**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

### Explicit Finite Difference Method of Subsonic Flows – Elliptical Equations

The governing equation of subsonic fluid flows and heat transfer problems can be reduced to an elliptic form for particular applications. Some of the examples are the steady state heat conduction equation, velocity potential equation for incompressible, inviscid flow, and stream function equation. Now consider Laplace equation,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \qquad \text{……Equation 1.74}$$

The finite difference formulation of the above equation can be written by using the point formula as,

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = 0 \qquad \text{……Equation 1.75}$$

The corresponding grid points are shown in figure 1.16.

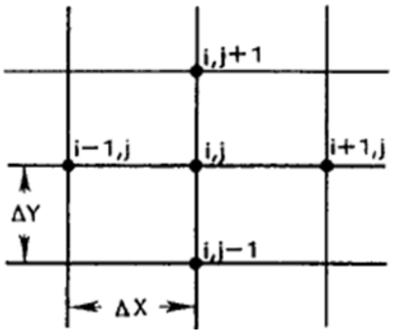


*Figure 1.16 Grid for five-point formula*

$$u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + \left(\frac{\Delta x}{\Delta y}\right)^2 (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = 0$$

……Equation 1.76

Define the ratio of step sizes as β = Δx/Δy and by rearranging the above equation we get,

$$u_{i+1,j} + u_{i-1,j} + \beta^2 u_{i,j+1} + \beta^2 u_{i,j-1} - 2(1+\beta^2)u_{i,j} = 0$$

……Equation 1.77

In order to explore various solution procedures, first consider a square domain with Dirichlet boundary conditions. For example let us simple example of 6 x 6 grid system subjected to the following boundary conditions.

$$x = 0 \quad u = u_2 \quad , \quad y = 0 \quad u = u_1$$

$$x = L \quad u = u_4 \quad , \quad y = H \quad u = u_3$$

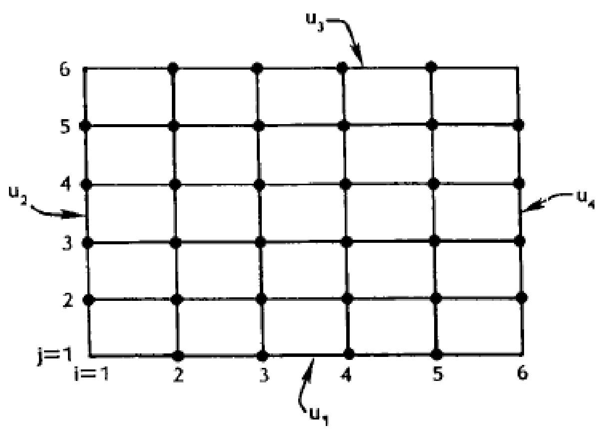


***Figure 1.17 Grid system used for solution***

By applying the above equation 1.77 to the interior grid points produce sixteen equations with sixteen unknowns.

$$u_{3,2} + u_{1,2} + \beta^2 u_{2,3} + \beta^2 u_{2,1} - 2(1+\beta^2)u_{2,2} = 0$$

$$u_{4,2} + u_{2,2} + \beta^2 u_{3,3} + \beta^2 u_{3,1} - 2(1+\beta^2)u_{3,2} = 0$$

$$u_{5,2} + u_{3,2} + \beta^2 u_{4,3} + \beta^2 u_{4,1} - 2(1+\beta^2)u_{4,2} = 0$$

$$u_{6,2} + u_{4,2} + \beta^2 u_{5,3} + \beta^2 u_{5,1} - 2(1+\beta^2)u_{5,2} = 0$$

$$u_{3,3} + u_{1,3} + \beta^2 u_{2,4} + \beta^2 u_{2,2} - 2(1+\beta^2)u_{2,3} = 0$$

$$u_{4,3} + u_{2,3} + \beta^2 u_{3,4} + \beta^2 u_{3,2} - 2(1+\beta^2)u_{3,3} = 0$$

$$u_{5,3} + u_{3,3} + \beta^2 u_{4,4} + \beta^2 u_{4,2} - 2(1+\beta^2)u_{4,3} = 0$$

$$u_{6,3} + u_{4,3} + \beta^2 u_{5,4} + \beta^2 u_{5,2} - 2(1+\beta^2)u_{5,3} = 0$$

$$u_{3,4} + u_{1,4} + \beta^2 u_{2,5} + \beta^2 u_{2,3} - 2(1+\beta^2)u_{2,4} = 0$$

$$u_{4,4} + u_{2,4} + \beta^2 u_{3,5} + \beta^2 u_{3,3} - 2(1+\beta^2)u_{3,4} = 0$$

$$u_{5,4} + u_{3,4} + \beta^2 u_{4,5} + \beta^2 u_{4,3} - 2(1+\beta^2)u_{4,4} = 0$$

