DEPARTMENT OF AEROSPACE ENGINEERING

Faculty Name	:	**Dr.A.Arun Negemiya, ASP/ Aero**	Academic Year	:	**2024-2025 (Even)**
Year & Branch	:	**III AEROSPACE**	Semester	:	**VI**
Course	:	**19ASB304 - Computational Fluid Dynamics for Aerospace Application**			

UNIT I - FUNDAMENTAL CONCEPTS

GRID GENERATION

The governing partial differential equations (PDEs) of the fluid mechanics are solved numerically by converting the partial equations into appropriate finite difference expressions which are used to rewrite PDEs as algebraic equations. These Finite Difference Equations (FDEs) are solved at the discrete points within the domain of interest; these discrete points are called Grids.

Structured Grid - The computational domain selected to be rectangular where the interior grid points are distributed along grid lines. The grid points can be identified easily concerning the appropriate grid lines. This type of grid is known as the ***structured grid.***

Unstructured Grid – The grid system that is constructed where the grid points cannot be associated with the orderly defined grid lines. This type of grid system is known as the ***unstructured grid.***

Initial and Boundary Conditions

In order to obtain a unique solution of a PDE, a set of supplementary conditions must be provided to determine the arbitrary functions which result from the integration of the PDE, The supplementary conditions are classified as boundary and initial conditions.

An initial condition is a requirement for which the dependent variable is specified at some initial state.

A boundary condition is a requirement that the dependent variable or its derivative must satisfy on the boundary of the domain of the PDE.

Various types of boundary conditions that will be encountered are,

1. **The Dirichlet boundary condition** – If the dependent variable along with the boundary is prescribed, it is known as the Dirichlet type.
2. **The Neumann boundary condition** – If the normal gradient of the dependent variable along with the boundary is specified, it is called the Neumann type.
3. **The Robin boundary condition** – If the imposed boundary condition is a linear combination of the Dirichlet and Neumann types, it is known as the Robin type.
4. **The Mixed boundary condition** – The boundary condition along a certain portion of the boundary is the Dirichlet type and on another portion of the boundary, a Neumann type. This type is known as a mixed boundary condition.

Structured Grids

Structured Grid - The computational domain selected to be rectangular in shape where the interior grid points are distributed along grid lines. The grid points can be identified easily with reference to the appropriate grid lines. This type of grid is known as the ***structured grid.***

The generation of grid with uniform spacing is the simplest within a rectangular physical domain. Grid points may be specified as coincident with the boundaries of the physical domain, thus making specification of boundary conditions considerably less complex. Unfortunately the majority of the physical domain of interest are nonrectangular, Therefore imposing rectangular computational domain on such physical domain will require some sort of interpolation for the implementation of the boundary condition. Since the boundary condition have a dominant influence on the solution of the equation, such an interpolation causes inaccuracies at the place of greatest sensitivity.

Types and Transformations

A transformation from physical space to computational space is introduce to overcome some of the difficulties such as unequal grid spacing near the boundaries and inaccuracies at the place of sensitivity. This transformation is accomplished by specifying a generalized coordinate system which will map the nonrectangular grid system in the physical space to a rectangular uniform grid spacing in the computational space.

