do
• Consider the location in $R^3$ of node $(i, j)$ (located on a boundary line previously meshed)
• Compute values of the associated parameters
end do for $j = 0$ to $M$;
end do for $i = 0$ and $i = N$
for $j = 0$ and $j = M$, do
for $i = 0$ to $N$, do
• Consider the location in $R^3$ of node $(i, j)$
• Compute values of the associated parameters
end do for $i = 0$ to $N$
end do for $i = 0$ and $i = M$;
end do for Step 1:
```

Step 2

```
Create the mesh in space $(t, u)$ of the unit square $[0, 1] \times [0, 1]$ as a function
    of its boundary discretazation
end do for Step 2
```

Step 3

```
do, for $i = 1$ to $N - 1$, do
for $j = 1$ to $M - 1$, do
• *Definition of Connectivity*: the vertices of the element created have the following couples as vertex numbers: $(i, j), (i + 1, j), (i + 1, j + 1)$ and $(i, j + 1)$, each of which will have a global number associated with it
• *Compute Vertex Location*: evaluate $t$ and $u$ corresponding to $i$ and $j$ and find the location using $C(t, u) = \sum_{i=0}^{n} \sum_{j=0}^{m} b_{ij} t^i u^j$ with $P_{ij}$ the matrix of control points
end do for $j = 1$ to $M - 1$;
end do for $i = 1$ to $N - 1$;
```

(3-2) Any two-dimensional method can be implemented in (t, u) space, the problem being to know if the mapping in R^3 of mesh points in the space of parameters is valid, close to the surface, and good quality.

Example 17.3.4

This example is based on the surface grid generation via Bezier curve polynomials [Warsi, 1992]. Figure E17.3.4a shows a generic forebody surface grid of an aircraft, with the number of points increased in the canopy region (Figure E17.3.4b). Discontinuities in a surface may be handled easily by selecting appropriate patches so that spline constructions do not occur at the discontinuities.

Figure E17.3.4c shows a set of curves generated for a generic re-entry vehicle as an example of curve generation and editing facilities. Since actual surface definition data are not available, each of the curves shown is generated with the curve segment generator in the program. The majority of the curves are generated using the Bezier generator, and the complex curves at the trailing edge of the wing are generated by appending multiple Bezier curves, elliptical, circular and straight line segments.

Figure E17.3.4d shows the initial surface grid generated for the generic re-entry vehicle using the previously designed curves shown in Figure E17.3.4c, and the surface generation facilities of splining cross-sectional data and transfinite interpolation with specified edge curves. The final surface grid for the generic pre-entry vehicle after using the surface editing facilities is shown in Figure E17.3.4e. Notice that grid distributions are now much smoother and point resolution in areas of interest is better, while the original surface geometry is maintained.

A sample far-field boundary and blocking arrangement for the entry vehicle after performing a domain decomposition is shown in Figure E17.3.4f, with the mesh on selected block faces around the re-entry vehicle shown in Figure E17.3.4g. Figure E17.3.4h shows a global view of the surface grids generated for a win/pylon/lead configuration.

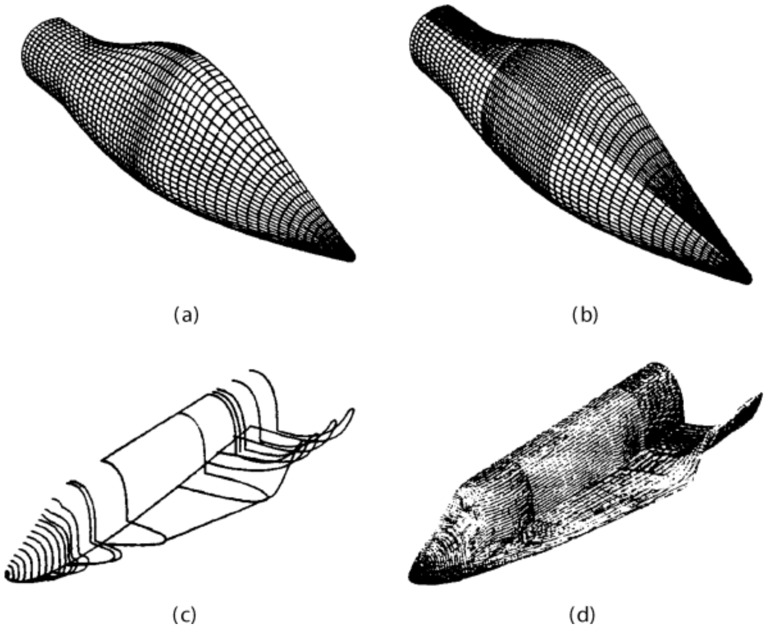


Figure E17.3.4 Surface grid generation via Bezier curve polynomials [Warsi, 1992]. **(a)** Generic forebody surface grid. **(b)** Enrichment of grid points in canopy region of forebody surface. **(c)** Surface definition curves for generic re-entry vehicle. **(d)** Initial surface grid for generic re-entry vehicle.