$$u_{6,4} + u_{4,4} + \beta^2 u_{5,5} + \beta^2 u_{5,3} - 2(1+\beta^2)u_{5,4} = 0$$

$$u_{3,5} + u_{1,5} + \beta^2 u_{2,6} + \beta^2 u_{2,4} - 2(1+\beta^2)u_{2,5} = 0$$

$$u_{4,5} + u_{2,5} + \beta^2 u_{3,6} + \beta^2 u_{3,4} - 2(1+\beta^2)u_{3,5} = 0$$

$$u_{5,5} + u_{3,5} + \beta^2 u_{4,6} + \beta^2 u_{4,4} - 2(1+\beta^2)u_{4,5} = 0$$

$$u_{6,5} + u_{4,5} + \beta^2 u_{5,6} + \beta^2 u_{5,4} - 2(1+\beta^2)u_{5,5} = 0$$

These equations are expressed in the matrix form as,

$$
\begin{bmatrix}
\alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 0 & 0 & 0 & \beta^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha
\end{bmatrix}
\begin{bmatrix}
u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{5,2} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{5,3} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \\ u_{5,4} \\ u_{2,5} \\ u_{3,5} \\ u_{4,5} \\ u_{5,5}
\end{bmatrix}
=
\begin{bmatrix}
-u_{1,2} - \beta^2 u_{2,1} \\
-\beta^2 u_{3,1} \\
-\beta^2 u_{4,1} \\
-u_{6,2} - \beta^2 u_{5,1} \\
-u_{1,3} \\
0 \\
0 \\
-u_{6,3} \\
-u_{1,4} \\
0 \\
0 \\
-u_{6,4} \\
-u_{1,5} - \beta^2 u_{3,6} \\
-\beta^2 u_{3,6} \\
-\beta^2 u_{4,6} \\
-u_{6,5} - \beta^2 u_{5,6}
\end{bmatrix}
$$

where $\alpha = -2(1+\beta^2)$.

**Solution Algorithms**

In general there are two methods of solution for the system of simultaneous linear algebraic equations they are Direct and Iterative Methods. Some of the familiar direct methods are, Cramer's Rule and Gaussian Elimination Method. The major disadvantage of this method is it has enormous amount of arithmetic operations to produce a solution. So in this chapter discusses only in the iterative method. Iterative procedures for solving a system of linear algebraic equations are simple and easy to program. The idea behind this method is to obtain the solution by iteration. The various formulations of the iterative method can be divided into two categories. If the formulations results only in one unknown this is called as Explicit/Point iterative method. If the formulation involves more than one unknown it is called as Implicit/Line iterative method.

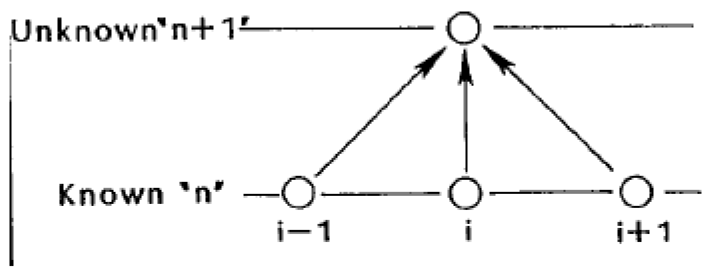


*Figure 1.18 Explicit Formulation*

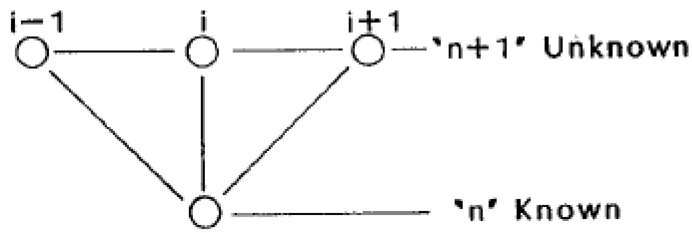


*Figure 1.19 Implicit Formulation*

**The Jacobi Iteration Method**

In this method the dependent variable at each grid point is solved using initial guessed values of the neighboring points or previously computed values. Therefore the equation is given by,

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k})\right]$$

……Equation 1.78

Which is used to compute $u_{ij}$ at the new iteration level of k+1 where k corresponds to the previously computed values. The computation is carried out until a specified convergence criteria is met. The results from the convergence can be called as ***Converged Solution*** if it has met the convergence criteria and as ***Steady-State Solution*** if the results does not vary with time.

**The Point Gauss-Seidel Iteration Method**

In this method the current values of the dependent variable is are used to compute the neighboring points as soon as they are available. This will certainly increase the convergence rate dramatically over the Jacobi method. The method is convergent if the largest elements are located in the main diagonal of the coefficient matrix.