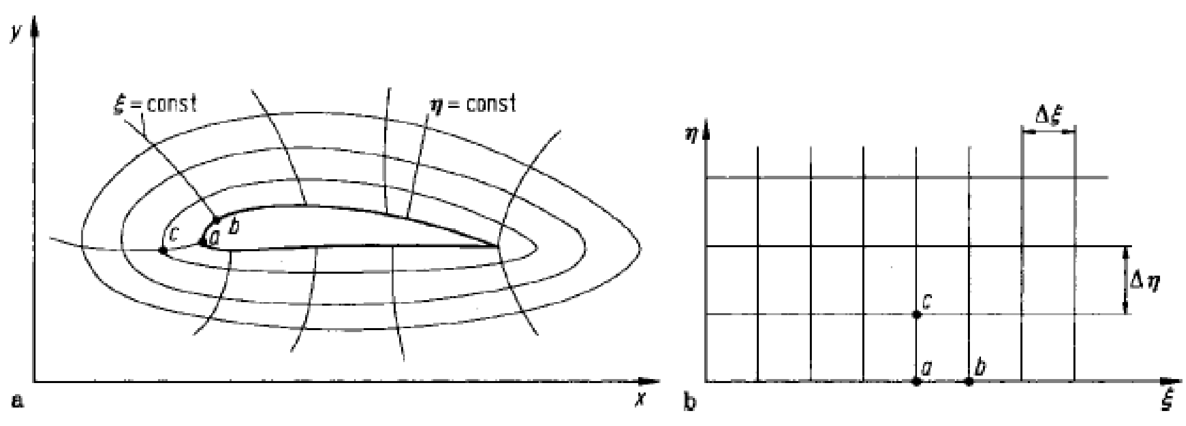

Figure 2.1 Transformation from physical space to computational space

In general grid generation techniques are classified based on complex variables as,

1. **Algebraic methods**
2. **Partial differential methods**
3. **Conformal mapping**

Conformal mapping is based on complex variables and are limited to 2-D problems so this will not be discussed.

Transformations of the Governing Partial Differential Equations

The relationship between the physical and computational space are given by,

$$\begin{aligned} \xi &= \xi(x,y) \\ \eta &= \eta(x,y) \end{aligned}$$

……….Equation 2.1

The chain rule for partial differentiation yields the following expression;

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta}$$

……….Equation 2.2

The above equation can be written as,

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta}$$

……….Equation 2.3

Now consider a model PDE

$$\frac{\partial u}{\partial x} + a\frac{\partial u}{\partial y} = 0$$

The above equation can be transformed from the physical to the computational space by using the equations 2.3, we get

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

This can be rearranged as

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

……….Equation 2.4

This equation is the one which will be solved in the computational domain. The transformation derivatives, ξ_x, ξ_y, η_x, and η_y are defined from the equation 2.1. Comparing the original equation and the transformed equation , the transformed equation is more complicated than the original equation. The advantage outweigh the complexity of the transformed PDE.

Metrics and the Jacobian of Transformation

Recall the equation 2.3. The terms such as ξ_x, ξ_y, η_x, and η_y appear.

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial y} = \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta}$$

These transformation derivatives are defined as the metrics of transformation or simply metrics. The interpolation of the metrics is obvious considering the following approximation;

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

……….Equation 2.5

This expression indicates that the metrics represent the ratio of the arc length in the computational space to that of the physical space.

From equation 2.1. the following differential expressions are obtained,

$$d\xi = \xi_x dx + \xi_y dy$$

$$d\eta = \eta_x dx + \eta_y dy$$

……….Equation 2.6

This can be written in the compact form as,

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

……….Equation 2.7

Reversing the role of independent variables, we get,

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

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

……….Equation 2.8

The above equation can be written as,

$$dx = x_\xi d\xi + x_\eta d\eta$$

$$dy = y_\xi d\xi + y_\eta d\eta$$

……….Equation 2.9

In a compact form it may be written as,

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

……….Equation 2.10

Comparing equation 2.7 and 2.10 it can be concluded as,

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

……….Equation 2.11

From which we can conclude that

$$\begin{aligned} \xi_x &= J y_\eta \\ \xi_y &= -J x_\eta \\ \eta_x &= -J y_\xi \\ \eta_y &= J x_\xi \end{aligned}$$

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

J is defined as the Jacobian of Transformation.

The Jacobian J is interpreted as the ration of the areas in 2D and ratio of volumes in 3D in the computational space to that of the physical space.

Generation of Structured Grids

Grid systems with the following features are desired;

1. A mapping which guarantees one-to-one correspondence ensuring grid lines of the same family do not cross each other;
2. Smoothness of the grid point distribution;
3. Orthogonality or near Orthogonality of the grid lines;
4. Options for grid point clustering;

Algebraic Grid Generation Techniques

The simplest grid generation is the algebraic method. The major advantage of this scheme is the speed with which a grid can be generated. An algebraic equation is used to relate the grid points in the computational domain to those of the physical domain. This objective is met by using an interpolation scheme between the specified boundary grid points to generate the interior grid points.