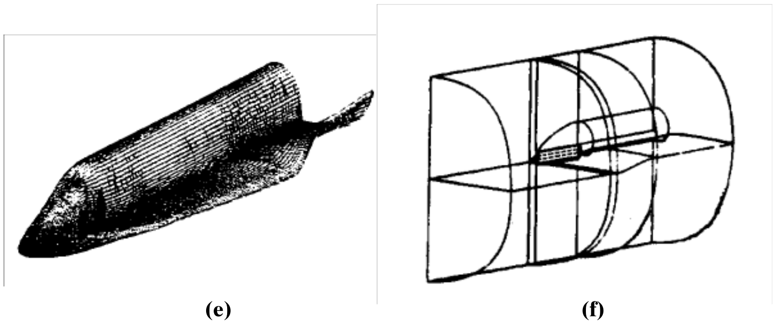
Figure E17.3.4i shows some details of the surface grids in the wing/pylon interaction region.

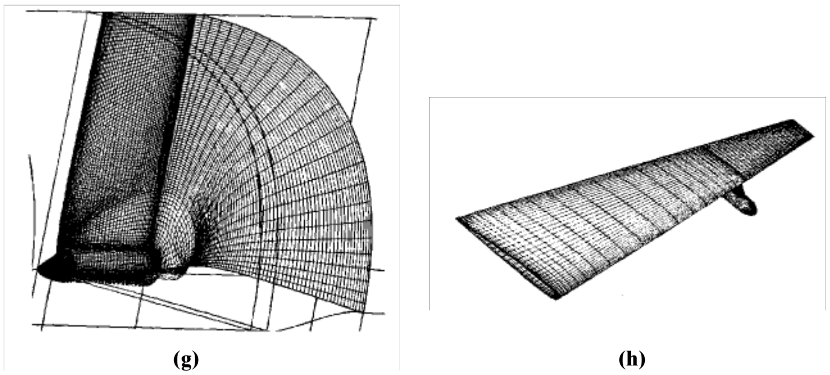
17.4 MULTIBLOCK STRUCTURED GRID GENERATION

An efficient approach to the grid generation in complex domain, particularly in three-dimensional geometries, is to establish block configurations initially, construct the grid with increasing details, and make modifications on an existing grid with minimum restrictions. Such a sequential procedure is known as multiblock grid generation, which is conducive to parallel processing to be discussed in Section 20.4. Ecer, Spyropoulos, and Maul [1985] presented the multiblock structured finite element grid generation. Brief descriptions of this approach are given below.

A convenient way of generating the finite element multigrid system is to use isoparametric elements in 2-D or 3-D. Linear, quadratic, or cubic interpolation functions may be used to divide the domain roughly by a desired number of blocks, each of which will then be subdivided into as many elements as required for computation. For geometries with a pointed nose or leading and trailing edges of an airfoil, it is necessary to use wedge type elements such as a triangle collapsed from a quadrilateral element for 2-D (see Example 9.3.5) or the counterpart for 3-D with a tetrahedron collapsed from a hexahedron.