The formal requirement for the convergence of the method is

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

And at least for one row,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

……Equation 1.79

Since this is a sufficient condition, the method may converge even though the condition is met for all rows. The finite difference equation 1.79 can be written as

$$u_{i,j} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j} + u_{i-1,j} + \beta^2(u_{i,j+1} + u_{i,j-1})\right]$$

……Equation 1.80

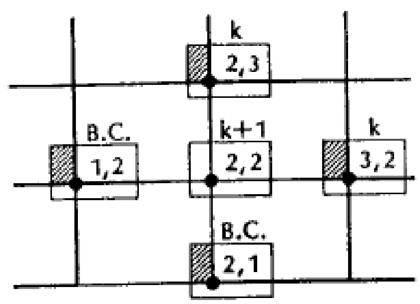


***Figure 1.20 Grid Points for the equation 1.30***

For the computation of the value at the point (2,2) the equation can be written as,

$$u_{2,2}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{3,2} + \underline{u_{1,2}} + \beta^2(u_{2,3} + \underline{u_{2,1}})\right]$$

……Equation 1.81

In the above equation $u_{12}$ and $u_{21}$ are provided by the boundary conditions and values $u_{23}$ and $u_{32}$ are the values from the previous iteration. Thus in terms of the iteration level the equation can be written as,

$$u_{2,2}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{3,2}^{k} + \underline{u_{1,2}} + \beta^2(u_{2,3}^{k} + \underline{u_{2,1}})\right]$$

……Equation 1.82

The general formulation is provided by the equation

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1})\right]$$

……Equation 1.83

This is a point iteration method since only one unknown is sought. The grid points are shown in the below figure.

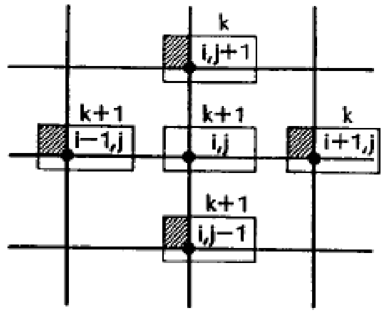


***Figure 1.21 Grid points for the equation 1.83***

**Point Successive Over-Relaxation Method (PSOR)**

In this solution process a trend in the computed values of the dependent variable is noticed, then the direction of change can be used to extrapolate for the next iteration and thereby accelerating the solution procedure. This procedure is known as successive over-relaxation(SOR).

Consider the point Gauss Seidel iteration method, which is given by

$$u_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1})\right]$$

……Equation 1.84

Adding $u_{i,j}^{k} - u_{i,j}^{k}$ to the right hand side and collecting the terms we obtain

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}) - 2(1+\beta^2)u_{i,j}^{k}\right]$$

……Equation 1.85

As the solution proceeds $u_{ij}^{k}$ must approach $u_{ij}^{k+1}$ . To accelerate the solution the values in the bracket is multiplied by ω, the relaxation parameter.

So the equation becomes,

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1}) - 2(1+\beta^2)u_{i,j}^{k}\right]$$

For the solution to converge it is necessary that 0 < ω < 2. If 0 < ω < 1 it is called under relaxation, the above equation is rearranged as,

$$u_{i,j}^{k+1} = (1-\omega)u_{i,j}^{k} + \frac{\omega}{2(1+\beta^2)}\left[u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + \beta^2(u_{i,j+1}^{k} + u_{i,j-1}^{k+1})\right]$$

……Equation 1.86

## 1.9 Explicit Finite Difference Method of Supersonic Flows – Hyperbolic Equations

The model equation considered for studying the Explicit FDM methods for the Hyperbolic equations is First order wave equation,

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x} \qquad a > 0$$

……Equation 1.87

Which is linear equation for constant speed a.

**Euler's FTFS method**

In this explicit method, forward time and forward space approximations of the first-order are used, the resulting Finite Difference Equation (FDE) is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_{i+1}^n - u_i^n}{\Delta x}$$

……Equation 1.88

**Euler's FTCS method**

In this formulation central differencing of special derivative is used, the resulting FDE is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$$

……Equation 1.89

**The First Upwind Differencing Method**

The backward differencing of the special derivative produces the FDE,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_i^n - u_{i-1}^n}{\Delta x}$$

……Equation 1.90

This method is stable when c is less than or equal to 1.

$$c \leq 1, \text{ where } c = a\Delta t/\Delta x$$

It is the Courant number. The FDE for the conditionally stable solution is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -a\frac{u_{i+1}^n - u_i^n}{\Delta x}$$

……Equation 1.91

**The Lax method**

If an average value of $u_i^n$ in the Euler's FTCS method is used , we get a FDE of the form,

$$u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^n) - \frac{a\Delta t}{2\Delta x}(u_{i+1}^n - u_{i-1}^n)$$

……Equation 1.92

This method is stable when, $c \leq 1$

**Midpoint Leapfrog method**

In this method, Central differencing of the second order is used of both the time and space derivative. This gives the FDE,

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = -a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$$

……Equation 1.93

This is of the order $[(\Delta t)^2,\ (\Delta x)^2]$. This method is stable when, $c \leq 1$

This requires the two sets of the initial values to start the solution. The Midpoint Leapfrog method has a higher order of accuracy.