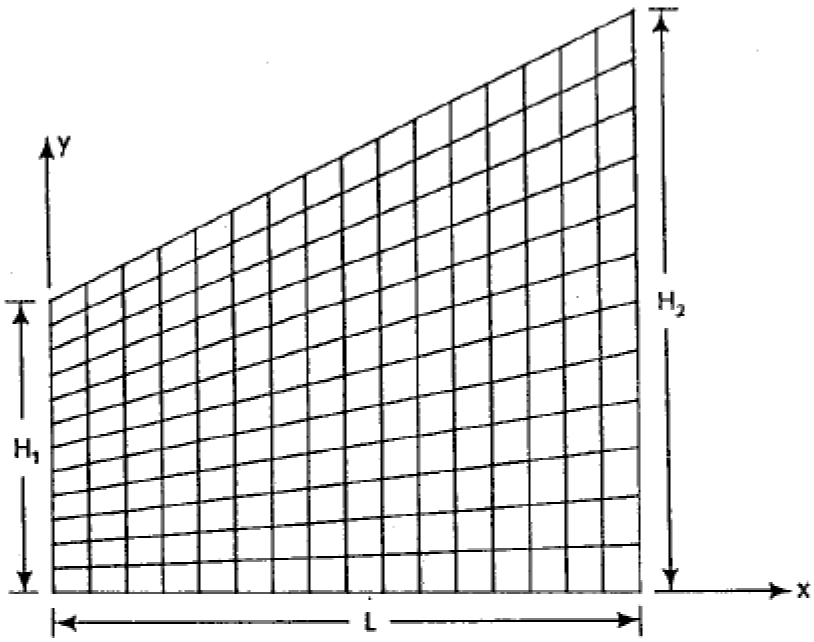

Figure 2.2 The physical space which must be transformed

Consider the simple physical domain depicted in the figure 2.2. Introducing the following algebraic relations will transform this non rectangular physical domain into a rectangular domain:

$$\xi = x$$
$$\eta = \frac{y}{y_t}$$

……….Equation 2.12

y_t represents the upper boundary which is expressed as

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

Thus the equation 1.22 can be written as

$$\xi = x$$
$$\eta = \frac{y}{H_1 + \frac{H_2 - H_1}{L}x}$$

By rearranging the terms we get,

$$x = \xi$$

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

……….Equation 2.13

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

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

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

……….Equation 2.14

In the above equation η is normalized, its value varies from zero to one.

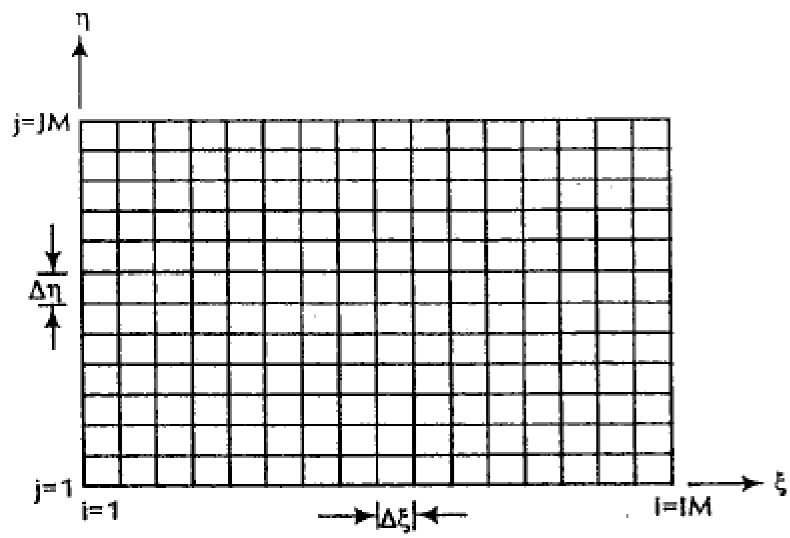

Figure 2.3 The rectangular computational domain with uniform grid spacing

The values of ξ and η are known at each grid point within the domain. The equation 2.13 can be used to identify the corresponding grid points in the physical space.

The metrics and the Jacobian of the transformation must be evaluated before any transformed PDEs can be solved. An algebraic model is used the metrics are calculated analytically. This aspect is an advantage of the algebraic methods since numerical computation of metrics will require additional computation time and may introduce some errors into the system of equations of motion that are to be solved.