Consider the modeling of a complete aircraft geometry as an example. The geometric modeling package provides information in three steps as shown in Figure 17.4.1a


(e) (f)


(g) (h)

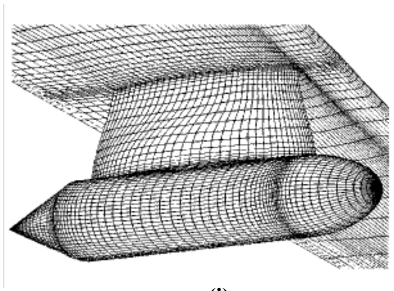

(i)

Figure E17.3.4 (*continued*) (**e**) Final surface grid for generic re-entry vehicle. (**f**) Example farfield and blocking arrangement for re-entry vehicle. (**g**) Grids on block faces around re-entry vehicle. (**h**) Surface grids for wing/pylon/load configuration. (**i**) Detail of surface grids in the wing/pylon intersection region.

by digitizing points on several sections [Figure 17.4.1a(1)]. These digitized points are connected by a series of B-splines and through Boolean operations [Figure 17.4.1a(2,3)].

The next step is to describe the blocks surrounding the aircraft. It is convenient to define four regions with two of them [I and II of Figure 17.4.1b(1)] for radial directions and another set of two regions [III and IV of Figure 17.4.1b(2)] for two pairs of wings along the aircraft. As a result, the block structured for the developed aircraft grid is obtained as shown in Figure 17.4.1b(3) with the coordinate system of radial, tangential, and longitudinal directions. It is shown that there are 336 blocks with the inner two subvolumes having smaller number of blocks than the outer subvolumes. Figure 17.4.1c shows the resulting grid generated around the nose.

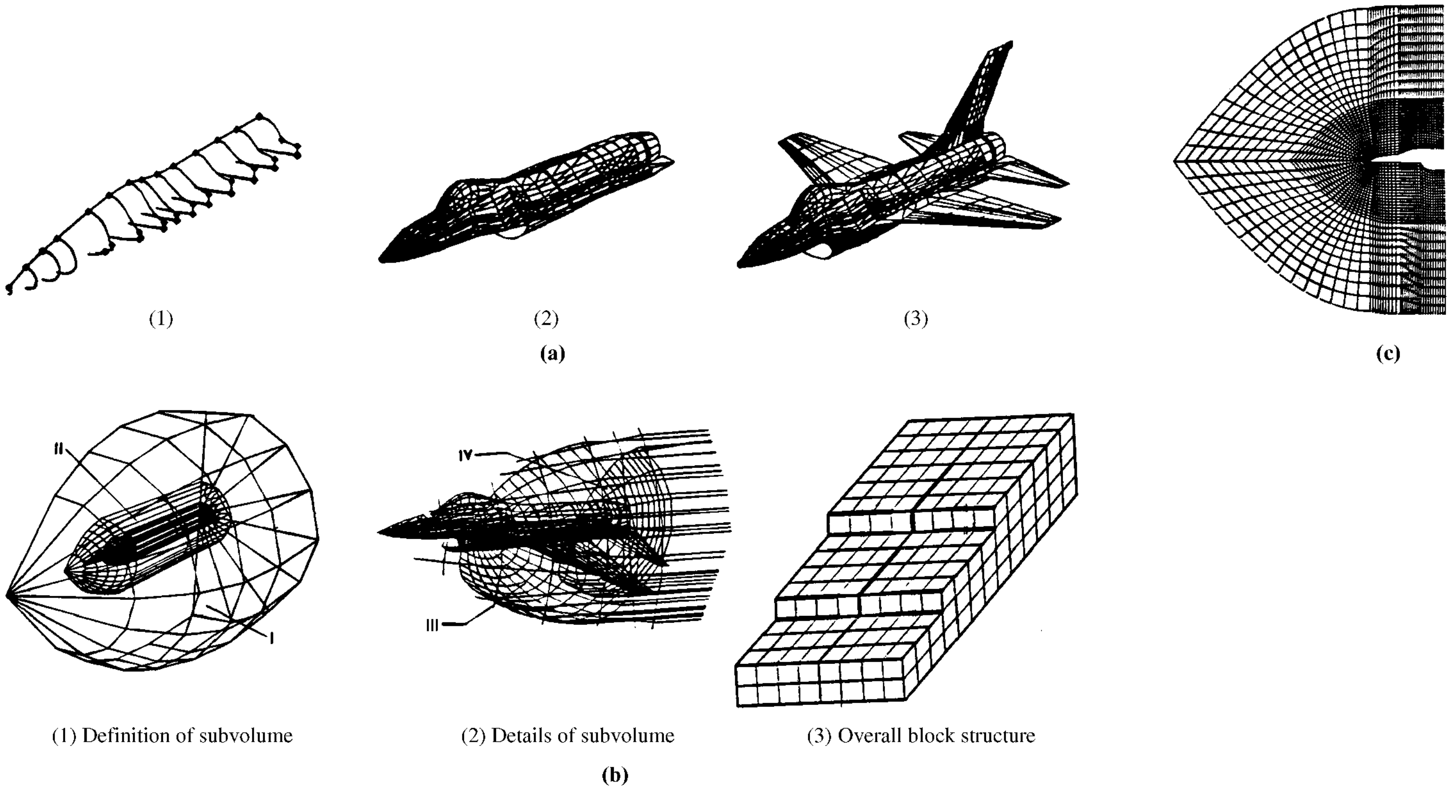


Figure 17.4.1 Multiblock structured grid generation [Ecer, 1986]. (**a**) Procedure of describing the aircraft geometry. (**b**) Geometric description of block structured around the aircraft. (**c**) Across section of the final grid for part of the aircraft geometry.