**The Lax-Wendroff method**

This finite difference approximation of the PDE is derived from the Taylor series expansion of the dependent variable as follows.

$$u(x,\ t + \Delta t) = u(x,t) + \frac{\partial u}{\partial t}\Delta t + \frac{\partial^2 u}{\partial t^2}\frac{(\Delta t)^2}{2!} + O(\Delta t)^3$$

……Equation 1.94

In the terms of the indices

$$u_i^{n+1} = u_i^n + \frac{\partial u}{\partial t}\Delta t + \frac{\partial^2 u}{\partial t^2}\frac{(\Delta t)^2}{2!} + O(\Delta t)^3$$

……Equation 1.95

Now consider the model equation

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x}$$

……Equation 1.96

By taking the time derivative we get,

$$\frac{\partial^2 u}{\partial t^2} = -a\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) = -a\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right) = a^2\frac{\partial^2 u}{\partial x^2}$$

……Equation 1.97

Substitute both the equations in the indices terms shown above we get,

$$u_i^{n+1} = u_i^n + \left(-a\frac{\partial u}{\partial x}\right)\Delta t + \frac{(\Delta t)^2}{2}\left(a^2\frac{\partial^2 u}{\partial x^2}\right)$$

……Equation 1.98

By applying the space derivative of the first and second order derivatives, we get

$$u_i^{n+1} = u_i^n - a\Delta t\left[\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right] + \frac{1}{2}a^2(\Delta t)^2\left[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}\right]$$

……Equation 1.99

This formulation is known as the Lax-Wendroff method, this method is stable for $c \leq 1$

## 1.10 Explicit Finite Difference Method of Viscous Flows – Parabolic Equations

**The Forward Time/Central Space (FTCS) method**

In this method forward difference approximation for the time derivative and central differencing for the space derivative which gives,

$$u_i^{n+1} = u_i^n + \frac{\alpha(\Delta t)}{(\Delta x)^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)$$

……Equation 1.100

The above equation is stable for $\alpha\Delta t/(\Delta x)^2 \leq 1/2.$

**The Richardson method**

In this approximation method central differencing is used for both time and space derivatives, the resulting FDE is,

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}$$

……Equation 1.101

The above equation is unconditionally unstable and has no practical value.

**The DuFort-Frankel method**

In this formulation the time derivative is approximated by a central differencing and the second order space derivative is also approximated by the central differencing method. Due to stability constrains $u_i^n$ in the right hand side is replaced by the average value. This is the modification of the Richardson method. The resulting FDE is

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha \frac{u_{i+1}^n - 2\frac{u_i^{n+1} + u_i^{n-1}}{2} + u_{i-1}^n}{(\Delta x)^2}$$

……Equation 1.102

From which

$$u_i^{n+1} = u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\left[u_{i+1}^n - u_i^{n+1} - u_i^{n-1} + u_{i-1}^n\right]$$

……Equation 1.103

This can be rewritten as

$$\left[1 + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n+1} = \left[1 - 2\frac{\alpha(\Delta t)}{(\Delta x)^2}\right] u_i^{n-1} + \frac{2\alpha(\Delta t)}{(\Delta x)^2}\left[u_{i+1}^n + u_{i-1}^n\right]$$

……Equation 1.104

This method is of the order of ,

$$[(\Delta t)^2,\ (\Delta x)^2,\ (\Delta t/\Delta x)^2].$$

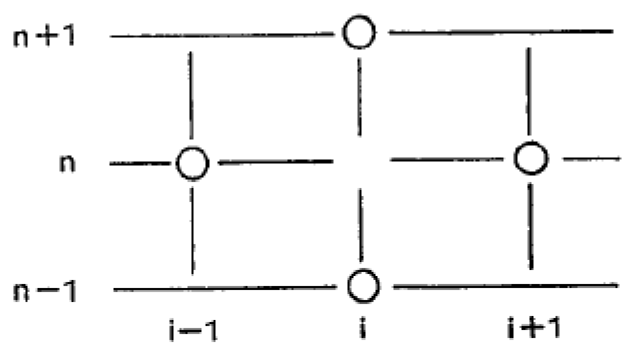


***Figure 1.22 Grid points for the DuFort-Frankel method***