17.5 SUMMARY

Algebraic methods and PDE mapping methods constitute the two major schemes used in the structured grid generation primarily for FDM applications. The algebraic methods consist of domain vertex methods and transfinite interpolation methods, whereas the PDE mapping methods require solutions of elliptic, hyperbolic, or parabolic partial differential equations. We examined the methods of surface grid generation, using both elliptic PDE methods and algebraic methods.

It was also shown that the use of multiblock structured grid generation is particularly effective in FEM applications. In some complex geometries, however, unstructured grid generation is advantageous, particularly in terms of adaptive mesh. This subject will be presented in the next chapter.

REFERENCES

Arina, R. and Casella, M. [1991]. A Harmonic Grid Generation Technique for Surfaces and Three-Dimensional Regions. In *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*. A. S. Arcilla et al. (eds.). North Holland, 935–46.

Bezier, P. [1986]. Courbes et surfaces, mathematiques et CAO. 4, Hermes.

Chung, T. J. [1988]. *Continuum Mechanics*. Englewood Cliffs, NJ: Prentice-Hall.

Cook, W. A. [1974]. Body oriented coordinates for generating 3-Dimensional meshes. *Int. J. Num. Meth. Eng*., 8, 27–43.

Coons, S. A. [1967]. Surfaces for Computer-Aided Design of Space Forms. Project MAC, Technical Rep. MAC-TR 44 MIT, MA, USA, Design Div., Dept. Mech. Eng., Available from: Clearinghouse for Federal Scientific-Technical Information, National Bureau of Standards, Springfield, VA, USA.

De Boor, C. [1972]. On calculating with B-splines. *J. Approx. Theory*, 6, 50–62.

Ecer, A., Spyropoulos, J., and Maul, J. D. [1985]. A three-dimensional, block-structured finite element grid generation scheme. *AIAA J.*, 23, 10, 1483–90.

Farin, G. [1987]. *Geometric Modeling: Algorithms and New Trends*. Philadelphia: *SIAM*.

———. [1988]. *Curves and Surfaces for Computer Aided Geometric Design*. New York: Academic Press.

Gordon, W. J. and Hall, C. A. [1973]. Construction of curvilinear coordinate systems and applications to mesh generation. *Int. J. Num. Meth. Eng*., 7, 461–77.

Nakamura, S., Fradl, D. D., Spradling M. L., and Kuwahara, K. [1991]. Mapping of curved surfaces onto a side boundary of the three-dimensional computational grid using two elliptic partial differential equations. In A. S. Arcilla et al. (eds.). *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*. New York: North Holland.

Steger, J. L. and Sorenson, R. L. [1980]. Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates, Numerical Grid Generation Techniques. NASA Conference Publication 2166, 463–78.

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

Warsi, S. [1992]. Algebraic surface grid generation in three-dimensional space. In Software Systems for Surface Modeling and Grid Generation, NASA Conference Publication 3143, Hampton: NASA Langley Research Center.

Warsi, Z. U. A. and Koomullil, G. P. [1991]. Application of spectral techniques in surface grid generation. In A. S. Arcilla et al. (eds.). *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*. North Holland, 955–64.